Bayesian inference on the order of stationary vector autoregressions with application to multivariate modelling of electroencephalography data

Rachel Louise Binks

Thesis submitted for the degree of
Doctor of Philosophy

School of Mathematics, Statistics & Physics
Newcastle University
Newcastle upon Tyne
United Kingdom

April 2025

Acknowledgements

Firstly, I would like to thank my supervisors Sarah Heaps and Darren Wilkinson for their support throughout my PhD. I couldn't have asked for a better pair of supervisors and I am grateful for all your help, encouragement and expertise over the last four years. It goes without saying that I couldn't have done it without you. Thank you also to Yujiang Wang for your help with the application sections of this thesis and to Jen Wood, Andrew Turnbull and Colin Gillespie for your help with any programme and supervision related admin.

Thank you to everybody in the CDT for making my PhD experience a thoroughly enjoyable one. Special mention goes to my cohort and in particular Jonny and Karoline. It's been a pleasure to share the PhD experience with you all. I'm also grateful to the rest of my friends who have provided plenty of entertainment over the past few years, especially Martha, Caz and Magda. Thank you also to Jack, Tom Lowe, Kieran and Cameron. I don't think I would have ever considered a PhD if I hadn't seen you start one first. Thanks for showing me that if you can do it, I can too.

To Mum, Dad, Oliver and Eleanor, thank you for your endless love and support and for helping me believe in myself. Finally, thank you to Tom for always finding a way to make me laugh when I'm stressed. I couldn't have done this without your constant support, love and encouragement.

Abstract

Vector autoregressions (VARs) are widely used for modelling multivariate time series. VARs have an associated order p; given observations at the preceding p time points, the variable at time t is conditionally independent of all earlier history. The model order is therefore intrinsic to the characterisation of the process. It is common to assume a VAR is stationary, which requires the means, variances and covariances of the process to be constant over time. This can be enforced by imposing the stationarity condition which restricts the parameter space of the autoregressive coefficients to the stationary region. However, implementing this constraint is difficult as the stationary region has a complex geometry. Fortunately, pioneering recent work has provided a solution for enforcing stationarity in autoregressions of fixed order p based on a reparameterisation in terms of a set of interpretable and unconstrained transformed partial autocorrelation matrices. In this research, focus is placed on the difficult problem of allowing p to be unknown, developing priors and computational inference that take full account of order uncertainty.

To this end, a comparison of existing approaches for determining the order of stationary univariate autoregressions is provided. An approach employing shrinkage priors for partial autocorrelations is then generalised for the multivariate case, using the cumulative shrinkage and multiplicative gamma process priors to increasingly shrink the partial autocorrelation matrices with increasing lag. Identifying the lag beyond which these matrices become equal to zero then determines p. Methods for identifying whether a partial autocorrelation matrix is effectively zero are developed.

The work is illustrated through application to neural activity data. In particular, a detailed discussion of methods to decompose a VAR into latent processes is provided, which is then used to investigate ultradian rhythms in the brain. Relationships between different regions of the brain are investigated through Granger causality plots.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation for thesis

Vector autoregressive (VAR) processes are widely used to model multivariate time series data in a range of application areas. In an autoregression of order p, the random variable at time t is conditionally independent of its values at lags $p+1$, $p+2$, ... given observations at the preceding p time points. Indeed the random variable at time t can be expressed as a noisy linear combination of these p values. The order of the autoregression is therefore intrinsic to the characterisation of the joint process and plays a vital role in forecasting. However, its value is typically not known *a priori*.

It is common to assume a vector autoregressive process is stationary, which requires the means, variances and covariances of the process to be constant over time. This can be enforced by imposing the stationarity condition which restricts the parameter space of the autoregressive coefficients to the stationary region. However, the highly complex geometry of this region hampers the process of specifying a prior and subsequent computational inference. Fortunately, in pioneering recent work, Heaps (2023) solved the problem for vector autoregressions of fixed order by introducing an unconstrained and interpretable reparameterisation of the stationary model. This is constructed by mapping the original model parameters to a set of partial autocorrelation matrices, which can be regarded as a vector analogue of the partial autocorrelation function of a univariate autoregression. A second transformation then scales the singular values of each of these partial autocorrelation matrices from $[0, 1)$ to the positive real line. The transformed partial autocorrelation matrices are interpretable and allow specification of a prior which is invariant with respect to the order of the components in the observation vector. Markov chain Monte Carlo (MCMC) methods for computational inference need only operate over a Euclidean space, making implementation routine. However, a clear limitation of this work is that inference is conditional on a fixed order of the process, with no account for the uncertainty in its

value.

In the context of univariate stationary autoregressions, Bayesian inference on the order of the process has been widely studied (Barnett *et al.*, 1996; Huerta & West, 1999; Vermaak *et al.*, 2004). Due to the geometric complexities of the stationary region in the multivariate case, extensions of these ideas to learning the order of stationary vector autoregressions are rare. Whilst Huerta & Prado (2006) consider determining the order of diagonal vector autoregressions and both Zhang *et al.* (2021) and Fan *et al.* (2022) have considered determining the order of vector autoregressions without enforcing stationarity, the problem of learning the order of the general class of stationary vector autoregressions remains unaddressed in the literature. In this thesis, we provide the first methodology for quantifying uncertainty in model order for the full class of stationary vector autoregressions. To this end, we enforce stationarity using the transformed partial autocorrelation parameterisation of Heaps (2023), exploiting a number of its properties to build shrinkage priors for an overparameterised model. In particular, under this parameterisation, the model of order k is nested within the model of order $k+1$. We can therefore fit an overparameterised model with purposefully more lags than are required and construct priors which increasingly shrink the transformed partial autocorrelation matrices at higher lags towards zero. By identifying the lag beyond which the partial autocorrelations become essentially equal to zero, we learn about the order of the process. The interpretability of the reparameterised model allows classical theory on the sampling distribution of the partial autocorrelation function to inform specification of the shrinkage priors and subsequent decision-making about whether a partial autocorrelation matrix is effectively zero.

A key motivation for learning the order of a stationary autoregression is that conditional on the order we can obtain key insights into the underlying mechanisms of the process. For example, if we know the model order, a stationary autoregression can be decomposed into latent processes accounting for low frequency trends, quasi-periodic behaviour and high frequency noise contributions. Furthermore, conditioning on the model order in a stationary vector autoregression allows interpretations of the relationships between variables, for example through Granger causality networks. These insights can provide greater understanding about data in a variety of fields. In particular, in the field of neuroscience, by modelling electroencephalography (EEG) data recorded across multiple regions of the brain as a vector autoregressive process and inferring the model order, insights into the relationships between different regions of the brain can be obtained. Furthermore, decomposing the observed series into latent processes can allow insight into ultradian rhythms in the brain. Ultradian rhythms are periodic biological rhythms in which the periods are shorter than 24 hours. However their mechanisms and function in the brain remain elusive. We investigate such biological processes in the brain in an application of our Bayesian methods to EEG data, as an example of the possible insights

that can be found as a benefit of our methods.

1.2 Contributions of thesis

In this section we highlight the contributions to the literature that are made in this thesis. We note that parts of this thesis, and particularly Chapter 5, are the basis of the paper *Bayesian inference on the order of stationary vector autoregressions* (Binks *et al.*, 2024).

In Chapter 2 we consider methods for decomposition of stationary vector autoregressions into latent processes, as discussed in Prado (1998). We add some additional detail to the information provided in Prado (1998) clarifying why the latent processes relating to the complex eigenvalues have similar behaviour to an autoregressive model of order two, which is missing some details in the original work.

In Chapter 4 we compare existing methods for Bayesian inference on the order of stationary univariate autoregressions. Additionally, we develop a new procedure for determining the order of stationary univariate autoregressions by adapting existing methods for variable selection in regression (Kuo & Mallick, 1998). This allows us to present a new representation of a spike-and-slab prior for order determination that is computationally faster than the spike-and-slab prior featured in existing work (Barnett *et al.*, 1996). The comparison of methods is illustrated through a simulation study. By considering how computational speed and mixing vary across methods we are able to make recommendations as to which performs best. Our preferred method is then applied to electroencephalography data.

The main contributions of this thesis are discussed in Chapter 5 where we provide the first Bayesian methodology for determining the order of the general class of stationary vector autoregressions. We make use of two increasing shrinkage priors to increasingly shrink the partial autocorrelations at higher lags towards zero and discuss associated methods for computational inference. Both of these priors rely on the concept of an *effective order* of the process which we define to be the highest lag for which the partial autocorrelation matrix is determined to be non-zero. The first increasing shrinkage prior discussed is the cumulative shrinkage process (Legramanti *et al.*, 2020). This is a spike-and-slab prior which we adapt for use in determining the order of stationary vector autoregressions. We present a method for calculating a Rao-Blackwellised estimate of the posterior mass function for the effective order. This is needed as without it the calculation of the effective order under this prior would require sampling of discrete-valued parameters which are not permitted in the Bayesian inference software Stan (Carpenter *et al.*, 2017), which we use for analysis. Based on classical time series theory, we also provide a method for choosing the value of the parameter θ_∞ in the cumulative shrinkage process prior which determines the location of the spike. The second increasing shrinkage prior we consider is the multi-

plicative gamma process (Bhattacharya & Dunson, 2011) which is a global-local shrinkage prior that requires a choice of truncation criterion to determine if parameters are effectively equal to zero. Based on classic time series theory, we suggest a principled choice of truncation criterion to identify whether a partial autocorrelation matrix is effectively zero. We also present a thorough simulation experiment investigating the success of each prior in inferring the true order of a VAR process.

In Chapter 6 we apply our methods to electroencephalography data to contribute to the literature on biological processes in the brain. When applying the cumulative shrinkage process to the EEG data we find that this choice of prior is highly sensitive to the choice of the hyperparameter θ_∞ when model misspecification is present. Nevertheless we are able to use the much more robust multiplicative gamma process prior to provide insights into relationships between different regions of the brain and investigate its ultradian rhythms.

1.3 Outline of thesis

The remaining chapters of this thesis are outlined as follows. Chapter 2 contains background materials on Bayesian inference and time series modelling. It includes introductory information on Markov chain Monte Carlo (MCMC) methods including the Metropolis-Hastings and Hamiltonian Monte Carlo algorithms. Univariate and vector autoregressions are defined and the stationarity conditions for such autoregressions are discussed. Finally, a discussion of methods for decomposing a vector autoregressive process into latent processes is provided which we use in Chapter 6.

In Chapter 3 we discuss the electroencephalography data used to illustrate our modelling and inferential procedures. We carry out some exploratory data analysis of the data and use classical methods for time series analysis to fit basic autoregressive moving average models to the univariate series in each brain region.

Chapter 4 discusses methods for Bayesian inference of stationary univariate autoregressions with a fixed order, before discussing existing methods of Bayesian inference of stationary univariate autoregressions where the model order is unknown. These methods reparameterise the model in terms of either the partial autocorrelations or the roots of the characteristic equation in order to simplify the stationarity condition. Both reparameterisations are discussed. To compare the different methods a simulation study is carried out. We then consider Bayesian order determination for the univariate series associated with each brain region of the electroencephalography data.

In Chapter 5 we discuss a reparametrisation of the autoregressive matrices in a VAR model in terms of a set of interpretable, unconstrained, partial autocorrelation matrices which generalises univariate results. We then discuss Bayesian inference of vector autoregressive processes with a fixed model order. This work is extended to allow for an

unknown order. Three choices of prior are considered to facilitate this: a simple spike-and-slab prior, the cumulative shrinkage process and the multiplicative gamma process. Posterior inference is discussed and carried out using Hamiltonian Monte Carlo through Stan. A thorough simulation study is presented to explore the behaviour of the posterior for the model order in the idealised setting where we know the data were generated from a VAR process.

In Chapter 6 we apply the methods discussed in Chapter 5 to electroencephalography data. We discuss order determination before conditioning on the posterior modal order to investigate biological processes in the brain. Relationships between different regions in the brain are investigated using Granger causality plots before we investigate ultradian rhythms in the brain by decomposing the data into latent processes.

Finally, in Chapter 7 we conclude the thesis and discuss further work.

Chapter 2

Background materials

2.1 Bayesian inference

There are two main frameworks in statistical analysis, classical statistical analysis, known as the frequentist framework, and Bayesian analysis. Throughout this thesis we will take a Bayesian approach to analysis. Bayesian inference focuses on quantifying the uncertainty associated with all unknowns through a probability distribution. In this section, we discuss the fundamentals of Bayesian inference. Further details can be found in Gelman *et al.* (2014) or Congdon (2006).

2.1.1 Bayes' theorem

There are a number of key components in a Bayesian analysis, namely the prior distribution, the likelihood function and the posterior distribution. Denote the observed data by $\boldsymbol{x} = (x_1, \ldots, x_n)$ and the unknown model parameters by $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ and assume that $\boldsymbol{x}$ can be modelled by a probability density function $p(\boldsymbol{x}|\boldsymbol{\theta})$ (or a probability mass function if the data is discrete). The prior distribution, denoted $\pi(\boldsymbol{\theta})$, is a probability distribution representing our beliefs about the possible values of the model parameters $\boldsymbol{\theta}$, before observing any data. The information obtained from the observed data is contained in the likelihood function

$$L(\boldsymbol{\theta}|\boldsymbol{x}) = p(\boldsymbol{x}|\boldsymbol{\theta}),$$

which can be regarded as a function of the unknown model parameters $\boldsymbol{\theta}$ when evaluated at $\boldsymbol{x}$. Bayesian inference then relies on the idea of using the information observed in the data to update our prior beliefs using Bayes' theorem, resulting in a posterior distribution. Using Bayes' theorem, the posterior density is such that

$$\pi(\boldsymbol{\theta}|\boldsymbol{x}) = \frac{\pi(\boldsymbol{\theta})L(\boldsymbol{\theta}|\boldsymbol{x})}{\int_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta})L(\boldsymbol{\theta}|\boldsymbol{x})d\boldsymbol{\theta}}. \tag{2.1}$$

The integral $\int_{\boldsymbol{\theta}} \pi(\boldsymbol{\theta})L(\boldsymbol{\theta}|\boldsymbol{x})d\boldsymbol{\theta}$ is a normalising constant that ensures the posterior density integrates to equal one. As this integral does not depend on $\boldsymbol{\theta}$, it is common to write Bayes' theorem as

$$\pi(\boldsymbol{\theta}|\boldsymbol{x}) \propto \pi(\boldsymbol{\theta})L(\boldsymbol{\theta}|\boldsymbol{x}).$$

In some cases, the prior distribution can be chosen so that when updating the prior through Bayes' theorem the posterior distribution is from the same family as the prior. For example, if a normal prior distribution is chosen for the mean of data which are normally distributed, the resulting posterior distribution for the mean will also be a normal distribution with updated parameter values. When the prior and resulting posterior distributions have the same distribution, the prior is referred to as a conjugate prior. The use of a conjugate prior generally makes evaluation of the posterior density very straightforward. However, in many cases the conjugate prior distribution may not accurately represent our prior beliefs about the parameters. As such, an alternative prior distribution should be chosen which better represents those prior beliefs, but this can result in more complicated analysis. In particular, it is often the case that the normalising constant in (2.1) cannot be evaluated in closed form, making the posterior density analytically intractable. In cases such as this it is common to resort to simulation techniques which can simulate draws from the posterior density without having to be able to evaluate the normalising constant. In particular, throughout this thesis we use a range of Markov chain Monte Carlo (MCMC) algorithms to sample from the posterior densities of our unknown parameters.

2.1.2 Markov chain Monte Carlo methods

Markov chain Monte Carlo (MCMC) methods are computational techniques which use Markov chains to sample from the joint posterior distribution of the parameters $\boldsymbol{\theta}$. They work by generating draws from a Markov chain whose stationary distribution is the same as the posterior distribution. The chain is initialised at a point with support in the posterior and over time converges to a stationary distribution. Once the chain has reached the stationary distribution, the samples from the Markov chain can be taken as (correlated) samples from the posterior distribution. There are a number of different MCMC algorithms which implement this process.

Metropolis-Hastings algorithm

The Metropolis-Hastings algorithm (Hastings, 1970), is a commonly used MCMC algorithm used to sample from the posterior distribution $\pi(\boldsymbol{\theta}|\boldsymbol{x})$. In the Metropolis-Hastings algorithm, we first generate a proposed value $\boldsymbol{\theta}^*$ from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ which typically has the same support as the posterior distribution and is easy to sample from. Then, we accept or reject the proposed value as a sample from the posterior distribution in

accordance with the acceptance probability, which depends on the posterior distribution. The full algorithm is detailed in Algorithm 1. It can be noted that the acceptance probability depends only on the ratio of probability densities and so the posterior density only needs to be known up to a proportionality constant. This is how the often very difficult task of evaluating the normalising constant in the posterior density is avoided. It can also be noted that if the proposal distribution is symmetric such that $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)$ then the acceptance probability simplifies to

$$\alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta}) = \min\left\{1, \frac{\pi\left(\boldsymbol{\theta}^*|\boldsymbol{x}\right)}{\pi\left(\boldsymbol{\theta}|\boldsymbol{x}\right)}\right\}.$$

Throughout this thesis we use a range of proposal distributions where the choice is guided largely by the support of the posterior. In the majority of cases we use random walk proposal distributions $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)})$ which are defined such that

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{(k-1)} + \boldsymbol{u}_k$$

where the $\boldsymbol{u}_k$ are independent and identically distributed noise terms. For all proposal distributions, the acceptance rate is controlled by the variance, such that larger variances result in bigger jumps from the current value and fewer proposed values being accepted. Parameters in the proposal distribution which control the variance are therefore referred to as tuning parameters, as they can be tuned to give a variance which results in a desired proportion of proposed values being accepted.

Algorithm 1 Metropolis-Hastings algorithm

1. Initialise the state of the chain to $\boldsymbol{\theta}^{(0)}$ and set the iteration counter to $k = 1$.
2. Generate a proposed value $\boldsymbol{\theta}^* \sim q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)})$.
3. Evaluate the acceptance probability $\alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(k-1)})$ where
$$\alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(k-1)}) = \min\left\{1, \frac{\pi\left(\boldsymbol{\theta}^*|\boldsymbol{x}\right) q\left(\boldsymbol{\theta}^{(k-1)}|\boldsymbol{\theta}^*\right)}{\pi\left(\boldsymbol{\theta}^{(k-1)}|\boldsymbol{x}\right) q\left(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{(k-1)}\right)}\right\}.$$
4. Set $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^*$ with probability $\alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta}^{(k-1)})$. Otherwise set $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)}$.
5. Set k equal to $k + 1$ and return to step 2.

Algorithm 1 presents the Metropolis-Hastings algorithm in the case where we propose and then accept or reject a value for $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_d)$ as a whole. However, it may not always be feasible to define a proposal distribution for the whole parameter set in one go.

If this is the case we resort to componentwise transitions in our MCMC algorithm, where we propose and accept or reject each parameter one-at-a-time with its own Metropolis-Hastings step. Here, each parameter $\theta_1, \ldots, \theta_d$ is sampled from its full conditional distribution (FCD) where the full conditional distribution for θ_i, $\pi(\theta_i|\theta_1, \ldots, \theta_{i-1}, \theta_{i+1}, \ldots, \theta_d)$, is the posterior density of θ_i conditional on all other components of $\boldsymbol{\theta}$ and the data $\boldsymbol{x}$. Algorithm 2 details the full MCMC algorithm in the case of componentwise Metropolis-Hastings transitions.

Algorithm 2 Componentwise Metropolis-Hastings algorithm

1. Initialise the state of the chain to $\boldsymbol{\theta}^{(0)}$ and set the iteration counter to $k = 1$.
2. Obtain $\boldsymbol{\theta}^{(k)}$ by iteratively updating each parameter:
   - Sample $\theta_1^{(k)} \sim \pi(\theta_1|\theta_2^{(k-1)}, \ldots, \theta_d^{(k-1)}, \boldsymbol{x})$ using a Metropolis-Hastings step with proposal distribution $q_1(\theta_1^*|\theta_1^{(k-1)})$.
   - Sample $\theta_2^{(k)} \sim \pi(\theta_2|\theta_1^{(k)}, \ldots, \theta_d^{(k-1)}, \boldsymbol{x})$ using a Metropolis-Hastings step with proposal distribution $q_2(\theta_2^*|\theta_2^{(k-1)})$.
   - $\vdots$
   - Sample $\theta_d^{(k)} \sim \pi(\theta_d|\theta_1^{(k)}, \ldots, \theta_{d-1}^{(k)}, \boldsymbol{x})$ using a Metropolis-Hastings step with proposal distribution $q_d(\theta_d^*|\theta_d^{(k-1)})$.
3. Set k equal to $k + 1$ and return to step 2.

Gibbs algorithm

The Gibbs algorithm (Geman & Geman, 1984; Gelfand & Smith, 1990) is a special case of the componentwise Metropolis-Hastings algorithm detailed in Algorithm 2, where the FCD for each parameter is available to sample from. Here, in each step the FCD for the parameter is used as the proposal distribution, resulting in an acceptance probability of 1. The full Gibbs algorithm is detailed in Algorithm 3.

It may be that a subset of the parameters $\theta_1, \ldots, \theta_d$ have FCDs which are easily available to sample from, but others do not. In this case we use the Metropolis-within-Gibbs algorithm in which each component can be updated using either a Gibbs step or a Metropolis-Hastings step depending on whether the FCD can be directly sampled from.

Hamiltonian Monte Carlo

A disadvantage with standard Metropolis-Hastings samplers is that it can be difficult to design efficient proposal distributions that result in an appropriate number of transitions

Algorithm 3 Gibbs algorithm

1. Initialise the state of the chain to $\boldsymbol{\theta}^{(0)}$ and set the iteration counter to $k = 1$.
2. Obtain $\boldsymbol{\theta}^{(k)}$ by iteratively updating each parameter:
   - Sample $\theta_1^{(k)} \sim \pi(\theta_1 | \theta_2^{(k-1)}, \ldots, \theta_d^{(k-1)}, \boldsymbol{x})$.
   - Sample $\theta_2^{(k)} \sim \pi(\theta_2 | \theta_1^{(k)}, \ldots, \theta_d^{(k-1)}, \boldsymbol{x})$.
   - $\vdots$
   - Sample $\theta_d^{(k)} \sim \pi(\theta_d | \theta_1^{(k)}, \ldots, \theta_{d-1}^{(k)}, \boldsymbol{x})$.
3. Set k equal to $k + 1$ and return to step 2.

being accepted. If the sampler proposes a new value which is a large jump from the current value then it is likely to be rejected, whereas proposing values closer to the current value may mean more proposals are accepted but can result in slower movement around the parameter space and slow convergence. An alternative algorithm that has often been found to be more efficient is the Hamiltonian Monte Carlo (HMC) algorithm (Duane *et al.*, 1987; Neal, 2011) which uses information about the shape of the posterior distribution to generate proposed values. As these proposals are more targeted, those proposals with bigger jumps from the current value are still likely to be accepted, resulting in faster movement around the parameter space.

Hamiltonian Monte Carlo works by exploiting Hamiltonian dynamics to model the movement of the sampler around the posterior distribution as the motion of a particle moving through unbounded frictionless space. HMC treats the vector of unknown parameters $\boldsymbol{\theta}$ as the location of the particle and introduces a vector of auxiliary variables $\boldsymbol{\varphi} = (\varphi_1, \ldots, \varphi_d)$ to represent the momentum vector corresponding to the position $\boldsymbol{\theta}$ of the particle. As discussed in Neal (2011) the potential energy of the particle is denoted $U(\boldsymbol{\theta})$ and the kinetic energy is usually defined as $K(\boldsymbol{\varphi}) = \boldsymbol{\varphi}^{\mathrm{T}} M^{-1} \boldsymbol{\varphi}/2$ where M is the mass of the particle. The Hamiltonian dynamics of the system are described by the Hamiltonian, which is

$$H(\boldsymbol{\theta}, \boldsymbol{\varphi}) = U(\boldsymbol{\theta}) + K(\boldsymbol{\varphi}). \tag{2.2}$$

The motion of the particle is determined by the partial derivatives of the Hamiltonian which describe how $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ change over time. These partial derivatives are known as

Hamilton's equations:

$$\frac{d\theta_i}{dt} = \frac{\partial H}{\partial \varphi_i}$$
$$\frac{d\varphi_i}{dt} = -\frac{\partial H}{\partial \theta_i}$$

for $i = 1, \ldots, d$. Hamiltonian dynamics have a number of properties which make them well suited for use in constructing MCMC updates, as discussed in Neal (2011). First, Hamiltonian dynamics are reversible, as the mapping from the state at time t, $(\boldsymbol{\theta}(t), \boldsymbol{\varphi}(\boldsymbol{t}))$, to the state at time $t+s$, $(\boldsymbol{\theta}(t+s), \boldsymbol{\varphi}(\boldsymbol{t+s}))$, is one-to-one and so has an inverse. Second, the total energy in the system is conserved which can be seen as follows:

$$\frac{dH}{dt} = \sum_{i=1}^{d} \left(\frac{\partial H}{\partial \theta_i}\frac{d\theta_i}{dt} + \frac{\partial H}{\partial \varphi_i}\frac{d\varphi_i}{dt} \right) = \sum_{i=1}^{d} \left(\frac{\partial H}{\partial \theta_i}\frac{\partial H}{\partial \varphi_i} - \frac{\partial H}{\partial \varphi_i}\frac{\partial H}{\partial \theta_i} \right) = 0.$$

The significance of this is that if we obey Hamiltonian dynamics, a trajectory in $(\boldsymbol{\theta}, \boldsymbol{\varphi})$-space will follow the contours of constant energy of the Hamiltonian. Finally, Hamiltonian dynamics preserve volume. Arnold (1989) shows that a vector field with zero divergence is volume preserving which can be checked for Hamiltonian dynamics as follows:

$$\sum_{i=1}^{d} \left(\frac{\partial}{\partial \theta_i}\frac{d\theta_i}{dt} + \frac{\partial}{\partial \varphi_i}\frac{d\varphi_i}{dt} \right) = \sum_{i=1}^{d} \left(\frac{\partial}{\partial \theta_i}\frac{\partial H}{\partial \varphi_i} - \frac{\partial}{\partial \varphi_i}\frac{\partial H}{\partial \theta_i} \right) = \sum_{i=1}^{d} \left(\frac{\partial^2}{\partial \theta_i \partial \varphi_i} - \frac{\partial^2}{\partial \varphi_i \partial \theta_i} \right) = 0.$$

As discussed in Gelman *et al.* (2014), in HMC we introduce an augmented target distribution

$$\tilde{\pi}(\boldsymbol{\theta}, \boldsymbol{\varphi}) \propto \exp\{-H(\boldsymbol{\theta}, \boldsymbol{\varphi})\}$$

such that moves which leave the Hamiltonian invariant will also leave the augmented target density invariant. So by following Hamiltonian dynamics, we can generate proposals which represent large (and reversible) moves around the parameter space that will be accepted with high probability. From (2.2) we have

$$\begin{aligned}
\tilde{\pi}(\boldsymbol{\theta}, \boldsymbol{\varphi}) &\propto \exp\{-H(\boldsymbol{\theta}, \boldsymbol{\varphi})\} \\
&= \exp\{-U(\boldsymbol{\theta})\} \exp\{-K(\boldsymbol{\varphi})\} \\
&= \exp\{-U(\boldsymbol{\theta})\} \exp\left(-\frac{1}{2} \boldsymbol{\varphi}^{\mathrm{T}} M^{-1} \boldsymbol{\varphi} \right)
\end{aligned}$$

and so if we take $U(\boldsymbol{\theta}) = -\pi(\boldsymbol{\theta}|\boldsymbol{x})$, this makes our target posterior distribution the $\boldsymbol{\theta}$-marginal. We also see that the $\boldsymbol{\varphi}$-marginal is such that

$$\boldsymbol{\varphi} \sim N_d(\mathbf{0}, M).$$

Clearly this is also the FCD of $\boldsymbol{\varphi}$ and so if $\boldsymbol{\varphi}$ is sampled from this FCD whilst leaving $\boldsymbol{\theta}$ unchanged, this is a Gibbs move. Resampling $\boldsymbol{\varphi}$ is necessary to allow movement to different contours of H and hence proper exploration of the target posterior. Ideally, therefore, HMC would proceed by first sampling $\boldsymbol{\varphi}$ exactly from its marginal, then updating $\boldsymbol{\theta}$ and $\boldsymbol{\varphi}$ jointly by following the Hamiltonian dynamics (which would be accepted with probability one because it leaves the augmented target invariant). However, we cannot usually solve Hamilton's equations analytically. Therefore, we use numerical methods instead. Since this is not exact we must then decide whether or not to accept the obtained values as a draw from the target density. Hence, the numerical solution to these differential equations is used as a proposal value $(\boldsymbol{\theta}^*, \boldsymbol{\varphi}^*)$ in the HMC algorithm, which is accepted using a Metropolis-Hastings acceptance step. In order to solve these equations numerically, most implementations of HMC use the leapfrog integrator which discretises the trajectory of the particle over small steps ϵ. The leapfrog integrator is reversible and preserves volume exactly and so it follows the Hamiltonian dynamics fairly well, making it a good method for generating proposal values. The full HMC algorithm is detailed in Algorithm 4.

In addition to the matrix M, both the step size ϵ and the number of leapfrog steps L (see Algorithm 4) influence the efficiency of the sampler and can be regarded as tuning parameters (Neal, 2011). If ϵ is too large then the simulation will be inaccurate and result in a low acceptance rate, whereas if ϵ is too small then there will be lots of small steps resulting in slow computation time. If the number of leapfrog steps L is too small then successive samples will be close together, resulting in the slow mixing which we hoped to avoid through the use of HMC. On the other hand, if L is too large then the computation time will be too slow. The matrix M is usually chosen to be diagonal so that the components of $\boldsymbol{\varphi}$ are independent.

Throughout this thesis, we use `cmdstanr` (Gabry & Cesnovar, 2021), a lightweight R interface to the Stan software (Carpenter *et al.*, 2017) to implement the HMC algorithm. Stan requires users to write a programme in the probabilistic Stan modelling language, the role of which is to provide instructions for computing the logarithm of the kernel of the posterior density function. The Stan software then automatically sets up a Markov chain simulation to sample from the resulting posterior. This includes calculation of the gradient of the logarithm of the posterior density, random initialisation of the chains, and the tuning of the sampler. Stan optimises ϵ to match a user defined acceptance rate and estimates M using warmup iterations (Stan Development Team, 2024). Additionally, Stan makes use of a no-U-turn (NUTS) sampler (Hoffman *et al.*, 2014) to adapt L at each iteration.

Algorithm 4 Hamiltonian Monte Carlo algorithm

1. Initialise the state of the chain to $\boldsymbol{\theta}^{(0)}$ and set the iteration counter to $k = 1$.
2. Update the momentum by sampling $\boldsymbol{\varphi} \sim N_d(\mathbf{0}, M)$.
3. Use a total of L leapfrog steps to simultaneously update $(\boldsymbol{\theta}, \boldsymbol{\varphi})$ where one leapfrog step is as follows:
   (a) Make a half-step update of $\boldsymbol{\varphi}$ using the gradient of the log posterior such that
   $$\boldsymbol{\varphi} = \tilde{\boldsymbol{\varphi}} + \frac{1}{2}\epsilon \frac{d \log \pi(\boldsymbol{\theta}|\boldsymbol{x})}{d\boldsymbol{\theta}}$$
   where $\tilde{\boldsymbol{\varphi}}$ is the value of $\boldsymbol{\varphi}$ before the half-step update and ϵ is the step size which is used to tune the algorithm.
   (b) Update $\boldsymbol{\theta}$ using
   $$\boldsymbol{\theta} = \tilde{\boldsymbol{\theta}} + \epsilon \Sigma_{\boldsymbol{\varphi}} \boldsymbol{\varphi}$$
   where $\tilde{\boldsymbol{\theta}}$ is the value of $\boldsymbol{\theta}$ before the update.
   (c) Make a half-step update of $\boldsymbol{\varphi}$ using
   $$\boldsymbol{\varphi} = \tilde{\boldsymbol{\varphi}} + \frac{1}{2}\epsilon \frac{d \log \pi(\boldsymbol{\theta}|\boldsymbol{x})}{d\boldsymbol{\theta}}.$$
   After completing the L leapfrog steps, set $(\boldsymbol{\varphi}^*, \boldsymbol{\theta}^*)$ equal to the resulting values of $(\boldsymbol{\varphi}, \boldsymbol{\theta})$.
4. Set $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^*$ with probability equal to
   $$\min\left\{1, \frac{\pi(\boldsymbol{\theta}^*|\boldsymbol{x})\pi(\boldsymbol{\varphi}^*)}{\pi(\boldsymbol{\theta}|\boldsymbol{x})\pi(\boldsymbol{\varphi})}\right\}.$$
   Otherwise set $\boldsymbol{\theta}^{(k)} = \boldsymbol{\theta}^{(k-1)}$.
5. Set k equal to $k + 1$ and return to step 2.

Diagnostics checks

Whilst MCMC methods should converge to the stationary distribution over time, convergence may may be slow and so before analysing any output we must assess whether or not the chain has converged. At the beginning of any MCMC algorithm, the Markov chain is initialised at a value with support in the posterior. A common method for initialising the chain is to randomly sample an initial value from the prior. Then, the chain will take a number of iterations to converge to the stationary distribution. The samples obtained whilst converging to the stationary distribution are referred to as the *burn in* or *warmup* period and should be removed. Convergence is usually assessed visually using trace plots.

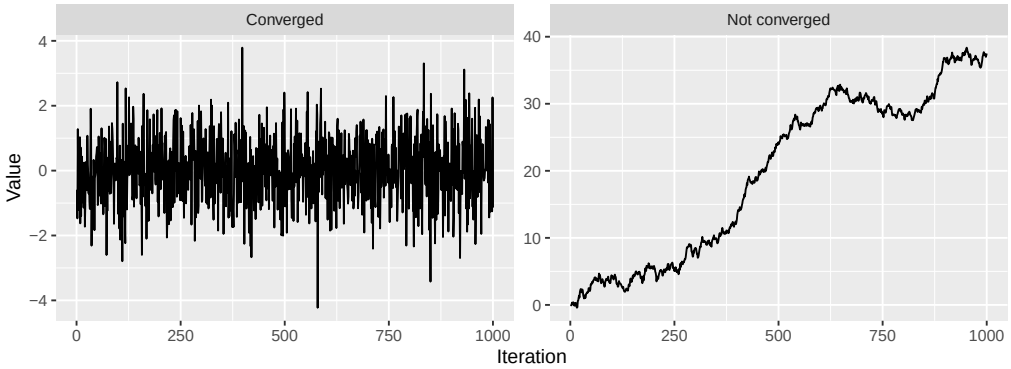


Figure 2.1: Trace plots for a Markov chain which has converged to the stationary distribution (left) and a Markov chain which has not converged to the stationary distribution (right).

The left hand panel of Figure 2.1 contains an example of a trace plot where the Markov chain has converged to the stationary distribution, whereas the right hand panel contains an example in which the Markov chain has not converged. Furthermore, whilst the plot on the left suggests that this chain has converged, it could be that it has become stuck at a local mode and is not actually exploring the full posterior density. In order to investigate whether a chain is exploring the full parameter space we can initialise multiple chains at different points and then check that the trace plots for each chain overlap. If the trace plots for each chain do not overlap, this suggests lack of convergence for at least one of the chains, whereas if all chains overlap we have no evidence to suggest that the chains have not converged to a global posterior mode. We can also overlay the marginal posterior densities obtained from each chain. If all posterior densities overlap then, as with the trace plots, we can be more confident that the chains have all converged to the global posterior mode. A numerical method for assessing convergence is the potential scale reduction factor known as $\hat{R}$ (Gelman & Rubin, 1992) which compares the average variance of the within-state samples across multiple chains to the variance of the pooled samples, with the idea that if the chains have converged then the within-state variance will equal the pooled variance. If all chains are in the equilibrium distribution then $\hat{R}$ will equal one. Vehtari *et al.* (2021) suggest that the sample should be investigated further for non convergence if $\hat{R} > 1.01$.

Additionally, whilst the Markov chain will converge to the stationary distribution eventually, it may then move around the stationary distribution slowly, with consecutive samples being highly autocorrelated. The movement of the chain around the stationary distribution is referred to as mixing. In an ideal world, there would be no autocorrelation between successive samples, resulting in a set of entirely independent samples from the posterior. However, in practice autocorrelation in successive samples is a common feature

of these Markov chains. An alternative interpretation of the trace plot in the right hand panel of Figure 2.1 could be that the chain has converged to the stationary distribution but the mixing is poor with the chain moving around the parameter space very slowly. In addition to helping establish whether a chain has converged to the global posterior mode, using multiple chains initialised at different points can help to establish whether a trace plot like this is indicating lack of convergence or poor mixing. A numerical way of assessing the mixing in a posterior sample is to calculate the effective sample size (ESS) (Plummer *et al.*, 2006; Goodman & Weare, 2010). The ESS is a measure of the effective number of independent samples that the chain is equivalent to. The ESS for a parameter θ is

$$N_{ESS} = \frac{N}{1 + 2\sum_{k=1}^{\infty} \hat{\rho}_k}$$

where $\hat{\rho}_k$ is the sample autocorrelation at lag k and N is the total number of samples. If the ESS is too low, it is good practice to run the algorithm for longer to gain a larger sample, which will result in a higher ESS. It may be the case that this results in a posterior sample which is computationally difficult to work with due to its size. In these cases, the posterior sample can be thinned by only retaining every k-th sample for a chosen value k, reducing computational overheads but still increasing the number of independent samples. Another way of assessing the mixing is to look at plots of the sample autocorrelation function $\hat{\rho}_k$ against the lag k. Typically, the moduli of the sample autocorrelations are compared to the value $1.96/\sqrt{N}$ and if the modulus at lag k is greater than $1.96/\sqrt{N}$ the n-th and $(n+k)$-th samples are correlated. As with a low ESS, if we find that there is a high level of autocorrelation between successive sample sizes it is often a good idea to run the sampler for a higher number of iterations, in order to achieve a desired number of approximately uncorrelated samples.

2.2 Time series modelling

Time series data are data which are recorded over time, often at regularly spaced intervals. In this thesis we focus on modelling multivariate time series data, where data are observed for multiple variables simultaneously at each time point.

2.2.1 Autoregressive moving average models

A popular choice of model for univariate time series data is the autoregressive moving average (ARMA) model, discussed in Box & Jenkins (1976). Autoregressive moving average models are comprised of two parts, an autoregressive component and a moving average

component. A zero-mean autoregressive process of order p, denoted AR(p), has the form

$$y_t = \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t$$

for $t = p+1, \ldots, n$. Here, the ϕ_s, $s = 1, \ldots, p$, are referred to as autoregressive coefficients and the random variable ε_t is white noise. As the focus of this thesis is on determining the order of stationary autoregressions, rather than the error terms, we only consider the most common case of Gaussian errors where $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$ independently for $t = p+1, \ldots, n$ with $\sigma^2 > 0$. Furthermore, as will be discussed in Chapter 3, in our application to EEG data extreme values were removed during preprocessing and so Gaussian errors are likely to be more suited to the data than heavier tailed distributions.

An alternative way of expressing an AR(p) model is

$$y_t - \phi_1 y_{t-1} - \ldots - \phi_p y_{t-p} = \phi(B) y_t = \varepsilon_t$$

where $\phi(B) = 1 - \phi_1 B - \ldots - \phi_p B^p$ is called the autoregressive operator and B, known as the backshift operator, is such that $B^s y_t = y_{t-s}$. A moving average process of order q, denoted MA(q), has the form

$$y_t = \varepsilon_t + \psi_1 \varepsilon_{t-1} + \ldots + \psi_q \varepsilon_{t-q}$$

where $\varepsilon_t \sim \mathrm{N}(0, \sigma^2)$ independently for $t = q+1, \ldots, n$ with $\sigma^2 > 0$. In backshift notation, this can be expressed as

$$y_t = \Psi(B) \varepsilon_t$$

where $\Psi(B) = 1 + \psi_1 B + \ldots \psi_q B^q$ is called the moving average operator. A zero-mean process that contains both an autoregressive component of order p and a moving average component of order q is denoted ARMA(p, q) and has the form

$$y_t = \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \varepsilon_t + \psi_1 \varepsilon_{t-1} + \ldots + \psi_q \varepsilon_{t-q}$$

equivalent to

$$\Phi(B) y_t = \Psi(B) \varepsilon_t$$

in backshift operator notation.

2.2.2 Vector autoregressive models

Vector autoregressive processes are a multivariate extension to univariate autoregressions and are widely used to model multivariate time series data in a variety of fields. These applications include modelling functional MRI (Chiang *et al.*, 2016) and electroencephalog-

raphy data in neuroscience (Herrera *et al.*, 1997; Goyal & Garg, 2020; Malinovskaia, 2022), modelling microbial dynamics in bioinformatics (Jiang *et al.*, 2013; Hannaford *et al.*, 2023), modelling the daily demand for gas in energy economics (Heaps *et al.*, 2020), and modelling macroeconomics (Sims, 1980; Koop & Korobilis, 2010). An m-variate zero-mean vector autoregressive process of order p, denoted $\text{VAR}_m(p)$, has the form

$$\boldsymbol{y}_t = \phi_1 \boldsymbol{y}_{t-1} + \ldots + \phi_p \boldsymbol{y}_{t-p} + \boldsymbol{\varepsilon}_t, \tag{2.3}$$

for $t = p+1, \ldots, n$. The parameters $\phi_i \in M_{m\times m}(\mathbb{R})$, $i = 1, \ldots, p$, are $m \times m$ autoregressive coefficient matrices denoted collectively as $\boldsymbol{\Phi} \in M_{m\times m}(\mathbb{R})^p$, where $M_{m\times n}(V)$ denotes the space of $m \times n$ matrices with entries in V. The errors $\boldsymbol{\varepsilon}_t$ form a sequence of uncorrelated, zero-mean random vectors. In this thesis we only consider Gaussian errors and so the errors have a multivariate normal distribution such that $\boldsymbol{\varepsilon}_t \sim \text{N}_m(\mathbf{0}, \Sigma)$, for $\Sigma \in \mathcal{S}_m^+$ where $\mathcal{S}_m^+$ denotes the space of $m \times m$ symmetric, positive definite matrices. Using the backshift operator discussed in Section 2.2.1, it is common to express (2.3) as

$$\boldsymbol{\varepsilon}_t = (I_m - \phi_1 B - \ldots - \phi_p B^p)\boldsymbol{y}_t = \phi(B)\boldsymbol{y}_t,$$

in which I_m is the $m \times m$ identity matrix and $\phi(B)$ is the autoregressive operator. Whilst we do not focus on modelling the multivariate extension to moving average processes in this thesis, vector moving average processes, we will discuss how our methods could be extended to account for this class of models in Chapter 7.

2.2.3 Stationarity in vector autoregressions

A common assumption when working with time series data is that of stationarity, which posits that the means, variances and covariances of the process do not change over time. Since the overall level of many time series exhibits periodic or systematic variation due to seasonality or time-trends, stationarity is often implausible as an assumption when modelling raw data. However, stationary vector autoregressions frequently form the core building block of more sophisticated models, for example for differenced data in integrated models, for innovations from a time-varying mean in a time series regression model or simply as components in state space models which are thought to be mean-reverting. From a practical perspective, enforcing stationarity prevents the predictive variance of the process from growing without bound into the future. This is often keenly motivated, for instance in applications where the goal is long-term forecasting or when modelling the dynamics of a linear system which is assumed to be in its equilibrium distribution. Moreover, stationarity admits various interpretations of the relationships between variables through the infinite-order moving average representation of the process, for example in Granger causality networks which we explore further in Chapter 6.

The matrix-valued polynomial $\phi(u) = (I_m - \phi_1 u - \ldots - \phi_p u^p)$, $u \in \mathbb{C}$, which is closely related to the autoregressive operator discussed in Section 2.2.1, is referred to as the characteristic polynomial. A vector autoregression is stable if and only if all the roots of the polynomial $\det\{\phi(u)\} = 0$ lie outside the unit circle. Appendix A.1 contains a sketch proof that this condition holds, summarising the work of Luetkepohl (2005). Furthermore, Luetkepohl (2005) discuss the fact that all stable processes are stationary, and unstable stationary processes are not generally of interest. As such, this result is often referred to as the stationarity condition for Φ and the subset of $M_{m\times m}(\mathbb{R})^p$ over which the condition is satisfied is referred to as the stationary region, denoted $\mathcal{C}_{p,m}$. For univariate autoregressions, when $m = 1$, the equation $\det\{\phi(u)\} = 0$ simplifies to $\phi(u) = 0$, which is typically referred to as the characteristic equation.

2.2.4 Order determination in vector autoregressions

The order p of a vector autoregression is intrinsic to the characterisation of the joint process and plays a vital role in forecasting. Furthermore, as will be discussed further in Chapter 6, by conditioning on the model order we can obtain further insight into the underlying mechanisms of a stationary vector autoregressive process, resulting in useful insights into real-world processes. However, the order of the process is not generally known *a priori*. As such, methods of order determination are of great practical use for those modelling time series data.

The majority of previous work on determining the order of vector autoregressions uses the frequentist framework, see for example Hurvich & Tsai (1993); Kilian & Ivanov (2001); Nielsen (2006); Canova (2007); Carriero *et al.* (2015); Han *et al.* (2017). Within this framework, a common approach to determining the model order is to sequentially increase or decrease the number of parameters, and as such the model order, before using model fit criteria to choose the best model. Common choices of model fit criteria include the Akaike information criterion (AIC) (Akaike, 1974) or the Bayesian information criterion (BIC) (Schwarz, 1978). Alternatively, nested models can be compared using the likelihood ratio test (Buse, 1982). One disadvantage of these approaches is that they result in a single choice of the "best" model and do not account for the associated uncertainty in the model order. Bayesian methods which incorporate uncertainty in the model order can overcome this drawback.

In the Bayesian framework, the problem of learning the order of the full class of stationary vector autoregressions remains hitherto unaddressed in the literature. However, as will be discussed further in Chapter 4, in the special case of univariate stationary autoregressions, the problem has been widely studied (Barnett *et al.*, 1996; Vermaak *et al.*, 2004; Monahan, 1983; Huerta & West, 1999). In the Bayesian framework, the only generalisation of these ideas to stationary vector autoregressions appears in Huerta & Prado

(2006) who extend their work on univariate autoregressions in Huerta & West (1999) by considering a multivariate generalization of the characteristic equation. However, because this generalisation is only available when the autoregressive coefficient matrices are diagonal, the approach is limited to the class of diagonal vector autoregressive processes. Other recent work which addresses the problem of order determination in vector autoregressions includes Zhang *et al.* (2021) and Fan *et al.* (2022) but their focus is on classes of rank-reduced models and stationarity is not enforced. In Chapter 5 we address the problem of learning the order of the full class of stationary vector autoregressions in the Bayesian framework.

2.2.5 Latent decomposition of vector autoregressions

Conditional on the model order, p, a $\text{VAR}_m(p)$ process can be decomposed into pm^2 latent processes accounting for low frequency trends, quasi-periodic behaviour and high frequency noise contributions. As discussed in Prado (1998), these latent series correspond to the pm eigenvalues of the state evolution matrix G which arises from the representation of the model as a multivariate dynamic linear model (DLM). For an m-variate process $\{\boldsymbol{x}_t\}$, a multivariate dynamic linear model is defined in Chapter 16 of West & Harrison (1997) as

$$\begin{aligned}\boldsymbol{x}_t &= \boldsymbol{y}_t + \boldsymbol{\nu}_t \\ \boldsymbol{y}_t &= F^{\mathrm{T}}\boldsymbol{\theta}_t \\ \boldsymbol{\theta}_t &= G_t\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t\end{aligned}$$

where $\boldsymbol{y}_t$ is the underlying process, $\boldsymbol{\theta}_t$ is an unobserved $d \times 1$ state vector, F is a known $d \times m$ matrix referred to as the observation matrix and G_t is a $d \times d$ state evolution matrix. Additionally, $\boldsymbol{\nu}_t$ is an m-dimensional vector of observation errors such that $\boldsymbol{\nu}_t \sim N_m(\mathbf{0}, V_t)$ and $\boldsymbol{\omega}_t$ is a d-vector of state innovations such that $\boldsymbol{\omega}_t \sim N_d(\mathbf{0}, W_t)$ where V_t and W_t are covariance matrices. A $\text{VAR}_m(p)$ process can be written in DLM form with $d = pm$, $\boldsymbol{\nu}_t = \mathbf{0}$ and F^{T} an $m \times pm$ matrix such that

$$F^{\mathrm{T}} = \begin{pmatrix} \boldsymbol{e}_1^{\mathrm{T}} & 0 & \dots & 0 \\ \boldsymbol{e}_2^{\mathrm{T}} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{e}_m^{\mathrm{T}} & 0 & \dots & 0 \end{pmatrix},$$

where the $\boldsymbol{e}_i$ are m-vectors with a one in the i-th position and zeros elsewhere. Clearly, $\boldsymbol{\theta}_t$ and $\boldsymbol{\omega}_t$ are both pm-dimensional vectors such that

$$\boldsymbol{\theta}_t = \begin{pmatrix} \boldsymbol{y}_t \\ \boldsymbol{y}_{t-1} \\ \vdots \\ \boldsymbol{y}_{t-p+1} \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{\omega}_t = \begin{pmatrix} \boldsymbol{\varepsilon}_t \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

and the $pm \times pm$ state evolution matrix G_t is constant over time such that

$$G_t = G = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_{p-1} & \phi_p \\ I_m & 0_m & \dots & 0_m & 0_m \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0_m & 0_m & \dots & I_m & 0_m \end{pmatrix}$$

where 0_m is an $m \times m$ matrix of zeros.

Methods for decomposing univariate autoregressions are well studied (Box & Jenkins, 1976; West & Harrison, 1997; Prado, 1998). In order to decompose a multivariate DLM into its latent processes, Prado (1998) first split the m-variate multivariate DLM into m scalar DLMs, one for each of the univariate components of $\boldsymbol{y}_t$. In the vector autoregressive case, this is such that

$$y_{ti} = F_i^{\mathrm{T}} \boldsymbol{\theta}_t \tag{2.4}$$

$$\boldsymbol{\theta}_t = G\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t \tag{2.5}$$

for $i = 1, \dots, m$, where F_i is the i-th column of the observation matrix F. For each component y_{ti} the model has the same state evolution matrix G and the same state and innovation vectors, $\boldsymbol{\theta}_t$ and $\boldsymbol{\omega}_t$, with the only difference across models being the vector of constants F_i. Prado (1998) then breaks each of the univariate components into latent processes using univariate results. The $pm \times pm$ evolution matrix G can have at most pm distinct eigenvalues. In the case where the values of Φ are inferred from data, the pm eigenvalues will be distinct with probability one. Suppose that this is the case and that the pm distinct eigenvalues are made up of c complex conjugate pairs denoted $r_j e^{\pm i\omega_j}$, $j = 1, \dots, c$, and $pm - 2c$ real eigenvalues denoted r_j, $j = 2c+1, \dots, pm$ where $r_j > 0$ and $\omega_j \in [0, \pi)$. The eigendecomposition of G is then such that $G = BAB^{-1}$ where A is a diagonal matrix containing the eigenvalues of G and B is the matrix containing the corresponding eigenvectors as columns. For each scalar DLM model Prado (1998) then defines the matrix $H_i = \mathrm{diag}(B^{\mathrm{T}} F_i) B^{-1}$, for $i = 1, \dots, m$, before reparameterising the scalar DLM models via $\boldsymbol{\gamma}_{ti} = H_i \boldsymbol{\theta}_t$ and $\boldsymbol{\delta}_{ti} = H_i \boldsymbol{\omega}_t$. This results in a new parameterisation

for each scalar DLM such that

$$y_{ti} = \mathbf{1}^{\mathrm{T}}\boldsymbol{\gamma}_{ti} \tag{2.6}$$

$$\boldsymbol{\gamma}_{ti} = A\boldsymbol{\gamma}_{t-1,i} + \boldsymbol{\delta}_{ti} \tag{2.7}$$

where $\mathbf{1}$ is a pm-vector of ones. This can be easily verified; see Appendix A.2. Consequently, each y_{ti} can be written as a sum of the pm components of $\boldsymbol{\gamma}_{ti}$ such that

$$y_{ti} = \sum_{j=1}^{c} z_{tij} + \sum_{j=2c+1}^{pm} x_{tij}$$

where the z_{tij} and x_{tij} are real-valued processes corresponding to the jth pair of complex eigenvalues and the jth real eigenvalue, respectively. For each of the $pm - 2c$ real eigenvalues r_j, the relevant component of (2.7) is

$$\gamma_{tij} = r_j\gamma_{t-1,ij} + \delta_{tij}$$

for $j = 2c+1, \ldots, pm$. Renaming $\gamma_{ti,j}$ as x_{tij} this becomes

$$x_{tij} = r_j x_{t-1,ij} + \delta_{tij}.$$

Therefore, the process x_{tij} follows an AR(1) structure with coefficient r_j for all $i = 1, \ldots, m$. For each of the c pairs of complex eigenvalues, the real-valued process z_{tij} is such that $z_{tij} = \gamma_{ti,2j-1} + \gamma_{ti,2j} = 2\mathrm{Re}(\gamma_{ti,2j-1})$, for $j = 1, \ldots, c$, where $\gamma_{ti,2j-1}$ and $\gamma_{ti,2j}$ are the components of $\boldsymbol{\gamma}_{ti}$ corresponding to the j-th pair of complex eigenvalues. Then, following Prado (1998), we can consider the subset of the matrix A which corresponds to the j-th pair of complex eigenvalues,

$$A_j = \begin{pmatrix} r_j e^{i\omega_j} & 0 \\ 0 & r_j e^{-i\omega_j} \end{pmatrix},$$

alongside $\boldsymbol{\gamma}_{tij} = (\gamma_{ti,2j-1}, \gamma_{ti,2j})^{\mathrm{T}}$ and write

$$z_{tij} = \gamma_{ti,2j-1} + \gamma_{ti,2j} = (1,1) \times \boldsymbol{\gamma}_{tij}$$
$$\boldsymbol{\gamma}_{tij} = A_j\boldsymbol{\gamma}_{t-1,ij} + \boldsymbol{\delta}_{tij}$$

where $\boldsymbol{\delta}_{tij} = (\delta_{ti,2j-1}, \delta_{ti,2j})^{\mathrm{T}}$ for $j = 1, \ldots, c$. Prado (1998) then considers the associated real canonical form (West & Harrison, 1997, Chapter 5) of this dynamic linear model

which is obtained by linearly transforming $\boldsymbol{\gamma}_{tij}$ using the matrix

$$H^* = \begin{pmatrix} 1 & 1 \\ i & -i \end{pmatrix}$$

such that

$$\boldsymbol{\gamma}^*_{tij} = H^* \boldsymbol{\gamma}_{tij} = \begin{pmatrix} 2\mathrm{Re}(\gamma_{ti,2j-1}) \\ -2\mathrm{Im}(\gamma_{ti,2j-1}) \end{pmatrix}.$$

Then the real canonical form of A_j is

$$H^* A_j H^{*-1} = r_j \begin{pmatrix} \cos(\omega_j) & \sin(\omega_j) \\ -\sin(\omega_j) & \cos(\omega_j) \end{pmatrix}$$

and $F^*_j = (1,1) \times H^{*-1} = (1,0)$ so that $z_{tij} = F^*_j \boldsymbol{\gamma}^*_{tij}$. It follows from the discussion at the start of this section that a (univariate) AR(2) process can be written in DLM form with

$$G = \begin{pmatrix} \phi_1 & \phi_2 \\ 1 & 0 \end{pmatrix}.$$

The eigenvalues of the state evolution matrix G are the solution of the equation

$$\lambda^2 - \phi_1 \lambda - \phi_2 = 0. \tag{2.8}$$

Consider the case where the eigenvalues of G are complex and call the roots $\lambda = r_j e^{\pm i\omega_j}$. Then the eigenvalues of G are the same as the eigenvalues of $H^* A_j H^{*-1}$ which are also equal to $r_j e^{\pm i\omega_j}$. West & Harrison (1997) define similar dynamic linear models to be those whose state evolution matrices have the same eigenvalues. Therefore, the real canonical DLM described above and the AR(2) process whose eigenvalues are equal to $\lambda = r_j e^{\pm i\omega_j}$ are similar models with similar behaviour. If the solutions to (2.8) are $\lambda = r_j e^{\pm i\omega_j}$ then

$$\begin{aligned} &(\lambda - r_j e^{i\omega_j})(\lambda - r_j e^{-i\omega_j}) = 0 \\ \implies &\lambda^2 - 2r_j \cos(\omega_j)\lambda + r_j^2 = 0. \end{aligned} \tag{2.9}$$

Comparing coefficients in (2.9) to (2.8), the real canonical DLM representation of z_{tij} is similar to an AR(2) model with autoregressive coefficients $2r_j \cos\omega_j$ and $-r_j^2$ and the process z_{tij} therefore has behaviour which is quasi-periodic with characteristic frequency ω_j and modulus r_j. This follows immediately from the form of the autocorrelation function of an AR(2) process whose characteristic equation has complex roots; see, for example, Box & Jenkins (1976). This holds for all dimensions $i = 1, \ldots, m$, though the time-varying amplitude and phase are different for each i. As the δ_{tij} are correlated, arising from a

transformation of the error terms $\boldsymbol{\varepsilon}_t$ in the original model, the innovations that drive the z_{tij} and x_{tij} processes are also correlated.

Chapter 3

Electroencephalography (EEG) data

3.1 Background

All stationary Gaussian processes can be arbitrarily well approximated by vector autoregressive-moving-average (VARMA) models of increasing order, making VARMA models a flexible class of models for modelling multivariate time series data. Furthermore, all VARMA models can be approximated by higher order vector autoregressions. This flexibility has led to vector autoregressions being successfully used to model multivariate time series data in a variety of fields, including neuroscience, as discussed in Section 2.2.2. In particular, within the field of neuroscience vector autoregressions have frequently been used to model multivariate electroencephalography (EEG) data, for example in Herrera *et al.* (1997); Goyal & Garg (2020); Malinovskaia (2022). These examples all use the frequentist framework and it has proven difficult to find previous examples which use Bayesian vector autoregressions to model multivariate EEG data. However, previous examples which used Bayesian autoregressions to model EEG data in the univariate case were more common Prado & West (1997); Prado (1998); Prado & Huerta (2002).

As an example application, we will apply the models and inferential procedures that we discuss in this thesis to a dataset of long-term intracranial electroencephalography (EEG) recordings with an aim to improve understanding of biological rhythms in the brain. Periodic biological rhythms on ultradian (sub-daily), circadian (daily), and longer timescales have been demonstrated in human physiology but particularly the ultradian rhythms remain elusive in mechanism and function in the brain (Goh *et al.*, 2019; Lloyd & Stupfel, 1991). Multiple lines of evidence suggest that some prominent ultradian rhythms exist in brain activity as measured by EEG (Hayashi *et al.*, 1994; Panagiotopoulou *et al.*, 2022), and may be related to rest-activity cycles, or even modulate disease symptoms.

In later chapters we aim to use the methods developed in this thesis to investigate the properties that such ultradian biological rhythms may display in human brain activity. In analysis of EEG data, it is common to first decompose the EEG signal into frequency bands such as delta ($\delta: \ 1-4$ Hz), theta ($\theta: \ 4-8$ Hz), alpha ($\alpha: \ 8-13$ Hz), beta ($\beta: \ 13-30$ Hz) and gamma ($\gamma: \ 30-80$ Hz). From this, the bandpower for each frequency band can be calculated, which is a measure of the contribution of that frequency to the overall signal. We use band power in two frequency bands (delta and beta) as our features of interest. As an example of what these frequency bands represent, a person in deep sleep is likely to have strong activity in the delta band whereas a person holding a conversation is likely to have stronger activity in the beta band.

3.2 Data preprocessing

The data was provided to us by the Computational Neurology, Neuroscience and Psychiatry (CNNP) lab at Newcastle University, having already been preprocessed. The full dataset is discussed in Wang *et al.* (2023) and consists of intracranial EEG recordings from 39 individuals with refractory focal epilepsy from the University College London Hospital (UCLH). The nature of the recording was chosen for its high signal-to-noise ratio without the need for extensive artefact detection and removal.

The preprocessing steps were as follows. Firstly, each subject's EEG data were divided into non-overlapping, consecutive segments of length 30 seconds. All channels within each segment were re-referenced to a common average reference. In the common average calculation, channels with extreme amplitude values were excluded. A notch filter (Tibdewal *et al.*, 2016) was then applied at 50 Hz for each 30 second time window to remove power line noise, after which the time windows were band-pass filtered from $0.5-80$ Hz using a Butterworth filter (Sen *et al.*, 2023) so that only frequencies in the frequency bands of interest (delta, theta, alpha, beta, gamma) were retained. Finally, the data was downsampled to 200 Hz.

Next, the EEG data were decomposed into commonly studied frequency bands. In particular, the EEG band power was calculated for each 30 second segment for all channels in two frequency bands ($\delta: \ 1-4$ Hz, $\beta: \ 13-30$ Hz) using Welch's method (Chiu *et al.*, 2023) with three-second non-overlapping windows. After taking logarithms to base 10 of the band power recordings in each channel, the channels were averaged into the brain regions from which they were recorded based on the Desikan-Killiany atlas (Desikan *et al.*, 2006) which is used to segment the brain into regions, similar to splitting the United Kingdom into counties.

In this thesis we consider a subset of the data consisting of four individuals which we give the anonymous identities of A, B, C and D. The four individuals were chosen as they

had data recorded in fewer than 20 regions of the brain which is a reasonable number of regions m for the methods we develop in Chapter 5 to handle. The number of brain regions m varied between individuals, with $m = 9$, 8, 8 and 13 for individuals A, B, C and D respectively. Table 3.1 summarises which brain regions recordings were taken from for each individual. The locations of these regions in the brain are depicted in Figure 3.1.

The full data sets were recorded over a number of days, with the exact length of the recording varying according to individual. The full recordings cover a period of 78.4, 168.1, 167.8 and 45.9 hours for individuals A, B, C and D respectively. However, large chunks of the data are missing and stationarity is not a plausible assumption over the whole timeline. For example, Figure 3.2 contains a plot of the data in the delta band from individual A over the full recorded timeframe of 78.4 hours, which consist of 9412 observations at 30 second intervals. We can see that the data are not stationary over the whole time period, with at least two regimes, with different mean and variance, clearly detectable by eye. To emphasise this point, we fit a Bayesian hidden Markov model to the data from the delta band from individual A; see, for example, Frühwirth-Schnatter (2006) for an introduction to hidden Markov models. Details of this model and the Stan programme used to fit it can be found in Appendices B and C.1 respectively. Figure 3.3 depicts the data coloured by the posterior modal state obtained when fitting this hidden Markov model. Note that whilst we permitted up to four hidden states, only two states appear as modal states in the posterior inference.

Since the work in this thesis focuses on determining the order of stationary vector autoregressions, we clearly cannot directly model the whole data set. However, when discussing future work in Chapter 7 we outline how our methods could be extended for data which are locally stationary, but where there are the types of reversible regime shifts that we see here.

As we cannot model the data over the full time period, for each individual we will instead analyse the longest possible contiguous time period of their band power time-series for which graphical interrogation of the data suggests stationarity is a plausible assumption. The length of the recording chosen for further analysis therefore varies across subjects. The number of observations in the recordings used were $n = 651$, 622, 685 and 231 for individuals A, B, C and D respectively, equivalent to 5.417, 5.175, 5.7 and 1.917 hours. These recordings were obtained during day-time hours. It is worth noting that we did not base our choice of stationary regions on the assigned states in the hidden Markov analysis depicted in Figure 3.3 as this simple hidden Markov model assumed the observations were conditionally independent given the state. As such, it did not account for the dependence between consecutive observations that we would expect to see in a vector autoregressive process, and so the hidden Markov model states may not be equivalent to stationary regions in a vector autoregressive process. Consequently, we made a choice of

Individual	Region	Region name
A	1	left hippocampus
	2	left amygdala
	3	right hippocampus
	4	right amygdala
	5	left inferiortemporal
	6	left fusiform
	7	left middletemporal
	8	left superiortemporal
	9	right middletemporal
B	1	right putamen
	2	right hippocampus
	3	right amygdala
	4	right lateralorbitofrontal
	5	right middletemporal
	6	right temporalpole
	7	right superiortemporal
	8	right insula
C	1	right bankssts
	2	right middletemporal
	3	right postcentral
	4	right superiortemporal
	5	right supramarginal
	6	right inferiorparietal
	7	right inferiortemporal
	8	right precentral
D	1	right bankssts
	2	right caudalmiddlefrontal
	3	right fusiform
	4	right inferiorparietal
	5	right lateraloccipital
	6	right lingual
	7	right middletemporal
	8	right superiortemporal
	9	right supramarginal
	10	right inferiortemporal
	11	right postcentral
	12	right precentral
	13	right superiorparietal

Table 3.1: Names of the regions where recordings were taken for individuals A, B, C and D. The locations of the regions are depicted in Figure 3.1.

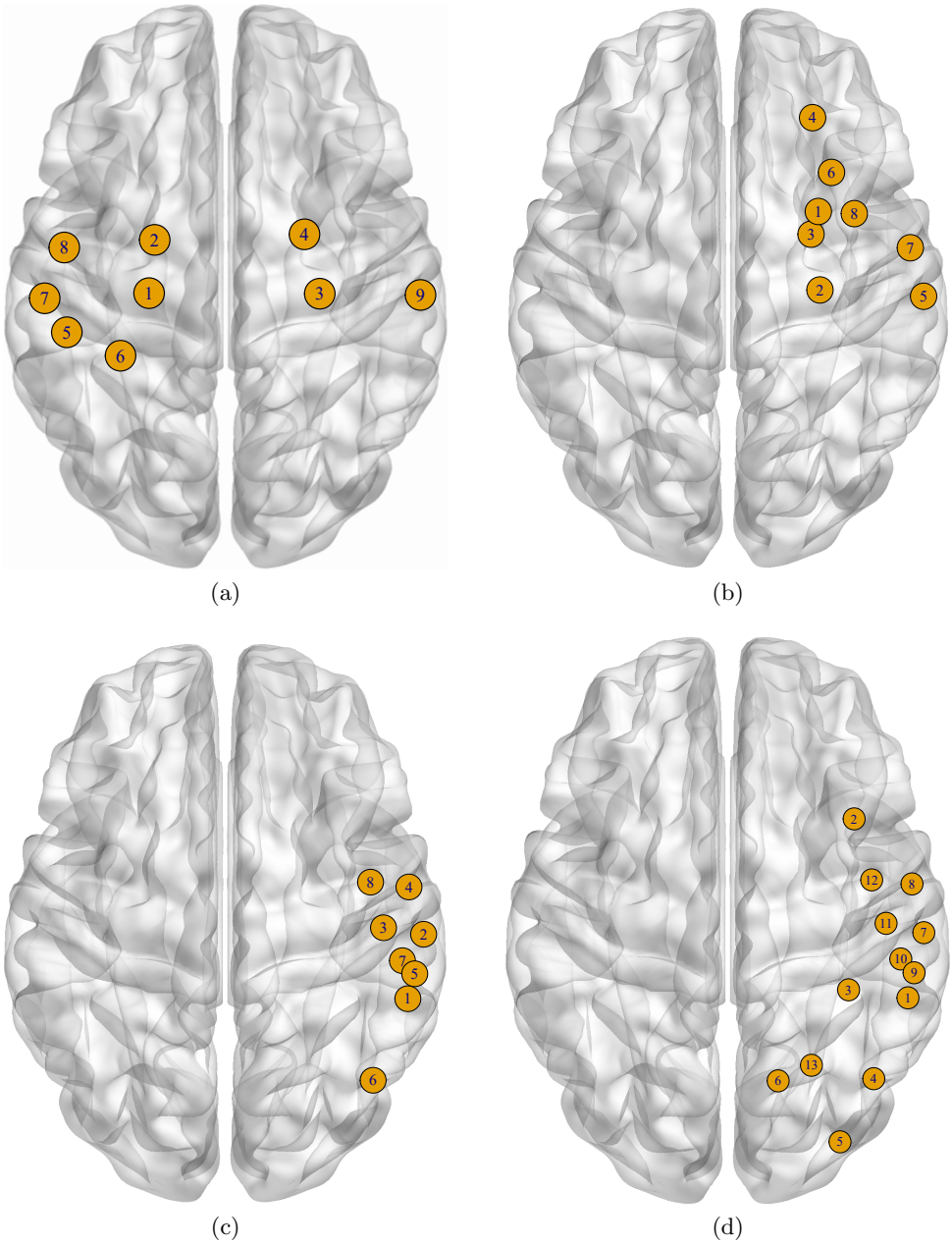


Figure 3.1: Glass brains showing the locations of the regions where recordings were taken for individuals (a) A, (b) B, (c) C and (d) D. The corresponding region names for each individual are detailed in Table 3.1.

which segments we deemed to be stationary independently of the results from the hidden Markov model. After mean-centering the data, Figure 3.4 contains a plot of the segment of data chosen for analysis in the delta band of individual A. A summary of the number of regions m and the length n of the data used for analysis for each individual is given in

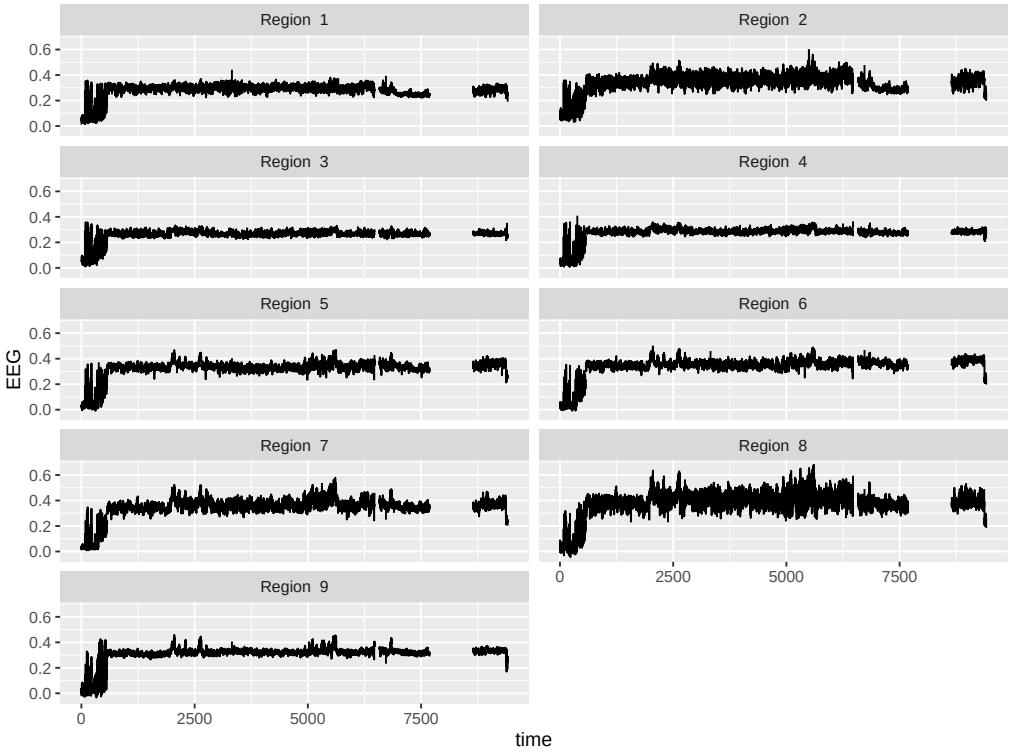


Figure 3.2: Time series plots of the full EEG recordings in each region of the brain for individual A in the delta band.

Individual	m	n
A	9	651
B	8	622
C	8	685
D	13	231

Table 3.2: Summary of the number of regions m and length n of the time series used for analysis for each individual considered in this thesis.

Table 3.2.

3.3 Box-Jenkins approach to model fitting

Before considering Bayesian approaches to determining the order of stationary autoregressions, we first explore the dependence structure in the data using classical methods for determining the order of univariate ARMA models. In particular, treating each region in each frequency band for each individual as univariate data, we take an iterative approach to the Box-Jenkins method for modelling time series data (Box & Jenkins, 1976)

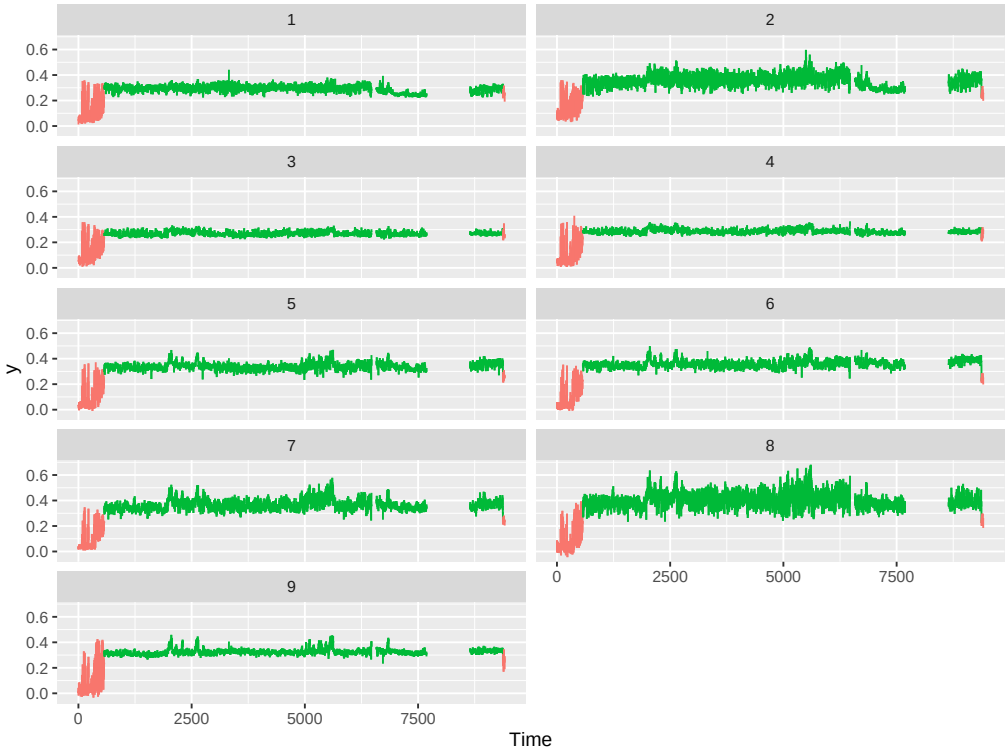


Figure 3.3: Time series plots of the full EEG recordings in each region of the brain for individual A in the delta band, coloured by the posterior model state obtained when fitting a Bayesian hidden Markov model to the data which permitted up to four hidden states.

to identify a suitable model.

Consider, initially, the data from the delta band in individual A. There are EEG recordings from 9 regions in the brain for this individual. For each region, we aim to fit a univariate time series model to the data. To identify an initial model that could be plausible for each region, we look at correlograms of the sample autocorrelation and partial autocorrelation functions. If the autocorrelation function tails off slowly and the partial autocorrelation function cuts off after lag p, then an autoregressive model of order p can be taken as an initial choice of model. On the other hand, if the partial autocorrelation function tails off slowly and the autocorrelation function cuts off after lag q then a moving average model of order q can be taken as the initial choice of model. If both the autocorrelation and partial autocorrelation functions tail off slowly then the model is likely to have both an autoregressive and a moving average component. Figures 3.5 and 3.6 contain the sample autocorrelation functions and the sample partial autocorrelation functions respectively, for the data from each region for the delta band in individual A. In all regions, the sample autocorrelation function tails off slowly, whereas the sample partial autocorrelation functions appear to cut off more abruptly, suggesting that autoregressive

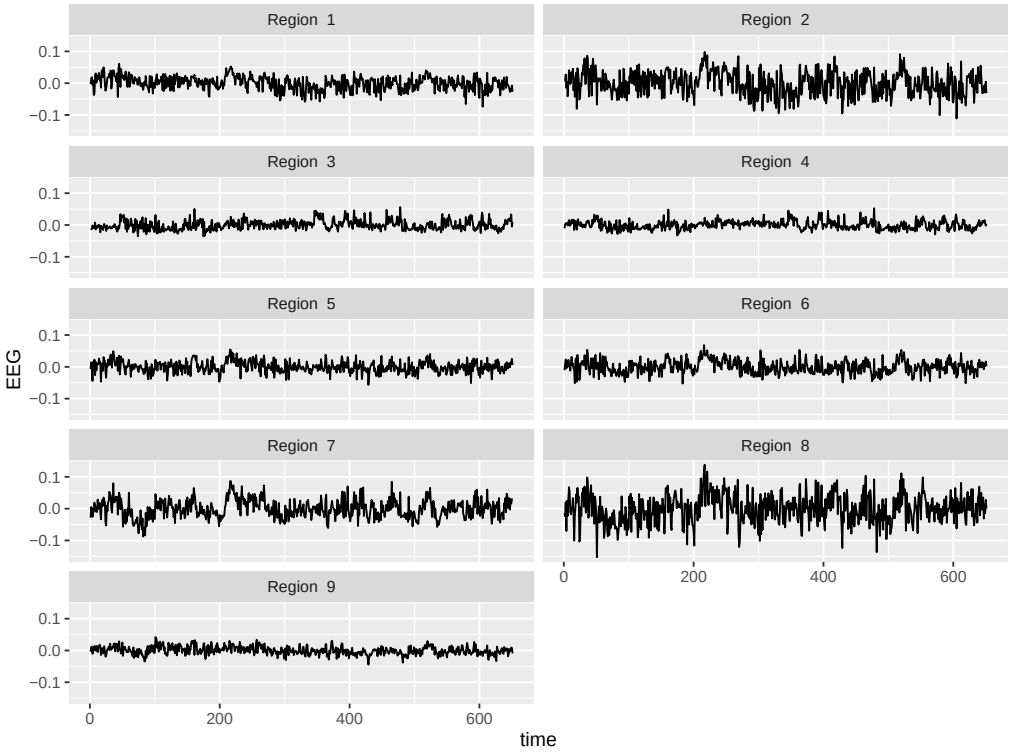


Figure 3.4: Time series plots of the portion of the EEG recordings chosen for analysis in each region of the brain for individual A in the delta band. This is the segment corresponding to time points 3500 to 4150 in Figure 3.2

.

models may be a good fit to the data. By considering the lag k after which the partial autocorrelation function cuts off, we identified initial models for the data in each region. We then sequentially added and removed autoregressive and moving average components, using significance tests and the Akaike information criterion (AIC) to compare the models in order to investigate whether the more complex model in each comparison provided an improved model fit. These iterative steps led to a final choice of model for each region, with these choices detailed in Table 3.3. This approach to model fitting was repeated with all data sets. The results are omitted for brevity but can be found in Appendix D.

Having identified suitable models for the data in each region, we can then investigate the behaviour of the model residuals. Figure 3.7 contains a pairs plot of the residuals from each region obtained from fitting the models in Table 3.3. Clearly, the residuals from the different regions are positively correlated, most notably those in the nearby left hippocampus and amygdala (regions 1 and 2) and those in the nearby right hippocampus and amygdala (regions 3 and 4). This is similar across all data sets, with the results for the other data sets presented in Appendix D. As such, it is not sensible to model the

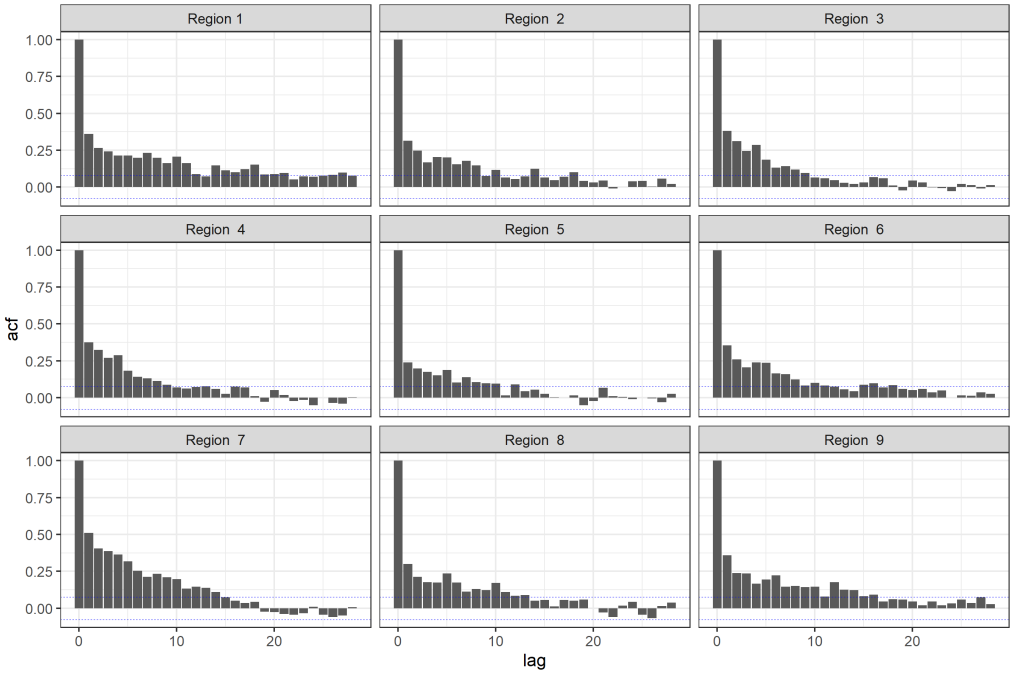


Figure 3.5: Plots of the sample autocorrelation function for each region for the delta band in individual A.

Region	Chosen model
1	AR(7)
2	AR(7)
3	AR(4)
4	AR(4)
5	AR(5)
6	AR(5)
7	AR(4)
8	AR(5)
9	AR(6)

Table 3.3: Models chosen for each region for the delta band in individual A using an iterative approach to the Box-Jenkins method.

data from the different regions independently, as we have done here. Instead, to better quantify our uncertainty on rhythms in brain activity, it is preferable to model the data as a multivariate time series. As the aim of this thesis is to develop Bayesian methods to determine the order of stationary vector autoregressive processes, we hope to use our methods to gain some interesting insight into these multivariate data.

Whilst all chosen models for the regions in the delta band of individual A are pure autoregressive processes, for some of the other individuals, models with moving average

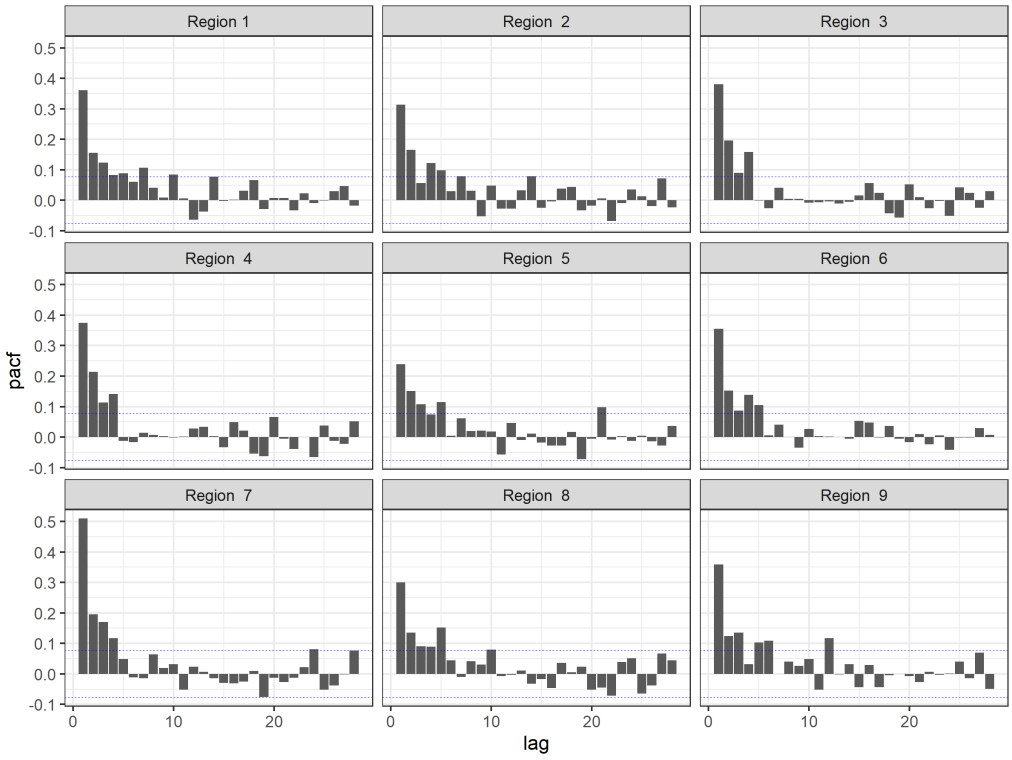


Figure 3.6: Plots of the sample partial autocorrelation function for each region for the delta band in individual A.

components were selected. However, for the remainder of this thesis we focus on autoregressive processes as the problem we aim to solve is difficult even without considering both components. Moreover, since all autoregressive-moving-average models can be approximated by a higher order autoregressive process, we do not regard this restriction as being limiting. Nevertheless, we discuss the extension to vector autoregressive-moving-average models in Chapter 7.

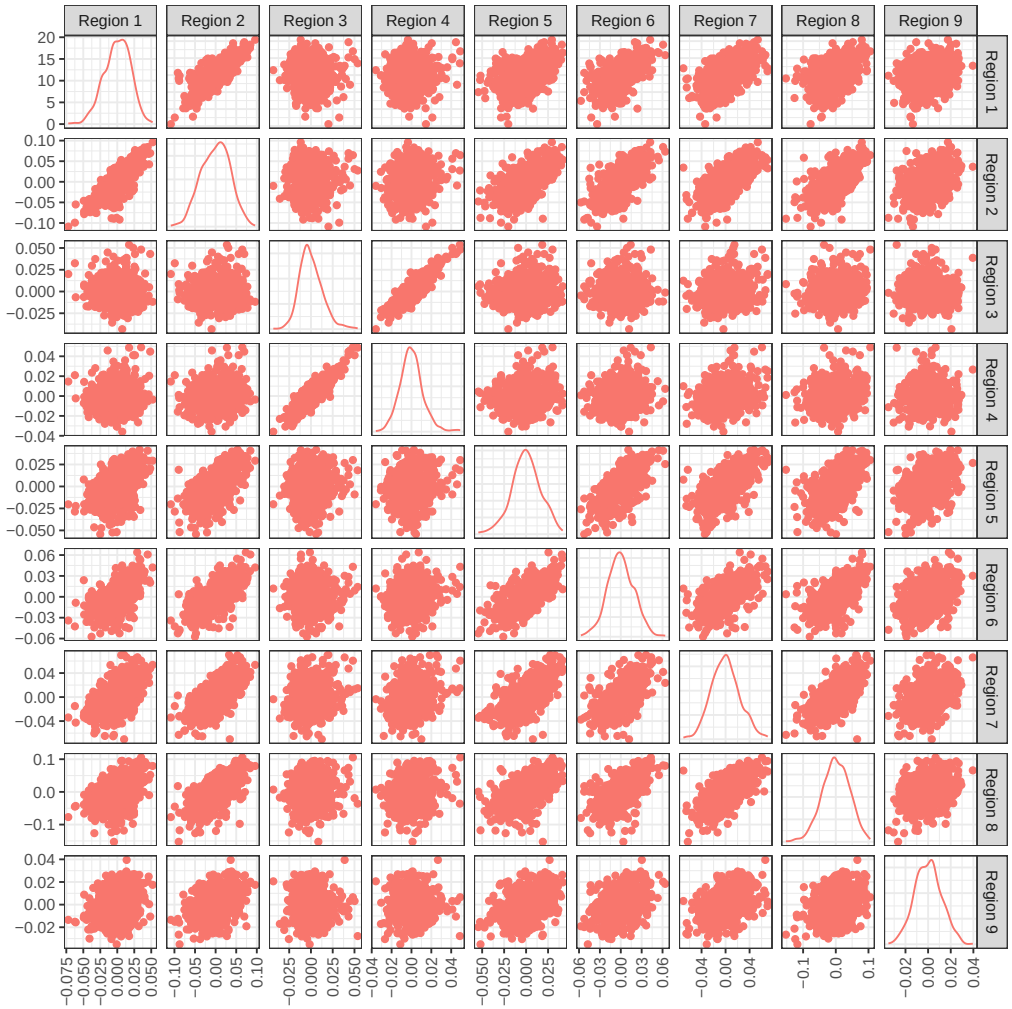


Figure 3.7: Pairs plot of the residuals obtained from each region after fitting the models in Table 3.3 for the delta band in individual A.

Chapter 4

Modelling stationary univariate autoregressions

Whilst the main objective of this thesis is to develop methods for determining the order of stationary vector autoregressive processes, in this chapter we first consider the simpler case of learning the order of univariate stationary autoregressions, which is already well documented in the literature (Barnett *et al.*, 1996; Huerta & West, 1999; Vermaak *et al.*, 2004). Initially, we discuss methods for enforcing stationarity when the model order is known, before discussing methods which allow for uncertainty in the model order.

4.1 Enforcing stationarity when p is known

In Section 2.2.3 we discussed how a Gaussian autoregressive process is stationary if and only if the roots of the characteristic equation

$$\phi(u) = 0$$

lie outside the unit circle. Values of the autoregressive parameters, $\phi_1, \ldots, \phi_p$, which satisfy this condition lie in a region referred to as the stationary region, denoted $\mathcal{C}_{p,m}$, where $m = 1$ in the univariate case. When $p = 1$, the region $\mathcal{C}_{1,1}$ is simply the interval (-1,1) and when $p = 2$ the region $\mathcal{C}_{2,1}$ is a triangle in the (ϕ_1, ϕ_2) plane. However, as p increases this region has an increasingly complex geometry. As an example, Figure 4.1 contains samples from a diffuse distribution over the stationary region for the autoregressive parameters ϕ_1, ϕ_2 and ϕ_3 in an AR(3) process. As p increases and the stationary region becomes more complex it becomes increasingly difficult to specify meaningful prior distributions for $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_p)$ whose support is constrained to this space. As such, a common approach in the literature is to consider reparameterising the autoregressive model in terms of a new set of parameters, for which the conditions of stationarity are easier to impose.

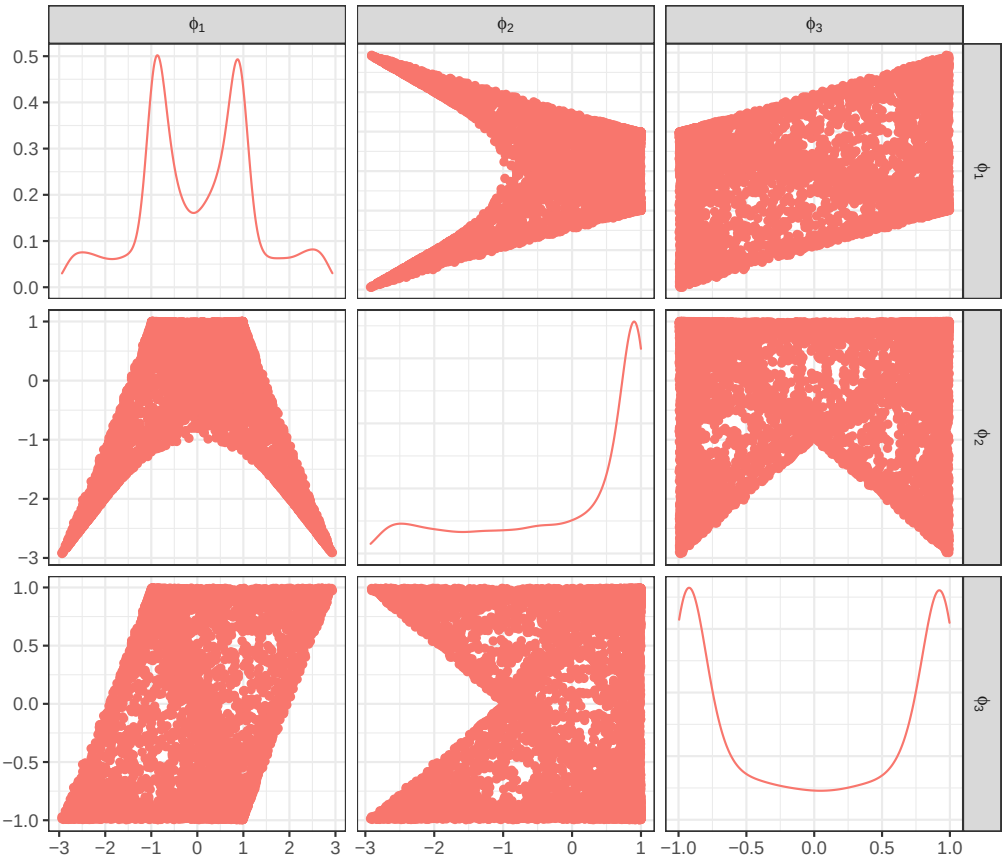


Figure 4.1: Draws from a diffuse distribution over the stationary region for ϕ_1, ϕ_2 and ϕ_3 in an AR(3) model. Plots along the diagonal show marginal densities, but of interest here are the plots off the diagonal which depict the bivariate densities for the pairs of parameters.

One such reparameterisation, defined by Barndorff-Nielsen & Schou (1973), is a mapping between the autoregressive parameters and the partial autocorrelations, where the $(s+1)$-th partial autocorrelation, ρ_{s+1}, is defined as a conditional correlation between y_{t+1} and y_{t-s} given $y_t, \ldots, y_{t-s+1}$. In the remainder of this section we define the mapping between $\boldsymbol{\Phi} = (\phi_1, \ldots, \phi_p)$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_p)$ and its inverse. We note that although versions of these algorithms exist which do not compute the autocovariance function $\gamma_i = \text{Cov}(y_t, y_{t+i})$ (see, for example, Marriott *et al.*, 1996) we present algorithms that calculate the γ_i for two reasons. First because we need the autocovariance function to calculate the marginal distribution of the initial term in the likelihood and second because these forms of the algorithms generalise readily to the vector case in Chapter 5.

The forward mapping from $(\boldsymbol{\Phi}, \sigma^2)$ to $(\boldsymbol{\rho}, \sigma^2)$ is as follows:

1. For $i = 0, \ldots, p$ compute the autocovariances $\gamma_i = \text{Cov}(y_t, y_{t+i})$ from $(\boldsymbol{\Phi}, \sigma^2)$. The autocovariances $\gamma_0, \ldots, \gamma_{p-1}$ can be found by representing the AR(p) process as

a $\text{VAR}_p(1)$ process, with an autoregressive matrix denoted ϕ and error variance matrix denoted Σ, then computing its stationary variance Γ_0. This results in a discrete Lyapunov equation (an equation of the form $\Gamma_0 = \phi\Gamma_0\phi^{\mathrm{T}} + \Sigma$) which can be solved using vectorisation and Kronecker product operators. The autocovariances $\gamma_0, \ldots, \gamma_{p-1}$ are the elements in the first column of Γ_0. Then, γ_p can be found using the Yule-Walker equations for the order p process ($\gamma_p = \phi_1\gamma_{p-1} + \ldots + \phi_1\gamma_0$). Further details on such calculations can be found in Chapter 2 of Luetkepohl (2005).

2. From $\boldsymbol{\phi}$ and $(\gamma_1, \ldots, \gamma_p)$ compute the partial autocorrelations $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_p)$ using a recursion: for each $s = 0, \ldots, p-1$

   (a) Compute $\phi_{s+1,s+1}$ using

   $$\phi_{s+1,s+1} = \frac{\gamma_{s+1} - \phi_{s1}\gamma_s - \cdots - \phi_{ss}\gamma_1}{\gamma_0 - \phi_{s1}\gamma_1 - \cdots - \phi_{ss}\gamma_s}$$

   which simplifies to $\phi_{11} = \gamma_1/\gamma_0$ when $s = 0$.

   (b) If $s > 0$, for $i = 1, \ldots, s$ compute $\phi_{s+1,i}$ using

   $$\phi_{s+1,i} = \phi_{si} - \phi_{s+1,s+1}\phi_{s,s-i+1}.$$

   (c) Compute the $(s+1)$-th partial autocorrelation ρ_{s+1} using

   $$\rho_{s+1} = \phi_{s+1,s+1}.$$

Then the following algorithm performs the reverse mapping from $(\boldsymbol{\rho}, \sigma^2)$ to (Φ, σ^2):

1. From σ^2 and $\boldsymbol{\rho}$ compute the stationary variance γ_0:

   (a) Initialise: let $\sigma_p^2 = \sigma^2$.

   (b) Recursion: for $s = p-1, \ldots, 0$, compute σ_s^2 using

   $$\sigma_s^2 = \frac{\sigma_{s+1}^2}{1 - \rho_{s+1}^2}.$$

   (c) Output: take $\gamma_0 = \sigma_0^2$.

2. From $\boldsymbol{\rho}$ compute the autoregressive coefficients.

   (a) Initialise: let $\sigma_0^2 = \gamma_0$.

   (b) Recursion: for each $s = 0, \ldots, p-1$,

      i. Compute $\phi_{s+1,s+1}$ using

      $$\phi_{s+1,s+1} = \rho_{s+1}.$$

ii. If $s > 0$, for $i = 1, \ldots, s$ compute $\phi_{s+1,i}$ using

$$\phi_{s+1,i} = \phi_{si} - \phi_{s+1,s+1}\phi_{s,s-i+1}.$$

iii. Compute σ^2_{s+1} using

$$\sigma^2_{s+1} = \sigma^2_s(1 - \phi^2_{s+1,s+1}).$$

iv. Compute γ_{s+1} using

$$\gamma_{s+1} = \phi_{s+1,s+1}\sigma^2_s + \phi_{s1}\gamma_s + \cdots + \phi_{ss}\gamma_1. \tag{4.1}$$

(c) Output: take $\phi_i = \phi_{pi}$ for $i = 1, \ldots, p$.

As mentioned previously, this reverse mapping also computes the autocovariance function. If only the autoregressive coefficients were required, steps 1, 2a, 2b(iii) and 2b(iv) could be omitted. Proofs of these mappings are included in Appendix A.3. These proofs are simplified versions of proofs provided in the Supplementary Materials of Heaps (2023), which prove the more complicated mapping from the parameters in a vector autoregression to a set of partial autocorrelation matrices, discussed further in Section 5.1.

A benefit of this reparameterisation is that the new set of parameters is interpretable as the partial autocorrelation function is a widely used tool in classical time series analysis. Furthermore, as discussed in Barndorff-Nielsen & Schou (1973), under this reparameterisation stationarity can be enforced by ensuring that all partial autocorrelations lie in the interval (-1,1), which gives simpler model constraints that are more easily accommodated in computational Bayesian inference. As such, specifying prior distributions for each ρ_s, $s = 1, \ldots, p$, with support in (-1,1) enforces model stationarity. We take this approach when considering Bayesian inference of the parameters in a stationary AR(p) process, for a known p.

4.1.1 Prior distribution

The unknown model parameters under the partial autocorrelation reparameterisation consist of the partial autocorrelations $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_p)$ and the error variance σ^2. Giving each ρ_s, $s = 1, \ldots, p$, an independent prior distribution, the overall prior distribution is of the form

$$\pi(\rho_1, \ldots, \rho_p, \sigma^2) = \pi(\sigma^2) \prod_{s=1}^{p} \pi(\rho_s).$$

For each of the partial autocorrelations, $\rho_1, \ldots, \rho_p$, we need to specify a prior distribution with support over the interval (-1,1). One option which satisfies this requirement is

the stretched beta distribution, which can be obtained by transforming the beta distribution to stretch its range of support from the interval (0,1) to the interval (-1,1). For each partial autocorrelation, $\rho_s \in (-1, 1)$, let

$$\tilde{\rho}_s = (\rho_s + 1)/2 \in (0, 1).$$

Then we take our prior distribution to be such that

$$\tilde{\rho}_s \sim \text{Beta}(a_{\rho_s}, b_{\rho_s}) \tag{4.2}$$

independently for $s = 1, \ldots, p$, which ensures that each ρ_s satisfies the stationarity condition and lies in the interval (-1,1). The hyperparameters a_{ρ_s} and b_{ρ_s} are to be chosen prior to analysis.

The prior distribution for σ^2 must have support on the positive real line. A popular choice of prior distribution for variance parameters which has support in this range is the inverse gamma distribution. As such, we specify a prior for σ^2 such that

$$\sigma^2 \sim \text{IG}(a_{\sigma^2}, b_{\sigma^2}) \tag{4.3}$$

where a_{σ^2} and b_{σ^2} are hyperparameters to be chosen prior to carrying out any analysis.

4.1.2 Posterior inference

For $i \leq j$, denote by $y_{i:j}$ the time series $y_i, \ldots, y_j$. The likelihood for a series of n observations, $y_{1:n}$, from a zero-mean AR(p) process can be expressed as

$$p(y_{1:n} \mid \sigma^2, \Phi) = p(y_{1:p} \mid \sigma^2, \Phi) \prod_{t=p+1}^{n} p(y_t \mid y_{(t-p):(t-1)}, \sigma^2, \Phi) \tag{4.4}$$

in which

$$Y_t \mid y_{(t-p):(t-1)}, \sigma^2, \Phi \sim \text{N}\left(\sum_{i=1}^{p} \phi_i y_{t-i}\,,\, \sigma^2\right)$$

and the initial distribution is

$$(Y_1, \ldots, Y_p)^{\text{T}} \mid \sigma^2, \Phi \sim \text{N}_p(\mathbf{0}, G).$$

Here G is given by

$$G = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{p-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{p-1} & \gamma_{p-2} & \cdots & \gamma_0 \end{pmatrix},$$

where the autocovariances $\gamma_0, \ldots, \gamma_{p-1}$ are available as by-products of the recursive mapping between the partial autocorrelations and the autoregressive parameters, see Equation (4.1).

Regarding the likelihood as a function of the partial autocorrelations and combining it with the prior via Bayes' theorem gives the posterior distribution as

$$\pi(\sigma^2, \boldsymbol{\rho} \mid y_{1:n}) \propto p(y_{1:n} \mid \sigma^2, \boldsymbol{\rho}) \pi(\sigma^2) \prod_{s=1}^{p} \pi(\rho_s). \tag{4.5}$$

Whilst the posterior distribution is not analytically tractable, the full conditional distributions for the parameters are easily obtainable and so we can use a Metropolis-within-Gibbs algorithm to sample from the posterior densities for these parameters. The full conditional distributions and proposal distributions for the parameters are described below, with the Metropolis-within-Gibbs algorithm described in Algorithm 5.

Full conditional for $\tilde{\rho}_s$

Let $\tilde{\boldsymbol{\rho}}_{-s} = (\tilde{\rho}_1, \ldots, \tilde{\rho}_{s-1}, \tilde{\rho}_{s+1}, \ldots, \tilde{\rho}_p)$. For each $\tilde{\rho}_s$, $s = 1, \ldots, p$, the full conditional distribution is

$$\begin{aligned}
\pi(\tilde{\rho}_s | y_{1:n}, \tilde{\boldsymbol{\rho}}_{-s}, \sigma^2) &\propto p(y_{1:n} | \tilde{\boldsymbol{\rho}}, \sigma^2) \pi(\tilde{\rho}_s) \\
&\propto \det(G)^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2} (y_{1:p}^T G^{-1} y_{1:p}) \right\} \\
&\quad \times \exp\left\{ -\frac{1}{2\sigma^2} \sum_{t=p+1}^{n} (y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2 \right\} \\
&\quad \times \tilde{\rho}_s^{a_{\rho_s}-1} (1 - \tilde{\rho}_s)^{b_{\rho_s}-1}
\end{aligned} \tag{4.6}$$

where the parameters $\phi_1, \ldots, \phi_p$ and G are a complicated function of $\tilde{\rho}_1, \ldots, \tilde{\rho}_p$ and σ^2 resulting from the recursive mapping between $(\boldsymbol{\rho}, \sigma^2)$ and (Φ, σ^2). A Metropolis-Hastings step is used to sample the scaled partial autocorrelations with acceptance probability

$$\alpha(\tilde{\rho}_s^*, \tilde{\rho}_s) = \min\left\{ 1, \frac{\pi(\tilde{\rho}_s^* | y_{1:n}, \tilde{\boldsymbol{\rho}}_{-s}, \sigma^2) q(\tilde{\rho}_s | \tilde{\rho}_s^*)}{\pi(\tilde{\rho}_s | y_{1:n}, \tilde{\boldsymbol{\rho}}_{-s}, \sigma^2) q(\tilde{\rho}_s^* | \tilde{\rho}_s)} \right\}$$

where $q(\cdot|\cdot)$ is a proposal distribution. As $\tilde{\rho}_s$ must lie in the interval (0,1), we choose a beta distribution for the proposal such that

$$\tilde{\rho}_s^* | \tilde{\rho}_s \sim \text{Beta}\{w_d \tilde{\rho}_s + \varepsilon, w_d(1 - \tilde{\rho}_s) + \varepsilon\}. \tag{4.7}$$

This has mean given by

$$\frac{w_d\tilde{\rho}_s + \varepsilon}{w_d + 2\varepsilon}$$

and variance

$$\frac{(w_d\tilde{\rho}_s + \varepsilon)\{w_d(1-\tilde{\rho}_s) + \varepsilon\}}{(w_d + 2\varepsilon)^2(w_d + 2\varepsilon + 1)},$$

where ε and w_d are tuning parameters. If $\varepsilon = 0$, the mean of the proposal would be the current value of $\tilde{\rho}_s$. However, if the current value of $\tilde{\rho}_s$ is either zero or one, and $\varepsilon = 0$ then the mean of the proposal distribution will also be zero or one respectively with variance equal to zero. This means that the sampler can get stuck proposing values of zero or one if $\varepsilon = 0$. Therefore, the purpose of the parameter ε is to slightly move the proposal mean away from zero or one in these cases and ensure that the variance is not equal to zero. The parameter w_d is used to control the proposal variance. The variance of the proposal distribution is quadratic in w_d in the numerator and cubic in w_d in the denominator. Therefore, increasing w_d decreases the variance and results in more proposed values being accepted.

Full conditional for σ^2

For σ^2 the full conditional distribution is:

$$\begin{aligned}
\pi(\sigma^2|y_{1:n}, \boldsymbol{\rho}) &\propto \pi(\sigma^2)p(y_{1:n}|\boldsymbol{\rho}, \sigma^2) \\
&\propto (\sigma^2)^{-a_{\sigma^2}-1} \exp\left(-\frac{b_{\sigma^2}}{\sigma^2}\right) \\
&\quad \times \det(G)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y_{1:p}^T G^{-1} y_{1:p})\right\} \\
&\quad \times (\sigma^2)^{-\frac{1}{2}(n-p)} \exp\left\{-\frac{1}{2\sigma^2}\sum_{t=p+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2\right\}. \qquad (4.8)
\end{aligned}$$

A Metropolis-Hastings step is used to sample σ^2 with acceptance probability

$$\alpha(\sigma^{2^*}, \sigma^2) = \min\left\{1, \frac{\pi(\sigma^{2^*}|y_{1:n}, \boldsymbol{\rho})q(\sigma^2|\sigma^{2^*})}{\pi(\sigma^2|y_{1:n}, \boldsymbol{\rho})q(\sigma^{2^*}|\sigma^2)}\right\}$$

where $q(\cdot|\cdot)$ is a proposal distribution. When choosing a proposal distribution for σ^2, it can be noticed that ignoring the initial p values of the time series results in an inverse

gamma full conditional distribution for σ^2. This can be shown as follows:

$$\begin{aligned}
\pi(\sigma^2|y_{(p+1):n}, \boldsymbol{\rho}) &\propto \pi(\sigma^2)p(y_{(p+1):n}|\boldsymbol{\rho}, \sigma^2) \\
&\propto (\sigma^2)^{-a_{\sigma^2}-1} \exp\left(-\frac{b_{\sigma^2}}{\sigma^2}\right) \\
&\quad \times (\sigma^2)^{-\frac{1}{2}(n-p)} \exp\left\{-\frac{1}{2\sigma^2}\sum_{t=p+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2\right\} \\
&= (\sigma^2)^{-a_{\sigma^2}-\frac{(n-p)}{2}-1} \exp\left\{-\frac{b_{\sigma^2}}{\sigma^2} - \frac{1}{2\sigma^2}\sum_{t=p+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2\right\} \\
&= (\sigma^2)^{-a_{\sigma^2}-\frac{(n-p)}{2}-1} \exp\left[-\frac{1}{2\sigma^2}\left\{2b_{\sigma^2} + \sum_{t=p+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2\right\}\right]
\end{aligned}$$

which gives

$$\sigma^2 \mid y_{(p+1):n}, \boldsymbol{\rho} \sim \text{IG}\left[a_{\sigma^2} + \frac{n-p}{2}, \frac{1}{2}\left\{2b_{\sigma^2} + \sum_{t=p+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p})^2\right\}\right].$$

We will use this inverse gamma distribution as the proposal distribution for σ^2 when considering the full data.

Algorithm 5 Metropolis-within-Gibbs algorithm for inference of a stationary AR(p) process with a known p

1. Initialise the state of the chain to $(\sigma^{2^{(0)}}, \tilde{\rho}_1^{(0)}, \ldots, \tilde{\rho}_p^{(0)})$ and set the iteration counter to $k = 1$.

2. Sample $\sigma^{2^{(k)}}$ from $\pi(\sigma^2|\sigma^{2^{(k-1)}}, \tilde{\rho}_1^{(k-1)}, \ldots, \tilde{\rho}_p^{(k-1)})$ using a Metropolis-Hastings step with the proposal distribution

$$\sigma^{2^*} \mid \sigma^2 \sim \text{IG}\left[a_{\sigma^2} + \frac{n-p}{2}, \frac{1}{2}\left\{2b_{\sigma^2} + \sum_{t=p+1}^{n}(y_t - \phi_1^{(k-1)} y_{t-1} - \cdots - \phi_p^{(k-1)} y_{t-p})^2\right\}\right].$$

3. For $s = 1, \ldots, p$, sample $\tilde{\rho}_s^{(k)}$ from $\pi(\tilde{\rho}_s|\sigma^{2^{(k)}}, \tilde{\rho}_1^{(k)}, \ldots, \tilde{\rho}_{s-1}^{(k)}, \tilde{\rho}_{s+1}^{(k-1)}, \ldots, \tilde{\rho}_p^{(k-1)})$ using a Metropolis-Hastings step with the proposal distribution

$$\tilde{\rho}_s^*|\tilde{\rho}_s \sim \text{Beta}\{w_d\tilde{\rho}_s + \varepsilon, w_d(1-\tilde{\rho}_s) + \varepsilon\}.$$

4. Set k equal to $k + 1$ and return to step 2.

Simulation experiment

Consider the case where we know that the data are generated from an AR(p) process. We apply our MCMC scheme to simulated data to investigate the behaviour of the posterior distribution in this case. We simulated an AR(3) process of length 10,000, where the true values of the partial autocorrelations are $\boldsymbol{\rho} = (0.8, 0.4, 0.1)$ and the true value of the error variance is $\sigma^2 = 0.8$. We fit the Metropolis-within-Gibbs algorithm described in Algorithm 5 to this data, running the algorithm for 5,000 iterations, with the first 1,000 discarded as burn-in. In the inverse gamma prior for σ^2 we take $a_{\sigma^2} = 2.5$ and $b_{\sigma^2} = 1.5$. These values are chosen to correspond with the values chosen in the inverse Wishart distribution used in the vector autoregressive case, discussed in Chapter 5. In the prior for the $\tilde{\rho}_s$, we choose $a_{\rho_s} = 1$ and $b_{\rho_s} = 1$ for the hyperparameters. This is equivalent to specifying a uniform distribution over the interval (0,1), which is chosen as we have no reason to believe *a priori* that any values of the partial autocorrelations are more likely than others. Whilst the results are not included here, we also repeated the experiment using values of $a_{\rho_s} = b_{\rho_s} = 0.1$ and $a_{\rho_s} = b_{\rho_s} = 10$ to investigate sensitivity to the choice of prior hyperparameters and did not find the results to be sensitive to this choice. The proposal variance for the Metropolis-Hastings updates of the $\tilde{\rho}_s$ is obviously sensitive to the choice of ω_d and ε as these are tuning parameters. We found that values of $\omega_d = 1,000$ and $\varepsilon = 0.05$ resulted in an appropriate number of proposals being accepted in the Metropolis-Hastings updates of the $\tilde{\rho}_s$. We found it straightforward to tune these parameters manually. However, if this had been difficult, we could have considered adaptive MCMC as an approach that avoids the need to manually tune the proposal parameters by automatically adjusting them during the sampling process to find suitable values (Haario *et al.*, 2001; Atchadé & Rosenthal, 2005; Andrieu & Thoms, 2008; Roberts & Rosenthal, 2009). We do not consider adaptive MCMC here or later in this thesis as in all cases where it was necessary to tune proposal parameters it was straightforward to tune them manually to achieve an appropriate acceptance rate. In addition to our Metropolis-within-Gibbs scheme, we also coded up a Stan programme to compare to the output from our MCMC scheme. The Stan programme is provided in Appendix C.2.

Figures 4.2 and 4.3 contain trace and posterior density plots respectively, for the Metropolis-within-Gibbs scheme output overlaid with the output from the Stan programme. The pink corresponds to the output from the Metropolis-within-Gibbs scheme described in Algorithm 5 whilst the other colours are from four different chains of the Stan output. The trace plots suggest that all chains have converged to the same mode and the posterior densities from all chains seem to match. Promisingly, the true values for the partial autocorrelations and error variance lie within their respective posterior distributions. The fact that the Metropolis-within-Gibbs scheme reasonably picks up the true values of the parameters as well as showing overlap with an independently coded MCMC algorithm

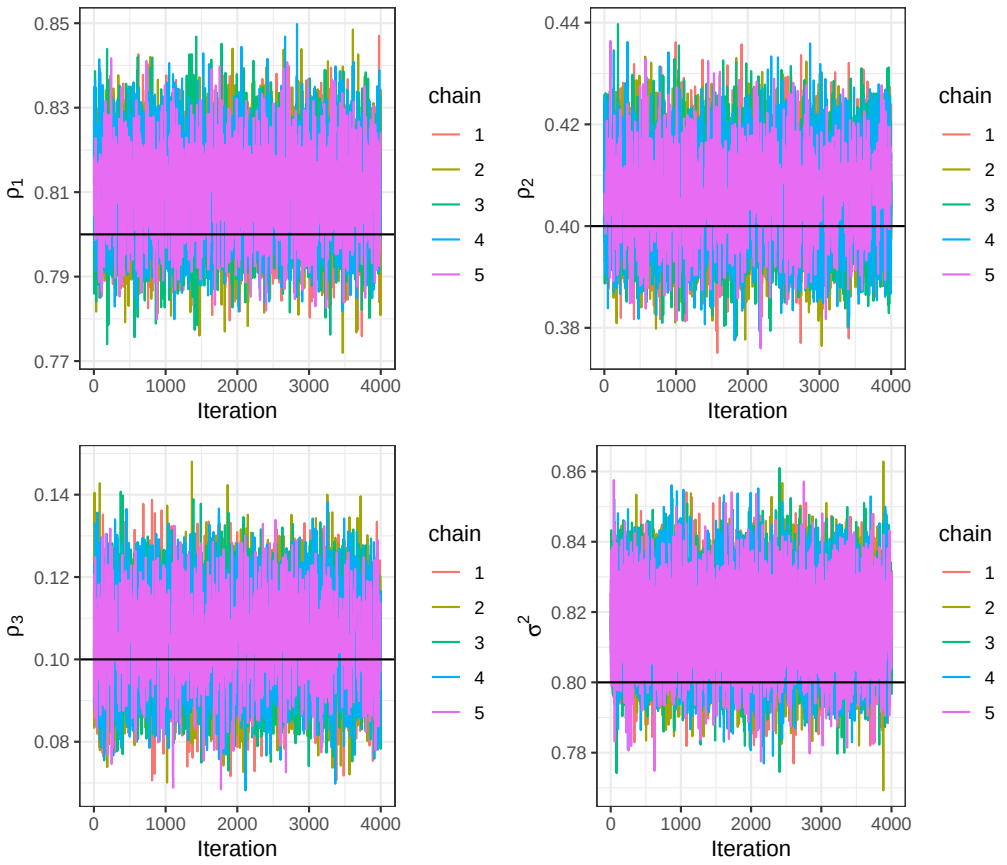


Figure 4.2: Trace plots of draws from the posterior density of the partial autocorrelations ρ_1, ρ_2, ρ_3 and error variance σ^2 for data which has been simulated from an AR(3) process. Chain 5, depicted in pink, was obtained using the Metropolis-within-Gibbs algorithm and chains 1 to 4, depicted by the other colours, were obtained using Stan. The true values are represented as black horizontal lines.

gives no evidence of any problems with the Metropolis-Hastings algorithm.

4.2 Enforcing stationarity when p is unknown

Having considered the case where the model order p is known, we then considered methods from the literature which investigate the order of stationary univariate autoregressions. Barnett *et al.* (1996) and Vermaak *et al.* (2004) consider learning the order of stationary AR(p) processes by first reparameterising the model in terms of its partial autocorrelations, as described in Section 4.1. A univariate stationary autoregression of order p has a non-zero partial autocorrelation at lag p and then zero partial autocorrelations at all higher lags. Therefore, by choosing a large (maximum) value for p and assigning each partial autocorrelation a spike-and-slab prior with a continuous distribution over $(-1, 1)$ and an

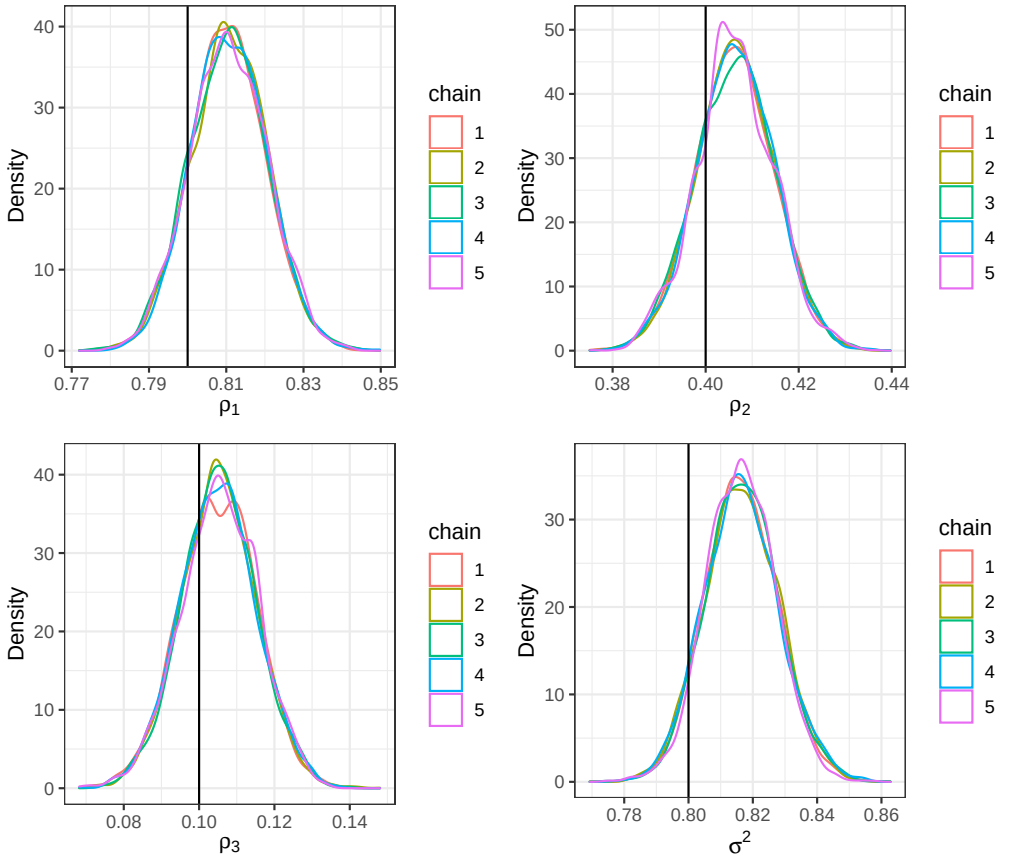


Figure 4.3: Posterior density plots for the partial autocorrelations ρ_1, ρ_2, ρ_3 and error variance σ^2 for data which was simulated from an AR(3) process. Chain 5, depicted in pink, was obtained using the Metropolis-within-Gibbs algorithm and chains 1 to 4, depicted by the other colours, were obtained using Stan. The true values are represented as black vertical lines.

atom of probability at zero, Barnett *et al.* (1996) allow inference on the order of the process. Vermaak *et al.* (2004) also enforce stationarity by reparameterising the model in terms of its partial autocorrelations but frame the problem of order determination as a model selection problem and use reversible jump Markov chain Monte Carlo to learn the order of the process. Since reversible jump algorithms are notoriously difficult to tune (Brooks *et al.*, 2003) and for more straightforward generalisation to the vector case, in order to carry out Bayesian inference on the order of an autoregressive process, we follow Barnett *et al.* (1996) in considering spike-and-slab priors for the partial autocorrelations.

4.2.1 Prior distribution over the partial autocorrelation reparameterisation

We consider two representations of a spike-and-slab prior for the partial autocorrelations which have a "slab" component with support on the interval (-1, 1) and a "spike" in probability at zero to allow order determination. Both representations make use of indicator variables to indicate whether each partial autocorrelation is non-zero. Specifically, I_s is an indicator variable such that $I_s = 1$ if ρ_s is non-zero and $I_s = 0$ otherwise, for $s = 1, \ldots, p$. In the first representation we take a similar approach to Barnett *et al.* (1996) who sample the indicator variables and the partial autocorrelations jointly, whilst in the second we consider an adaptation of methods developed for variable selection in regression by Kuo & Mallick (1998) which sample the indicator variables and the partial autocorrelations from their full conditional distributions in two blocks. In both cases, we allow a maximum order for the model, $p_{\max}$, and as such a maximum of $p_{\max}$ non-zero partial autocorrelations, where $p_{\max}$ is a large value which should be conservative enough that we believe $p < p_{\max}$. The two representations of the spike-and-slab prior which we consider both result in the same overall prior distribution for ρ_s, but they are expressed differently. In each case, we adopt an overall prior specification of the form

$$\pi(\rho_1, \ldots, \rho_{p_{\max}}, I_1, \ldots, I_{p_{\max}}, \sigma^2) = \pi(\sigma^2) \prod_{s=1}^{p_{\max}} \pi(\rho_s | I_s) \pi(I_s).$$

In each representation of the spike-and-slab prior, the indicator variables are given a Bernoulli prior distribution such that

$$I_s \sim \text{Bern}(p_{I_s}) \tag{4.9}$$

for $s = 1, \ldots, p_{\max}$, where p_{I_s} is the prior probability that ρ_s is non-zero, chosen prior to analysis.

In the representation of the spike-and-slab prior discussed in Barnett *et al.* (1996), the prior density for a partial autocorrelation is conditional on the value of its respective indicator variable. The s-th partial autocorrelation, ρ_s for $s = 1, \ldots, p_{\max}$, is 0 if $I_s = 0$ and if $I_s = 1$ then Barnett *et al.* (1996) suggest giving ρ_s a uniform prior on the interval (-1,1). We achieve an identical specification by giving ρ_s the same stretched beta prior density as in the known p case, defined in terms of $\tilde{\rho}_s$ in Equation (4.2), with hyperparameters $a_{\rho_s} = b_{\rho_s} = 1$.

In the context of variable selection in regression, Kuo & Mallick (1998) set up the regression model as

$$y_i = \sum_{j=1}^{q} \beta_j I_j x_{ij} + \epsilon_i$$

where y_i is the response variable for $i = 1, \ldots, n$, x_{ij} is the j-th explanatory variable for $j = 1, \ldots, q$, the β_j are the regression coefficients, and ϵ_i is an error term. Here, the I_j, $j = 1, \ldots, q$, are indicator variables such that if $I_j = 1$ then the jth explanatory variable is included in the model, whereas if $I_j = 0$, the jth explanatory variable is excluded from the model. We adapt this method for use in determining the order of an autoregressive process by constructing our partial autocorrelations such that

$$\rho_s = I_s \rho'_s$$

for $s = 1, \ldots, p_{\max}$. This results in $\rho_s = 0$ if $I_s = 0$, and $\rho_s \neq 0$ if $I_s = 1$. Here, the I_s have the prior described in (4.9) and we give the ρ'_s the same stretched beta distribution as used in both the known p case and the first representation of the spike-and-slab prior, specified in terms of $\tilde{\rho}_s$ in (4.2).

Alongside both representations of the spike-and-slab prior we give σ^2 the same inverse gamma prior density as in the known p case, given in (4.3).

4.2.2 Posterior inference over the partial autocorrelations

Having specified the prior distribution for σ^2 and two different representations of a spike-and-slab prior for $\boldsymbol{\rho}$, we can now consider posterior inference. The likelihood for the parameters is given in Equation (4.4). Due to the different representations of the spike-and-slab prior, the representation of the posterior is slightly different under each method.

Let $\boldsymbol{I} = (I_1, \ldots, I_{p_{\max}})$. When using the spike-and-slab representation described in Barnett *et al.* (1996), we can combine the likelihood with the prior distribution via Bayes' theorem to obtain the posterior distribution as

$$\pi(\sigma^2, \boldsymbol{\rho}, \boldsymbol{I} \mid y_{1:n}) \propto p(y_{1:n} \mid \sigma^2, \boldsymbol{\rho}, \boldsymbol{I}) \pi(\sigma^2) \prod_{s=1}^{p_{\max}} \pi(\rho_s \mid I_s) \pi(I_s).$$

The full conditional distributions of each unknown parameter are straightforward to obtain and so we can use a Metropolis-within-Gibbs algorithm to sample from this posterior distribution. As in Barnett *et al.* (1996), the partial autocorrelations, ρ_s for $s = 1 \ldots, p_{\max}$, and indicator variables, I_s for $s = 1, \ldots, p_{\max}$, will be sampled jointly, conditional on the other variables. The idea is that by sampling as a block, rather than in two Gibbs steps for $\rho_s | I_s, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}$ and then $I_s | \rho_s, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}$, where $\boldsymbol{I}_{-s} = (I_1, \ldots, I_{s-1}, I_{s+1}, \ldots, I_{p_{\max}})$, mixing might be improved. For ρ_s and I_s we have

$$\pi(\rho_s, I_s | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) \propto p(y_{1:n} | \boldsymbol{\rho}, \boldsymbol{I}, \sigma^2) \pi(\rho_s | I_s) \pi(I_s). \tag{4.10}$$

A proposal for the partial autocorrelation ρ_s and indicator variable I_s is generated jointly

from a proposal distribution with density

$$q(\rho_s^*, I_s^* | \boldsymbol{\rho}, \boldsymbol{I}, \sigma^2, y_{1:n}) = q_1(I_s^* | \boldsymbol{\rho}, \boldsymbol{I}, \sigma^2, y_{1:n}) q_2(\rho_s^* | I_s^*, \boldsymbol{\rho}, \boldsymbol{I}, \sigma^2, y_{1:n}) \tag{4.11}$$

where

$$q_1(I_s^* | \boldsymbol{\rho}, \boldsymbol{I}, \sigma^2, y_{1:n}) = \pi(I_s^* | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})$$

is the full conditional distribution (4.10) with the partial autocorrelation ρ_s^* integrated out and ρ_s^* is sampled from the density $q_2(\rho_s^* | I_s^*, \rho_s)$ such that $\Pr(\rho_s^* = 0 | I_s^* = 0) = 1$ and if $I_s^* = 1$ the proposal density is implied by (4.7) in the known p case. The pair is then accepted or rejected together with acceptance probability

$$\alpha\left\{(\rho_s^*, I_s^*), (\rho_s, I_s)\right\} = \min\left\{1, A\right\}$$

where

$$\begin{aligned}
A &= \frac{\pi(I_s^*, \rho_s^* | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})}{\pi(I_s, \rho_s | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})} \times \frac{q(I_s, \rho_s | I_s^*, \rho_s^*, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})}{q(I_s^*, \rho_s^* | I_s, \rho_s, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})} \\
&= \frac{\pi(I_s^* | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) \pi(\rho_s^* | I_s^*, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})}{\pi(I_s | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) \pi(\rho_s | I_s, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})} \\
&\quad \times \frac{\pi(I_s | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) q_2(\rho_s | I_s, \rho_s^*)}{\pi(I_s^* | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) q_2(\rho_s^* | I_s^*, \rho_s,)} \\
&= \frac{\pi(\rho_s^* | I_s^*, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) q_2(\rho_s | I_s, \rho_s^*)}{\pi(\rho_s | I_s, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) q_2(\rho_s^* | I_s^*, \rho_s)}
\end{aligned}$$

As the indicator variable I_s can only take the values of zero or one, its full conditional distribution, marginalised over ρ_s, ie $\pi(I_s | \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})$ which is our proposal distribution q_1, is Bernoulli. We can calculate its success probability by integrating (4.10) over ρ_s:

$$\begin{aligned}
\Pr(I_s = 1 | y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) &= \int_{-1}^{1} p(I_s = 1, \rho_s | y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) \mathrm{d}\rho_s \\
&\propto \int_{-1}^{1} p(y_{1:n} | I_s = 1, \rho_s, \boldsymbol{I}_{-s}, \boldsymbol{\rho}_{-s}, \sigma^2) \\
&\quad \times p(I_s = 1, \rho_s | \boldsymbol{I}_{-s}, \boldsymbol{\rho}_{-s}, \sigma^2) \mathrm{d}\rho_s \\
&= \int_{-1}^{1} p(y_{1:n} | I_s = 1, \rho_s, \boldsymbol{I}_{-s}, \boldsymbol{\rho}_{-s}, \sigma^2) p(I_s = 1, \rho_s) \mathrm{d}\rho_s \\
&= \int_{-1}^{1} p(y_{1:n} | I_s = 1, \rho_s, \boldsymbol{I}_{-s}, \boldsymbol{\rho}_{-s}, \sigma^2) \\
&\quad \times p(\rho_s | I_s = 1) \Pr(I_s = 1) \mathrm{d}\rho_s.
\end{aligned} \tag{4.12}$$

The integrand in (4.12) can be close to zero and so the log-sum-exp trick for integrals is used. Given the integral

$$K = \int_a^b p(x)\mathrm{d}x,$$

let

$$L = \log \int_a^b p(x)\mathrm{d}x.$$

The log-sum-exp trick for integrals calculates L using

$$L = l_0 + \log \int_a^b q(x)\mathrm{d}x$$

where $l(x) = \log p(x)$, $l_0 = \max\{l(x)\}$ over the interval $[a, b]$ and $q(x) = \exp\{l(x) - l_0\}$. The integral of $q(x)$ can be calculated numerically using the `integrate` function in R. Then, taking the exponential of L gives $\exp(L) = K$, allowing calculation of the original integral. This trick is used to calculate the integral in (4.12). As this integral is proportional to $\Pr(I_s = 1|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2)$, we also need a normalising constant. As there are only two values that I_s can take we also need the probability that $I_s = 0$. We have

$$\begin{aligned}
\Pr(I_s = 0|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) &= p(I_s = 0, \rho_s = 0|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) \\
&\propto p(y_{1:n}|I_s = 0, \rho_s = 0, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) \\
&\quad \times p(I_s = 0, \rho_s = 0|\boldsymbol{I}_{-s}, \boldsymbol{\rho}_{-s}, \sigma^2) \\
&= p(y_{1:n}|I_s = 0, \rho_s = 0, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) p(I_s = 0, \rho_s = 0) \\
&= p(y_{1:n}|I_s = 0, \rho_s = 0, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) p(\rho_s = 0|I_s = 0) \Pr(I_s = 0) \\
&= p(y_{1:n}|I_s = 0, \rho_s = 0, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) \Pr(I_s = 0). \qquad (4.13)
\end{aligned}$$

Letting $\Pr(I_s = 1|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) \propto B$ where B is given in (4.12) and $\Pr(I_s = 0|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) \propto C$ where C is given in (4.13) then we have

$$\Pr(I_s = 1|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2) = \frac{B}{B + C}$$

and so we propose I_s^* from a $\mathrm{Bern}\{\Pr(I_s = 1|y_{1:n}, \boldsymbol{\rho}_{-s}, \boldsymbol{I}_{-s}, \sigma^2)\}$ distribution.

Additionally, the prior distribution for σ^2 is the same as in the known p case. This results in the full conditional distribution for σ^2 having the same form as in the known p case, given in (4.8). Therefore, we use the same Metropolis-Hastings step and proposal distribution to sample σ^2 as described in the known p case in Algorithm 5. The full Metropolis-within-Gibbs algorithm is provided in Algorithm 6.

Let $\boldsymbol{\rho'} = (\rho'_1, \ldots, \rho'_p)$. In the representation of the spike-and-slab prior adapted from methods described in Kuo & Mallick (1998), combining the likelihood with the prior

Algorithm 6 Metropolis-within-Gibbs algorithm for inference of an AR(p) process with an unknown p, reparameterised in terms of the partial autocorrelations, under the representation described in Barnett *et al.* (1996)

1. Initialise the state of the chain to $(\sigma^{2^{(0)}}, \rho_1^{(0)}, \ldots, \rho_{p_{\max}}^{(0)}, I_1^{(0)}, \ldots, I_{p_{\max}}^{(0)})$ and set the iteration counter to $k = 1$.
2. Sample $\sigma^{2^{(k)}}$ from $\pi(\sigma^2|\sigma^{2^{(k-1)}}, \rho_1^{(k-1)}, \ldots, \rho_{p_{\max}}^{(k-1)}, I_1^{(k-1)}, \ldots, I_{p_{\max}}^{(k-1)})$ using a Metropolis-Hastings step with the proposal distribution
$$\sigma^{2^*} \mid \sigma^2 \sim \text{IG}\left[a_{\sigma^2} + \frac{n - p_{\max}}{2}, \frac{1}{2}\left\{2b_{\sigma^2} + \sum_{t=p_{\max}+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p_{\max}})^2\right\}\right].$$
3. For $s = 1, \ldots, p$, sample $(\rho_s^{(k)}, I_s^{(k)})$ from $\pi(\rho_s, I_s|\rho_1^{(k)}, \ldots, \rho_{s-1}^{(k)}, \rho_{s+1}^{(k-1)}, \ldots, \rho_{p_{\max}}^{(k-1)}, I_1^{(k)}, \ldots, I_{s-1}^{(k)}, I_{s+1}^{(k-1)}, \ldots, I_{p_{\max}}^{(k-1)}, \sigma^{2^{(k)}}, y_{1:n})$ using a Metropolis-Hastings step with the proposal distribution given in (4.11).
4. Set k equal to $k + 1$ and return to step 2.

distribution via Bayes' theorem gives the posterior distribution as

$$\pi(\sigma^2, \boldsymbol{\rho}', \boldsymbol{I} \mid y_{1:n}) \propto p(y_{1:n} \mid \sigma^2, \boldsymbol{\rho}', \boldsymbol{I})\pi(\sigma^2)\prod_{s=1}^{p_{\max}}\pi(\rho_s')\pi(I_s).$$

Once again, obtaining the full conditional distributions for the unknown parameters is straightforward, and so we use a Metropolis-within-Gibbs algorithm to sample from the posterior. The full conditional distribution for each I_s, $s = 1, \ldots, p_{\max}$, is such that

$$\begin{aligned}
\pi(I_s|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) &\propto p(y_{1:n}|\boldsymbol{\rho}', \boldsymbol{I}, \sigma^2)\pi(I_s) \\
&\propto \det(G)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(y_{1:p_{\max}}^T G^{-1} y_{1:p_{\max}})\right\} \\
&\times (\sigma^2)^{-\frac{1}{2}(n-p_{\max})} \exp\left\{-\frac{1}{2\sigma^2}\sum_{t=p_{\max}+1}^{n}(y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p_{\max}})^2\right\} \\
&\times p_{I_s}^{I_s}(1 - p_{I_s})^{(1-I_s)}
\end{aligned}$$

where Φ and G are functions of $(\boldsymbol{\rho}', \boldsymbol{I})$. As each I_s can only take one of two values, namely zero or one, the posterior probability of each value can be calculated up to a proportionality constant. Then we have

$$\Pr(I_s = 1|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) = \frac{\pi(I_s = 1|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})}{\pi(I_s = 1|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n}) + \pi(I_s = 0|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})}.$$

The posterior distribution for I_s is therefore such that

$$I_s|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n} \sim \text{Bern}\{\Pr(I_s = 1|\boldsymbol{\rho}', \boldsymbol{I}_{-s}, \sigma^2, y_{1:n})\}$$

and each I_s can be sampled using a Gibbs sampling step. If $I_s = 0$ then $\rho_s = \rho'_s I_s = 0$ and there will be no information in the likelihood from ρ'_s. Therefore, the likelihood will be independent of ρ'_s and the full conditional distribution for ρ'_s will be its prior distribution. Hence, if $I_s = 0$ we can use a Gibbs step to sample ρ'_s from its prior. If $I_s = 1$ then as ρ'_s has the same prior distribution as the partial autocorrelations in the known case, the full conditional distribution for ρ'_s will have the same form as the full conditional for the partial autocorrelations in the known p case, and so the Metropolis-Hastings step and proposal distribution described in Algorithm 5 can be used to sample from the posterior for ρ'_s. Finally, as σ^2 has the same prior distribution as in both the known p case and the first representation of the spike-and-slab prior, the full conditional distribution for σ^2 also has the same form as the distribution in Equation (4.8), and so we once again use the same Metropolis-Hastings step and proposal distribution to sample σ^2 as described in Algorithm 5. The full Metropolis-within-Gibbs algorithm for this representation of the prior is given in Algorithm 7.

Simulation experiment

In order to investigate the behaviour of the posterior distributions under both representations of the spike-and-slab prior, we applied the MCMC schemes to simulated time series data. In particular, for each $p \in \{1, 2, 3, 4\}$ we simulated 10 stationary AR(p) processes, of length 1,000. The process used to simulate the data is discussed further in Section 5.3.6, where we conduct a thorough simulation study for the vector autoregressive case. We use a subset of the data sets simulated in this later work here, using those data sets where $m = 1$.

For each of the 10 data sets for each value of p, we fit our model under both representations with a maximum permitted order of $p_{\max} = 7$, and as such a maximum of $p_{\max} = 7$ non-zero partial autocorrelations. In the prior for the indicator variables, we give each p_{I_s}, for $s = 1, \ldots, p_{\max}$, a value of

$$p_{I_s} = \left(\frac{\alpha}{\alpha + 1}\right)^s, \tag{4.14}$$

for a specified value of $\alpha > 0$. This results in a set of probabilities p_{I_s} which decrease as the lag increases, representing a prior belief that partial autocorrelations at higher lags are more likely to be zero. The particular choice of p_{I_s} given in (4.14) is chosen as it corresponds with the prior expectation of a partial autocorrelation matrix being non-zero

Algorithm 7 Metropolis-within-Gibbs algorithm for inference of an AR(p) process with an unknown p, reparameterised in terms of the partial autocorrelations, under the representation adapted from Kuo & Mallick (1998)

1. Initialise the state of the chain to $(\sigma^{2^{(0)}}, \tilde{\rho}_1^{(0)}, \ldots, \tilde{\rho}_{p_{\max}}^{(0)}, I_1^{(0)}, \ldots, I_{p_{\max}}^{(0)})$ and set the iteration counter to $k = 1$.

2. Sample $\sigma^{2^{(k)}}$ from $\pi(\sigma^2|\sigma^{2^{(k-1)}}, \tilde{\rho}_1^{(k-1)}, \ldots, \tilde{\rho}_{p_{\max}}^{(k-1)}, I_1^{(k-1)}, \ldots, I_{p_{\max}}^{(k-1)})$ using a Metropolis-Hastings step with the proposal distribution

$$\sigma^{2^*} \mid \sigma^2 \sim \text{IG}\left[a_{\sigma^2} + \frac{n - p_{\max}}{2}, \frac{1}{2}\left\{2b_{\sigma^2} + \sum_{t=p_{\max}+1}^{n} (y_t - \phi_1 y_{t-1} - \cdots - \phi_p y_{t-p_{\max}})^2\right\}\right].$$

3. For $s = 1, \ldots, p$,

   (a) Sample

$$I_s^{(k)} \sim \text{Bern}\{\Pr(I_s = 1|\rho_1'^{(k)}, \ldots, \rho_{s-1}'^{(k)}, \rho_s'^{(k-1)}, \ldots, \rho_p'^{(k-1)}, I_1^{(k)}, \ldots, I_{s-1}^{(k)}, I_{s+1}^{(k-1)}, \ldots, I_p^{(k-1)}, \sigma^{2^{(k)}}, y_{1:n})\}.$$

   (b) i. If $I_s = 0$, sample $\tilde{\rho}_s^{(k)}$ from $\pi(\tilde{\rho}_s)$ and set $\rho_s'^{(k)} = 2\tilde{\rho}_s^{(k)} - 1$.

   ii. If $I_s = 1$, sample $\tilde{\rho}_s^{(k)}$ from $\pi(\tilde{\rho}_s|\sigma^{2^{(k)}}, \tilde{\rho}_1^{(k)}, \ldots, \tilde{\rho}_{s-1}^{(k)}, \tilde{\rho}_{s+1}^{(k-1)}, \ldots, \tilde{\rho}_p^{(k-1)})$ using a Metropolis-Hastings step with proposal distribution

$$\tilde{\rho}_s^*|\tilde{\rho}_s \sim \text{Beta}\{w_d\tilde{\rho}_s + \varepsilon, w_d(1 - \tilde{\rho}_s) + \varepsilon\}$$

   and set $\rho_s'^{(k)} = 2\tilde{\rho}_s^{(k)} - 1$.

   (c) Set $\rho_s^{(k)} = \rho_s'^{(k)} I_s^{(k)}$.

4. Set k equal to $k + 1$ and return to step 2.

under the cumulative shrinkage process (CUSP) prior (Legramanti *et al.*, 2020), which we discuss in Section 5.3.3 for modelling vector autoregressions. We take $\alpha = 3$, to correspond with the choice of α used when fitting the CUSP prior in Section 5.3.3 where α is the prior expectation for the order of the process. In the stretched beta priors and inverse gamma prior, we use the same prior specification as in the known p case, with $a_{\rho_s} = b_{\rho_s} = 1$, $a_{\sigma^2} = 2.5$ and $b_{\sigma^2} = 1.5$. Similarly, we retained the same values for the tuning parameters as in the known p case, taking $\omega_d = 1,000$ and $\varepsilon = 0.05$. Each MCMC scheme was run for a burn-in period of 4,000 iterations followed by a further 16,000 iterations for analysis. The time taken for the 16,000 iterations following burn-in was also recorded for each method. The usual graphical diagnostics gave no evidence of any lack of convergence.

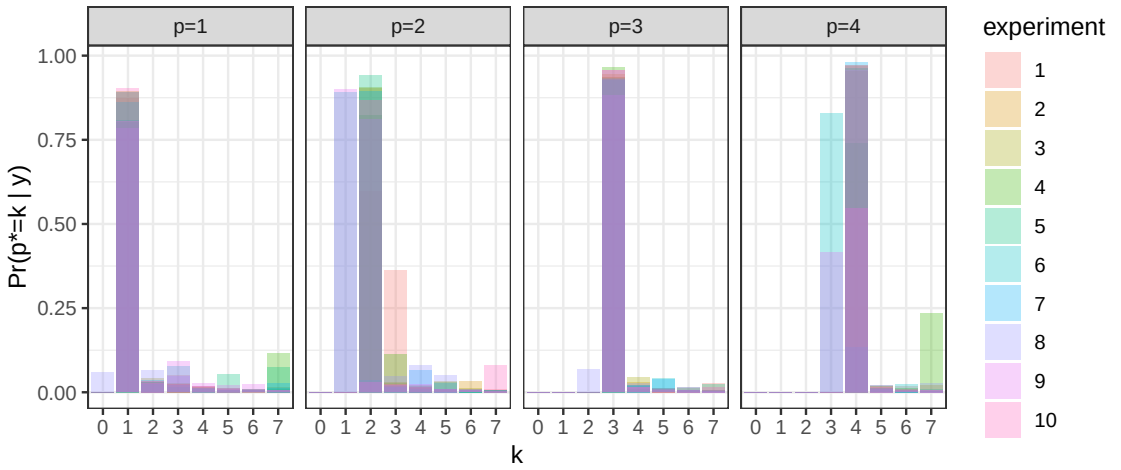


Figure 4.4: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $p \in \{1, 2, 3, 4\}$ under the Barnett *et al.* (1996) representation of the spike-and-slab prior, with $n = 1000$.

Let the lag of the final non-zero partial autocorrelation sampled at each draw of the MCMC schemes be denoted by p^*, which we refer to as the *effective order* of the process. Figures 4.4 and 4.5 contain the posterior mass functions for p^* for each data set obtained from the Barnett *et al.* (1996) and Kuo & Mallick (1998) approaches respectively. For each value of p, the posterior mass functions are represented as a set of 10 overlaid bar charts, one for each data set. Both representations give very similar output, with the posterior mode being at the true order of the process in nearly all cases. This is promising, as both representations of the spike-and-slab prior result in the same prior density, and so we would expect the posterior densities to be identical up to Monte Carlo error. The only experiment which does not have near identical output is experiment four when $p = 4$. In this case, we suspect that under one of the representations the sampler may have become stuck at a local mode, hence resulting in a different output. However, as this difference in results only occurs in one out of forty experiments, we are satisfied that there is no suggestion that either of the samplers are coded up incorrectly.

Furthermore, for a fixed value of $p = 3$, we also simulated 10 stationary AR(p) processes of length $n = 100$ and another 10 of length $n = 500$, to allow comparison of the posterior mass functions across $n \in \{100, 500, 1000\}$ under both representations. We repeated our analysis using the same prior specifications and number of iterations as in the previous set of experiments. Figures 4.6 and 4.7 contain the posterior mass functions for each value of n using the Barnett *et al.* (1996) and Kuo & Mallick (1998) representations respectively. Once again, the posterior mass functions are very similar across both representations. In nearly all cases, the true order is again the posterior mode, though there is a higher

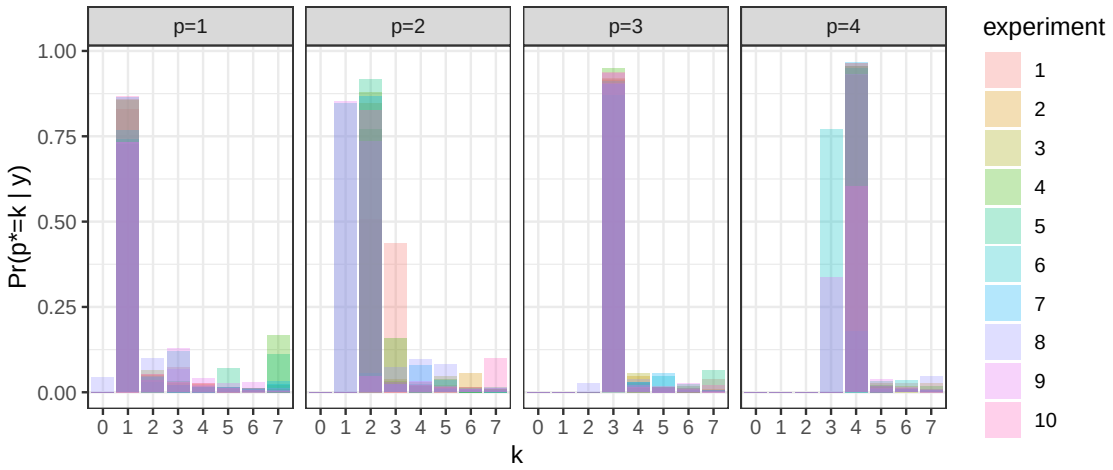


Figure 4.5: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $p \in \{1, 2, 3, 4\}$ under the Kuo & Mallick (1998) representation of the spike-and-slab prior, with $n = 1000$.

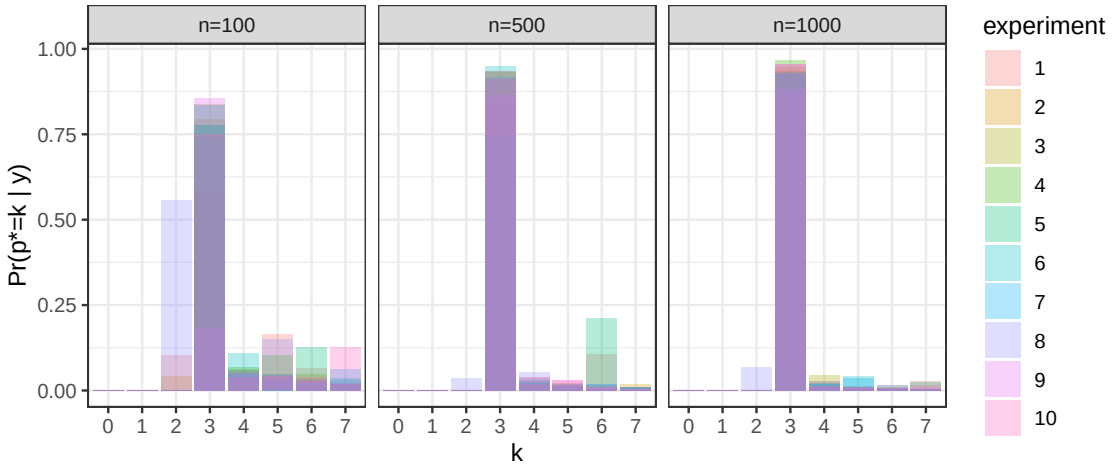


Figure 4.6: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $n \in \{100, 500, 1000\}$ under the Barnett *et al.* (1996) representation of the spike-and-slab prior, with $p = 3$.

level of uncertainty as n decreases, which is to be expected with the reduced amount of information available from the data.

Whilst the posterior is the same across the two representations (up to Monte Carlo error), the time taken for each of the MCMC schemes to run differs greatly. This is largely due to sampling in the Barnett *et al.* (1996) approach being slowed down by the numerical integration over the partial autocorrelations. For example, in the $p = 1$ case, the average time taken across the 10 data sets to sample the 16,000 iterations following burn-in was

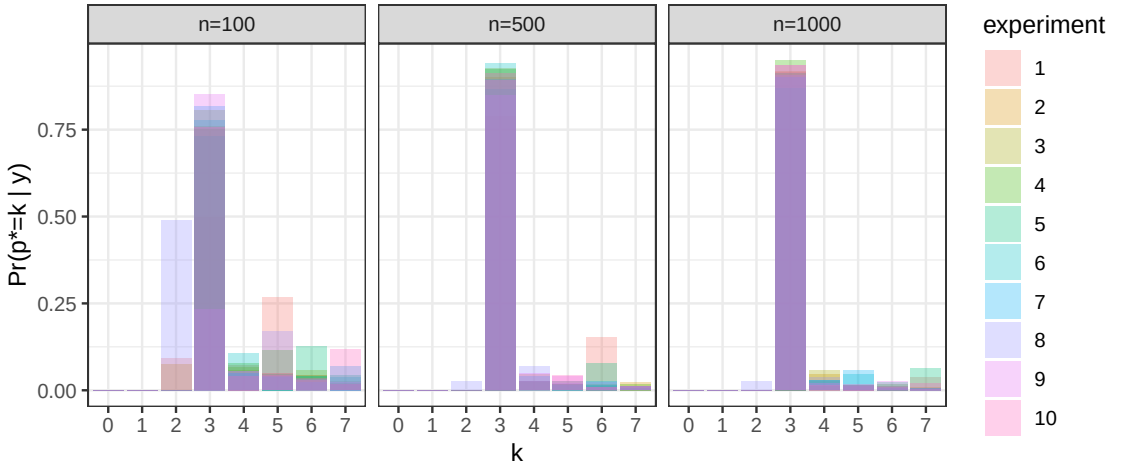


Figure 4.7: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $n \in \{100, 500, 1000\}$ under the Kuo & Mallick (1998) representation of the spike-and-slab prior, with $p = 3$.

Method	$p = 1$	$p = 2$	$p = 3$	$p = 4$
Kuo and Mallick	1.5456 (1796.79)	1.8459 (1861.40)	1.0607 (1019.52)	1.0067 (1032.67)
Barnett	0.0155 (1080.69)	0.0134 (1529.12)	0.0098 (959.45)	0.0095 (1144.20)

Table 4.1: Average minimum ESS/s across 10 data sets for each $p \in \{1, 2, 3, 4\}$ for two representations of a spike-and-slab prior under the partial autocorrelation parameterisation. The average minimum ESS obtained across the 10 data sets for all iterations is provided in brackets.

1162.52s for the Kuo & Mallick (1998) approach but 69,721.81s for the Barnett *et al.* (1996) approach. To consider whether the extra time taken to run the latter approach is worthwhile, the minimum effective sample size per second (ESS/s) across variables for each representation was considered. Table 4.1 contains the average minimum ESS/s across 10 data sets for each value of p under the two representations. We can see that the minimum effective sample sizes per second are much higher for the representation adapted from Kuo & Mallick (1998), suggesting that it is not of benefit to use the much slower representation from Barnett *et al.* (1996). As such, when using this reparameterisation for model order determination with real data, we will use the representation adapted from methods in Kuo & Mallick (1998).

4.2.3 Characteristic root reparameterisation

An alternative approach that has been used for order determination in the case of univariate autoregressions is discussed in Huerta & West (1999). In this work, the authors reparameterise the autoregressive model in terms of the reciprocal roots of the character-

istic equation. Under this parameterisation, the characteristic polynomial $\phi(x)$, discussed in Section 2.2.3, can be written as

$$\phi(x) = \prod_{s=1}^{p}(1 - \alpha_s x) \tag{4.15}$$

where the α_s, for $s = 1, \ldots, p$, are the reciprocals of the roots of the characteristic equation. Since a univariate autoregressive process is stationary if and only if the roots of the characteristic equation lie outside the unit circle, an equivalent condition of stationarity is that $|\alpha_s| < 1$ for $s = 1, \ldots, p$. In their work, Huerta & West (1999) also allow unit roots, however when using their work this is not something that we permit, instead choosing to enforce strict stationarity. The p reciprocal roots can be a combination of real reciprocals and complex conjugate pairs of reciprocal roots and we follow Huerta & West (1999) by assuming the (non-zero) α_s to be distinct. Repeated (non-zero) roots are not considered as the prior distributions chosen give distinct (non-zero) roots almost surely. If all roots are non-zero and there are C complex pairs of reciprocal roots then there must be $R = p - 2C$ real reciprocal roots. The complex reciprocals are then written as $r_s e^{\pm i\omega_s}$ for $s = 1, \ldots, C$ and the real reciprocals are written as r_s for $s = 2C + 1, \ldots, p$. To satisfy the stationarity condition we have $|r_s| < 1$ for $s = 1, \ldots, p$. Additionally, we have $\omega_s > 0$ in the argument for the complex pairs. Under this reparameterisation, if zero roots were allowed then the model order p would be equivalent to the number of non-zero reciprocal roots of the characteristic equation. To allow uncertainty in the model order, we therefore assign priors to the real and complex reciprocal roots with atoms of probability at moduli 0 in each case. We follow Huerta & West (1999) in fixing a value C as the maximum number of non-zero pairs of complex reciprocals and a value R as the maximum number of non-zero real reciprocals. Then, the maximum number of non-zero reciprocals, and as such the maximum model order is $p_{\max} = 2C + R$. .

4.2.4 Prior distribution

Real reciprocals

Each real reciprocal r_s, for $s = 2C+1, ..., p_{\max}$, must lie within the interval (-1,1) to satisfy the stationary condition, and can be allowed to be zero to allow for uncertainty in the model order. Huerta & West (1999) give each r_s, for $s = 2C + 1, \ldots, p_{\max}$, the following prior

$$r_s|\pi_{r_0} \sim \pi_{r0} I_0(r_s) + (1 - \pi_{r0}) g_r(r_s)$$

where $g_r(r_s)$ is a continuous density over the interval (-1,1), $I_0(r_s)$ is an indicator function for r_s equalling one if $r_s = 0$ and zero otherwise and π_{r0} is the prior probability of r_s being equal to zero. This density gives an atom of probability at zero, allowing for

some of the real reciprocals to be zero and as such allowing for model order uncertainty. We treat π_{r0} as an unknown, and give this a Beta$(1, 1)$ prior distribution, equivalent to U$(0, 1)$, representing prior indifference about whether the real reciprocals are zero or not. Futhermore, we follow Huerta & West (1999) in giving $g_r(r_s)$ a U$(-1, 1)$ distribution, which makes all possible values equally likely if $r_s \neq 0$.

Complex reciprocals

Each pair of complex reciprocals has the form $r_s e^{\pm i\omega_s}$ for $s = 1, \ldots, C$. We follow Huerta & West (1999) in parametrising in terms of the period, $\lambda_s = 2\pi/\omega_s$, rather than the argument ω_s. Here, the modulus of the complex pair must be less than one, and so $0 \leq r_s < 1$, and the period λ_s is bounded below by 2, to ensure that ω_s is less than π (for identifiability) and bounded above by $\lambda_u = n/2$ as this is the maximum observable period for a time series of length n. For each pair of complex reciprocals, Huerta & West (1999) place independent priors on the modulus and the period. The modulus is given a prior such that

$$r_s|\pi_{c_0} \sim \pi_{c0} I_0(r_s) + (1 - \pi_{c0}) g_c(r_s)$$

for $s = 1, \ldots, C$. Here, $I_0(r_s)$ is an indicator function equalling one if $r_s = 0$ and zero otherwise, π_{c0} is the prior probability of r_s being equal to zero and $g_c(r_s)$ is a continuous distribution with support on the interval (0, 1). As in the real reciprocal case, we treat π_{c0} as an unknown, giving it a Beta(1, 1) prior. The period of the complex pair is given a marginal density $h(\lambda_s)$ with support in $(2, \lambda_u)$.

As discussed in Section 2.2.5, an AR(p) process can be decomposed into p latent processes corresponding to its real and complex roots. In particular, the latent processes corresponding to the pairs of complex roots have similar behaviour to AR(2) processes with autoregressive coefficients $2r_s \cos(2\pi/\lambda_s)$ and $-r_s^2$. Huerta & West (1999) assume a uniform prior on these autoregressive parameters, restricted to the stationary region, which induces densities for $g_c(r_s)$ and $h(\lambda_s)$. Omitting the subscripts, let $X = 2r\cos(2\pi/\lambda)$ and $Y = -r^2$. Then the joint density of X and Y is

$$f_{X,Y}(x, y) \propto c$$

for some constant c, with $y < 1 - x$, $y < 1 + x$ and $y > -1$ to restrict the distribution to the stationary region. Using a bivariate transformation of variables, the joint distribution of r and λ is

$$f_{r,\lambda}(r, \lambda) = f_{X,Y}\left(2r\cos\left(\frac{2\pi}{\lambda}\right), -r^2\right)|J|.$$

Calculating the Jacobian gives

$$|J| = \begin{vmatrix} \frac{\partial x}{\partial r} & \frac{\partial x}{\partial \lambda} \\ \frac{\partial y}{\partial r} & \frac{\partial y}{\partial \lambda} \end{vmatrix} = \begin{vmatrix} 2\cos(2\pi/\lambda) & 4r\pi \sin(2\pi/\lambda)/\lambda^2 \\ -2r & 0 \end{vmatrix} = \frac{8\pi r^2}{\lambda^2} \sin(2\pi/\lambda).$$

Therefore, the joint distribution for r and λ is

$$f_{r,\lambda}(r, \lambda) \propto \frac{r^2}{\lambda^2} \sin(2\pi/\lambda).$$

The marginal densities for r and λ are clearly independent and so the marginal densities for r and λ are readily derived. We obtain $g_c(r) \propto r^2$ with support in (0, 1) to enforce stationarity, and so $r \sim \text{Beta}(3, 1)$. Additionally, $h(\lambda) \propto \sin(2\pi/\lambda)/\lambda^2$ with support over $(2, \lambda_u)$.

Prior for σ^2

Huerta & West (1999) suggest choosing a conjugate inverse gamma prior for σ^2 to make computation of the posterior density easy. As such, we use the same prior for σ^2 as under the partial autocorrelation reparameterisation. That is, we take

$$\sigma^2 \sim \text{IG}(a_{\sigma^2}, b_{\sigma^2}).$$

4.2.5 Posterior inference

Huerta & West (1999) introduce $p_{\max}$ initial values of the series, $Y_{\text{init}} = \{y_0, y_{-1}, \ldots, y_{-p_{\max}+1}\}$ as latent variables to simplify calculation of the likelihood. Note that in the partial autocorrelation reparameterisation, the first p values in the observed time series were assigned the joint stationary distribution, whereas here, we follow Huerta & West (1999) in treating the initial values as unobserved and sampling the latent initial values from the stationary distribution. Treating the first $p_{\max}$ values, Y_{init}, as latent variables, the likelihood has the form

$$p(y_{1:n} \mid \sigma^2, \Phi, y_{\text{init}}) = \prod_{t=1}^{n} p(y_t \mid y_{(t-p):(t-1)}, \sigma^2, \Phi)$$

in which

$$Y_t \mid y_{(t-p):(t-1)}, \sigma^2, \Phi \sim \text{N}\left(\sum_{i=1}^{p} \phi_i y_{t-i}\,,\, \sigma^2\right).$$

Let the collection of all unknown parameters in the priors for the real and complex reciprocals be denoted as $\boldsymbol{\vartheta}$. Regarding the likelihood as a function of the reciprocal roots

and combining it with the prior via Bayes' theorem gives the posterior distribution as

$$\pi(\sigma^2, \boldsymbol{r}, \boldsymbol{\lambda}, \boldsymbol{\vartheta} \mid y_{1:n}) \propto p(y_{1:n} \mid \sigma^2, \boldsymbol{r}, \boldsymbol{\lambda}, \boldsymbol{\vartheta}) \pi(\sigma^2) \pi(\boldsymbol{\vartheta}) \prod_{s=1}^{p} \pi(r_s) \prod_{s=1}^{C} \pi(\lambda_s)$$

where $\boldsymbol{r} = (r_1, \ldots, r_{p_{\max}})$ and $\boldsymbol{\lambda} = (\lambda_1, \ldots, \lambda_C)$.

Huerta & West (1999) use a Metropolis-within-Gibbs algorithm to sample from the full conditional distributions for each parameter one-at-a-time. We denote the full collection of unknown parameters as ψ and use the notation $\psi \backslash \zeta$ to denote the set of all unknown parameters excluding the subset of parameters ζ. We will now consider the full conditional distributions for each of the unknown model parameters.

Real reciprocals

Each real reciprocal, $\alpha_s = r_s$ for $s = 2C+1, \ldots, p_{\max}$, is sampled individually. Conditional on $\psi \backslash \alpha_s$, Huerta & West (1999) consider the filtered time series

$$u_{st} = \prod_{i=1, i \neq s}^{p_{\max}} (1 - \alpha_i B) y_t$$

for $t = 0, \ldots, n$, to aid in the calculation of the full conditional distribution for the real reciprocals. It can be shown as follows that u_{st} is an AR(1) process with autoregressive coefficient $\alpha_s = r_s$:

$$\begin{aligned}
u_{st} &= \prod_{i=1, i \neq s}^{p_{\max}} (1 - \alpha_i B) y_t \\
(1 - \alpha_s B) u_{st} &= (1 - \alpha_s B) \prod_{i=1, i \neq s}^{p_{\max}} (1 - \alpha_i B) y_t \\
(1 - \alpha_s B) u_{st} &= \prod_{i=1}^{p_{\max}} (1 - \alpha_i B) y_t \\
(1 - \alpha_s B) u_{st} &= \varepsilon_t,
\end{aligned}$$

where the final line follows from Equation (4.15). As such, we have

$$(1 - r_s B) u_{st} = \varepsilon_t$$

and so

$$u_{st} \sim N(r_s u_{st-1}, \sigma^2) \text{ for } t = 1, \ldots, n.$$

Then

$$\begin{aligned}
L(r_s|\boldsymbol{u}) &\propto \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{1}{2\sigma^2}(u_{st} - r_s u_{st-1})^2 \right\} \\
&\propto \prod_{t=1}^{n} \exp\left\{ -\frac{1}{2\sigma^2}(u_{st}^2 - 2r_s u_{st} u_{st-1} + r_s^2 u_{st-1}^2) \right\} \\
&\propto \exp\left\{ -\frac{1}{2\sigma^2} \sum_{t=1}^{n} (r_s^2 u_{st-1}^2 - 2r_s u_{st} u_{st-1}) \right\} \\
&\propto \exp\left\{ -\frac{1}{2\sigma^2} \left(r_s^2 \sum_{t=1}^{n} u_{st-1}^2 - 2r_s \sum_{t=1}^{n} u_{st} u_{st-1} \right) \right\} \\
&\propto \exp\left\{ -\frac{\sum_{t=1}^{n} u_{st-1}^2}{2\sigma^2} \left(r_s - \frac{\sum_{t=1}^{n} u_{st} u_{st-1}}{\sum_{t=1}^{n} u_{t-1^2}} \right)^2 \right\} \\
&\propto \exp\left\{ -\frac{1}{2M_s}(r_s - m_s)^2 \right\}
\end{aligned}$$

where

$$m_s = \frac{\sum_{t=1}^{n} u_{st} u_{st-1}}{\sum_{t=1}^{n} u_{st-1}^2}$$

and

$$M_s = \frac{\sigma^2}{\sum_{t=1}^{n} u_{st-1}^2}.$$

However, we also have the stationary condition that $-1 < r_s < 1$, so the density needs truncating to (-1,1), making the conditional likelihood for r_s proportional to a truncated Normal density, $\phi_{\text{tr}}(r_s|m_s, M_s)$, where $\phi_{\text{tr}}(r_s|m_s, M_s)$ is the normal density with parameters m_s and M_s, truncated to (-1,1). Under this likelihood, the conditional posterior for each real reciprocal is a mixture distribution given by

$$\{\pi_{r0} I_0(r_s) + (1 - \pi_{r0}) g_r(r_s)\} \phi_{\text{tr}}(r_s|m_s, M_s)$$

where $g_r(r_s)$ is a $\text{U}(-1, 1)$ density. This reduces to

$$p_{s0} I_0(r_s) + (1 - p_{s0}) \phi_{\text{tr}}(r_s|m_s, M_s)$$

where

$$p_{s0} = \frac{\pi_{r0} \exp\{-m_s^2/2M_s\}}{\pi_{r0} \exp\{-m_s^2/2M_s\} + \frac{1}{2}(1 - \pi_{r0})\sqrt{2\pi M_s} \left\{ \Phi\left(\frac{1-m_s}{\sqrt{M_s}}\right) - \Phi\left(\frac{-1-m_s}{\sqrt{M_s}}\right) \right\}}.$$

We can sample from this posterior distribution using a Gibbs step. The parameters, r_s, are exchangeable in the prior and the likelihood is invariant under permutation of the labels, $s = 2C + 1, \ldots, p_{\max}$. Consequently, the marginal posterior for the r_s has $R!$ symmetric

modes and so the real roots are not, therefore, identifiable in the posterior. As discussed in Richardson & Green (1997), this could lead to occasional label switching during MCMC sampling when the labelling of the roots suddenly switches (i.e the sampler jumps to another mode). In turn, this can make it difficult to assess convergence and mixing. We follow Huerta & West (1999) in addressing the label switching problem by using an online relabelling algorithm, ordering the roots according to $r_{2C+1} < r_{2C+2} < \ldots < r_{p_{\max}}$.

Complex reciprocal pairs

The modulus, r_s, and period, λ_s, are sampled jointly for each complex root pair. Let C_s be the set of indices of all root reciprocals except the sth complex pair. Huerta & West (1999) consider the filtered time series

$$w_{st} = \prod_{i \in C_s} (1 - \alpha_i B) y_t$$

for $t = -1, 0, \ldots, n$ to aid in the derivation of the posterior distribution for the complex reciprocals. It can be shown as follows that w_{st} is an AR(2) process with parameters $\phi_{s1} = 2r_s \cos(2\pi/\lambda_s)$ and $\phi_{s2} = -r_s^2$:

$$\begin{aligned}
w_{st} &= \prod_{i \in C_s} (1 - \alpha_i B) y_t \\
(1 - \alpha_s B)(1 - \bar{\alpha}_s B) w_{st} &= (1 - \alpha_s B)(1 - \bar{\alpha}_s B) \prod_{i \in C_s} (1 - \alpha_i B) y_t \\
(1 - \alpha_s B)(1 - \bar{\alpha}_s B) w_{st} &= \prod_{i=1}^{p_{\max}} (1 - \alpha_i B) y_t \\
(1 - r_s e^{i\omega_s} B)(1 - r_s e^{-i\omega_s} B) w_{st} &= \varepsilon_t \\
\left\{1 - r_s (e^{i\omega_s} + e^{-i\omega_s}) B + r_s^2 B^2\right\} w_{st} &= \varepsilon_t \\
\left\{1 - r_s (2 \cos \omega_s) B + r_s^2 B^2\right\} w_{st} &= \varepsilon_t \\
(1 - 2 r_s \cos \omega_s B + r_s^2 B^2) w_{st} &= \varepsilon_t.
\end{aligned}$$

Huerta & West (1999) define D_s to be the matrix with columns $\boldsymbol{d}_{s1}$ and $\boldsymbol{d}_{s2}$ where $\boldsymbol{d}_{s1}^T = (w_{s0}, w_{s1}, \ldots, w_{s,n-1})$ and $\boldsymbol{d}_{s2}^T = (w_{s,-1}, w_{s0}, \ldots, w_{s,n-2})$. They then let

$$H_s = (D_s^T D_s)^{-1}$$

and

$$\boldsymbol{h}_s = (h_{s1}, h_{s2})^T = H_s D_s^T \boldsymbol{w}_s$$

where $\boldsymbol{w}_s = (w_{s1}, \ldots, w_{sn})^T$. Here, $\boldsymbol{h}_s$ is a least squares estimate of the parameters (ϕ_{s1}, ϕ_{s2}) in the AR(2) model for w_{st}, which has variance equal to $\sigma^2 H_s$. Denote the element in row l and column k of H_s as $H_{s,lk}$. At each iteration of the MCMC scheme Huerta & West (1999) propose new values of ϕ_{s1} and ϕ_{s2} as follows:

1. With probability 1/2, propose $\phi^*_{s1} = \phi^*_{s2} = 0$, i.e the reciprocal root pair has modulus zero.

2. With probability 1/2, propose $\phi^*_{s2} \sim \mathrm{N_{tr}}(h_{s2}, \sigma^2 H_{s,22})$ truncated to (-1,0) and $\phi^*_{s1}|\phi^*_{s2} \sim \mathrm{N_{tr}}(m^*_s, M^*_s)$ truncated to $(-2\sqrt{-\phi^*_{s2}}, 2\cos(2\pi/\lambda_u)\sqrt{-\phi^*_{s2}})$ where

$$m^*_s = h_{s1} + H_{s,12}(\phi_{s2} - h_{s2})/H_{s,22},$$

and

$$M^*_s = \sigma^2(H_{s,11} - H^2_{s,12}/H_{s,22}).$$

If the proposal distribution is assumed to have a multivariate normal distribution such that

$$(\phi^*_{s1}, \phi^*_{s2})^{\mathrm{T}} \sim \mathrm{N}_2(\boldsymbol{h}_s, \sigma^2 H_s),$$

then the proposal distribution for ϕ_2 used by Huerta & West (1999), described above, is the marginal distribution of ϕ^*_{s2}. The proposal distribution for ϕ_{s1} is then obtained using conditional theory of the multivariate normal distribution, where the proposed value of ϕ_{s1} is conditioned on the proposed value of ϕ_{s2}. Sampling ϕ^*_{s2} from its marginal proposal distribution first followed by sampling $\phi^*_{s1}|\phi^*_{s2}$ allows simpler truncation of the proposed values to satisfy the constraints detailed previously on r_s and λ_s. Each proposed value is then accepted with probability α where

$$\alpha = \min\left\{1, \frac{\pi(r^*_s)\pi(\lambda^*_s)L(\phi^*_{s1}, \phi^*_{s2}|\boldsymbol{w})q(\phi_{s1}, \phi_{s2}|\boldsymbol{w})}{\pi(r_s)\pi(\lambda_s)L(\phi_{s1}, \phi_{s2}|\boldsymbol{w})q(\phi^*_{s1}, \phi^*_{s2}|\boldsymbol{w})}\right\}.$$

Here, the proposed values r^*_s and λ^*_s can be obtained from ϕ^*_{s1} and ϕ^*_{s2} and $q(\cdot)$ denotes the density of the mixture proposal distribution given above. As in the real reciprocal case, an online relabelling algorithm is used to avoid label switching. Here, Huerta & West (1999) suggest that the complex reciprocal pairs can be ordered by either the size of the modulus or the size of the period. We choose to order the complex reciprocal pairs by the size of the period as we found it led to clearer separation of the posterior modes.

Full conditionals for π_{r0} and π_{c0}

Let r_0 be the number of sampled real reciprocals that are equal to zero. The full conditional distribution for π_{r0} is

$$\begin{aligned}\pi(\pi_{r0}|\psi\backslash\pi_{r0}) &\propto \pi(\pi_{r0}) \prod_{s=2C+1}^{p_{\max}} \pi(r_s|\pi_{r0}) \\ &= \prod_{s=2C+1}^{p_{\max}} \{\pi_{r0} I_0(r_s) + (1-\pi_{r0}) g_r(r_s)\} \\ &= \pi_{r0}^{r_0}(1-\pi_{r0})^{R-r_0}\end{aligned}$$

and so, conditional on the sampled reciprocals, the prior distribution for π_{r0} has a conjugate update such that

$$\pi_{r0}|\boldsymbol{\psi}\backslash\pi_{r0} \sim \text{Beta}\{r_0+1, (R-r_0+1)\}.$$

In a similar way, the prior distribution for π_{c0} has a conjugate update such that

$$\pi_{c0}|\boldsymbol{\psi}\backslash\pi_{c0} \sim \text{Beta}\{c_0+1, (C-c_0+1)\},$$

where c_0 is the number of sampled complex reciprocal pairs with modulus equal to zero. A Gibbs step can be used to sample from these full conditional distributions.

Full conditional for σ^2

Having sampled the reciprocals, α_s for $s = 1, \ldots, p_{\max}$, we can calculate the corresponding set of autoregressive parameters and as such we can find the conditional likelihood for σ^2 as follows

$$\begin{aligned}L(\sigma^2|\boldsymbol{\psi}\backslash\sigma^2, y_{1:n}) &\propto \prod_{t=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{-\frac{1}{2\sigma^2}(y_t - \phi_1 y_{t-1} - \ldots - \phi_{p_{\max}} y_{t-p_{\max}})^2\right\} \\ &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2}\left\{\sum_{t=1}^{n}(y_t - \phi_1 y_{t-1} - \ldots - \phi_{p_{\max}} y_{t-p_{\max}})^2\right\}\right]\end{aligned}$$

and so the conditional likelihood for σ^2 is proportional to an IG(a, b) density where

$$a = \frac{n}{2} - 1$$

and

$$b = \frac{1}{2}\sum_{t=1}^{n}(y_t - \phi_1 y_{t-1} - \ldots - \phi_{p_{\max}} y_{t-p_{\max}})^2.$$

Given the prior distribution for σ^2 is such that $\sigma^2 \sim \text{IG}(a_{\sigma^2}, b_{\sigma^2})$, the full conditional distribution for σ^2 is

$$\begin{aligned}\pi(\sigma^2 \mid \psi \backslash \sigma^2, y_{1:n}) &\propto (\sigma^2)^{-\frac{n}{2}} \exp\left[-\frac{1}{2\sigma^2}\left\{\sum_{t=1}^{n}(y_t - \phi_1 y_{t-1} - \ldots - \phi_{p_{\max}} y_{t-p_{\max}})^2\right\}\right] \\ &\quad \times (\sigma^2)^{-a_{\sigma^2}-1} \exp\left(-\frac{b_{\sigma^2}}{\sigma^2}\right) \\ &\propto (\sigma^2)^{-\frac{n}{2}-a_{\sigma^2}-1} \exp\left[-\frac{1}{\sigma^2}\left\{b_{\sigma^2} + \frac{1}{2}\sum_{t=1}^{n}(y_t - \phi_1 y_{t-1} - \ldots - \phi_{p_{\max}} y_{t-p_{\max}})^2\right\}\right].\end{aligned}$$

This results in a conjugate update such that

$$\sigma^2 | \boldsymbol{\psi} \backslash \sigma^2 \sim \text{IG}\left\{a_{\sigma^2} + \frac{n}{2}, b_{\sigma^2} + \frac{1}{2}\sum_{t=1}^{n}(y_t - \phi_1 y_{t-1} - \ldots - \phi_{p_{\max}} y_{t-p_{\max}})^2\right\},$$

which can be sampled using a Gibbs step.

Sampling of initial values

As univariate stationary autoregressive processes are time reversible, we can use the reverse time model

$$y_t = \sum_{j=1}^{p_{\max}} \phi_j y_{t+j} + \epsilon_t$$

for $t = 0, \ldots, -p_{\max} + 1$ to iteratively sample the latent initial values for the time series, where the autoregressive parameters $(\phi_1, \ldots, \phi_{p_{\max}})$ are computed from the current sample of reciprocals and the current sampled value of σ^2 is used to sample the innovations.

Simulation experiment

To investigate the behaviour of this posterior distribution in the idealised case where we know that the data have been generated from an autoregressive process of a known order, we applied these methods to simulated time series data. As in the partial autocorrelation reparameterisation, we considered data for values of $p \in \{1, 2, 3, 4\}$, using the same 10 data sets for each p as in the partial autocorrelation reparameterisation.

In our analysis, we allow a maximum of 5 real reciprocals and 5 pairs of complex reciprocals, giving a maximum model order of $p_{\max} = 15$. In the prior for σ^2, we use the same prior specification as under the partial autocorrelation reparameterisation and take $a_{\sigma^2} = 2.5$ and $b_{\sigma^2} = 1.5$. The hyperparameters in the prior distributions for all other parameters were discussed in Section 4.2.4. The MCMC scheme was run for 24,000 iterations with the first 10,000 discarded as burn-in. Whilst plots are not included here due to the large number of parameters across data sets, the usual graphical diagnostics

Method	$p = 1$	$p = 2$	$p = 3$	$p = 4$
Huerta and West	1.6305 (1973.65)	1.0300 (1249.92)	1.0148 (1204.76)	0.8858 (1072.84)

Table 4.2: Average minimum ESS/s across 10 data sets for each $p \in \{1, 2, 3, 4\}$ for the prior described in Huerta & West (1999) under the characteristic root parameterisation. The average minimum ESS obtained across the 10 data sets for all iterations is provided in brackets.

gave no evidence of any lack of convergence, however a larger number of burn-in iterations were needed before the sampler converged under this prior, in comparison to the priors in the partial autocorrelation parameterisation.

Let the effective order p^* of the process be the number of non-zero root reciprocals sampled at each draw of the MCMC scheme. Figures 4.8 and 4.9 depict the posterior mass functions for p^* for each of the 10 data sets for each value of p as a set of overlaid bar charts. Whilst the maximum permitted order was $p_{\max} = 15$, the bar charts have been truncated to only include up to lag 9 to aid visualisation, as there was no posterior mass above lag 9 in any data set. In nearly all cases the true order is the posterior mode, with considerable support. These results are promising and seem to demonstrate that the posterior behaves in the manner we would expect.

As for the priors in the partial autocorrelation parameterisation, we also calculated the average minimum ESS/s for the parameters across all 10 data sets for a given p. The results are provided in Table 4.2. For each value of p, the minimum ESS/s was similar to that obtained in the Kuo & Mallick (1998) approach under the partial autocorrelation parameterisation.

Unfortunately, whilst this prior has promising results in the univariate case, there is no natural extension of this parameterisation to the vector autoregressive case. An exception is discussed in the work of Huerta & Prado (2006) who extend Huerta & West (1999) by considering a multivariate generalisation of the characteristic equation. However, because this is only available when the autoregressive matrices are diagonal, their work is limited to the class of diagonal vector autoregressive processes. As such, due to the lack of a natural extension for the general class of vector autoregressions, we do not consider this parameterisation beyond this set of simulation experiments.

4.3 Application to EEG data

Whilst the simulation experiments in Sections 4.2.2 and 4.2.5 are useful to investigate the posterior behaviour of the discussed methods, of more interest is how these methods can be used to gain insight in real-life applications. As discussed in 2.2.5, conditional on the model order p, stationary autoregressions can be decomposed into latent series which account for low frequency trends, quasiperiodic behaviour and high frequency noise contributions.

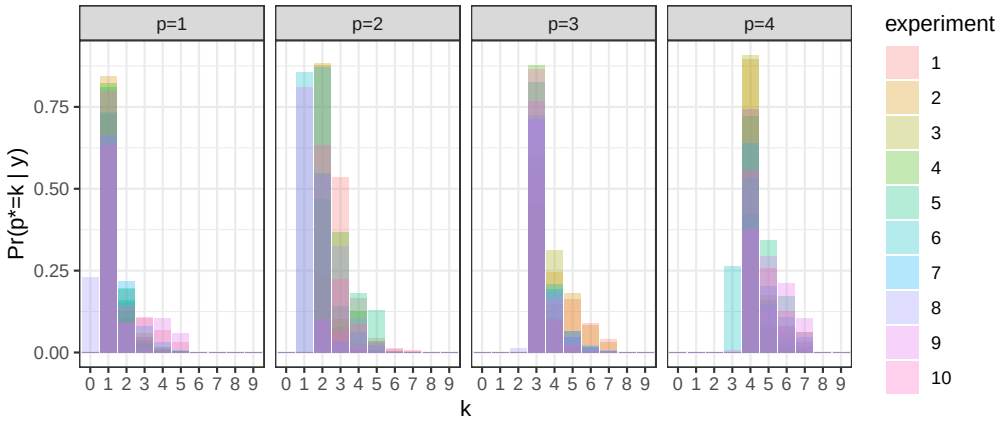


Figure 4.8: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $p \in \{1, 2, 3, 4\}$, with $n = 1000$, using the prior discussed by Huerta & West (1999) under the characteristic root parameterisation.

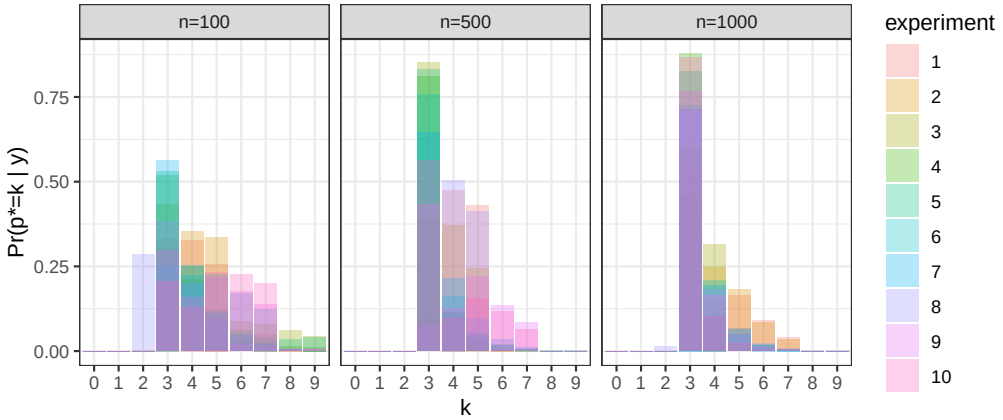


Figure 4.9: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $n \in \{100, 500, 1000\}$, with $p = 3$, using the prior discussed by Huerta & West (1999) under the characteristic root parameterisation.

These processes can give useful insights into real-life processes. In particular, these latent processes could give insight into biological rhythms in the brain when applied to EEG data, as discussed in Chapter 3. In this section we aim to investigate any underlying behaviour exhibited in our EEG recordings, treating the data as univariate.

For the simulated data sets we considered three methods for determining the order of univariate autoregressions. Two of these methods involved reparameterising the autoregressive model in terms of its partial autocorrelation function. Both the Kuo & Mallick (1998) and Barnett *et al.* (1996) approaches targeted the same posterior but the Kuo & Mallick (1998) approach ran much faster, and gave a much higher minimum ESS/s, and so

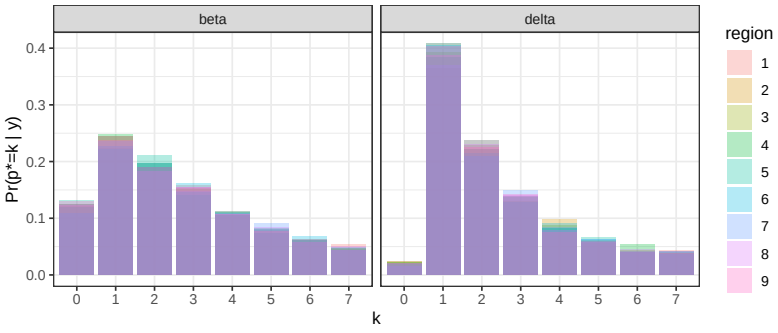
we will not apply the Barnett *et al.* (1996) method here. The approach defined by Huerta & West (1999) involved reparameterising the model in terms of the reciprocal roots of the characteristic equation. However, as discussed in Section 4.2.5, there is no natural extension of this method to the vector autoregressive case, and so we do not consider this method beyond the simulation experiments in Section 4.2.5. As such, in this section we apply the method adapted from Kuo & Mallick (1998) to the EEG data and discard the other two methods.

We have EEG recordings from four individuals living with epilepsy, referred to as individuals A, B, C and D. The number of regions m where data was recorded and the length n of the data segments to be analysed vary across individuals, and are detailed in Table 3.2. Treating each region for each individual as univariate time series data, we applied the MCMC scheme detailed in Algorithm 7 from the Kuo & Mallick (1998) approach to data from each of the regions for the four individuals, with a maximum order of $p_{\max} = 7$. In each case, the sampler was run with four chains for 4,000 burn-in iterations which was a sufficient number for all four chains to converge to the same mode. Following this, 4,000 iterations were sampled and retained for analysis which was sufficient to obtain a minimum effective sample size of 1,000. We retained the same prior specification as in the simulation experiments in Section 4.2.2 with $a_{\rho_s} = b_{\rho_s} = 1$, $a_{\sigma^2} = 2.5$ and $b_{\sigma^2} = 1.5$. The values of p_{I_s} were given in (4.14), where we again take $\alpha = 3$. Individual C has one missing data point in each of the regions. This missing data point was inferred at each iteration by sampling its value from the autoregressive model using the current values of Φ and σ^2.

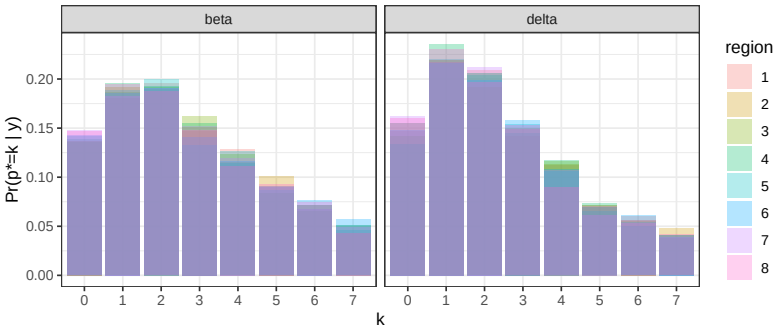
For each region for each of the individuals, we used the draws from the posterior distribution to calculate the posterior mass function for the effective order p^*, taking p^* at each iteration to be the lag of the maximum non-zero partial autocorrelation in the current draw. For each individual, the posterior mass functions, displayed in Figure 4.10, are presented as a set of overlaid bar charts, one for each region. For each individual, the posterior mass function is very similar across all regions. The posterior mode varies across individuals, typically taking values of either 1 or 2, though there is some posterior mass on all values between 0 and $p_{\max} = 7$. In Section 3.3, we used the Box-Jenkins approach to identify a set of autoregressive models that appeared to give a good fit to the data in the delta band of individual A. Whilst the posterior mode obtained here is lower than the order identified using the Box-Jenkins approach, the orders identified in Section 3.3 all have some posterior mass attached to them. However, the output here suggests that perhaps a lower order model could be appropriate, resulting in a less complicated choice of model than obtained using the Box-Jenkins approach.

Having obtained a posterior sample of the effective order at each iteration of the MCMC scheme, we can condition on the order at each iteration to decompose the time

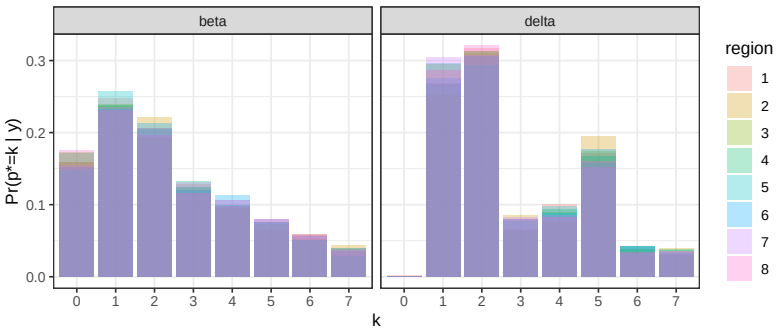
series into its p latent processes, one for each of the roots of the characteristic equation. The processes corresponding to any complex roots of the characteristic equation are of particular interest as they can provide insight into hidden quasi-periodic behaviour of the process. The complex roots of the characteristic equation are not identifiable, as the characteristic equation is the same under any permutation of the order of the labels $s = 1, \ldots, C$, so we apply an ordering constraint such that the complex roots are ordered by decreasing period. Figure 4.11 contains box plots of the posterior samples for the period of the quasi-periodic series with the largest period, for each band in each region for each individual. In all cases, the period of the quasi-periodic series with the largest period was around 3.5 minutes. This was the case across all regions in all individuals, in both the beta and delta bands. The similarity in the periods across the beta and delta bands indicate that there is a global cycle in the band power pattern, rather than a local cycle within a specific band. The results of this investigation into brain rhythms in regions of the brain will be discussed further in Chapter 6, where comparisons will be made to the results of analysis of the data when treating the data as multivariate.



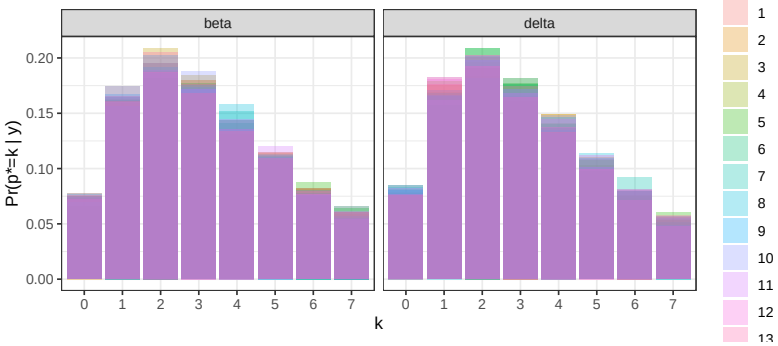
(a)



(b)



(c)



(d)

Figure 4.10: Overlaid posterior mass functions for the effective order p^* for each region in individuals (a) A, (b) B, (c) C and (d) D for both the beta (left) and delta (right) bands. The region names for each individual are detailed in Table 3.1 and depicted in Figure 3.1.

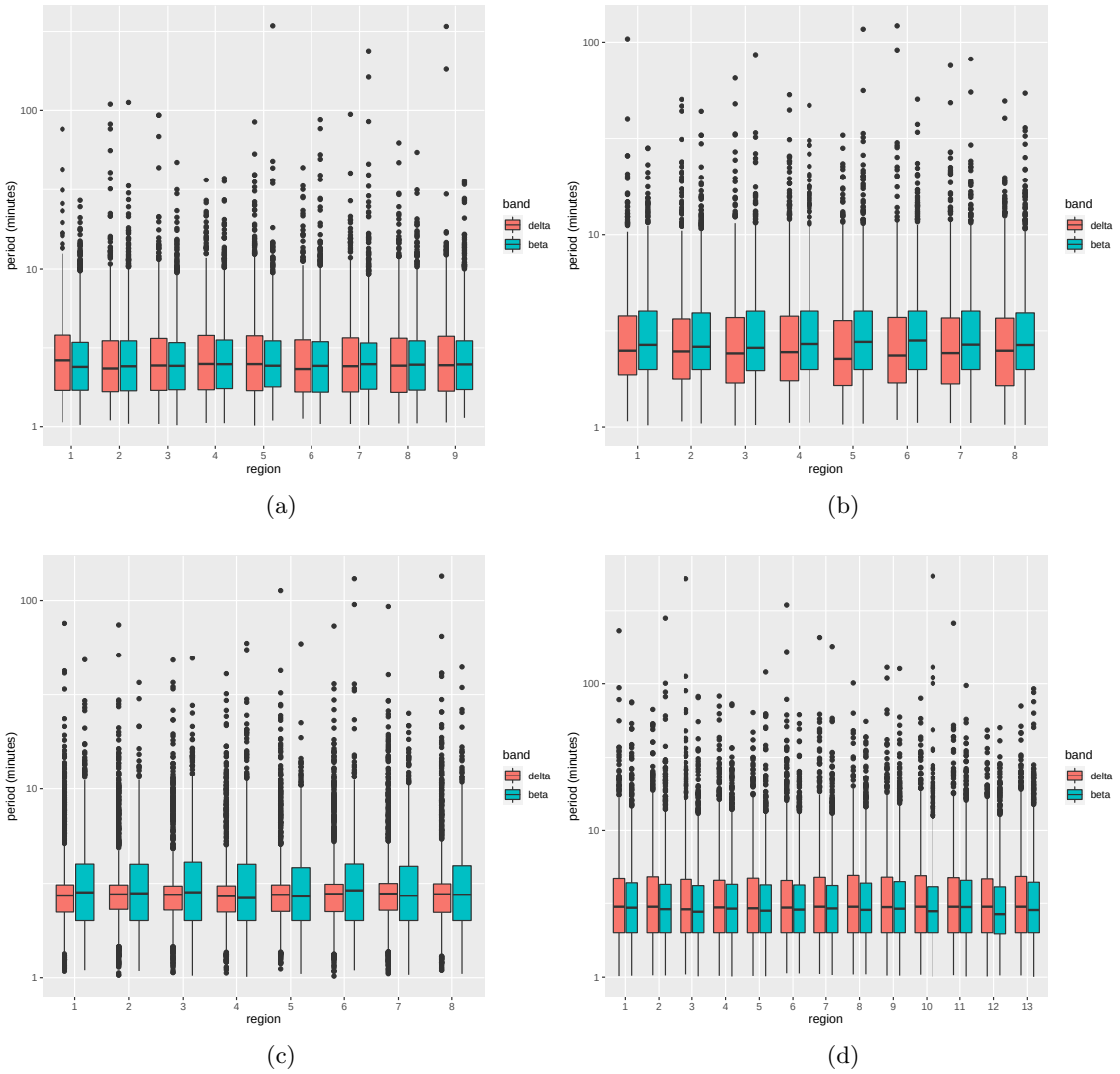


Figure 4.11: Box plots of posterior sample of the period in the quasi-periodic series with the largest period, for each region in individuals (a) A, (b) B, (c) C and (d) D for both the beta (blue) and delta (pink) bands. The region names for each individual are detailed in Table 3.1 and the locations of the regions in the brain are depicted in Figure 3.1.

Chapter 5

Modelling stationary vector autoregressions

The main objective of this thesis is to develop methods to determine the order of stationary vector autoregressions. In Chapter 4 we discussed methods for determining the order of stationary univariate autoregressive processes. In this chapter, we consider extensions to these methods for the vector autoregressive case.

5.1 Reparameterisation over the stationary region

As discussed in Chapter 2, an m-variate Gaussian vector autoregression of order p is stationary if and only if the roots of the equation

$$\det\{\phi(u)\} = 0$$

lie outside the unit circle, where

$$\phi(u) = I_m - \phi_1 u - \ldots - \phi_p u^p, \qquad u \in \mathbb{C}$$

is the characteristic polynomial. This condition constrains the parameter space to a region referred to as the stationary region, $\mathcal{C}_{p,m}$, in which the geometry becomes increasingly complex as either p or m increase. For example, consider the simplest vector autoregressive process where $m = 2$ and $p = 1$:

$$\boldsymbol{y}_t = \phi_1 \boldsymbol{y}_{t-1} + \boldsymbol{\varepsilon_t}$$

where $\boldsymbol{\epsilon_t} \sim \mathrm{N}_2(\mathbf{0}, \Sigma)$. The complex geometry of the stationary region for this case is visualised in Figure 5.1. With no standard distributions over $\mathcal{C}_{p,m}$, this complicates the process of specifying a prior that conveys meaningful information, for example, concerning

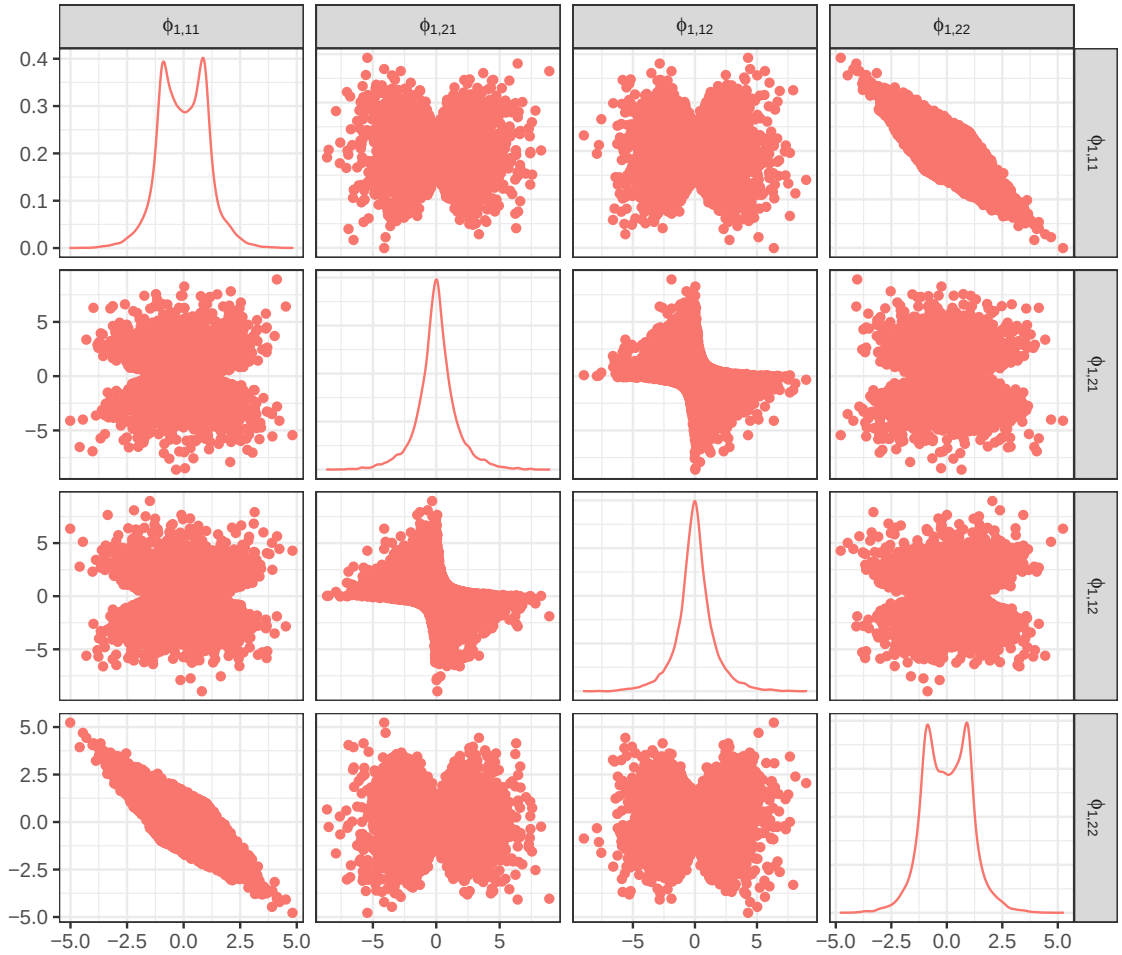


Figure 5.1: Draws from a diffuse distribution over the stationary region for the elements of ϕ_1 in the $\text{VAR}_2(1)$ model. Plots along the diagonal show marginal densities, but of interest here is the plots off the diagonal which depict the bivariate densities for all pairs of parameters.

the relative sizes of the autocorrelations at different lags. Moreover, it is difficult to design an efficient MCMC sampler which targets a distribution with support constrained to $\mathcal{C}_{p,m}$. In recent work, Heaps (2023) proposed a solution which addresses both issues, reparameterising the model over the stationary region in terms of a set of interpretable, unconstrained parameters. The reparameterisation involves two bijective mappings. First, the original model parameters $(\Sigma, \Phi) \in \mathcal{S}_m^+ \times \mathcal{C}_{p,m}$ are mapped to a new parameter set $\{\Sigma, (P_1, \ldots, P_p)\} \in \mathcal{S}_m^+ \times \mathcal{V}^p$ in which $\mathcal{V}$ denotes the subset of matrices in $M_{m\times m}(\mathbb{R})$ whose singular values are less than one. The matrix P_s is referred to as the s-th partial autocorrelation matrix. This reparameterisation is a multivariate generalisation of the mapping from the autoregressive parameters to the partial autocorrelations $\rho_1, \ldots, \rho_p$ in the case of univariate autoregressions, discussed in Chapter 4. The matrix P_{s+1} is defined

as the conditional cross-covariance matrix between $\boldsymbol{y}_{t+1}$ and $\boldsymbol{y}_{t-s}$ given $\boldsymbol{y}_t, \ldots, \boldsymbol{y}_{t-s+1}$ which has been standardised through

$$P_{s+1} = \Sigma_s^{-1/2} \text{Cov}(\boldsymbol{y}_{t+1}, \boldsymbol{y}_{t-s} | \boldsymbol{y}_t, \ldots, \boldsymbol{y}_{t-s+1}) \Sigma_s^{*\,-1/2},$$

$s = 0, \ldots, p-1$, in which Σ_s and Σ_s^* are the conditional variances

$$\Sigma_s = \text{Var}(\boldsymbol{y}_{t+1} | \boldsymbol{y}_t, \ldots, \boldsymbol{y}_{t-s+1}) \quad \text{and} \quad \Sigma_s^* = \text{Var}(\boldsymbol{y}_{t-s} | \boldsymbol{y}_{t-s+1}, \ldots, \boldsymbol{y}_t)$$

and $\Sigma^{1/2}$ denotes the symmetric matrix-square-root.

The forward mapping from $(\Sigma, \Phi) \in \mathcal{S}_m^+ \times \mathcal{C}_{p,m}$ to $\{\Sigma, (P_1, \ldots, P_p)\} \in \mathcal{S}_m^+ \times \mathcal{V}^p$ proceeds in the same way as the mapping originally described in Ansley & Kohn (1986). However, whilst Ansley & Kohn (1986) use Cholesky factorisations in the matrix-square-roots, we follow Heaps (2023) in using symmetric matrix-square-roots. As discussed in Heaps (2023), using symmetric matrix-square-roots results in a reparameterisation in which we can construct priors which are closed under orthogonal transformation of the observation vectors. This is beneficial as it permits the use of exchangeable priors which are invariant to the ordering of the elements in the observation vector. The full algorithm for this mapping is below.

1. For $i = 0, \ldots, p$, compute the autocovariances $\Gamma_i = \text{Cov}(\boldsymbol{y}_t, \boldsymbol{y}_{t+i})$ from (Σ, Φ). The autocovariances $\Gamma_0, \ldots, \Gamma_{p-1}$ can be found by representing the $\text{VAR}_m(p)$ process as a $\text{VAR}_{mp}(1)$ process, with an autoregressive matrix denoted ϕ and error variance matrix denoted Σ', then computing its stationary variance Γ_0'. This results in a discrete Lyapunov equation (an equation of the form $\Gamma_0' = \phi \Gamma_0' \phi^{\mathrm{T}} + \Sigma'$) which can be solved using vectorisation and Kronecker product operators. The autocovariances $\Gamma_0, \ldots, \Gamma_{p-1}$ are the $m \times m$ matrices obtained from the first m columns of Γ_0'. Then, Γ_p can be found using the Yule-Walker equations for the order p process ($\Gamma_p = \phi_1 \Gamma_{p-1} + \ldots + \phi_p \Gamma_0$). Further details on such calculations can be found in Chapter 2 of Luetkepohl (2005).

2. Compute the partial autocorrelation matrices $(P_1, \ldots, P_p)$ from Φ and $(\Gamma_0, \ldots, \Gamma_p)$:

   (a) Initialise: construct $\Sigma_0 = \Sigma_0^* = \Gamma_0$ then calculate the symmetric matrix-square-root factorisations such that

   $$\Sigma_0 = \Sigma_0^{\frac{1}{2}} \Sigma_0^{\frac{1}{2}} = \Sigma_0^{*\frac{1}{2}} \Sigma_0^{*\frac{1}{2}} = \Sigma_0^*.$$

   (b) Recursion: for each $s = 0, \ldots, p-1$:

i. Compute $\phi_{s+1,s+1}$ using

$$\phi_{s+1,s+1} = (\Gamma_{s+1}^{\mathrm{T}} - \phi_{s1}\Gamma_s^{\mathrm{T}} - \cdots - \phi_{ss}\Gamma_1^{\mathrm{T}})\Sigma_s^{*-1}$$

and $\phi_{s+1,s+1}^*$ using

$$\phi_{s+1,s+1}^* = (\Gamma_{s+1} - \phi_{s1}^*\Gamma_s - \cdots - \phi_{ss}^*\Gamma_1)\Sigma_s^{-1}.$$

In the case where $s = 0$ this simplifies to give

$$\phi_{11} = \Gamma_1^{\mathrm{T}}\Sigma_0^{*-1} = \Gamma_1^{\mathrm{T}}\Gamma_0^{-1}$$

and

$$\phi_{11}^* = \Gamma_1\Sigma_0^{-1} = \Gamma_1\Gamma_0^{-1}.$$

ii. If $s > 0$, for $i = 1, \ldots, s$, compute $\phi_{s+1,i}$ using

$$\phi_{s+1,i} = \phi_{si} - \phi_{s+1,s+1}\phi_{s,s-i+1}^*$$

and $\phi_{s+1,i}^*$ using

$$\phi_{s+1,i}^* = \phi_{si}^* - \phi_{s+1,s+1}^*\phi_{s,s-i+1}.$$

iii. Compute the $(s+1)$th partial autocorrelation P_{s+1} using either

$$P_{s+1} = \Sigma_s^{-\frac{1}{2}}\phi_{s+1,s+1}\Sigma_s^{*\frac{1}{2}}$$

or

$$P_{s+1} = (\Sigma_s^{*-\frac{1}{2}}\phi_{s+1,s+1}^*\Sigma_s^{\frac{1}{2}})^{\mathrm{T}}.$$

iv. If $s < p - 1$, compute Σ_{s+1} using

$$\Sigma_{s+1} = \Gamma_0 - \phi_{s+1,1}\Gamma_1 - \cdots - \phi_{s+1,s+1}\Gamma_{s+1}$$

and Σ_{s+1}^* using

$$\Sigma_{s+1}^* = \Gamma_0 - \phi_{s+1,1}^*\Gamma_1^{\mathrm{T}} - \cdots - \phi_{s+1,s+1}^*\Gamma_{s+1}^{\mathrm{T}}.$$

Then calculate the symmetric matrix-square-roots such that

$$\Sigma_{s+1} = \Sigma_{s+1}^{\frac{1}{2}}\Sigma_{s+1}^{\frac{1}{2}}$$

and

$$\Sigma_{s+1}^* = \Sigma_{s+1}^{*\frac{1}{2}}\Sigma_{s+1}^{*\frac{1}{2}}.$$

There are two recursions in the backwards mapping from $\{\Sigma, (P_1, \ldots, P_p)\} \in \mathcal{S}_m^+ \times \mathcal{V}^p$ to $(\Sigma, \Phi) \in \mathcal{S}_m^+ \times \mathcal{C}_{p,m}$. The first was developed by Heaps (2023) to update the algorithm in Ansley & Kohn (1986) to allow symmetric matrix-square-roots, whilst the second is based on Lemma 2.1 in Ansley & Kohn (1986). The full mapping algorithm is as follows.

1. Compute the stationary variance Γ_0 from $\{\Sigma, (P_1, \ldots, P_p)\}$:
   (a) Initialise: let $\Sigma_p = \Sigma$ with the corresponding symmetric matrix-square-root factorisation being such that $\Sigma_p = \Sigma_p^{\frac{1}{2}} \Sigma_p^{\frac{1}{2}}$.
   (b) Recursion: for $s = p-1, \ldots, 0$, construct the symmetric matrix $\Sigma_s^{\frac{1}{2}}$ such that
   $$\Sigma_{s+1} = \Sigma_s^{\frac{1}{2}} (I_m - P_{s+1} P_{s+1}^{\mathrm{T}}) \Sigma_s^{\frac{1}{2}}, \tag{5.1}$$
   then compute $\Sigma_s = \Sigma_s^{\frac{1}{2}} \Sigma_s^{\frac{1}{2}}$.
   (c) Output: set $\Gamma_0 = \Sigma_0$.
2. Compute the autoregressive coefficient matrices in Φ from $(P_1, \ldots, P_p)$ and Γ_0:
   (a) Initialise: set $\Sigma_0 = \Sigma_0^* = \Gamma_0$ with corresponding symmetric matrix-square-root factorisation such that $\Sigma_0 = \Sigma_0^{\frac{1}{2}} \Sigma_0^{\frac{1}{2}} = \Sigma_0^{*\frac{1}{2}} \Sigma_0^{*\frac{1}{2}} = \Sigma_0^*$.
   (b) Recursion: for each $s = 0, \ldots, p-1$:
      i. Compute $\phi_{s+1,s+1}$ using
      $$\phi_{s+1,s+1} = \Sigma_s^{\frac{1}{2}} P_{s+1} \Sigma_s^{*-\frac{1}{2}}$$
      and $\phi_{s+1,s+1}^*$ using
      $$\phi_{s+1,s+1}^* = \Sigma_s^{*\frac{1}{2}} P_{s+1}^{\mathrm{T}} \Sigma_s^{-\frac{1}{2}}.$$
      ii. If $s > 0$, for $i = 1, \ldots, s$ compute $\phi_{s+1,i}$ using
      $$\phi_{s+1,i} = \phi_{si} - \phi_{s+1,s+1} \phi_{s,s-i+1}^*$$
      and $\phi_{s+1,i}^*$ using
      $$\phi_{s+1,i}^* = \phi_{si}^* - \phi_{s+1,s+1}^* \phi_{s,s-i+1}.$$
      iii. Compute Σ_{s+1} using
      $$\Sigma_{s+1} = \Sigma_s - \phi_{s+1,s+1} \Sigma_s^* \phi_{s+1,s+1}^{\mathrm{T}}$$
      and Σ_{s+1}^* using
      $$\Sigma_{s+1}^* = \Sigma_s^* - \phi_{s+1,s+1}^* \Sigma_s \phi_{s+1,s+1}^{*\mathrm{T}}.$$

Then calculate the symmetric matrix-square-roots such that

$$\Sigma_{s+1} = \Sigma_{s+1}^{\frac{1}{2}} \Sigma_{s+1}^{\frac{1}{2}}$$

and

$$\Sigma_{s+1}^{*} = \Sigma_{s+1}^{*\frac{1}{2}} \Sigma_{s+1}^{*\frac{1}{2}}.$$

iv. Compute Γ_{s+1} using

$$\Gamma_{s+1}^{\mathrm{T}} = \phi_{s+1,s+1}\Sigma_{s}^{*} + \phi_{s1}\Gamma_{s}^{\mathrm{T}} + \cdots + \phi_{ss}\Gamma_{1}^{\mathrm{T}}. \tag{5.2}$$

(c) Output: take $\phi_i = \phi_{pi}$ for $i = 1, \ldots, p$ and from step 1(a) we have $\Sigma = \Sigma_p$.

Proofs of the forward and reverse mappings between the autoregressive matrices and the partial autocorrelation matrices are provided in the supplementary materials of Heaps (2023). Within these proofs, Heaps (2023) discusses how to solve (5.1) for $\Sigma_s^{\frac{1}{2}}$. In particular, if $\Sigma_s^{\frac{1}{2}}$ is the symmetric matrix-square-root of Σ_s, then denote the symmetric matrix-square-root of $(I_m - P_{s+1}P_{s+1}^{\mathrm{T}})$ by B_{s+1}^{-1} and solve (5.1) for $\Sigma_s^{\frac{1}{2}}$ using

$$\Sigma_s^{\frac{1}{2}} = B_{s+1}(B_{s+1}^{-1}\Sigma_{s+1}B_{s+1}^{-1})^{\frac{1}{2}}B_{s+1}.$$

Unfortunately, there are also no standard distributions over the stationary region under this parameterisation, in which the stationarity condition demands that all partial autocorrelation matrices, $P_1, \ldots P_p$, have singular values of less than one. As such Heaps (2023) considers a second transformation which maps each partial autocorrelation matrix $P \in \mathcal{V}$ to an unconstrained square matrix $A \in M_{m\times m}(\mathbb{R})$. This mapping is a generalisation of a mapping originally defined by Ansley & Kohn (1986) but modified to allow alternative matrix-square-roots to the Cholesky factorisations used by Ansley & Kohn (1986). In particular, we follow Heaps (2023) in using symmetric matrix-square-roots. To map from a partial autocorrelation matrix P to its corresponding unconstrained matrix A using the mapping defined in Heaps (2023) let

$$B^{-1}B^{-1\mathrm{T}} = I_m - PP^{\mathrm{T}}$$

be a matrix-square-root factorisation and then let $A = BP$. Similarly, for the reverse mapping from the unconstrained matrices A to the partial autocorrelation matrices P let

$$BB^{\mathrm{T}} = I_m + AA^{\mathrm{T}}$$

and then let $P = B^{-1}A$. The use of symmetric matrix-square-roots in this mapping leads

to the simple equations

$$A = (I_m - PP^{\mathrm{T}})^{-\frac{1}{2}}P$$

and

$$P = (I_m + AA^{\mathrm{T}})^{-\frac{1}{2}}A.$$

This results in a set of transformed partial autocorrelation matrices which have a more natural interpretation than would be the case if Cholesky factorisations were used, particularly through the relative sizes of the partial autocorrelation matrices across lags. Specifically, denote the singular value decomposition of P by

$$P = URV^{\mathrm{T}},$$

where $R = \mathrm{diag}(r_1, \ldots, r_m)$ and the singular values satisfy $1 > r_1 \geq r_2 \geq \cdots \geq r_m \geq 0$. The corresponding factorisation of A is given by

$$\begin{aligned}
A &= (I_m - PP^{\mathrm{T}})^{-1/2}P \\
&= (I_m - URV^{\mathrm{T}}VRU^{\mathrm{T}})^{-1/2}URV^{\mathrm{T}} \\
&= (UU^{\mathrm{T}} - UR^2U^{\mathrm{T}})^{-1/2}URV^{\mathrm{T}} \\
&= \left\{U(I_m - R^2)U^{\mathrm{T}}\right\}^{-1/2}URV^{\mathrm{T}} \\
&= \left\{(U^{\mathrm{T}})^{-1}(I_m - R^2)^{-1}U^{-1}\right\}^{1/2}URV^{\mathrm{T}} \\
&= \left\{U(I_m - R^2)^{-1}U^{\mathrm{T}}\right\}^{1/2}URV^{\mathrm{T}} \\
&= U(I_m - R^2)^{-1/2}U^{\mathrm{T}}URV^{\mathrm{T}} \\
&= U(I_m - R^2)^{-1/2}RV^{\mathrm{T}} \\
&= U\tilde{R}V^{\mathrm{T}}
\end{aligned}$$

where

$$\tilde{R} = (I_m - R^2)^{-\frac{1}{2}}R = \mathrm{diag}(\tilde{r}_1, \ldots, \tilde{r}_m)$$

and

$$\tilde{r}_i = \frac{r_i}{(1 - r_i^2)^{1/2}} \geq 0$$

for $i = 1, \ldots, m$. Therefore, as P and A share the same singular vectors and the singular values of A are a strictly increasing function of the singular values of P, the second transformation can be regarded as an orientation-preserving mapping which simply scales the singular values of P from $[0, 1)$ to the positive real line. As such, the relative sizes of the partial autocorrelation matrices across lags are retained under the unconstrained parameterisation, which is a useful property that will be discussed further in Section 5.3.1. As the transformed partial autocorrelation matrices, $A_1, \ldots, A_p$, are unconstrained,

this reparameterisation results in a set of parameters $\{(A_1, \ldots, A_p), \Sigma\}$ for which prior specification is simpler, as there are is a wide range of distributional choices available for the unconstrained A matrices or their corresponding vectorisations, $\text{vec}(A)$, including the multivariate normal distribution or the multivariate t-distribution. As such, in the remainder of this chapter we consider possible choices of prior distributions under this parameterisation, considering various implementations of the normal distribution for its simplicity.

5.2 Enforcing stationarity when p is known

Before considering the case where the order of the process p is unknown, we first consider Bayesian inference for a VAR(p) process with a known p. For the known p case Heaps (2023) discusses methods for Bayesian inference of vector autoregressions when the model is reparameterised in terms of the transformed partial autocorrelation matrices, $A_1, \ldots, A_p$, as described above. In this section we aim to summarise this work, describing methods for sampling from the posterior densities of $A_1, \ldots, A_p$ and Σ using an exchangeable prior distribution described in Heaps (2023).

5.2.1 Prior distribution

The unknown parameters in our model consist of the transformed partial autocorrelation matrices $A_1, \ldots, A_p$ and the error variance Σ. Denoting the collection of any unknown hyperparameters in the prior for $A_1, \ldots, A_p$ by $\boldsymbol{\vartheta}$, we adopt an overall prior specification of the form

$$\pi(\Sigma, A_1, \ldots, A_p, \boldsymbol{\vartheta}) = \pi(\Sigma)\pi(\boldsymbol{\vartheta}) \prod_{s=1}^{p} \pi(A_s|\boldsymbol{\vartheta}). \tag{5.3}$$

Choices of prior distribution for the parameters $A_1, \ldots A_p$ and Σ are discussed below.

Prior distribution for Σ

Various options are available for the error variance matrix Σ and distributions which offer the property of invariance with respect to the order of the variables in the observation vector are discussed in Heaps (2023). In the applications in this thesis, we use one such distribution, taking Σ to be inverse Wishart, with degrees of freedom ν and a scale matrix S that has a common element on the diagonal and a common element off the diagonal:

$$\Sigma \sim \text{W}^{-1}(S, \nu).$$

An alternative choice of prior suggested by Heaps (2023) could be the multivariate normal distribution for the matrix-logarithm of Σ.

Prior distribution for A_s matrices

As for Σ, an exchangeable prior distribution for $A_1, \ldots, A_p$ is useful to represent invariance in a modeller's prior beliefs with respect to the order of the variables in the observation vector. In addition, Heaps (2023) discusses how a prior which allows borrowing strength between the diagonal elements and the off-diagonal elements of each A_s may be desirable due to the potential for a very large number of parameters ($O(m^2)$) in a vector autoregression. Allowing borrowing strength would permit sharing of information between each of the diagonal elements to improve inference when there are a large number of parameters. Similarly, information could be shared between each of the off-diagonal elements. To satisfy these requirements, Heaps (2023) adopts a hierarchical prior for the diagonal and off-diagonal elements of A_s such that

$$
\begin{aligned}
a_{s,ii}|\mu_{s1}, \omega_{s1} &\sim \mathrm{N}(\mu_{s1}, \omega_{s1}^{-1}) \text{ independently for } i = 1, \ldots, m, \\
a_{s,ij}|\mu_{s2}, \omega_{s2} &\sim \mathrm{N}(\mu_{s2}, \omega_{s2}^{-1}) \text{ independently for } i, j = 1, \ldots, m \text{ with } i \neq j
\end{aligned}
$$

and

$$
\begin{aligned}
\mu_{s1} &\sim \mathrm{N}(e_{s1}, f_{s1}^2), & \omega_{s1} &\sim \mathrm{Gam}(g_{s1}, h_{s1}), \\
\mu_{s2} &\sim \mathrm{N}(e_{s2}, f_{s2}^2), & \omega_{s2} &\sim \mathrm{Gam}(g_{s2}, h_{s2}).
\end{aligned}
$$

Here, $a_{s,ii}$ denotes the i-th diagonal element of A_s and $a_{s,ij}$ denotes the (i,j)-th (off-diagonal) element of A_s. In order to consider suitable choices of hyperparameters for this prior, we can consider the marginal expectations, variances and correlations which are such that:

$$
\begin{aligned}
\mathrm{E}(a_{s,ii}) &= e_{s1}, \\
\mathrm{Var}(a_{s,ii}) &= f_{s1}^2 + \frac{h_{s1}}{(g_{s1} - 1)}, \quad g_{s1} > 1, \qquad (5.4) \\
\mathrm{Cor}(a_{s,ii}, a_{s,jj}) &= \frac{f_{s1}^2 (g_{s1} - 1)}{f_{s1}^2 (g_{s1} - 1) + h_{s1}}, \quad i \neq j. \qquad (5.5)
\end{aligned}
$$

Similarly, for the off-diagonal elements, marginally

$$
\begin{aligned}
\mathrm{E}(a_{s,ij}) &= e_{s2}, \\
\mathrm{Var}(a_{s,ij}) &= f_{s2}^2 + \frac{h_{s2}}{(g_{s2} - 1)}, \quad g_{s2} > 1, \qquad (5.6) \\
\mathrm{Cor}(a_{s,ij}, a_{s,kl}) &= \frac{f_{s2}^2 (g_{s2} - 1)}{f_{s2}^2 (g_{s2} - 1) + h_{s2}}, \quad i \neq k \text{ and/or } j \neq l, \qquad (5.7)
\end{aligned}
$$

where $i \neq j$ and $k \neq l$. Heaps (2023) discusses how these moments can be used as guidelines for the choice of hyperparameters e_{si}, f_{si}, g_{si} and h_{si}, $i = 1, 2$. In particular, we can set $e_{s1} = e_{s2} = 0$ to centre our prior for each of the elements on zero. Additionally, the Supplementary Materials of Heaps (2023) discuss how allowing the marginal variance for the elements of the A_s matrices to be too large can result in multimodality in the prior induced for the elements in the partial autocorrelation matrices P_s. Heaps (2023) presents a table of the maximum marginal standard deviation for the elements of the A_s matrices for which multimodality in the prior is avoided for different values of m, and we follow these guidelines when considering suitable hyperparameter values. Furthermore, a high value for the correlations between the elements of the A_s matrices can be chosen to encourage borrowing strength, though the correlations shouldn't be so high that complete shrinkage to the common value is near-enforced. For example, Heaps (2023) sets the correlations equal to 0.7, and we follow this in the applications in this thesis. Using Equations (5.4) - (5.7) in combination with the table in Heaps (2023), values of f_{si}, g_{si} and h_{si}, $i = 1, 2$, can be chosen to ensure that the marginal standard deviation of the elements of A_s is small enough to avoid multimodality whilst also ensuring an appropriate level of correlation between the elements of the A_s matrices.

We use this exchangeable prior when proceeding with posterior inference for a $\text{VAR}_m(p)$ model with a known p. The unknown hyperparameters in the prior, $(\mu_{s1}, \mu_{s2}, \omega_{s1}, \omega_{s2})$ for $s = 1, \ldots, p$, are denoted collectively as $\boldsymbol{\vartheta}$ in the prior (5.3) and the posterior discussed in the next section (5.8).

5.2.2 Posterior inference

For $i \leq j$, denote by $\boldsymbol{y}_{i:j}$ the time series $\boldsymbol{y}_i, \ldots, \boldsymbol{y}_j$. The likelihood for a series of n observations, $\boldsymbol{y}_{1:n}$, from a zero-mean $\text{VAR}_m(p)$ process can be expressed as

$$p(\boldsymbol{y}_{1:n} \mid \Sigma, \Phi) = p(\boldsymbol{y}_{1:p} \mid \Sigma, \Phi) \prod_{t=p+1}^{n} p(\boldsymbol{y}_t \mid \boldsymbol{y}_{(t-p):(t-1)}, \Sigma, \Phi)$$

in which

$$\boldsymbol{Y}_t \mid \boldsymbol{y}_{(t-p):(t-1)}, \Sigma, \Phi \sim \text{N}_m \left(\sum_{i=1}^{p} \phi_i \boldsymbol{y}_{t-i} \,,\, \Sigma \right)$$

and the initial distribution is

$$(\boldsymbol{Y}_1^{\text{T}}, \ldots, \boldsymbol{Y}_p^{\text{T}})^{\text{T}} \mid \Sigma, \Phi \sim \text{N}_{mp}(\boldsymbol{0}, G).$$

Here G is given by

$$G = \begin{pmatrix} \Gamma_0 & \Gamma_1 & \cdots & \Gamma_{p-1} \\ \Gamma_1^{\mathrm{T}} & \Gamma_0 & \cdots & \Gamma_{p-2} \\ \vdots & \vdots & \ddots & \vdots \\ \Gamma_{p-1}^{\mathrm{T}} & \Gamma_{p-2}^{\mathrm{T}} & \cdots & \Gamma_0 \end{pmatrix},$$

where the matrices $\Gamma_0, \ldots, \Gamma_{p-1}$ are available as by-products of the recursive mapping between the partial autocorrelation matrices and the original model parameters, see Equation (5.2).

Regarding the likelihood as a function of the new parameters and combining it with the prior (5.3) via Bayes' theorem yields the posterior distribution as

$$\pi(\Sigma, A_1, \ldots, A_p, \boldsymbol{\vartheta} \mid \boldsymbol{y}_{1:n}) \propto p(\boldsymbol{y}_{1:n} \mid \Sigma, A_1, \ldots, A_p)\pi(\Sigma)\pi(\boldsymbol{\vartheta}) \prod_{s=1}^{p} \pi(A_s \mid \boldsymbol{\vartheta}). \tag{5.8}$$

The full posterior distribution is of a complicated nature which is not tractable. However, by considering the full conditional distributions of each parameter we can consider a Metropolis-Hastings algorithm with one-at-a-time updates to sample from the posterior distribution. The full conditional distributions for each parameter are considered below with the Metropolis-within-Gibbs algorithm outlined in Algorithm 8.

Full conditional for Σ

The full conditional distribution for Σ is:

$$\begin{aligned} \pi(\Sigma|\boldsymbol{y}_{1:n}, A_1, \ldots, A_p, \boldsymbol{\vartheta}) &\propto \pi(\Sigma)p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta}) \\ &\propto |\Sigma|^{-(\nu+m+1)/2} \exp\left\{-\frac{1}{2}\mathrm{tr}(S\Sigma^{-1})\right\} \\ &\times |G|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_{1:p}^{\mathrm{T}} G^{-1} \boldsymbol{y}_{1:p})\right\} \\ &\times |\Sigma|^{-(n-p)/2} \\ &\times \exp\left[-\frac{1}{2}\sum_{t=p+1}^{n}\left\{\left(\boldsymbol{y}_t - \sum_{i=1}^{p}\phi_i \boldsymbol{y}_{t-i}\right)^{\mathrm{T}} \Sigma^{-1} \left(\boldsymbol{y}_t - \sum_{i=1}^{p}\phi_i \boldsymbol{y}_{t-i}\right)\right\}\right]. \end{aligned}$$

As the stationary variance matrix G depends on Σ in a non-trivial way and the mapping from $A_1, \ldots A_p$ to $\phi_1, \ldots, \phi_p$ depends on Σ, this full conditional distribution is not a standard distribution. Therefore, we use a Metropolis-Hastings step to sample Σ with acceptance probability

$$\alpha(\Sigma^*, \Sigma) = \min\left\{1, \frac{p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma^*, \boldsymbol{\vartheta})\pi(\Sigma^*)q_\Sigma(\Sigma|\Sigma^*)}{p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta})\pi(\Sigma)q_\Sigma(\Sigma^*|\Sigma)}\right\},$$

where an inverse Wishart distribution with the current value of Σ as the mean is used as a random walk proposal distribution for $q_{\Sigma}(\cdot|\cdot)$.

Full conditional for $a_{s,ii}$

Let $a_{s,ii}$ denote the ith diagonal element of A_s and $A_{-(s,ii)}$ be the set of all the elements of $A_1, \ldots, A_p$ except $a_{s,ii}$. For each $a_{s,ii}$, $i = 1, \ldots, m$, $s = 1, \ldots, p$, the full conditional distribution is

$$\begin{aligned}
\pi(a_{s,ii}|\boldsymbol{y}_{1:n}, A_{-(s,ii)}, \Sigma, \boldsymbol{\vartheta}) &\propto p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ii}|\boldsymbol{\vartheta}) \\
&\propto \det(G)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_{1:p}^{\mathrm{T}} G^{-1} \boldsymbol{y}_{1:p})\right\} \\
&\times \det(\Sigma)^{-(n-p)/2} \\
&\times \exp\left[-\frac{1}{2}\sum_{t=p+1}^{n}\left\{\left(\boldsymbol{y}_t - \sum_{i=1}^{p}\phi_i \boldsymbol{y}_{t-i}\right)^{\mathrm{T}} \Sigma^{-1}\left(\boldsymbol{y}_t - \sum_{i=1}^{p}\phi_i \boldsymbol{y}_{t-i}\right)\right\}\right] \\
&\times \sqrt{\frac{\omega_{s1}}{2\pi}} \exp\left\{-\frac{\omega_{s1}}{2}\left(a_{s,ii} - \mu_{s1}\right)^2\right\}.
\end{aligned}$$

Once again, this full conditional distribution is not a standard distribution and so a Metropolis-Hastings step is used to sample the diagonal elements of the transformed partial autocorrelations with acceptance probability

$$\alpha(a_{s,ii}^*, a_{s,ii}) = \min\left\{1, \frac{p(\boldsymbol{y}_{1:n}|a_{s,ii}^*, A_{-(s,ii)}, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ii}^*|\boldsymbol{\vartheta})q_{a_{s,ii}}(a_{s,ii}|a_{s,ii}^*)}{p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ii}|\boldsymbol{\vartheta})q_{a_{s,ii}}(a_{s,ii}^*|a_{s,ii})}\right\},$$

where $q_{a_{s,ii}}(\cdot|\cdot)$ is a Gaussian random walk proposal distribution such that $\alpha(a_{s,ii}^*, a_{s,ii})$ simplifies to

$$\alpha(a_{s,ii}^*, a_{s,ii}) = \min\left\{1, \frac{p(\boldsymbol{y}_{1:n}|a_{s,ii}^*, A_{-(s,ii)}, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ii}^*|\boldsymbol{\vartheta})}{p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ii}|\boldsymbol{\vartheta})}\right\}.$$

Full conditional for $a_{s,ij}$

Let $a_{s,ij}$ denote the (i,j)th element of A_s and $A_{-(s,ij)}$ be the set of all the elements of $A_1, \ldots, A_p$ except $a_{s,ij}$. For each $a_{s,ij}$, $i = 1, \ldots, m$, $j = 1, \ldots, m$, $i \neq j$, $s = 1, \ldots, p$, the

full conditional distribution is

$$\begin{aligned}
\pi(a_{s,ij}|\boldsymbol{y}_{1:n}, A_{-(s,ij)}, \Sigma, \boldsymbol{\vartheta}) &\propto p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ij}|\boldsymbol{\vartheta}) \\
&\propto \det(G)^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(\boldsymbol{y}_{1:p}^{\mathrm{T}} G^{-1} \boldsymbol{y}_{1:p})\right\} \\
&\times \det(\Sigma)^{-(n-p)/2} \\
&\times \exp\left[-\frac{1}{2}\sum_{t=p+1}^{n}\left\{\left(\boldsymbol{y}_t - \sum_{i=1}^{p}\phi_i \boldsymbol{y}_{t-i}\right)^{\mathrm{T}} \Sigma^{-1}\left(\boldsymbol{y}_t - \sum_{i=1}^{p}\phi_i \boldsymbol{y}_{t-i}\right)\right\}\right] \\
&\times \sqrt{\frac{\omega_{s2}}{2\pi}} \exp\left\{-\frac{\omega_{s2}}{2}\left(a_{s,ij} - \mu_{s2}\right)^2\right\}.
\end{aligned}$$

A Metropolis-Hastings step is used to sample the off-diagonal elements of the transformed partial autocorrelations from this non-standard distribution, with acceptance probability

$$\alpha(a^*_{s,ij}, a_{s,ij}) = \min\left\{1, \frac{p(\boldsymbol{y}_{1:n}|a^*_{s,ij}, A_{-(s,ij)}, \Sigma, \boldsymbol{\vartheta})\pi(a^*_{s,ij}|\boldsymbol{\vartheta})}{p(\boldsymbol{y}_{1:n}|A_1, \ldots, A_p, \Sigma, \boldsymbol{\vartheta})\pi(a_{s,ij}|\boldsymbol{\vartheta})}\right\}$$

based on a Gaussian random walk proposal distribution.

Full conditional for μ_{s1}

Let $\boldsymbol{a}_{s1}$ be the set of all diagonal elements of A_s. For each μ_{s1}, $s = 1, \ldots, p$, the full conditional distribution is

$$\begin{aligned}
\pi(\mu_{s1}|\boldsymbol{a}_{s1}, \omega_{s1}) &\propto \pi(\mu_{s1}) \prod_{i=1}^{m} \pi(a_{s,ii}|\mu_{s1}, \omega_{s1}) \\
&\propto \exp\left\{-\frac{1}{2}\left(\frac{\mu_{s1} - e_{s1}}{f_{s1}}\right)^2\right\} \exp\left\{-\frac{\omega_{s1}}{2}\sum_{i=1}^{m}(a_{ii} - \mu_{s1})^2\right\} \\
&\propto \exp\left[-\frac{1}{2}\left\{\frac{\mu_{s1}^2}{f_{s1}^2} - \frac{2\mu_{s1}e_{s1}}{f_{s1}^2} + \omega_{s1}\sum_{i=1}^{m}(\mu_{s1}^2 - 2a_{s,ii}\mu_{s1})\right\}\right] \\
&\propto \exp\left\{-\frac{1}{2}\left(\frac{\mu_{s1}^2}{f_{s1}^2} - \frac{2\mu_{s1}e_{s1}}{f_{s1}^2} + m\omega_{s1}\mu_{s1}^2 - 2\mu_{s1}\omega_{s1}\sum_{i=1}^{m} a_{s,ii}\right)\right\} \\
&\propto \exp\left[-\frac{1}{2}\left\{\left(m\omega_{s1} + \frac{1}{f_{s1}^2}\right)\mu_{s1}^2 - 2\left(\frac{e_{s1}}{f_{s1}^2} + \omega_{s1}\sum_{i=1}^{m} a_{s,ii}\right)\mu_{s1}\right\}\right] \\
&\propto \exp\left[-\frac{1}{2}\left(m\omega_{s1} + \frac{1}{f_{s1}^2}\right)\left\{\mu_{s1} - \frac{\left(\frac{e_{s1}}{f_{s1}^2} + \omega_{s1}\sum_{i=1}^{m} a_{s,ii}\right)}{\left(m\omega_{s1} + \frac{1}{f_{s1}^2}\right)}\right\}^2\right].
\end{aligned}$$

That is,

$$\mu_{s1}|\boldsymbol{a}_{s1}, \omega_{s1} \sim \mathrm{N}\left(\frac{\frac{e_{s1}}{f_{s1}^2} + \omega_{s1}\sum_{i=1}^{m} a_{s,ii}}{m\omega_{s1} + \frac{1}{f_{s1}^2}}, \frac{1}{m\omega_{s1} + \frac{1}{f_{s1}^2}}\right),$$

permitting the use of a Gibbs step to sample from the posterior for μ_{s1}, $s = 1, \ldots, p$.

Full conditional for μ_{s2}

The calculation for the full conditional for μ_{s2} proceeds in the same way as the calculation for the full conditional of μ_{s1}, except that it is conditional on the off-diagonal elements of A_s. Let $\boldsymbol{a}_{s2}$ denote the set of all off-diagonal elements of A_s. As there are $m^2 - m$ off-diagonal elements of A_s compared to m diagonal elements, we also replace m with $m(m-1)$ in the case of μ_{s2}. For each μ_{s2}, $s = 1, \ldots, p$, we have

$$\mu_{s2}|\boldsymbol{a}_{s2}, \omega_{s2} \sim \mathrm{N}\left\{\frac{\frac{e_{s2}}{f_{s2}^2} + \omega_{s2}\sum_{i,j=1, i\neq j}^{m} a_{s,ij}}{m(m-1)\omega_{s2} + \frac{1}{f_{s2}^2}}, \frac{1}{m(m-1)\omega_{s2} + \frac{1}{f_{s2}^2}}\right\}$$

which we sample using a Gibbs step.

Full conditional for ω_{s1}

For each ω_{s1}, $s = 1, \ldots, p$, the full conditional distribution is

$$\begin{aligned}
\pi(\omega_{s1}|\boldsymbol{a}_{s1}, \mu_{s1}) &\propto \pi(\omega_{s1}) \prod_{i=1}^{m} \pi(a_{s,ii}|\omega_{s1}, \mu_{s1}) \\
&\propto \omega_{s1}^{g_{s1}-1} \exp\left(-\omega_{s1} h_{s1}\right) \omega_{s1}^{\frac{m}{2}} \exp\left\{-\frac{\omega_{s1}}{2}\sum_{i=1}^{m}(a_{s,ii} - \mu_{s1})^2\right\} \\
&\propto \omega_{s1}^{g_{s1}+\frac{m}{2}-1} \exp\left[-\omega_{s1}\left\{h_{s1} + \frac{1}{2}\sum_{i=1}^{m}(a_{s,ii} - \mu_{s1})^2\right\}\right].
\end{aligned}$$

That is,

$$\omega_{s1}|\boldsymbol{a}_{s1}, \mu_{s1} \sim \mathrm{Gam}\left\{g_{s1} + \frac{m}{2}, h_{s1} + \frac{1}{2}\sum_{i=1}^{m}(a_{s,ii} - \mu_{s1})^2\right\}$$

which can be sampled using a Gibbs step.

Full conditional for ω_{s2}

The calculation for the full conditional distribution of ω_{s2} proceeds in the same way as for the full conditional distribution of ω_{s1} except that it is conditional on the off-diagonal elements of A_s, rather than the diagonal elements. Additionally, there are $m(m-1)$ off-diagonal elements in the case of ω_{s2} compared to m diagonal elements in the case of ω_{s1}.

The full conditional for ω_{s2} is

$$\omega_{s2}|\boldsymbol{a}_{s2}, \mu_{s2} \sim \text{Gam}\left\{ g_{s2} + \frac{m(m-1)}{2}, h_{s2} + \frac{1}{2} \sum_{i,j=1, i\neq j}^{m} (a_{s,ij} - \mu_{s2})^2 \right\}$$

which we sample using a Gibbs step.

Algorithm 8 Metropolis-within-Gibbs algorithm for inference of a $\text{VAR}_m(p)$ process with a known p

1. Initialise the state of the chain to $(\Sigma^{(0)}, A_1^{(0)}, \ldots, A_p^{(0)}, \mu_{s1}^{(0)}, \mu_{s2}^{(0)}, \omega_{s1}^{(0)}, \omega_{s2}^{(0)})$ and set the iteration counter to $k = 1$.
2. Sample $\Sigma^{(k)}$ from $\pi(\Sigma|\Sigma^{(k-1)}, A_1^{(k-1)}, \ldots, A_p^{(k-1)}, \mu_{s1}^{(k-1)}, \mu_{s2}^{(k-1)}, \omega_{s1}^{(k-1)}, \omega_{s2}^{(k-1)})$ using a Metropolis-Hastings step with an inverse Wishart random walk proposal distribution, $q_\Sigma(\Sigma^*|\Sigma^{(k-1)})$.
3. For $s = 1, \ldots, p$, $i = 1, \ldots, m$, sample $a_{s,ii}^{(k)}$ from $\pi(a_{s,ii}|\Sigma^{(k)}, \boldsymbol{A}_{-(s,ii)}^{(k-1)}, \mu_{s1}^{(k-1)}, \mu_{s2}^{(k-1)}, \omega_{s1}^{(k-1)}, \omega_{s2}^{(k-1)})$ using a Metropolis-Hastings step with a Gaussian random walk proposal, $q_{a_{s,ii}}(a_{s,ii}^*|a_{s,ii}^{(k-1)})$, updating $\boldsymbol{A}_{-(s,ii)}^{(k-1)}$ after each sub-step.
4. For $s = 1, \ldots, p$, $i = 1, \ldots, m$, $j = 1, \ldots, m$, $i \neq j$, sample $a_{s,ij}^{(k)}$ from $\pi(a_{s,ij}|\Sigma^{(k)}, \boldsymbol{A}_{-(s,ij)}^{(k-1)}, \mu_{s1}^{(k-1)}, \mu_{s2}^{(k-1)}, \omega_{s1}^{(k-1)}, \omega_{s2}^{(k-1)})$ using a Metropolis-Hastings step with a Gaussian random walk proposal, $q_{a_{s,ij}}(a_{s,ij}^*|a_{s,ij}^{(k-1)})$, updating $\boldsymbol{A}_{-(s,ij)}^{(k-1)}$ after each sub-step.
5. Sample $\mu_{s1}^{(k)} \sim \text{N}\left(\frac{\frac{e_{s1}}{f_{s1}^2} + \omega_{s1}^{(k-1)} \sum_{i=1}^m a_{s,ii}^{(k)}}{m\omega_{s1}^{(k-1)} + \frac{1}{f_{s1}^2}}, \frac{1}{m\omega_{s1}^{(k-1)} + \frac{1}{f_{s1}^2}} \right)$.
6. Sample $\mu_{s2}^{(k)} \sim \text{N}\left\{ \frac{\frac{e_{s2}}{f_{s2}^2} + \omega_{s2}^{(k-1)} \sum_{i,j=1, i\neq j}^m a_{s,ij}^{(k)}}{m(m-1)\omega_{s2}^{(k-1)} + \frac{1}{f_{s2}^2}}, \frac{1}{m(m-1)\omega_{s2}^{(k-1)} + \frac{1}{f_{s2}^2}} \right\}$.
7. Sample $\omega_{s1}^{(k)} \sim \text{Gam}\left\{ g_{s1} + \frac{m}{2}, h_{s1} + \frac{1}{2} \sum_{i=1}^m (a_{s,ii}^{(k)} - \mu_{s1}^{(k)})^2 \right\}$.
8. Sample $\omega_{s2}^{(k)} \sim \text{Gam}\left\{ g_{s2} + \frac{m(m-1)}{2}, h_{s2} + \frac{1}{2} \sum_{i,j=1, i\neq j}^m (a_{s,ij}^{(k)} - \mu_{s2}^{(k)})^2 \right\}$.
9. Set k equal to $k + 1$ and return to step 2.

Simulation experiment

We are interested in the efficiency of our Metropolis-within-Gibbs sampler, particularly with respect to the parameters in $A_1, \ldots, A_p$. To investigate this behaviour the Metropolis-

Hastings scheme described in Algorithm 8 is applied to simulated data where the true values of the parameters are known, using a bespoke sampler coded in R. For a given m and p, we simulate a set of matrices $A_1, \ldots, A_p$ with elements sampled independently from a standard normal distrubution. Setting the error variance as $\Sigma = I_m$ we use the set of matrices $A_1, \ldots, A_p$ to simulate a $\text{VAR}_m(p)$ process of length $n = 1000$. We initially consider simulated data in the simplest vector autoregressive case where $m = 2$ and $p = 1$. As discussed in Section 5.2.1, we follow the guidelines in Heaps (2023) to choose appropriate values of the hyperparameters in the exchangeable prior for $A_1, \ldots, A_p$. For $m = 2$ the choices of hyperparameters

$$e_{si} = 0,\ f_{si} = \sqrt{0.35},\ g_{si} = 1.1,\ h_{si} = 0.015, \quad (i = 1, 2)$$

result in correlations between the elements of the matrices of 0.7 and a marginal variance of 1/2, which avoids multimodality in the prior. In the inverse Wishart prior for Σ we take the scale matrix to be the identity matrix, I_m, and the degrees of freedom to be $m + 4$ which ensures that the variance is finite. We consider three different methods for proposing new elements of the A_s matrices, $s = 1, \ldots, p$:

1. Sample each element of A_s individually using a normal random walk proposal, as described above.

2. Sample all elements in each column of A_s jointly using a multivariate normal random walk proposal with a diagonal tuning matrix.

3. Sample all elements of A_s jointly using a multivariate random walk proposal with a diagonal tuning matrix.

Options 2 and 3 involve combining steps 3 and 4 of Algorithm 8. Figures 5.2 and 5.3 contain trace plots and posterior density plots respectively for each element of A_1 in the case where $m = 2$ and $p = 1$. The analysis has been repeated for each sampling method, tuned such that acceptance rates were 20-60%, with results based on sampling the matrix as a whole depicted in red, results based on sampling by columns in gold and results based on sampling each element individually depicted in blue. We additionally coded the model up in Stan and the output from fitting the model via HMC using Stan is depicted in purple. In each case the Metropolis-Hastings algorithm was run for 21000 iterations of which 1000 were discarded as burn-in. All samplers appear to have converged and there are no obvious mixing issues. Each of the samplers was initialised at a different value and all three sampling methods give trace plots and posterior densities which match up well and additionally match the output from Stan, suggesting the samplers are not sensitive to the choice of initial value. Whilst all three methods give similar results, the method in which whole matrices are sampled jointly runs much faster (26,738.85 seconds) than when

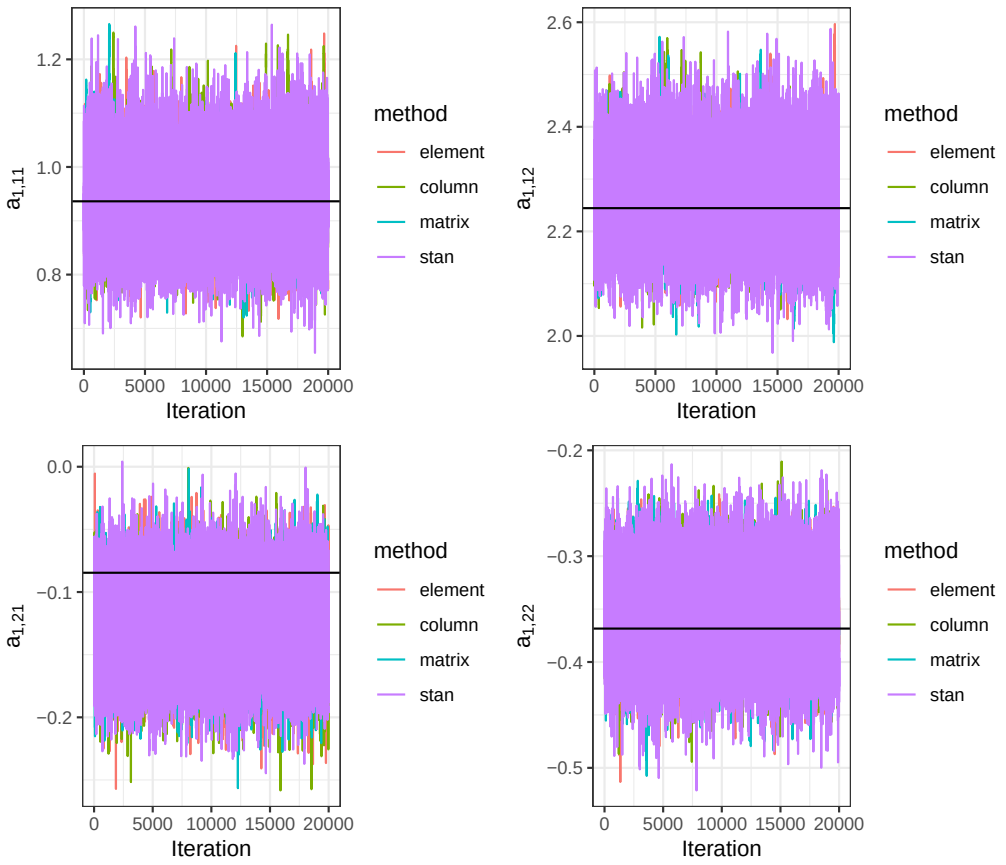


Figure 5.2: Trace plots of the posterior samples of each element of A_1, obtained from inference of a $\text{VAR}_2(1)$ process. The different colours depict when the matrix is sampled as a whole (blue), by columns (green) by element (red) and via Stan (purple), with the true values represented by a black horizontal line.

sampling by columns (39,945.32 seconds) or elements (65,594.18 seconds), suggesting that as all methods show similar convergence and mixing the fastest method should be used. The true values of the elements of A_1, represented by a horizontal line on the trace plots and a vertical line on the density plots, are reasonably likely under the posterior samples which gives no reason to doubt that the samplers are coded up correctly.

Unfortunately, whilst our bespoke Metropolis-Hastings sampler works well for smaller values of m, as we increase m the sampler quickly develops convergence problems. The MCMC scheme was also applied to data which was simulated using $m = 3$ and $p = 3$, with the data simulated using the same method described above. We retained the same prior specification for Σ but in this case values for the hyperparameters in the prior for A_1, A_2, and A_3 which satisfy the requirements for avoiding multimodality when $m = 3$ are

$$e_{si} = 0,\ f_{si} = \sqrt{0.455},\ g_{si} = 1.365,\ h_{si} = 0.071, \qquad (i = 1, 2).$$

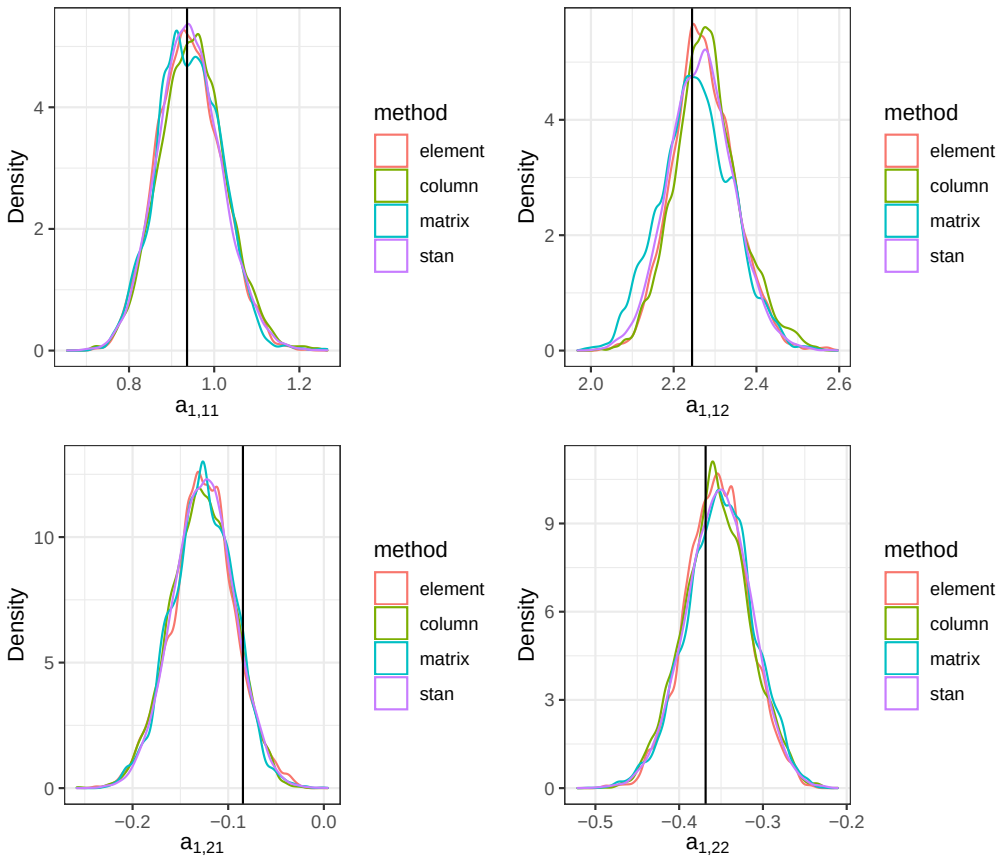


Figure 5.3: Posterior densities for each element of A_1, obtained from inference of a simulated $\text{VAR}_2(1)$ process. The different colours depict when the matrix is sampled as a whole (blue), by columns (green), by element (red) and via Stan (purple), with the true values represented by a black vertical line.

Figure 5.4 contains a trace plot of the MCMC sample of $a_{1,11}$. The chain does not appear to have converged after 10^5 samples. The true value is overlaid as a red horizontal line. In this case whole A_s matrices were sampled in one go but the convergence issues appear for all three sampling methods. The other elements of A_1, A_2 and A_3 are not included but show similar results. This suggests that whilst this bespoke Metropolis-Hastings sampler could be used for small toy examples, it is not feasible for use in general.

Motivated by the poor performance of our Metropolis-within-Gibbs sampler, we additionally considered Hamiltonian Monte Carlo (HMC) (Neal, 2011) implemented using Stan. As discussed in Section 2.1.2, rather than appealing to conditional independence structure in the posterior for one-at-a-time parameter updates, HMC uses information on the gradient of the logarithm of the posterior density to generate global proposals that update all parameters simultaneously. The Stan programme for this model is included in Appendix C.3. A further advantage of fitting the model using HMC in Stan is that it

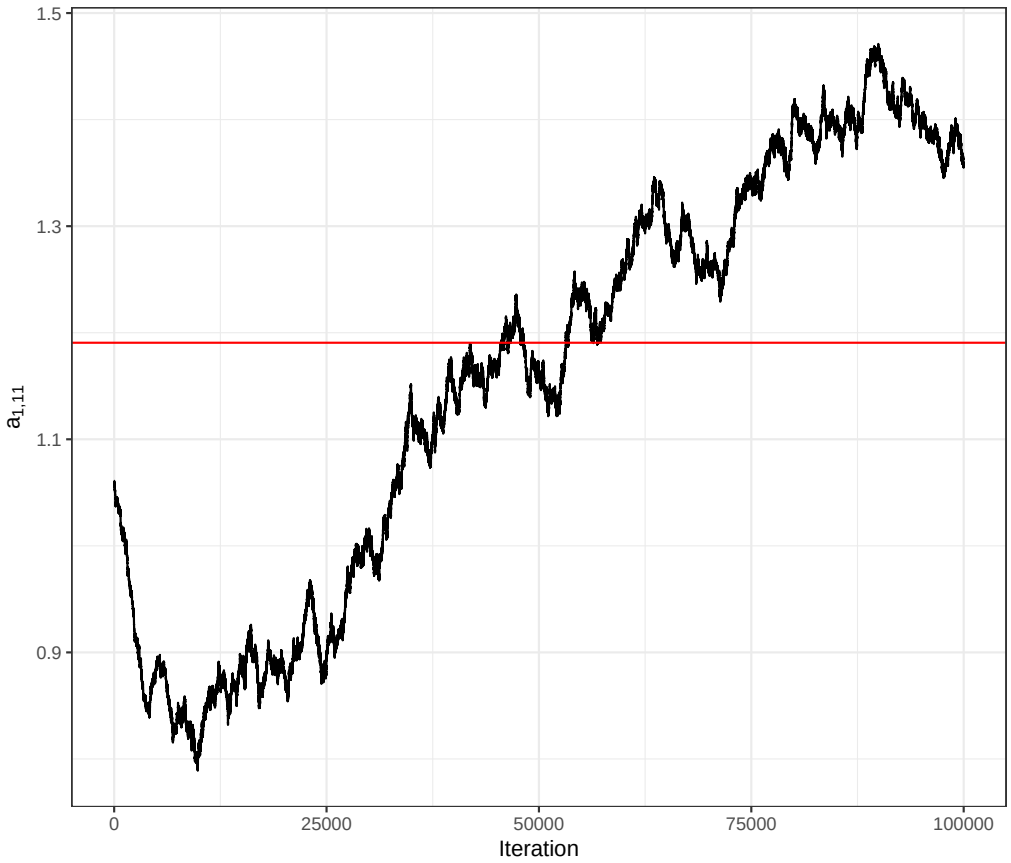


Figure 5.4: Trace plot of the posterior samples of $a_{1,11}$, the first element of A_1, obtained from inference of a simulated $\text{VAR}_3(3)$ process, with the true value represented as a red horizontal line.

results in a much faster running time. For example, in the case where $m = 2$ and $p = 1$ the run time was 1,103.6 seconds compared to 26,738.85 seconds for the fastest Metropolis-within-Gibbs sampler. However, this comparison is not totally fair since the Stan software translates the algorithm into fast, compiled C++ code; if our bespoke Metropolis-within-Gibbs sampler had been coded in C++, rather than R, we may have seen more comparable run times.

Repeating the analysis of the simulated data with $m = 3$ and $p = 3$, Figures 5.5 and 5.6 contain trace plots and density plots respectively for the first element, $a_{1,11}$, $a_{2,11}$ and $a_{3,11}$, of each of the matrices A_1, A_2, and A_3. These are representative of all parameters. The model was run for 1000 warm-up iterations followed by 4000 sampling iterations, using four chains which are represented by four different colours in the plots. The true values for each of the elements, represented as a black horizontal line in the trace plots and as a black vertical line in the density plots, seem reasonable under the posterior and there are no reasons to doubt the convergence of the sampler.

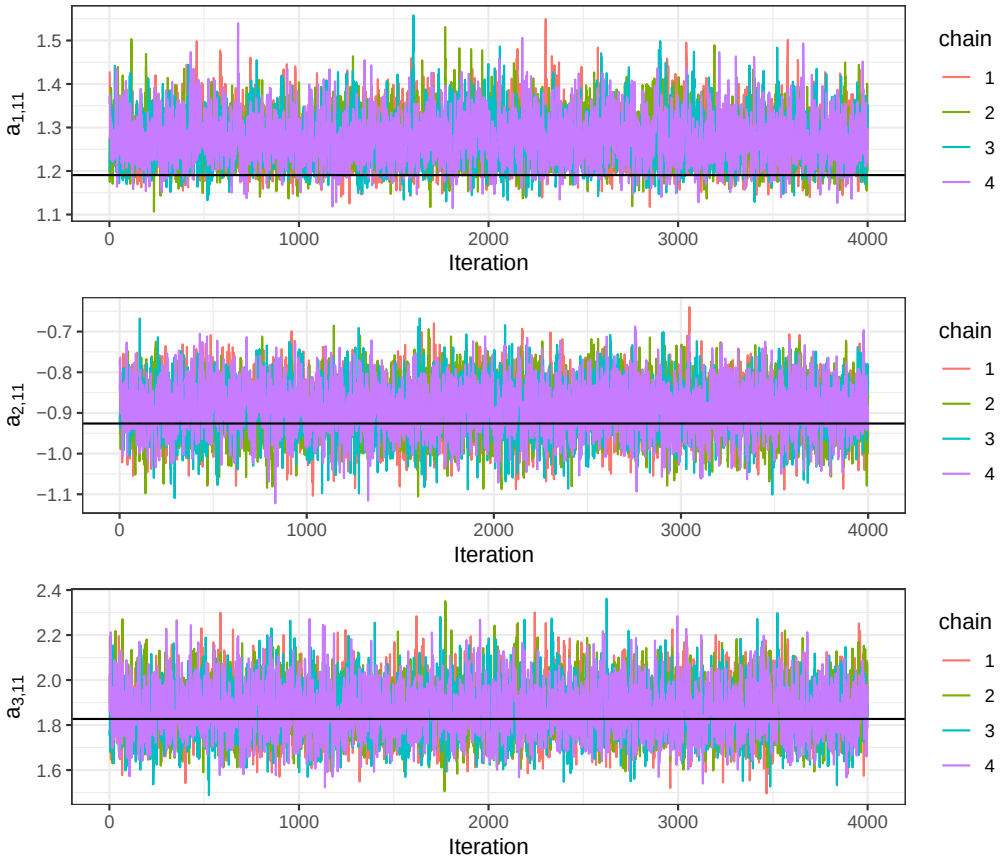


Figure 5.5: Trace plots for the first element, $a_{1,11}, a_{2,11}$ and $a_{3,11}$ of the matrices A_1, A_2 and A_3, obtained from inference of a $\text{VAR}_3(3)$ process using Stan. The different colours (red, green, blue and purple) represent the four different chains and the black horizontal line represents the true value.

5.3 Enforcing stationarity when p is unknown

In Section 5.1, we discussed a reparameterisation of a stationary vector autoregressive process in terms of a set of transformed partial autocorrelation matrices which are unconstrained. In Section 5.2, by choosing a prior for these unconstrained matrices, we induced a prior for the original model parameters that is constrained to the stationary region. Hence stationarity is enforced. However, a clear limitation with this approach is that inference is conditional on a given value for the model order, p, with no account for the associated uncertainty. Whilst in some cases the model order may be known, in the vast majority of real-life situations it is not known *a priori*. In this section, we discuss possible prior distributions for the transformed partial autocorrelation matrices which allow us to enforce stationarity whilst also learning about the order of the process. Throughout, as in Chapter 4, we denote by $p_{\max}$ the maximum order of the process that we are prepared

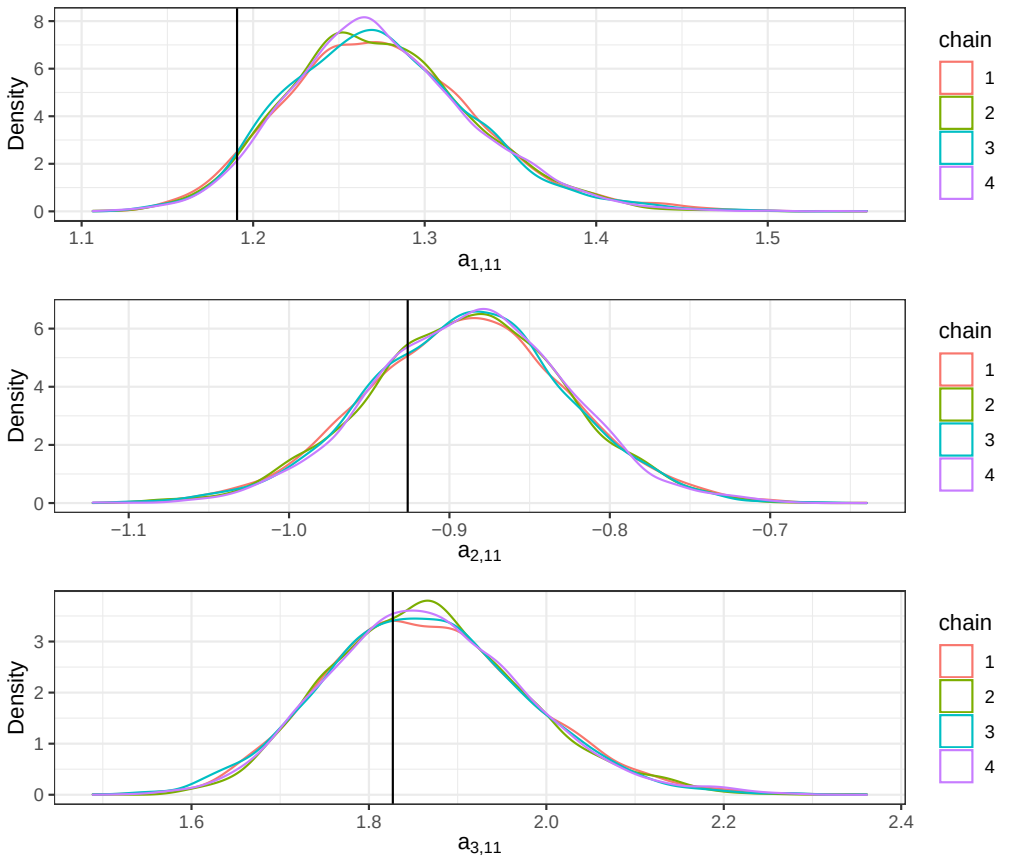


Figure 5.6: Posterior density plots for the first element, $a_{1,11}, a_{2,11}$ and $a_{3,11}$ of the matrices A_1, A_2 and A_3, obtained from inference of a $\text{VAR}_3(3)$ process using Stan. The different colours (red, green, blue and purple) represent the four different chains and the black vertical line represents the true value.

to consider.

5.3.1 Shrinkage prior for transformed partial autocorrelations

When modifying our prior and inferential procedures to learn the model order, we can exploit two important implications of the relationship between the singular value decompositions of P and A, described in Section 5.1. First, the spectral norms of P and A, $r_1 = \|P\|_2$ and $\tilde{r}_1 = \|A\|_2$, are clearly related through the monotonic mapping:

$$\tilde{r}_1 = r_1/(1 - r_1^2)^{1/2}.$$

The relative sizes of the unconstrained parameters A_s across lags $s = 1, \ldots, p_{\max}$ therefore relate directly to the relative sizes of the partial autocorrelation matrices P_s across lags. Second, $P = 0_m$ if and only if $A = 0_m$ in which 0_m denotes the $m \times m$ matrix of zeros.

It follows from the definition of the partial autocorrelation matrices that for $k < p_{\max}$, $P_k \neq 0_m$ and $P_{k+s} = 0_m$ for $s = 1, \ldots, p_{\max} - k$ if and only if $\phi_k \neq 0_m$ and $\phi_{k+s} = 0_m$ for $s = 1, \ldots, p_{\max} - k$. The order of a vector autoregression is therefore $k < p_{\max}$ if and only if $A_k \neq 0_m$ and $A_{k+s} = 0_m$ for $s = 1, \ldots, p_{\max} - k$. Under the unconstrained parameterisation, it follows that the model of order $k < p_{\max}$ is nested within the model of order $k + 1$.

A Bayesian approach to quantifying uncertainty on the dimension of nested models is to fit an overparameterised model with purposefully more components than are required. By using a shrinkage prior, components that are shrunk to zero, or shrunk enough to be deemed negligible in the likelihood, can then be discarded. Consequently, inference on both the continuous model parameters and the model dimension are available from a single within-model MCMC sampler, without recourse to transdimensional MCMC methods like reversible jump for which the design of efficient across-model proposal distributions is notoriously difficult. For other examples of this approach applying shrinkage priors in an overfitted model, see Rousseau & Mengersen (2011) for an example involving mixture models and Bhattacharya & Dunson (2011) for an example involving factor models. We can therefore borrow ideas from this literature by adopting a shrinkage prior for $A_1, \ldots, A_{p_{\max}}$ with a large value for $p_{\max}$. By learning about the lag beyond which the A_s can be taken as zero matrices, we learn about the order p of the process. It can therefore be regarded as an extension of the prior in the univariate case, discussed in Section 4.2.1, that uses spike-and-slab distributions for the partial autocorrelation parameters (Barnett *et al.*, 1996). Moreover, we can convey the very reasonable idea that the partial autocorrelations at higher lags are likely to be smaller than those at lower lags by choosing a shrinkage prior for the A_s, $s = 1, \ldots, p_{\max}$, whose degree of shrinkage increases with the lag s.

We discuss three such priors with the desired increasing shrinkage property, a spike-and-slab prior, the cumulative shrinkage process (CUSP) and the multiplicative gamma process (MGP). In all cases, our joint prior for the model parameters can be written as

$$\pi(\Sigma, A_1, \ldots, A_{p_{\max}}, \boldsymbol{\vartheta}) = \pi(\Sigma)\pi(\boldsymbol{\vartheta}) \prod_{s=1}^{p_{\max}} \pi(A_s|\boldsymbol{\vartheta}). \tag{5.9}$$

As previously, $\boldsymbol{\vartheta}$ denotes the collection of unknown hyperparameters (or latent variables) in the prior for the transformed partial autocorrelation matrices. The error variance matrix is, again, given an inverse Wishart prior distribution with a scale matrix that has a common element on the diagonal and off the diagonal. In the remainder of this chapter, we focus on the prior for $(A_1, \ldots, A_{p_{\max}}, \boldsymbol{\vartheta})$.

5.3.2 Spike-and-slab prior

A natural extension to our work in the univariate case in Section 4.2.1 is to incorporate $p_{\max}$ indicator variables in the model, $I_1, \ldots, I_{p_{\max}}$. In the approach adapted from Kuo & Mallick (1998) we would then define

$$A_s = I_s A'_s, \qquad s = 1, \ldots, p_{\max}$$

with associated prior

$$\pi(A'_1, \ldots, A'_{p_{\max}}, I_1, \ldots, I_{p_{\max}}, \boldsymbol{\vartheta}) = \pi(A'_1, \ldots, A'_{p_{\max}}, \boldsymbol{\vartheta})\pi(I_1, \ldots, I_{p_{\max}})$$

where $\pi(I_1, \ldots, I_{p_{\max}}) = \prod_{s=1}^{p_{\max}} \pi(I_s)$. Here we would take $I_s \sim \text{Bern}(p_s)$ with $p_1 \geq p_2 \geq \ldots \geq p_{p_{\max}}$ and assign the same prior to $(A'_1, \ldots, A'_{p_{\max}}, \boldsymbol{\vartheta})$ as that described in Section 5.2.1. In principle, we could extend our Metropolis-within-Gibbs sampler from Algorithm 8 to incorporate the indicator variables. However, as the sampler already performed poorly in the fixed-p case, this approach will clearly be untenable. Under the approach described in Barnett *et al.* (1996), we would instead construct the prior

$$\pi(A_1, \ldots, A_{p_{\max}}, I_1, \ldots, I_{p_{\max}}, \boldsymbol{\vartheta}) = \pi(\boldsymbol{\vartheta}) \prod_{s=1}^{p_{\max}} \pi(A_s \mid I_s, \boldsymbol{\vartheta})\pi(I_s)$$

where $I_s \sim \text{Bern}(p_s)$ with $p_1 \geq p_2 \geq \ldots p_{p_{\max}}$, $\Pr(A_s = 0_m \mid I_s = 0) = 1$ and the prior for each $(A_s \mid I_s = 1, \boldsymbol{\vartheta})$ is as described in Section 5.2.1. Practical implementation of the Barnett-inspired Metropolis-within-Gibbs sampler would then require sampling each pair (A_s, I_s) jointly by initially sampling the indicator variable from a Bernoulli distribution, with A_s integrated out, and then sampling A_s conditional on the indicator variable. However, even in the univariate case, numerical marginalisation over the (single) partial autocorrelation parameter slowed down sampling enough to make the algorithm inefficient. Marginalising numerically over all m^2 parameters in each A_s matrix will be computationally infeasible.

Although we found HMC, implemented via Stan, to work well in the fixed p case, it cannot be used to generate samples from the posterior of the hierarchical model under either representation of the spike-and-slab prior. This is because it is not possible to analytically integrate out the indicator variables and so they would need to be incorporated as unknowns in the model. Unfortunately, HMC requires the posterior to be continuously differentiable everywhere and so Stan cannot handle discrete-valued parameters. We therefore focus for the remainder of this chapter on priors that can be represented without the incorporation of discrete-valued unknowns.

5.3.3 Cumulative shrinkage process

Background

An increasing shrinkage prior which can be expressed without discrete-valued unknowns is the cumulative shrinkage process (CUSP) (Legramanti *et al.*, 2020). Consider a general set of parameters $\theta_1, \theta_2, \ldots$. The cumulative shrinkage process is based on a sequence of spike-and-slab distributions for the θ_s which assign increasing prior mass to the spike as s increases. In the general case, Legramanti *et al.* (2020) give each θ_s, $s = 1, 2, \ldots$, a conditionally independent spike-and-slab prior as follows:

$$\theta_s|\pi_s \sim C_s = (1-\pi_s)C_0 + \pi_s\delta_{\theta_\infty}, \qquad \pi_s = \sum_{l=1}^{s}\omega_l, \qquad \omega_l = \nu_l\prod_{m=1}^{l-1}(1-\nu_m) \tag{5.10}$$

where the ν_m are independent Beta(1, α) random variables, C_0 is a diffuse continuous slab distribution and the notation δ_{θ_∞} denotes the Dirac delta distribution at θ_∞. The prior can be restricted to a finite, maximum number of terms by setting $\nu_H = 1$ for some H. This forces π_H to be equal to 1 and forces θ_H to be assigned to the spike. As such, the maximum number of terms coming from the slab is $H-1$. Legramanti *et al.* (2020) denote the number of *active* terms, those terms coming from the slab rather than the spike, as H^*. The prior expectation of ω_l, $l = 1, \ldots, H-1$, is such that

$$\begin{aligned}
\mathrm{E}(\omega_l) &= \mathrm{E}\left\{\nu_l\prod_{m=1}^{l-1}(1-\nu_m)\right\} \\
&= \frac{1}{1+\alpha}\prod_{m=1}^{l-1}\left(1-\frac{1}{1+\alpha}\right) \\
&= \frac{1}{1+\alpha}\left(\frac{\alpha}{1+\alpha}\right)^{l-1} \\
&= \frac{\alpha^{l-1}}{(1+\alpha)^l}.
\end{aligned} \tag{5.11}$$

Then the prior expectation of π_s, $s = 1, \ldots, H-1$, is such that

$$\begin{aligned}
\mathrm{E}(\pi_s) &= \mathrm{E}\left(\sum_{l=1}^{s} \omega_l\right) \\
&= \sum_{l=1}^{s} \frac{\alpha^{l-1}}{(1+\alpha)^l} \\
&= \frac{1}{1+\alpha} \sum_{l=1}^{s} \frac{\alpha^{l-1}}{(1+\alpha)^{l-1}} \\
&= \left(\frac{1}{1+\alpha}\right)\left\{1 - \left(\frac{\alpha}{1+\alpha}\right)^s\right\}\left(1 - \frac{\alpha}{1+\alpha}\right)^{-1} \\
&= \left(\frac{1}{1+\alpha}\right)\left\{1 - \left(\frac{\alpha}{1+\alpha}\right)^s\right\}\left(\frac{1}{1+\alpha}\right)^{-1} \\
&= 1 - \left(\frac{\alpha}{1+\alpha}\right)^s. \qquad (5.12)
\end{aligned}$$

The rate of shrinkage is controlled by α, which can be interpreted as the prior mean of the number of terms modelled by the slab. Legramanti *et al.* (2020) show this by writing the prior for $\theta_s|\pi_s$ in terms of augmented indicator variables:

$$c_s|\pi_s \sim \mathrm{Bern}(1 - \pi_s).$$

Then the prior for $\theta_s|c_s$ can be written as

$$\theta_s|c_s \sim c_s C_0 + (1 - c_s)\delta_{\theta_\infty}.$$

If H^* is the number of terms coming from the slab then, in the untruncated case (i.e as $H \to \infty$)

$$
\begin{aligned}
\mathrm{E}(H^*) &= \sum_{s=1}^{\infty} \mathrm{E}(c_s) \\
&= \sum_{s=1}^{\infty} \mathrm{E}\{\mathrm{E}(c_s|\pi_s)\} \\
&= \sum_{s=1}^{\infty} \mathrm{E}(1-\pi_s) \\
&= \sum_{s=1}^{\infty} 1 - \mathrm{E}(\pi_s) \\
&= \sum_{s=1}^{\infty} 1 - \left\{1 - \left(\frac{\alpha}{1+\alpha}\right)^s\right\} \\
&= \sum_{s=1}^{\infty} \left(\frac{\alpha}{1+\alpha}\right)^s \\
&= \sum_{s=0}^{\infty} \left(\frac{\alpha}{1+\alpha}\right)^s - \left(\frac{\alpha}{1+\alpha}\right)^0 \\
&= 1 + \alpha - 1 \\
&= \alpha.
\end{aligned}
$$

The CUSP prior makes use of the stick-breaking construction of the Dirichlet process (Sethuraman, 1994; Ishwaran & James, 2001). Under this construction, the probability π_s of being assigned to the spike increases with s. As the dimension grows, C_s becomes increasingly concentrated around the spike. In many applications where the CUSP prior is used, θ_s represents the unknown scale in a hierarchical prior for the unknowns parameterising the s-th dimension in a model in which the corresponding mean is zero. In order to facilitate the removal of higher order terms in such cases, the spike θ_∞ can be set close to zero. Then, as increasing mass is assigned to the spike as the complexity grows, the terms of higher complexity are more likely to be assigned to the spike and deemed redundant. Whilst setting $\theta_\infty = 0$ is possible, Legramanti *et al.* (2020) recommend setting $\theta_\infty > 0$, arguing that this results in improved mixing.

Application to stationary VAR models

In the context of determining the order of vector autoregressive processes, we assign the A_s matrices a normal prior centred at zero where the degree of shrinkage is controlled by a variance parameter, θ_s. Given a maximum order for the time series, $p_{\max} = H - 1$, and

denoting the (i, j)th element in A_s by $a_{s,ij}$ the prior given to the A_s matrices is

$$a_{s,ij}|\theta_s \sim \mathrm{N}(0, \theta_s), \qquad s = 1, 2, \ldots, H \tag{5.13}$$

where the θ_s follow the cumulative shrinkage process described above with a finite number of terms and where C_0 is an inverse gamma distribution with fixed hyperparameters a and b. As discussed in the previous section, Stan cannot be used to fit models containing discrete valued parameters and, in this case, θ_s has a mixed discrete-continuous distribution with an atom of probability at θ_∞. As such, this prior cannot be used in its current form in an analysis with Stan. However, by integrating out θ_s we can find the marginal prior for $a_{s,ij}|\pi_s$ which is continuous provided $\theta_\infty > 0$. Specifically, we have $a_{s,ij}|\theta_s \sim \mathrm{N}(0, \theta_s)$, $s = 1, 2, \ldots, H$, and $\theta_s|\pi_s \sim (1 - \pi_s)\mathrm{IG}(a, b) + \pi_s\delta_{\theta_\infty}$. Then

$$\begin{aligned}
\pi(a_{s,ij}, \theta_s|\pi_s) =& \pi(a_{s,ij}|\theta_s)\pi(\theta_s|\pi_s) \\
=& \frac{\theta_s^{-1/2}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{a_{s,ij}^2}{\theta_s}\right) \times \left\{(1 - \pi_s)\frac{b^a\theta_s^{-a-1}}{\Gamma(a)} \exp\left(-\frac{b}{\theta_s}\right) + \pi_s\delta_{\theta_\infty}\right\} \\
=& (1 - \pi_s)\frac{b^a}{\Gamma(a)\sqrt{2\pi}}\theta_s^{-a-3/2} \exp\left\{-\frac{1}{\theta_s}\left(\frac{a_{s,ij}^2}{2} + b\right)\right\} \\
&+ \pi_s\frac{\theta_s^{-1/2}}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}\frac{a_{s,ij}^2}{\theta_s}\right)\delta_{\theta_\infty}.
\end{aligned}$$

Taking the integral of each half of the sum separately:

$$\begin{aligned}
&\int_0^\infty (1-\pi_s)\frac{b^a}{\Gamma(a)\sqrt{2\pi}}\theta_s^{-a-3/2}\exp\left\{-\frac{1}{\theta_s}\left(\frac{a_{s,ij}^2}{2}+b\right)\right\}d\theta_s \\
&\quad=(1-\pi_s)\frac{b^a}{\Gamma(a)\sqrt{2\pi}}2^{a+1/2}\Gamma\left(a+\frac{1}{2}\right)\left(a_{s,ij}^2+2b\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{1}{2}\right)}{\Gamma(a)\sqrt{2\pi}}b^a b^{-1/2}b^{1/2}2^{a+1/2}\left(a_{s,ij}^2+2b\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{1}{2}\right)}{\Gamma(a)\sqrt{2\pi}}b^{-1/2}(2b)^{a+1/2}\left(a_{s,ij}^2+2b\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{1}{2}\right)}{\Gamma(a)\sqrt{2\pi}}b^{-1/2}\left(\frac{a_{s,ij}^2+2b}{2b}\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{1}{2}\right)}{\Gamma(a)\sqrt{2\pi}}b^{-1/2}\left(1+\frac{a_{s,ij}^2}{2b}\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{1}{2}\right)}{\Gamma(a)\sqrt{2\pi}}a^{-1/2}\left(\frac{b}{a}\right)^{-1/2}\left(1+\frac{aa_{s,ij}^2}{2ab}\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{1}{2}\right)}{\Gamma(a)\sqrt{2\pi a(b/a)}}\left(1+\frac{a_{s,ij}^2}{2a(b/a)}\right)^{-a-1/2} \\
&\quad=(1-\pi_s)\mathrm{t}_{2a}(a_{s,ij};0,b/a).
\end{aligned}$$

Then

$$\int_0^\infty \frac{\pi_s\theta_s^{-1/2}}{\sqrt{2\pi}}\exp\left(-\frac{a_{s,ij}^2}{2\theta_s}\right)\delta_{\theta_\infty}(\theta_s)d\theta_s=\int_0^\infty \frac{\pi_s\theta_s^{-1/2}}{\sqrt{2\pi}}\exp\left(-\frac{a_{s,ij}^2}{2\theta_s}\right)\delta(\theta_s-\theta_\infty)d\theta_s.$$

Using,

$$\int_a^b f(x)\delta(x-c)dx=f(c)\ \text{ if }\ c\in[a,b]$$

we have

$$\begin{aligned}\int_0^\infty \frac{\pi_s\theta_s^{-1/2}}{\sqrt{2\pi}}\exp\left(-\frac{a_{s,ij}^2}{2\theta_s}\right)\delta(\theta_s-\theta_\infty)d\theta_s&\\ =\frac{\pi_s\theta_\infty^{-1/2}}{\sqrt{2\pi}}\exp\left(-\frac{a_{s,ij}^2}{2\theta_\infty}\right)&\\ =\frac{\pi_s}{\sqrt{2\pi\theta_\infty}}\exp\left(-\frac{a_{s,ij}^2}{2\theta_\infty}\right)&\\ =\pi_s\mathrm{N}(a_{s,ij};0,\theta_\infty).&\end{aligned}$$

Therefore

$$\pi(a_{s,11},\ldots,a_{s,m1},a_{s,12},\ldots,a_{s,mm}|\pi_s)=\prod_{i=1}^m\prod_{j=1}^m\pi(a_{s,ij}|\pi_s)$$

where

$$\pi(a_{s,ij}|\pi_s)=(1-\pi_s)\mathrm{t}_{2a}(0,b/a)+\pi_s\mathrm{N}(0,\theta_\infty).$$

Furthermore, this result generalises to allow prior dependence amongst the elements of the A_s matrices. Specifically, if we replace (5.13) with

$$\boldsymbol{a}_s=\mathrm{vec}(A_s)=(a_{s,11},\ldots,a_{s,mm})|\theta_s\sim\mathrm{N}_{m^2}(\mathbf{0},V\theta_s)$$

for $s=1,\ldots,H$, where V is an $m\times m$ positive definite matrix, then

$$\pi(\boldsymbol{a}_s|\pi_s)=(1-\pi_s)\mathrm{t}_{m^2,2a}(\mathbf{0},b/aV)+\pi_s\mathrm{N}_{m^2}(\mathbf{0},\theta_\infty V). \tag{5.14}$$

The calculation to show this is included in Appendix A.4 and proceeds in a similar way to the calculation above.

As discussed in Legramanti *et al.* (2020), if $b/a>\theta_\infty$, then

$$\Pr(|a_{s+1,ij}|\leq\epsilon)>\Pr(|a_{s,ij}|\leq\epsilon)$$

for each $i=1,\ldots,m$, $j=1,\ldots,m$, $s=1,\ldots,H-1$ and $\epsilon>0$. Therefore, choosing $b/a>\theta_\infty$ ensures cumulative shrinkage in distribution across lags. We must also choose a and b to ensure a marginal prior variance for the elements of the slab of less than or equal to one to avoid the multimodality in the prior induced for the partial autocorrelation matrices discussed in Section 5.2. Specifically, values of $a=3$ and $b=2$ satisfy this requirement and we use these values in Section 5.3.6. The choice for θ_∞ is more involved and we return to this point at the end of this subsection.

Effective order of the model

An A_s matrix is deemed active if it is assigned to the slab *a posteriori* and inactive if it is assigned to the spike. We refer to the number of active A_s matrices as the *effective order* of the model and denote it by p^*. From a computational perspective Legramanti *et al.* (2020) keep track of which parameter blocks are deemed active and inactive online through augmentation of the parameter space with discrete-valued auxiliary variables. These variables also facilitate construction of a simple Gibbs sampler (in the factor model setting in which it is presented) by making full conditional distributions analytically tractable. Specifically, denote by z_s, $s = 1, \dots, H$, a discrete valued variable with probability mass function $\Pr(z_s = l|\boldsymbol{\omega}) = \omega_l$, where $\boldsymbol{\omega} = (\omega_1, \dots, \omega_H)^{\mathrm{T}}$. These indicator variables are set up such that marginalising out the indicator variables in

$$\theta_s|z_s \sim \{1 - \mathbb{1}(z_s \leq s)\}\mathrm{IG}(a, b) + \mathbb{1}(z_s \leq s)\delta_{\theta_\infty}, \qquad z_s|\boldsymbol{\omega} \sim \mathrm{Mult}(1, \boldsymbol{\omega}),$$

results in the cumulative shrinkage distribution in (5.10). The z_s can be interpreted as discrete valued auxiliary variables that indicate whether a θ_s is sampled from the spike or the slab. In particular, if $z_s \leq s$ then θ_s is sampled from the spike, and if $z_s > s$ then θ_s is sampled from the slab. Therefore

$$\mathbb{1}(z_s \leq s) = \begin{cases} 1 & \text{if } \theta_s \text{ sampled from spike,} \\ 0 & \text{if } \theta_s \text{ sampled from slab.} \end{cases}$$

Then p^* can be calculated as the total number of θ_s, $s = 1, \dots, H$, which have been sampled from the slab. That is

$$\begin{aligned} p^* &= \sum_{s=1}^{H} \{1 - \mathbb{1}(z_s \leq s)\} \\ &= H - \sum_{s=1}^{H} \mathbb{1}(z_s \leq s). \end{aligned} \tag{5.15}$$

Legramanti *et al.* (2020) sample the z_s, allowing computation of p^* for every posterior draw. The posterior draws can later be tabulated to compute the posterior mass function for p^*. As we are using Stan for computational inference, we cannot use the auxiliary variable approach. Moreover we integrate the mixed discrete-continuous variable θ_s from our prior and so we cannot keep track of p^* online. However, we can still compute the posterior mass function for p^* offline, using a completed MCMC run by using some statistical theory and Rao-Blackwellisation. In particular, we can define

$$p^* = H - \sum_{s=1}^{H} J_s$$

where $J_1, \ldots, J_H$ are Bernoulli random variables which are conditionally independent given the parameters of the model. Conditional on the model parameters, J_s has success probability

$$\Pr(z_s \leq s | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma) = \sum_{k=1}^{s} \Pr(z_s = k | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma)$$

where

$$\begin{aligned}
\Pr(z_s = k | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma) &= \frac{\Pr(A_s | z_s = k, \boldsymbol{\omega}) \Pr(z_s = k | \boldsymbol{\omega})}{\Pr(A_s | \boldsymbol{\omega})} \\
&\propto \Pr(A_s | z_s = k, \boldsymbol{\omega}) \Pr(z_s = k | \boldsymbol{\omega}) \\
&\propto \Pr(A_s | z_s = k, \boldsymbol{\omega}) \omega_k \\
&\propto \begin{cases} \omega_k \mathrm{N}_{m^2}(A_s; \mathbf{0}, \theta_\infty V) & \text{for } k = 1, \ldots, s \\ \omega_k \mathrm{t}_{m^2, 2a}(A_s; \mathbf{0}, b/aV) & \text{for } k = s+1, \ldots, H. \end{cases}
\end{aligned}$$

Therefore, conditional on the model parameters, $J_1, \ldots, J_H$ form a sequence of independent but not identically distributed Bernoulli random variables and hence $J^* = \sum_{s=1}^{H} J_s$ has a Poisson binomial distribution. Then we can define $p^* = H - J^*$ and so

$$\Pr(p^* = k | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma) = \Pr(J^* = H - k | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma)$$

for each $k = 0, \ldots, H-1$. Note that $\Pr(p^* = H) = 0$ by construction. For a given sample of the parameters we can evaluate $\Pr(p^* = k | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma)$ for each $k = 0, \ldots, H-1$. Then, given a sample

$$(A_1^{(m)}, \ldots, A_H^{(m)}, \boldsymbol{\omega}^{(m)}, \Sigma^{(m)} | \boldsymbol{y}), \quad m = 1, \ldots, M$$

of size M from the posterior distribution $p(A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma | \boldsymbol{y})$, we can calculate the Rao-Blackwellised estimate of $\Pr(p^* = k | \boldsymbol{y})$, $k = 0, \ldots, H$, as

$$\begin{aligned}
\Pr(p^* = k | \boldsymbol{y}) &= \int \Pr(p^* = k, A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma | \boldsymbol{y}) (dA_1) \ldots (dA_H) (d\boldsymbol{\omega}) (d\Sigma) \\
&= \int \Pr(p^* = k | A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma) \\
&\quad \times p(A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma | \boldsymbol{y}) (dA_1) \ldots (dA_H) (d\boldsymbol{\omega}) (d\Sigma) \\
&\approx \frac{1}{M} \sum_{m=1}^{M} \Pr(p^* = k | A_1^{(m)}, \ldots, A_H^{(m)}, \boldsymbol{\omega}^{(m)}, \Sigma^{(m)}),
\end{aligned}$$

giving the posterior mass function for p^*. If all we want is an approximation of $\mathrm{E}(p^*|\boldsymbol{y})$, rather than the full posterior mass function, then there is no need to calculate the posterior mass function first. Instead, a simpler calculation is available, and is detailed in Appendix A.5.

Choice of θ_∞

The choice of θ_∞ under the CUSP prior is important due to its influence on inference of the order of the process. Ideally, we want our choice of θ_∞ to reflect the length and dimension of the time series. To this end, we begin by considering the partial autocorrelation matrices P_s in order to provide a steer on what might constitute a value that is "near zero".

In classical time series analysis, the partial autocorrelation plot, with its associated confidence intervals, plays an important role in the choice of order for a univariate autoregression. Under the hypothesis that the process is AR(p), the estimators for the partial autocorrelations of order $p+1, p+2, \ldots$ based on a sample of size n are approximately independent with mean equal to zero and variance equal to $1/n$. As a guide, we can therefore approximate the posterior for the m^2 components $p_{s,ij}$ of P_s under this hypothesis as independent $\mathrm{N}(0, 1/n)$ random variables. If we let $\tilde{\boldsymbol{p}}_s = \mathrm{vec}(P_s)$ then this leads to the approximation

$$\tilde{\boldsymbol{p}}_s \sim \mathrm{N}_{m^2}\left(\mathbf{0}, \frac{1}{n} I_{m^2}\right).$$

Unfortunately, the CUSP prior operates on the unconstrained A_s matrices, rather than the P_s matrices, and so what is actually required is the approximate distribution of A_s under this hypothesis. To this end, let $\tilde{\boldsymbol{a}}_s = \mathrm{vec}(A_s)$. Then

$$\tilde{\boldsymbol{a}}_s = f(\tilde{\boldsymbol{p}}_s) = \mathrm{vec}\left\{(I_m - PP^{\mathrm{T}})^{-1/2} P\right\}.$$

The Taylor expansion of $f(\tilde{\boldsymbol{p}}_s)$ up to first order, about $\tilde{\boldsymbol{p}}_s = 0$, is

$$f(\tilde{\boldsymbol{p}}_s) \approx f(\mathbf{0}) + \frac{\partial}{\partial \tilde{\boldsymbol{p}}_s^{\mathrm{T}}} f(\mathbf{0}) \tilde{\boldsymbol{p}}_s.$$

Heaps & Jermyn (2023) show that

$$\begin{aligned}
\frac{\partial \tilde{\boldsymbol{a}}_s}{\partial \tilde{\boldsymbol{p}}_s^{\mathrm{T}}} &= \frac{\partial f(\tilde{\boldsymbol{p}}_s)}{\partial \tilde{\boldsymbol{p}}_s^{\mathrm{T}}} \\
&= (P_s \otimes I_m)^{\mathrm{T}} \left\{(I_m - P_s P_s^{\mathrm{T}}) \otimes (I_m - P_s P_s^{\mathrm{T}})^{1/2} + (I_m - P_s P_s^{\mathrm{T}})^{1/2} \otimes (I_m - P_s P_s^{\mathrm{T}})\right\}^{-1} \\
&\times \left\{(P_s \otimes I_m) + (I_m \otimes P_s) I_{(m,m)}\right\} + \left\{I_m \otimes (I_m - P_s P_s^{\mathrm{T}})^{1/2}\right\}^{-1}
\end{aligned}$$

where $I_{(m,m)}$ is the $m^2 \times m^2$ commutation matrix such that $\mathrm{vec}(X) = I_{(m,m)} \mathrm{vec}(X)^{\mathrm{T}}$.

When evaluated at $P_s = 0_m$, this results in

$$\frac{\partial f(\mathbf{0})}{\partial \tilde{\boldsymbol{p}}_s^{\mathrm{T}}} = I_{m^2}$$

and as such

$$\tilde{\boldsymbol{a}}_s = f(\tilde{\boldsymbol{p}}_s) \approx \mathbf{0} + I_{m^2}\tilde{\boldsymbol{p}}_s = \tilde{\boldsymbol{p}}_s.$$

Therefore, up to the first order, $\tilde{\boldsymbol{a}}_s = \tilde{\boldsymbol{p}}_s$, and so under this hypothesis we can approximate the posterior for $\mathrm{vec}(A_s)$ as $\mathrm{N}_{m^2}\left(0, \frac{1}{n} I_{m^2}\right)$. We then wish to find θ_∞ such that under this hypothesis, the hypothetical posterior probability of assigning A_s to the spike is large. As such, we need to solve

$$\Pr_{\theta_\infty}(z_s \leq s | \boldsymbol{y}_0) = \beta \tag{5.16}$$

for θ_∞, where $\boldsymbol{y}_0$ is a hypothetical data set of length n which leads to the posterior for $\mathrm{vec}(A_s)$ being well approximated by $\mathrm{N}_{m^2}\left(0, \frac{1}{n} I_{m^2}\right)$ and where θ_∞ in the subscript on the left-hand-side indicates that the probability is a function of θ_∞. The calculation will also depend on the choices of the prior hyperparameters a and b in the slab and on the prior for $\boldsymbol{\omega}$. Unfortunately, the calculation becomes complicated if we allow $\boldsymbol{\omega}$ to be unknown, as we don't have any theory for the approximate distribution of $\boldsymbol{\omega}|\boldsymbol{y}_0$. Therefore, to simplify the calculation we fix $\boldsymbol{\omega}$ at its prior mean, given in (5.11), which we denote by $\boldsymbol{\omega}_0$. Then,

$$\begin{aligned}
\Pr_{\theta_\infty}(z_s \leq s | \boldsymbol{y}_0, \boldsymbol{\omega}_0) &= \int_{A_s} \Pr_{\theta_\infty}(z_s \leq s | A_s, \boldsymbol{\omega}_0) p(A_s | \boldsymbol{y}_0, \boldsymbol{\omega}_0)(dA_s) \\
&= \int_{A_s} \sum_{k=1}^{s} \Pr_{\theta_\infty}(z_s = k | A_s, \boldsymbol{\omega}_0) p(A_s | \boldsymbol{y}_0, \boldsymbol{\omega}_0)(dA_s) \\
&= \int_{A_s} \frac{\sum_{k=1}^{s} \Pr_{\theta_\infty}(A_s | z_s = k, \boldsymbol{\omega}_0) \Pr(z_s = k | \boldsymbol{\omega}_0)}{\Pr_{\theta_\infty}(A_s | \boldsymbol{\omega}_0)} p(A_s | \boldsymbol{y}_0, \boldsymbol{\omega}_0)(dA_s) \\
&= \int_{A_s} \left[\frac{\sum_{k=1}^{s} \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s); \mathbf{0}, \theta_\infty I_{m^2}\right\} \omega_{0,k}}{\pi_s \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s); \mathbf{0}, \theta_\infty I_{m^2}\right\} + (1 - \pi_s) t_{2a}\left\{\mathrm{vec}(A_s); \mathbf{0}, \frac{b}{a} I_{m^2}\right\}} \right] \\
&\quad \times p(A_s | \boldsymbol{y}_0, \boldsymbol{\omega}_0)(dA_s) \\
&= \int_{A_s} \left[\frac{\pi_{0,s} \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s); \mathbf{0}, \theta_\infty I_{m^2}\right\}}{\pi_{0,s} \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s); \mathbf{0}, \theta_\infty I_{m^2}\right\} + (1 - \pi_{0,s}) t_{2a}\left\{\mathrm{vec}(A_s); \mathbf{0}, \frac{b}{a} I_{m^2}\right\}} \right] \\
&\quad \times p(A_s | \boldsymbol{y}_0, \boldsymbol{\omega}_0)(dA_s) \\
&\approx \frac{1}{M} \sum_{j=1}^{M} \left[\frac{\pi_{0,s} \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s^{(j)}); \mathbf{0}, \theta_\infty I_{m^2}\right\}}{\pi_{0,s} \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s^{(j)}); \mathbf{0}, \theta_\infty I_{m^2}\right\} + (1 - \pi_{0,s}) t_{2a}\left\{\mathrm{vec}(A_s^{(j)}); \mathbf{0}, \frac{b}{a} I_{m^2}\right\}} \right]
\end{aligned}$$

where the $A_s^{(j)}$, $j = 1, \ldots, M$, are a sample from the approximate distribution

$$p(A_s|\boldsymbol{y}_0, \boldsymbol{\omega}_0) \approx \mathrm{N}_{m^2}\left\{\mathrm{vec}(A_s); \mathbf{0}, \frac{1}{n} I_{m^2}\right\}$$

and $\pi_{0,s} = \sum_{k=1}^{s} \omega_{0,k}$ is the prior mean for π_s, given in (5.12). For any particular m and n we can then calculate this probability over a grid of values for θ_∞ in order to find a value of θ_∞ which gives approximately the value chosen for β, say $\beta = 0.99$. Clearly, however, for any particular value of θ_∞, the probability $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0, \boldsymbol{\omega}_0)$ will be different for each $s = 1, \ldots, H-1$. Therefore the value of θ_∞ which solves (5.16) will be different for different values of s. As such, we must choose a single A_s matrix to use to calculate an appropriate θ_∞. We choose to calculate θ_∞ for $s = \alpha + 1$, as α is the prior expectation for the number of matrices modelled by the slab, or equivalently the prior expectation for the order of the process. Therefore, the first A_s matrix we could reasonably consider to be approximated by $\mathrm{N}_{m^2}\left\{\mathbf{0}, \frac{1}{n} I_{m^2}\right\}$ based on our prior expectation is $A_{\alpha+1}$.

5.3.4 Multiplicative gamma process

An alternative increasing shrinkage prior is the multiplicative gamma process (MGP) (Bhattacharya & Dunson, 2011) originally developed as a structured sequence of global-local shrinkage priors for the loadings matrix in infinite factor models. We adopt a prior of this form by choosing

$$a_{s,ij}|\lambda_{s,ij}, \tau_s \sim \mathrm{N}(0, \lambda_{s,ij}^{-1}\tau_s^{-1}),$$

independently for $i, j = 1, \ldots, m$, $s = 1, \ldots, p_{\max}$, where the local precision parameters at lag s are assigned the prior

$$\lambda_{s,ij} \sim \mathrm{Gam}(a/2, a/2),$$

independently for $i, j = 1, \ldots, m$, $s = 1, \ldots, p_{\max}$, and the global precision parameter at lag s is constructed as

$$\tau_s = \prod_{k=1}^{s} \delta_k, \quad \delta_1 \sim \mathrm{Gam}(a_1, 1), \quad \delta_k \sim \mathrm{Gam}(a_2, 1), \quad k \geq 2$$

in which the δ_k are independent. The global precisions τ_s are therefore a cumulative product of gamma random variables whose prior expectation $\mathrm{E}(\tau_s)$ increases with s when $a_2 > 1$. A set of random variables, X_s, $s = 1, 2, \ldots$, are said to be stochastically decreasing if

$$\Pr(X_s < \epsilon) \leq \Pr(X_{s+1} < \epsilon)$$

for some ϵ. Unfortunately, as discussed in Durante (2017), the $\theta_s = 1/\tau_s$ are not stochastically decreasing in s in the general case. However guidelines on the choice of hyperpa-

rameter a_1 and a_2 can be found in Durante (2017) who presents a numerical method for checking that the choice of hyperparameters a_1 and a_2 lead to global variances θ_s that are stochastically decreasing in s near zero, that is, $\Pr\{\theta_s \in (0, \theta]\}$ is non-decreasing in s for any θ in a small neighbourhood of zero. Whilst the θ_s are not stochastically decreasing in the general case, being stochastically decreasing in a small neighbourhood of zero can still be sufficient to facilitate increasing shrinkage. This property holds for all $a_1 > 0$ and $a_2 > 0$ but for some values of a_1 and a_2 the neighbourhood of zero may be particularly small. The numerical method described in Durante (2017) can be used to ensure that the neighbourhood $(0, \theta]$ is reasonably large to allow for increasing shrinkage in practice.

The multiplicative gamma process prior does not place any mass at zero and so none of the A_s, and hence P_s, matrices are shrunk exactly to zero. As such, under this prior a choice is needed on when to truncate the A_s matrices to zero. This choice of truncation criteria effectively replaces the choice of θ_∞ in the cumulative shrinkage prior, discussed in Section 5.3.3. We define the *effective order* p^* of the model as the value of $s \leq p_{\max}$ such that P_s fails a criterion for truncation to zero when $s = p^*$ but passes for $s = p^* + 1, \ldots, p_{\max}$. As in the choice of θ_∞ in the CUSP prior, we appeal to classical time series theory on partial autocorrelation estimators to guide our choice of truncation criteria such that it is robust with respect to the length n and dimension m of the data. It is, however, more straightforward in this case as the truncation criteria can be applied directly to the P_s matrices rather than the unconstrained A_s matrices.

Following Bhattacharya & Dunson (2011), we choose to truncate P_s to a zero matrix if the absolute value of all of its elements lie below some threshold, say ε. As discussed in Section 5.3.3, under the hypothesis that the process is AR(p) we can approximate the posterior for the m^2 components $p_{s,ij}$ of P_s, for $s > p$, as independent $\mathrm{N}(0, 1/n)$ random variables. We can then compute the quantile $q_m(\beta)$ such that

$$\Pr\left\{\max_{i,j} |p_{s,ij}| < q_m(\beta)\right\} = \beta$$

for some large value of β. Under the hypothesis that each of the components $p_{s,ij}$ are identically distributed independent $\mathrm{N}(0, 1/n)$ random variables, define

$$Y = \max\{|p_{s,ij}|, i = 1, \ldots, m, j = 1, \ldots, m\} = \max\{|X_k|, k = 1, \ldots, m^2\}$$

where $X_k \sim \mathrm{N}(0, 1/n)$ for $k = 1, \ldots, m^2$. Then, the cumulative distribution function of Y

is

$$\begin{aligned} F(y) &= P(Y \leq y) \\ &= P\left(\prod_{k=1}^{m^2} |X_k| \leq y\right) \\ &= \prod_{k=1}^{m^2} P\left(|X_k| \leq y\right) \\ &= \prod_{k=1}^{m^2} \left\{2\Phi\left(\frac{y}{\sqrt{1/n}}\right) - 1\right\} \\ &= \left\{2\Phi\left(\frac{y}{\sqrt{1/n}}\right) - 1\right\}^{m^2} \end{aligned}$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution. Taking $y = q_m(\beta)$ and $F\{q_m(\beta)\} = \beta$ gives

$$\beta = \left[2\Phi\left\{\frac{q_m(\beta)}{\sqrt{1/n}}\right\} - 1\right]^{m^2}. \tag{5.17}$$

Taking the inverse of (5.17), we can set the threshold

$$\varepsilon = q_m(\beta) = \frac{\Phi^{-1}\{(\beta^{1/m^2} + 1)/2\}}{\sqrt{n}}.$$

In the applications in this chapter and Chapter 6, we use $\beta = 0.99$. By choosing the threshold in this way, we account for both the length n and dimension m of the data, in addition to operating on a parameter which is unit-free.

Practical implementation of this criterion to learn about the order of the process is straightforward. We simply apply it to each draw from the posterior to obtain posterior samples of p^*. These can be summarised to yield a numerical approximation of the posterior for p^* which provides a proxy for the posterior for p.

5.3.5 Posterior inference

For a $\text{VAR}_m(p_{\max})$ process, regarding the likelihood described in Section 5.2.2 as a function of the transformed partial autocorrelation matrices and combining it with the prior (5.9) via Bayes' theorem yields the posterior distribution as

$$\pi(\Sigma, A_1, \ldots, A_{p_{\max}}, \boldsymbol{\vartheta} \mid \boldsymbol{y}_{1:n}) \propto p(\boldsymbol{y}_{1:n} \mid \Sigma, A_1, \ldots, A_{p_{\max}})\pi(\Sigma)\pi(\boldsymbol{\vartheta}) \prod_{s=1}^{p_{\max}} \pi(A_s \mid \boldsymbol{\vartheta}). \tag{5.18}$$

As discussed in Section 5.3.2, we use HMC via Stan to sample from the posterior distribution under both the CUSP and MGP priors. In the next section we consider analysis of simulated data. We consider real data in Chapter 6.

5.3.6 Simulation experiments

Consider the idealised setting in which we know that the data were generated from a stationary vector autoregression of known order, p. In order to explore the behaviour of the posterior distribution for p^* in this context, we carried out simulation experiments that considered data generated from processes whose orders took various values.

Under the CUSP and MGP priors our choice of θ_∞ and truncation criterion, respectively, make allowance for the dimension of the observation vector m and the length of the time series n. We might therefore expect some degree of robustness in the more challenging inferential situations when n is small or, in particular, when m is large. This was investigated by considering simulations under a variety of values of m and n.

Cumulative shrinkage process

The Stan programme for this model is provided in Appendix C.4. For each $m \in \{1, 3, 5, 7\}$ and $p \in \{1, 2, 3, 4\}$ we simulated ten sets of $m \times m$ matrices $A_1, \ldots, A_p$ with elements sampled independently from a standard normal distribution. Taking the error variance matrix to be $\Sigma = I_m$, these were used to simulate ten $\text{VAR}_m(p)$ processes of length $n = 1000$. Conditional on each data set, we then generated samples from the posterior distribution using Stan, as described in Section 5.3.5, restricting the inference to a maximum, finite number of terms by setting $H = 8$. As discussed in Section 5.3.3, this is equivalent to setting the maximum possible order to $p_{\max} = 7$. In our analysis we set $V = I$ in (5.14) and as discussed in Section 5.3.3, we set $a = 3$ and $b = 2$. Additionally we take $\alpha = 3$, to represent a prior expectation for the order of the VAR process that may be larger, smaller or equal to the true value. For these values of a, b and α, and for $m = 3$, 5, and 7, setting $\theta_\infty = 0.17$, 0.18 and 0.135 respectively gives $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) \approx 0.99$ for $s = \alpha + 1$, as discussed in Section 5.3.3. In the univariate case, where $m = 1$, all values of θ_∞ give $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) < 0.99$. In this case, we choose the value of θ_∞ which gives a probability that is as close to 0.99 as possible. In particular, for $m = 1$, we take $\theta_\infty = 0.002$, which gives $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) \approx 0.97$. In the inverse Wishart prior for Σ we use the same prior specification as in Section 5.2, taking the scale matrix to be the identity matrix, I_m, and the degrees of freedom to be $m + 4$. In all cases, we used four chains each with 1000 iterations of warm-up followed by 4000 sampling iterations.

To summarise our results, the posterior mass function for p^* in each experiment was calculated. For a given (m, p) the posterior mass functions for the ten data sets are pre-

sented as a set of overlaid bar charts in Figure 5.7. In nearly all cases, the posterior mode is at the true order, p. In addition, as m increases the level of uncertainty decreases. One reason for this could be that in the univariate case there is only one parameter $a_{s,11}$ providing information about each θ_s and so the assignment of θ_s to either the spike or slab is very uncertain. In contrast, as m increases the number of parameters providing information about each θ_s increases quadratically and so there is an increased level of certainty for the assignment to the spike or slab. As an example run time for the experiments, on a CentOS Linux 7 (Core) 64bit operating system with an Intel Xeon E5-2699 v4 processor (2.2 GHz), the average run time across all chains and experiments when $m = 5$ and $p = 4$ was 106.1 hours. The average minimum ESS achieved across the ten experiments when $m = 5$ and $p = 4$ was 1374. Whilst run times and minimum ESS are not provided for all simulations, due to the large number (160) of experiments, in general the run time increased with m or p, whilst the minimum ESS decreased.

Fixing $m = 3$ and considering $p \in \{1, 2, 3, 4\}$, we then simulated ten $\text{VAR}_m(p)$ processes of length $n = 100$ and another ten of length $n = 500$ using the same ten sets of matrices $A_1, \ldots, A_p$ as in the previous experiments, facilitating comparisons across $n \in \{100, 500, 1000\}$. Using the same prior specification as in the previous experiments we fit the model using Stan. The results for this study are given in Figure 5.8. Even for a short time series of length $n = 100$, the true order is the mode in the posterior mass function in nearly all cases. Whilst a long time series isn't needed to be able to recover the true order, an increase in the length of the process results in a reduced level of uncertainty, as would be expected given the availability of more information to update the prior.

Under the cumulative shrinkage prior the truncation value H not only limits the support of the prior but also influences how the prior mass is spread across values of p. Therefore the posterior may be notably influenced by the choice of H. At the very least, it is important to choose H such that it is conservative enough to ensure that $p_{\max} = H - 1$ is larger than any p with reasonable support from the data. Considering the simulated data sets described above for the case where $m = 3$, $p = 3$ and $n = 1000$, we fit the model in Stan using the same prior specifications as in the previous experiment but varying the truncation value with $H \in \{2, 4, 6, 8\}$. The results are summarised in Figure 5.9. In all cases where $p_{\max} \geq p$ the true order $p = 3$ is recovered. As would be expected, in the case where $p_{\max} = 1$ and so the maximum permitted order is less than the true order, the true order of the process is not recovered. Instead, the posterior suggests that the order is equal to $p_{\max}$. This highlights the necessity of ensuring that H is sufficiently large to ensure that $p_{\max} \geq p$. Naturally, as the order of the process in real-life situations is not usually known, the initial choice of H may end up being too small. In cases where there is a high posterior probability of the order being equal to $p_{\max}$ then this is an indication that H may not be large enough and that inference should be repeated with a larger value.

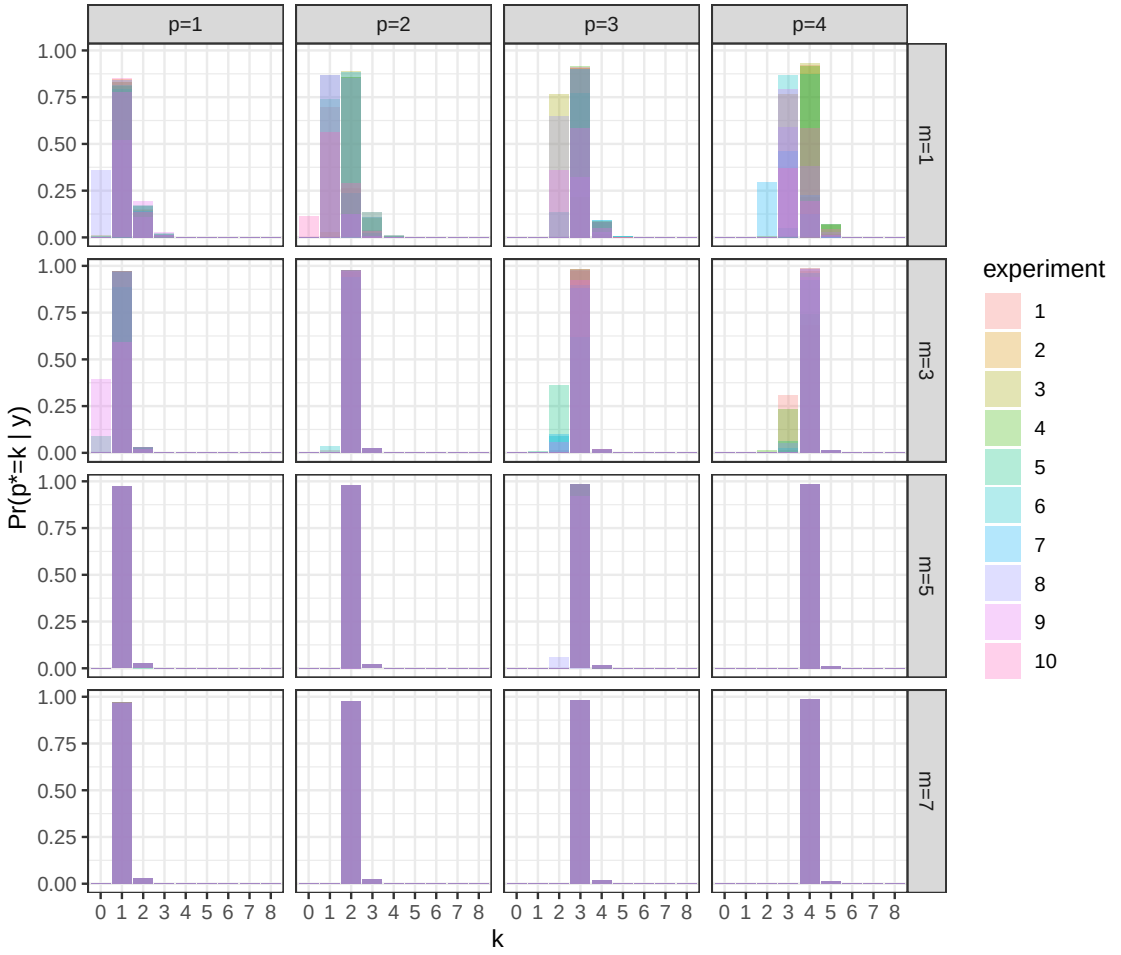


Figure 5.7: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each combination of $m \in \{1, 3, 5, 7\}$ and $p \in \{1, 2, 3, 4\}$, with $n = 1000$ under the CUSP prior.

Multiplicative gamma process

As for the cumulative shrinkage process, we investigate the posterior behaviour of p^* under the multiplicative gamma process using a range of simulation experiments. Using the same sets of $\text{VAR}_m(p)$ processes ($m \in \{1, 3, 5, 7\}, p \in \{1, 2, 3, 4\}$) as under the cumulative shrinkage process above, we generated samples from the posterior distribution using Stan, setting the maximum possible order as $p_{\max} = 7$. The Stan programme for this model is provided in Appendix C.5. We followed guidelines provided in Durante (2017) to choose the hyperparameters of the multiplicative gamma process prior, taking $a_1 = 2.5$ and $a_2 = 3$. Additionally, we set $a = 6$. These hyperparameter choices result in a marginal prior for the elements of the unconstrained A_s matrices which has variance of less than one to avoid multimodality in the prior, as discussed in Section 5.2. In the inverse Wishart distribution used as a prior for Σ, the scale matrix is taken as I_m and the degrees of

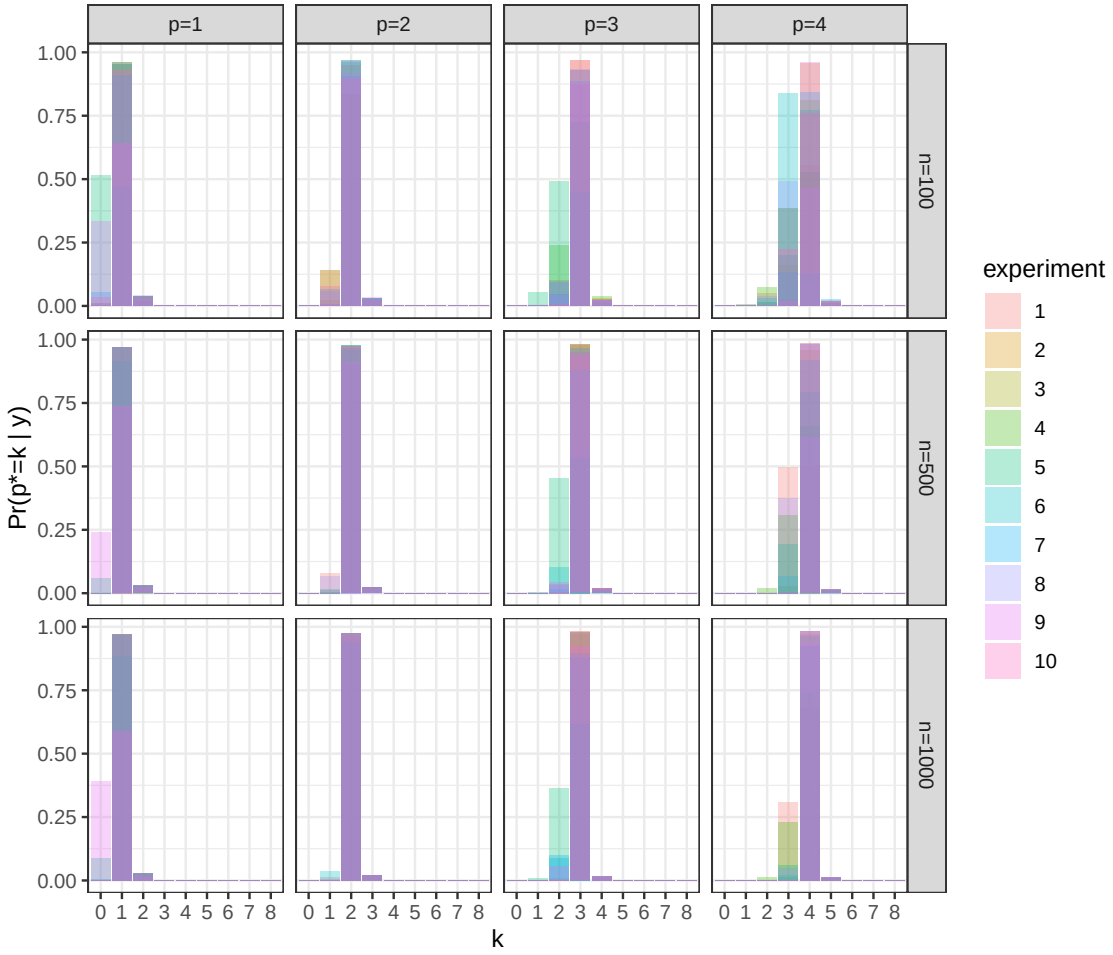


Figure 5.8: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $n \in \{100, 500, 1000\}$ under the CUSP prior, with $m = 3$, $p = 3$ and $H = 8$.

freedom as $m + 4$, which is the same prior specification used in the simulation studies under the cumulative shrinkage process and for the known p case in Section 5.2. In all cases, we used four chains each with 1000 iterations of warm-up followed by 4000 sampling iterations. Using the truncation criteria with $\beta = 0.99$, we calculated the limits $q_m(\beta)$ as 0.081, 0.103, 0.112 and 0.117 for $m = 1$, 3, 5, and 7 respectively, and obtained a posterior mass function for the effective order p^* of each process. The posterior mass functions are summarised in Figure 5.10 across all simulation experiments. In nearly all cases, the true order p of the process is the mode in the posterior for p^*, with considerable posterior support. The results are similar across different values of m and p, though the posterior uncertainty reduces slightly as m increases. The similarity across the range of values of m and p suggests robustness to the dimension of the data through our choice of truncation criteria. Across the experiments, most of the posterior mass that is not at the mode

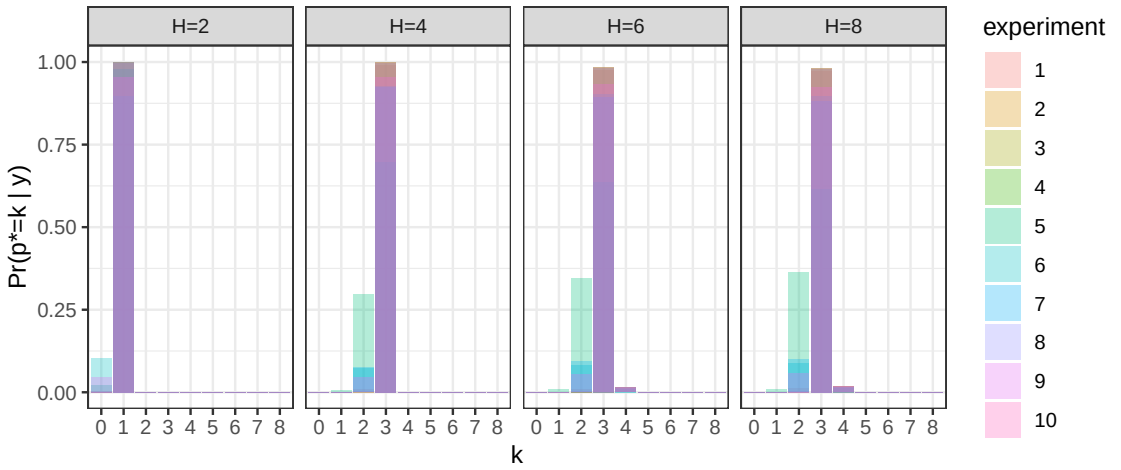


Figure 5.9: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each of $H \in \{2, 4, 6, 8\}$ under the CUSP prior, with $m = 3$, $p = 3$ and $n = 1000$. Recall that $p_{\max} = H - 1$.

appears to favour values of $p^* > p$. In contrast, the posterior mass under the CUSP prior in Figure 5.7 appeared to favour values of $p^* < p$. This is influenced by the prior mass functions which have more mass at higher values under the MGP prior but more mass at lower values under the CUSP prior. As an example run time for the experiments, on the same CentOS Linux 7 (Core) 64bit operating system with an Intel Xeon E5-2699 v4 processor (2.2 GHz) as in the CUSP simulations, the average run time across all chains and experiments when $m = 5$ and $p = 4$ was 218.7 hours with an average minimum ESS of 1048. The time taken for the simulations under the MGP prior is notably longer than under the CUSP prior and results in a lower minimum ESS. However, in Chapter 6, we find that the run times for the MGP in the case of real data are much quicker than with simulated data. As with the CUSP simulations, in general the run time increased with m or p, whilst the minimum ESS decreased.

Fixing $m = 3$, considering $p \in \{1, 2, 3, 4\}$ and using the same ten sets of matrices $A_1, \ldots, A_p$ as in the previous experiment, we then simulated ten $\text{VAR}_m(p)$ processes of length $n = 100$ and another ten of length $n = 500$, facilitating comparison across $n \in \{100, 500, 1000\}$. Retaining the same prior specification in the new experiments, we fit the model using HMC via Stan, as discussed above. Again, using the truncation criteria with $\beta = 0.99$ led to limits $q_m(\beta)$ equal to 0.326, 0.146 and 0.103 for $n = 100$, 500 and 1000, respectively. This yielded the posterior mass functions for p^* which are displayed in Figure 5.11. Across all experiments for the different values of n, the posterior mode for the effective order p^* recovers the true order p of the process, again, with considerable support. This holds for all values of n, suggesting robustness through the choice of truncation

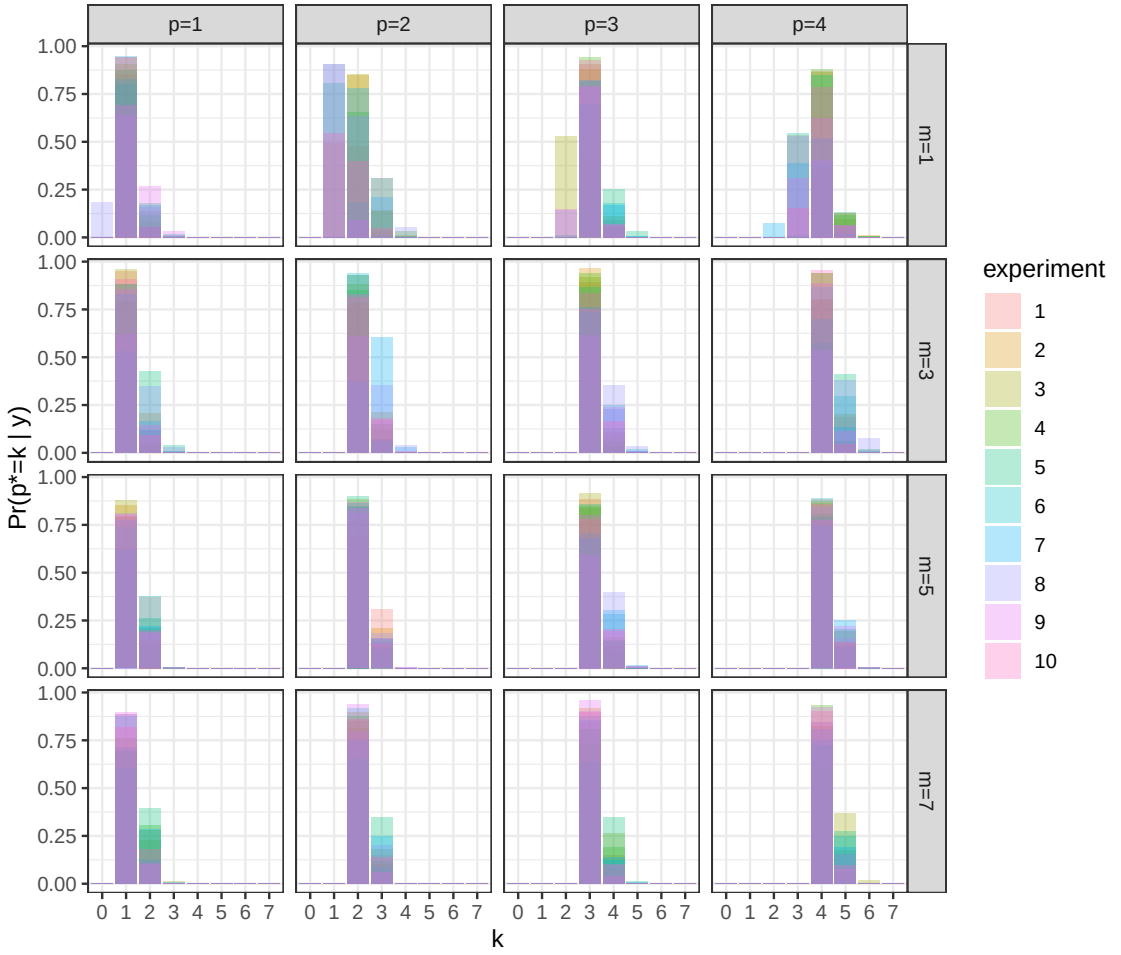


Figure 5.10: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each combination of $m \in \{1, 3, 5, 7\}$ and $p \in \{1, 2, 3, 4\}$, with $n = 1000$, under the MGP prior.

criteria, even for short time series.

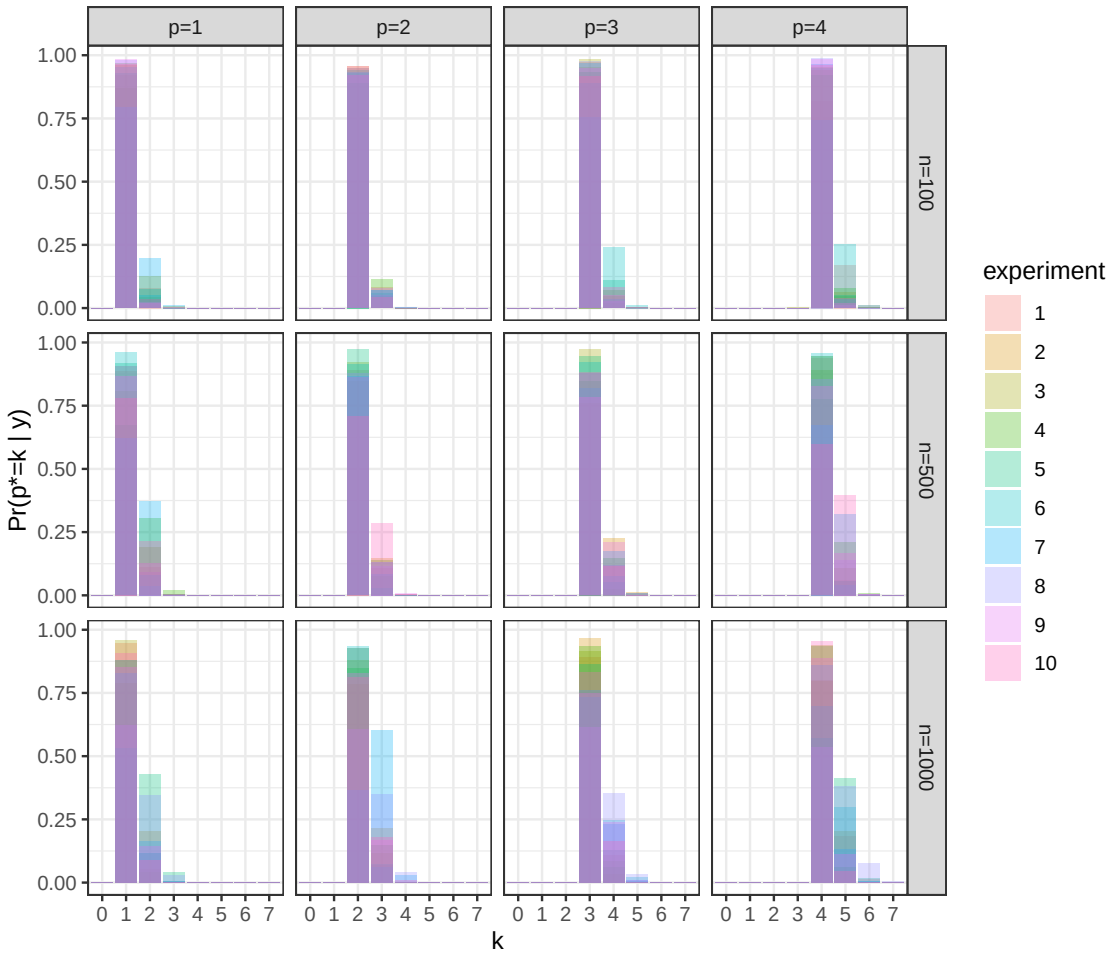


Figure 5.11: Overlaid posterior mass functions for the effective order p^* from 10 experiments for each combination of $n \in \{100, 500, 1000\}$ and $p \in \{1, 2, 3, 4\}$, with $m = 3$, under the MGP prior.

Chapter 6

Application to EEG data

Having observed promising results when determining the order of simulated vector autoregressive processes using both the cumulative shrinkage and the multiplicative gamma process priors, in this chapter we will consider the more interesting case of real world data. In Chapter 3 we found that when fitting univariate models to individual regions of our EEG data the resulting residuals were positively correlated suggesting the data should be modelled as a multivariate time series. Therefore, in this chapter we will model the EEG data across regions jointly as VAR processes with an unknown order that we wish to learn. As discussed in Chapter 3, our data comprises eight data sets, made up of four individuals (A, B, C and D) with EEG bandpower recordings from two frequency bands (beta and delta) in each individual. The number of regions m in which data were recorded and the length n of the recording varied across the individuals and are detailed in Table 3.2.

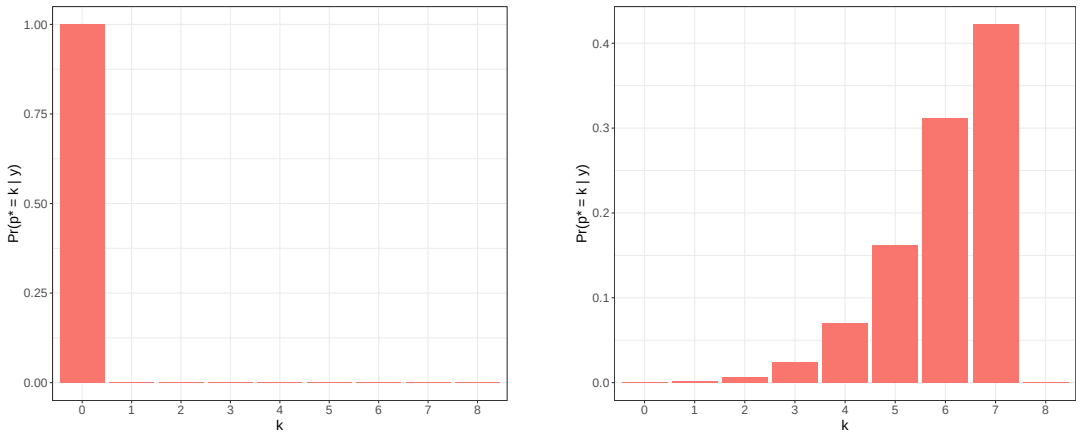
6.1 Missing data

In the data sets chosen for analysis for individual A, there are some missing observations. For the time points with missing observations, the data are missing in all regions. In our inference we assume that the data are missing at random so that the joint distribution of the missing and observed data given the parameters is simply determined by the vector autoregressive model. Then, as part of our MCMC scheme, we can obtain a posterior sample for each of the missing values by sampling them along with the parameters of the model, from the joint posterior of all the unknowns.

6.2 Cumulative shrinkage process

When applying the cumulative shrinkage process prior to simulated data in Section 5.3.6, we obtained promising results. However, when using this prior to perform inference using the EEG data, we observed peculiar behaviour in the posterior mass function for the

effective order, p^*. As an example, we consider inference on the data from the delta band in individual B. For this individual, $m = 8$ and $n = 622$. In order to carry out inference we set $\theta_\infty = 0.115$ which gave $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) \approx 0.99$ at lag 4 for these values of m and n, as discussed in Section 5.3.3. For the other hyperparameter values in the CUSP prior, we used the same values as in Section 5.3.6, taking $a = 3$, $b = 2$, $\alpha = 3$, $H = 8$ and $V = I$. In the posterior density for p^* obtained from fitting this model, depicted in Figure 6.1(a), a large portion of the posterior mass is stacked at zero. This result, which was replicated across the other data sets, is not consistent with our exploratory data analysis, discussed in Chapter 3, or the results obtained when using the multiplicative gamma process prior, discussed in the next section. As a result of these unexpected posteriors, we conducted an experiment to investigate the sensitivity in the posterior to the prior hyperparameters, in particular the value of θ_∞. In Section 5.3.3, we discussed the difficulties in choosing an appropriate value for θ_∞. We suggested choosing a value of θ_∞ which resulted in $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) \approx 0.99$ at lag $s = \alpha + 1$, resulting in the choice of $\theta_\infty = 0.115$ for the data sets for individual B. However, upon further investigation, when performing inference on the EEG data we discovered small changes in θ_∞ can result in big changes in the probability $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0)$. Figure 6.2, shows the value of $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0)$ for a range of values of θ_∞ at each lag up to $H = 8$, when $m = 8$ and $n = 622$ as is the case for the data obtained for individual B. We can clearly see that, with the exception of lag 8 where sampling from the spike is enforced in the prior, the probability of being assigned to the spike switches very quickly from one extreme to the other. This probability switch can be observed in all individuals, though the results for the other individuals are omitted. For the data for individual B, $\theta_\infty = 0.115$ gives $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) \approx 0.99$ at lag 4, but a slightly bigger value of $\theta_\infty = 0.195$ gives $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0) \approx 0.6 \times 10^{-6}$. When performing inference with $\theta_\infty = 0.195$ and keeping all other hyperparameters the same, we obtain the posterior mass function displayed in Figure 6.1(b). In this case, the posterior mode is $p_{\max} = 7$, with most of the mass supporting higher orders. This suggests that the posterior mass function is highly sensitive to the choice of the hyperparameter θ_∞. We note that this sensitivity to the choice of hyperparameter was not apparent when carrying out inference on the simulated data sets. Therefore, we conclude that whilst sensible posterior inferences can be obtained when there is no model misspecification (for example in the case of simulated data), inference on the model order is very sensitive to the choice of prior hyperparameters in analyses where the model does not perfectly describe the data. This suggests a lack of robustness to the kind of model misspecification that is inevitable in analyses of real time series. As such, we do not consider this method any further when analysing the EEG data, instead focusing on analysis using the multiplicative gamma process.



(a) (b)

Figure 6.1: Posterior mass function for the effective order p^* for the data in the delta band of individual B, using the CUSP prior with (a) $\theta_\infty = 0.115$ and (b) $\theta_\infty = 0.195$.

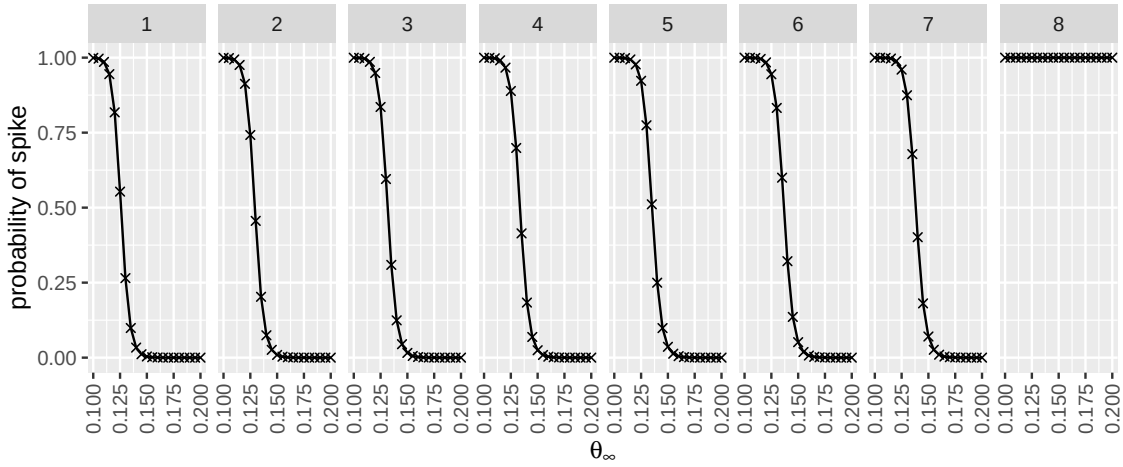


Figure 6.2: Value of $\Pr_{\theta_\infty}(z_s \leq s|\boldsymbol{y}_0)$ at lags $s = 1, \ldots, 8$ for different values of θ_∞ in the interval $[0.1, 0.2]$, when $m = 8$ and $n = 622$, as is the case for the data for individual B.

6.3 Multiplicative gamma process

Having discussed the difficulties observed when applying the cumulative shrinkage process prior to the EEG data, we now consider applying the multiplicative gamma process prior to the data, which gives much more promising results. When performing inference on the EEG data using the multiplicative gamma process prior, we used the same hyperparameter values as in the simulation studies in Section 5.3.6, taking $a_1 = 2.5$, $a_2 = 3$, $a = 6$ and $p_{\max} = 7$. For each individual and frequency band, we ran 4 chains, initialized at

different starting points. The chains were run for 17,000 iterations, discarding the first 1,000 as burn-in, and thinning to retain every fourth draw in order to reduce computational overheads. The usual graphical and numerical diagnostics gave no evidence of any lack of convergence and, after pooling the chains, the effective sample size was at least 2,023 for every model parameter. Using a CentOS Linux 7 (Core) 64bit operating system with an Intel Xeon E5-2699 v4 processor (2.2 GHz), the average run time across chains for the beta frequency band in individual A was 31.3 hours, with similar run times obtained for the other data sets.

6.3.1 Order determination

For each of the individuals, the posterior distributions for the effective order p^* for both the beta and delta series were calculated using the truncation criterion described in Section 5.3.4 with $\beta = 0.99$. The resulting posterior mass functions are shown in Figure 6.3. The results are quantitatively similar across all individuals, with a high level of posterior support on a mode of two in all cases, for both the delta and beta series. The similarity across individuals could possibly indicate similar generative processes for their ultradian rhythms. The posterior mass functions for p^* obtained here are consistent with the results obtained in our univariate analysis in Section 4.3, but with a reduction in variance. When modelling the data as multivariate, the support for higher order models appears to have been reduced, suggesting information from other variables at low lags supersedes information from the variable itself at larger lags.

6.3.2 Granger causality

Conditioning on the modal order of the process for both series in each patient, we can obtain samples from the posterior distributions of the autoregressive coefficient matrices. These posterior samples can be used to obtain further insight into biological rhythms in the brain. In multivariate time series analysis Granger causality investigates whether past observations of one variable can help predict future observations of another variable (Granger, 1969). In neuroscience literature, Granger causality has been used to investigate brain connectivity by considering whether past observations of brain activity in one region of the brain can help predict future observations in other regions of the brain (Manomaisaowapak *et al.*, 2022). The (i, j)-th element in the autoregressive matrix at lag-s, $\phi_{s,ij}$, governs the effect of the j-th variable at time $t - s$ on the i-th variable at time t. If $\phi_{s,ij}$ is non-zero we say that variable j Granger-causes variable i at lag s; this causal connection can be represented in a directed network, called a Granger causality plot, through an edge from vertex j to vertex i. In the context of our EEG data, an edge from region j to region i at lag s indicates that region j Granger-causes region i at lag

s. That is, an observation in region j at time $t-s$ can help to predict an observation in region i at time t. Conditional on the posterior modal order, $p^* = 2$, Figures 6.4–6.7 show the Granger causality plots at lags 1 and 2 in the beta and delta bands for individuals A to D. The coordinates of the vertices, representing the different brain regions, correspond to the x and y coordinates of the centre of the region using the Deskian-Killiany atlas. In these plots, an autoregressive coefficient is visualised as non-zero whenever zero lies outside the 50% equi-tailed Bayesian credible interval; the thickness of the edges representing non-zero coefficients are representations of the absolute value of the posterior mean. In some cases, such as in the beta band of individual B depicted in Figure 6.5(a), the model is reasonably confident that the model order is two, but not confident that any particular element of the autoregressive matrices are non-zero. A noticeable feature of these Granger causality plots is the higher number of connections in the delta band compared to the beta band. This was common across all individuals, and may indicate more localised processes underpinning the delta rhythms that interact with each other, whereas the beta rhythms in each region may be more driven by common processes.

Our prior is designed to facilitate order selection, not variable selection, and we used very broad, 50% credible intervals to identify potential zeros for graphical illustration in our Granger causality plots. Nevertheless, it is reassuring that the results obtained using the graphicalVAR package (Epskamp, 2024), which implements regularised-likelihood estimation with a lasso penalty on the individual autoregressive coefficients, gives broadly consistent inferences on the Granger causality network. Specifically, the non-zero entries identified by the graphicalVAR package are generally a subset of those we identified with our broad credible intervals. For example, for individual C in the beta band, the graphicalVAR package identifies only $\phi_{1,8,8}$ as non-zero, which is a subset of the non-zero coefficients we identify, and also the coefficient in our analysis with the posterior mean which is largest in absolute value. The Granger causality plots obtained using the graphicalVAR package are provided in Appendix E.

6.3.3 Decomposition into latent series

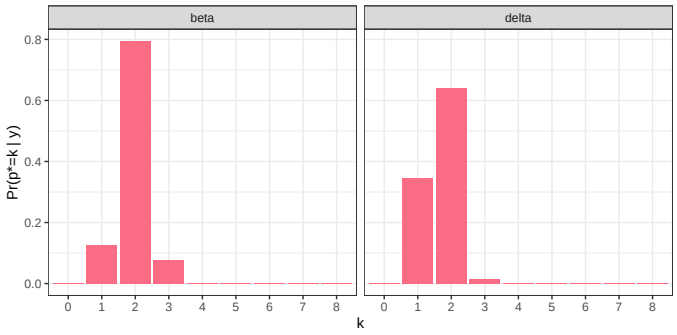
As discussed in Section 2.2.5, a $\text{VAR}_m(p)$ process can be decomposed into pm^2 latent series corresponding to the pm eigenvalues of the companion matrix obtained by representing the model as a $\text{VAR}_{mp}(1)$ process. When decomposing the EEG data the quasi-periodic series arising from the complex conjugate pairs of eigenvalues are of particular interest as they can capture the cyclical patterns that are key to understanding variation in brain activity. Inference on the period of these quasi-periodic series can give insights into the presence of periodic rhythms in the brain, such as the ultradian rhythms discussed in Chapter 3. The pairs of complex eigenvalues, $r_j e^{\pm i\omega_j}$, $j = 1, \ldots, c$, are not identifiable as the model remains unchanged under any permutation of their labelling. However, identification can

be achieved by applying an ordering constraint, for example, based on the modulus or the argument. Imposing the constraint $\omega_1 < \omega_2 < \cdots < \omega_c$, the quasi-periodic series z_{tij} are ordered by decreasing period $2\pi/\omega_j$.

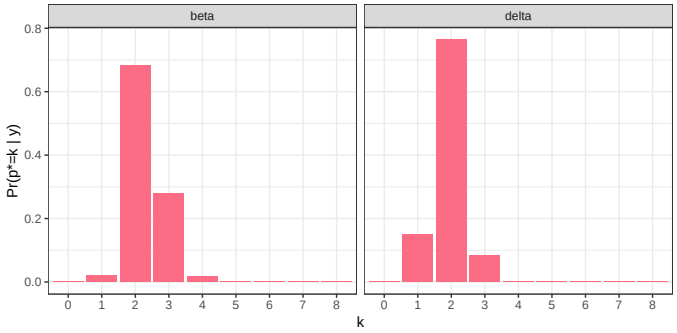
The posteriors for the periods and moduli of the first four quasi-periodic series are presented in Figures 6.8 to 6.11 for individuals A to D. We note that the z_{tij} with highest period also have highest modulus and might therefore be regarded as the dominating latent series. Across individuals, a common feature is that the posterior for the period of the dominating latent series in each band has its mean at around 20 minutes; for example, for individual A, the posterior means in the beta and delta bands are 19.619 and 26.921 minutes, with 95% equi-tailed Bayesian credible intervals of $(3.859, 80.882)$ and $(4.535, 111.871)$ minutes, respectively. The posterior means and credible intervals for the period of the dominating latent series in the other individuals are detailed in Table 6.1. This observation is consistent with ultradian rhythms of around 20 minutes which have previously been observed (Panagiotopoulou *et al.*, 2022). Whilst the 95% credible intervals are fairly wide, across all individuals the upper limit of the interval is below 2.5 hours. Ultradian rhythms have a period of less than 24 hours and so whilst there is a high level of uncertainty on the period, as there is still a high probability that the dominating latent series have a period of less than 2.5 hours, this is still an indication that ultradian rhythms are present. Additionally, across all individuals there appears to be strong evidence that there are multiple latent series with a period of greater than a minute. It is also noticeable that though there are some differences between the moduli of the series in the delta band compared to the beta band, there is very little difference between the corresponding periods. Again, this feature is replicated across all individuals. The similarity in the periods across the beta and delta bands indicate that there is a global cycle in the band power pattern, rather than a local cycle within a specific band. The similarities between subjects are striking, particularly the period of 20 minutes, and warrants future investigations into the possible biological mechanisms and potentially endogenous drivers (Goh *et al.*, 2019). Furthermore, it is interesting to note that when modelling the data as univariate processes in Section 4.3, we did not observe the same period length of 20 minutes, instead observing maximum periods of approximately 3.5 minutes, suggesting that modelling the data as univariate is inadequate for reliable detection of ultradian rhythms. However, as we only considered four subjects in this work, a larger study would be needed to confirm any biological interpretations, with a larger number of patients, longer recordings and accounting for the potential pathology present in these subjects.

Individual	Band	Posterior mean	Credible interval
A	beta	19.619	(3.859, 80.882)
	delta	26.921	(4.535, 111.871)
B	beta	15.234	(3.317, 63.667)
	delta	15.882	(2.970, 66.890)
C	beta	22.843	(4.031, 96.558)
	delta	29.453	(4.315, 130.936)
D	beta	21.068	(4.449, 86.117)
	delta	21.198	(4.069, 82.719)

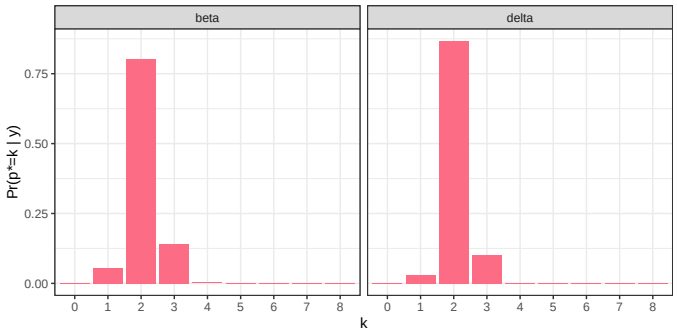
Table 6.1: Posterior means and credible intervals for the period of the dominating latent series in each band for individuals A to D.



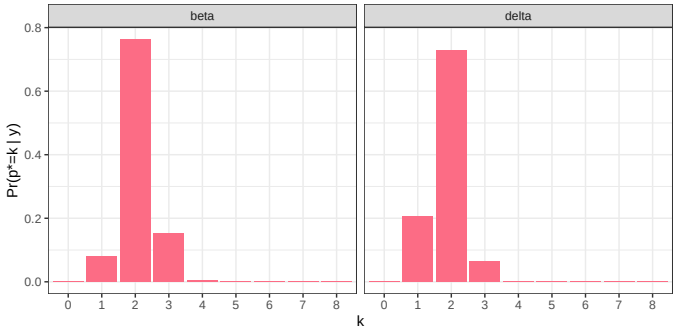
(a)



(b)

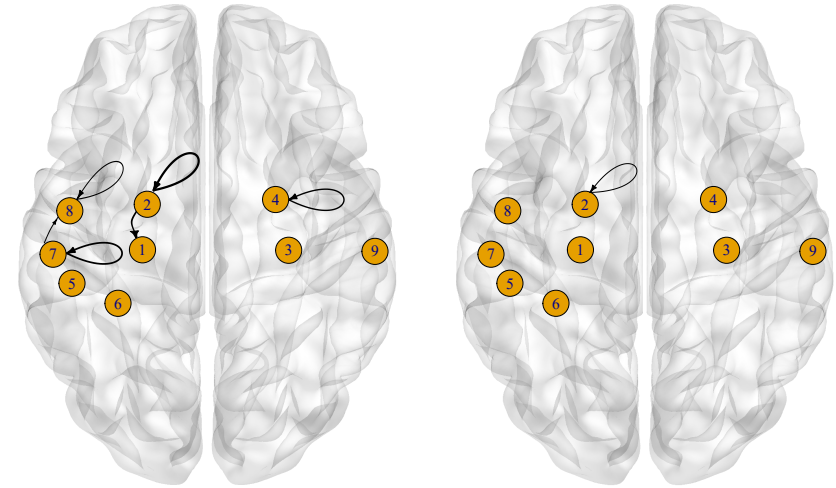


(c)

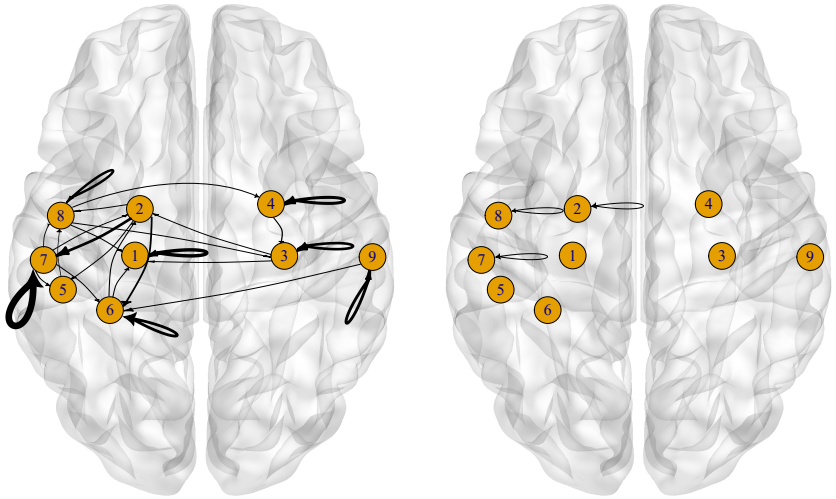


(d)

Figure 6.3: Posterior mass functions for the effective order p^* for the data from individuals (a) A, (b) B, (c) C and (d) D for both the beta (left) and delta (right) bands.

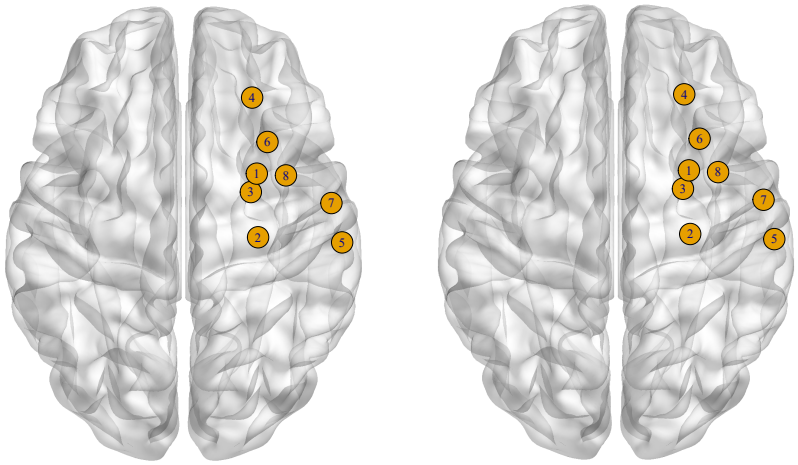


(a)

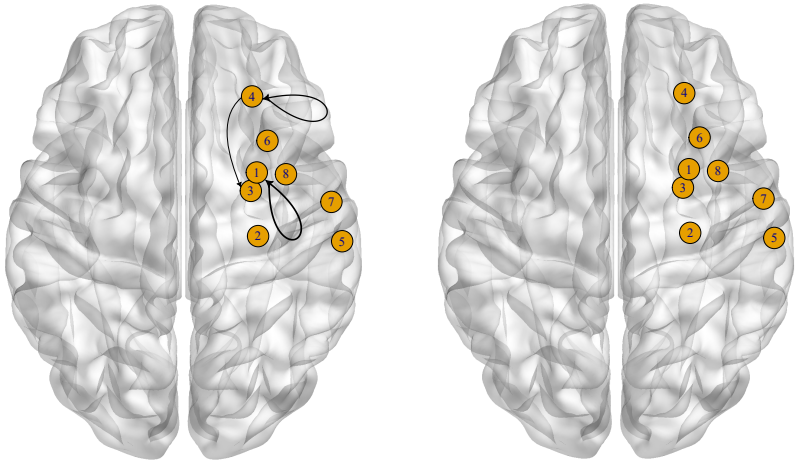


(b)

Figure 6.4: Granger causality plots of the posterior mean of the autoregressive coefficient matrices overlaid on glass brains showing the locations of the regions, for the VAR process of individual A in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

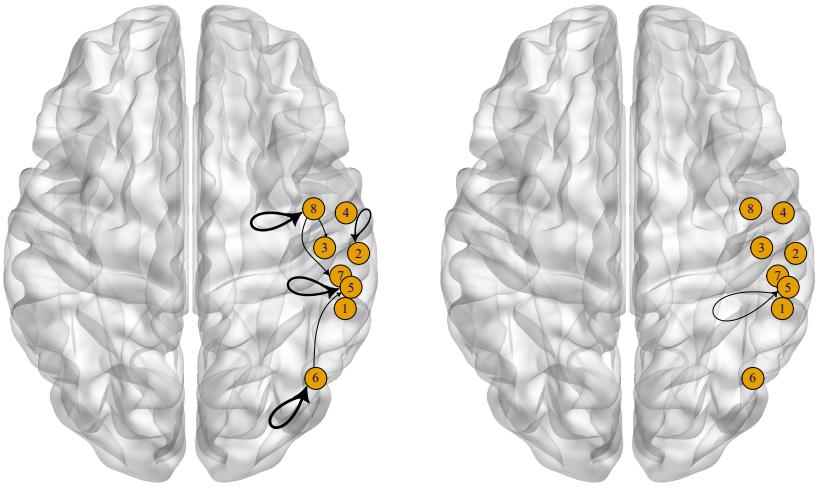


(a)

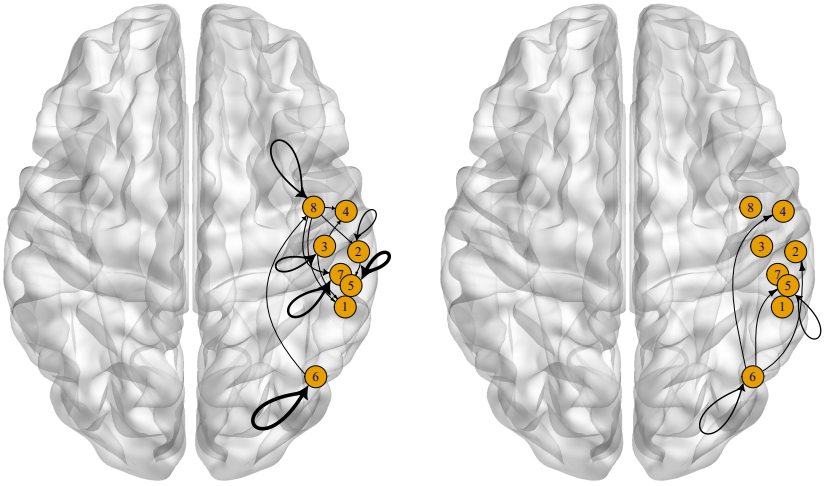


(b)

Figure 6.5: Granger causality plots of the posterior mean of the autoregressive coefficient matrices overlaid on glass brains showing the locations of the regions, for the VAR process of individual B in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

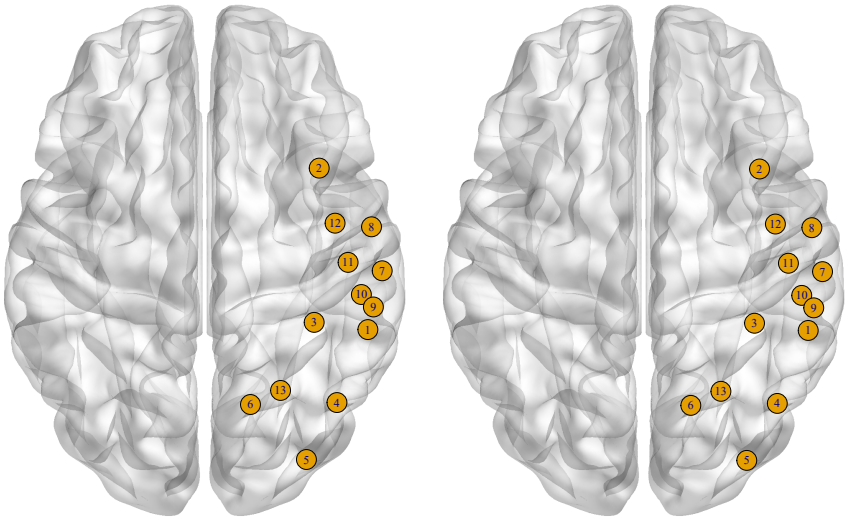


(a)

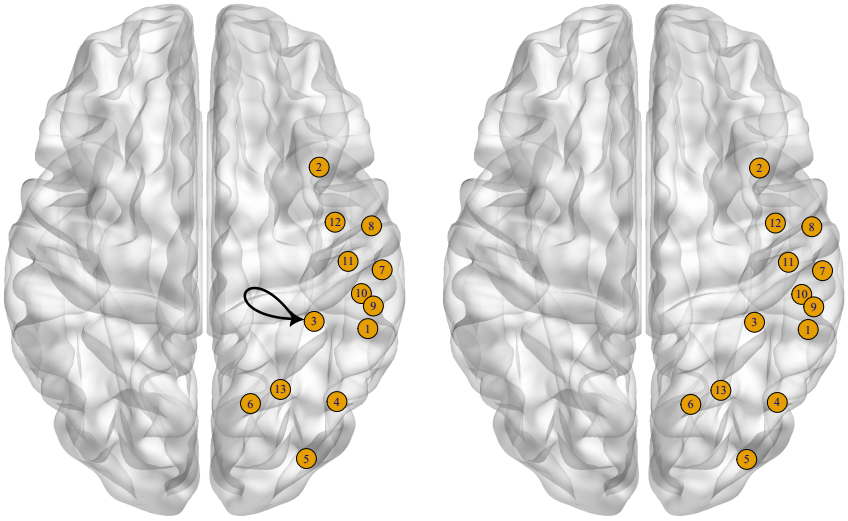


(b)

Figure 6.6: Granger causality plots of the posterior mean of the autoregressive coefficient matrices overlaid on glass brains showing the locations of the regions, for the VAR process of individual C in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

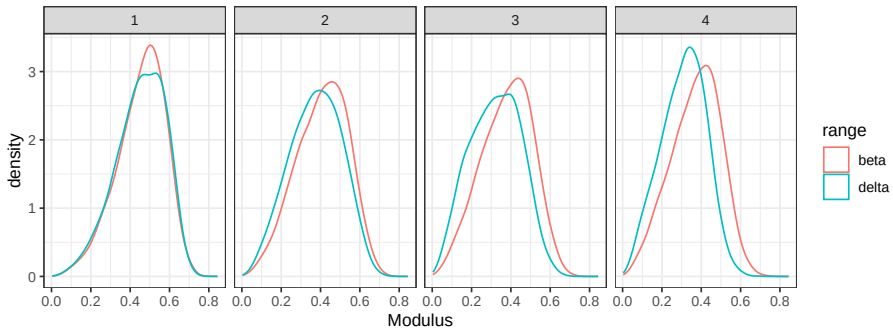


(a)

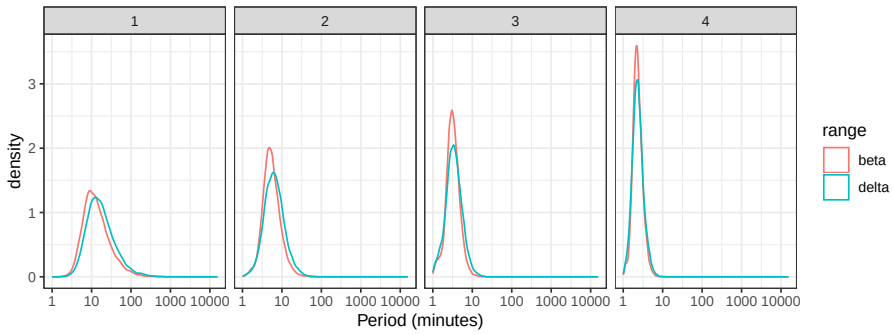


(b)

Figure 6.7: Granger causality plots of the posterior mean of the autoregressive coefficient matrices overlaid on glass brains showing the locations of the regions, for the VAR process of individual D in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

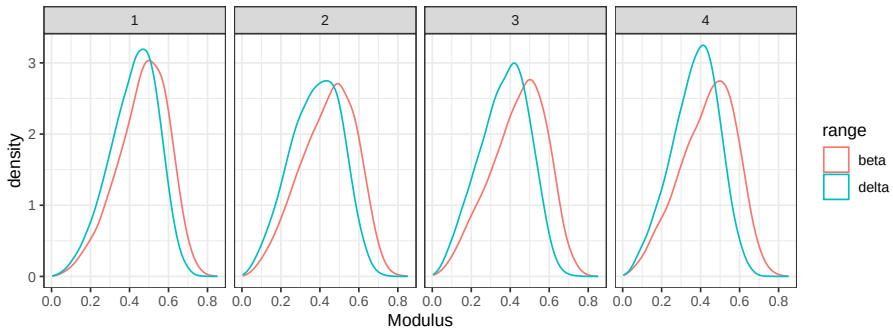


(a)

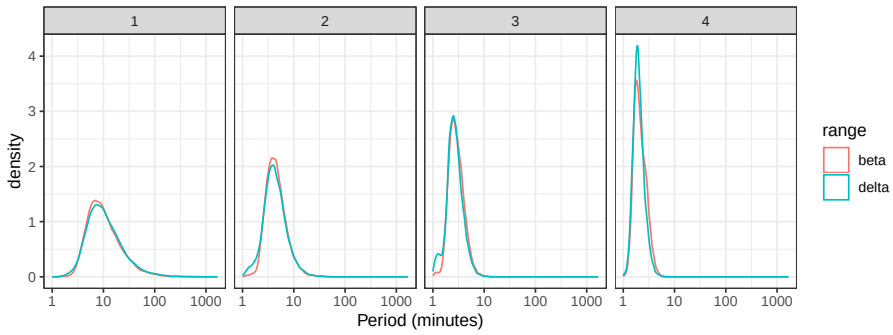


(b)

Figure 6.8: Posterior densities for (a) the moduli and (b) the period of the first four quasi-periodic series for individual A.

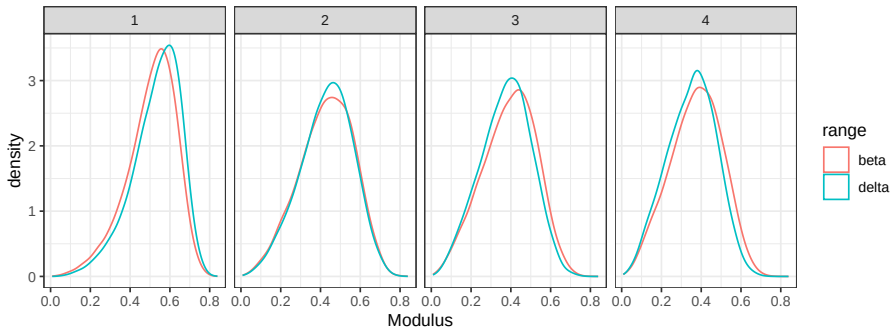


(a)

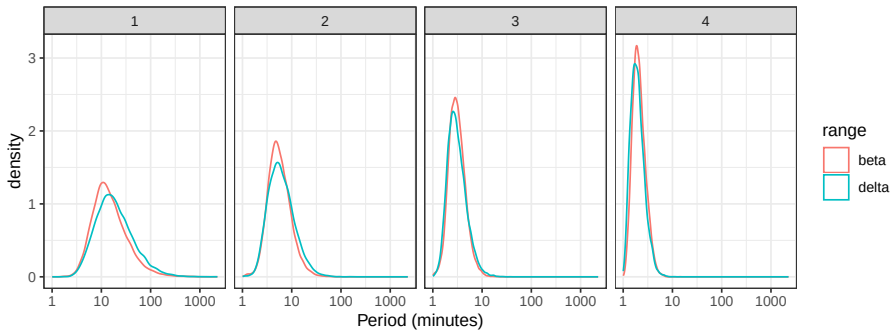


(b)

Figure 6.9: Posterior densities for (a) the moduli and (b) the period of the first four quasi-periodic series for individual B.

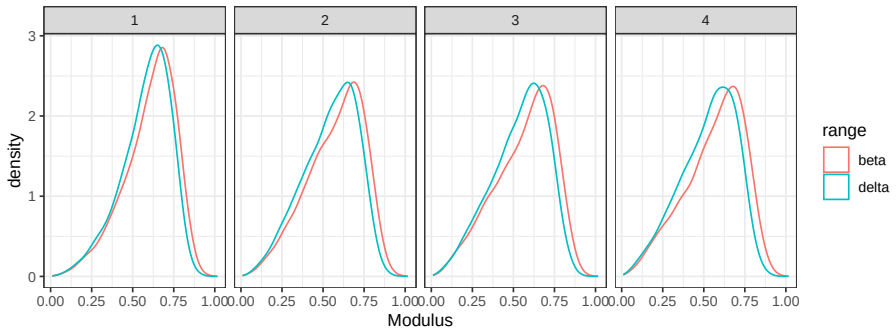


(a)

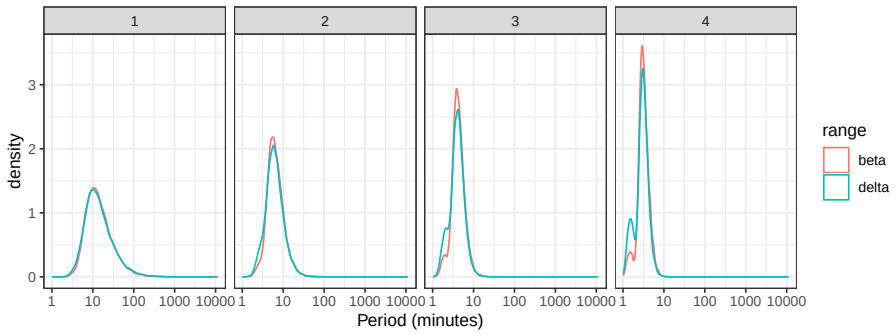


(b)

Figure 6.10: Posterior densities for (a) the moduli and (b) the period of the first four quasi-periodic series for individual C.



(a)



(b)

Figure 6.11: Posterior densities for (a) the moduli and (b) the period of the first four quasi-periodic series for individual D.

Chapter 7

Conclusions and further work

In this chapter we summarise the work discussed in this thesis, highlight its contributions to the literature and discuss further work.

7.1 Conclusions

In Chapter 1 we introduced our motivations and aims before discussing the contributions made, on which we elaborate in the next section.

In Chapter 2 we provided background information on Bayesian analysis and time series modelling. We discussed the benefits of Markov chain Monte Carlo methods for sampling from a posterior distribution and introduced relevant algorithms including the Metropolis-Hastings and Hamiltonian Monte Carlo samplers. Vector autoregressions were also introduced, and the stationary condition defined. Finally, we discussed the decomposition of a stationary vector autoregression into latent processes.

In Chapter 3 we discussed the EEG data that was used to demonstrate a use case for the methods developed in this thesis. We carried out some exploratory data analysis before fitting basic ARMA models to the data obtained from individual regions in the brain, using traditional time series model fitting techniques.

Chapter 4 began with a discussion of methods for enforcing stationarity when carrying out Bayesian inference of univariate autoregressions with a known model order. Following this we discussed existing methods for Bayesian inference on the model order of stationary univariate autoregressions, which relied on reparameterising the model in terms of either the partial autocorrelations or the roots of the characteristic equation. We provided a comparison of methods through the use of a simulation study and applied our preferred method to the EEG data obtained from individual regions of the brain.

In Chapter 5 we provided the first methodology for quantifying uncertainty in model order for the full class of stationary vector autoregressions, through a hierarchical Bayesian

model with accompanying model-fitting methodology. The methodology was based on an unconstrained reparameterisation of the stationary model in terms of a set of transformed partial autocorrelation matrices (Heaps, 2023) whose properties were exploited in the design of the prior. In particular, we capitalised on the nested structure of the new parameterisation by constructing an overparameterised hierarchical model which shrinks unnecessary, high-order terms to zero; by identifying the lag beyond which the partial autocorrelation parameters become effectively equal to zero, we could then learn the order of the process. Further, using the relationship between the spectral norm of a partial autocorrelation matrix and its unconstrained counterpart, the prior was chosen to increasingly shrink the partial autocorrelation matrices at higher lags towards zero. Two choices of increasing shrinkage priors for the unconstrained matrices were discussed, the cumulative shrinkage and multiplicative gamma processes. Additionally, associated methods for determining whether a partial autocorrelation matrix was effectively zero were presented, hence allowing determination of the model order. An efficient Hamiltonian Monte Carlo sampler for computational inference was implemented through Stan, with accompanying code to allow easy dissemination of the research. We applied our methodology to a series of simulation experiments in which data sets of various lengths n were sampled from various stationary $\text{VAR}_m(p)$ models. For all values of m, p and n considered, the posterior for the effective order of the process was highly concentrated around the known model order for both the cumulative shrinkage and multiplicative gamma processes.

We then applied our methodology to EEG data from recordings at various locations in the brain in Chapter 6. When using the cumulative shrinkage process we found that the prior was highly sensitive to the choice of hyperparameters when model misspecification was present, but we found the multiplicative gamma process to be robust when applied to real data. Conditioning on the posterior modal order of these EEG processes, obtained through the multiplicative gamma process, allowed physiological insight in a number of directions. By constructing Granger causality plots, we were able to highlight relationships between activity in different regions of the brain. Similarly, by constructing the latent decomposition of the series, we were able to identify underlying quasi-periodic structure. In particular, we found that the dominant latent component had a period that was around 20 minutes across all individuals in both the beta and the delta bands.

7.2 Contributions of the thesis

In this section we discuss the contributions made throughout this thesis.

In Chapter 2 we provided additional clarification on how the behaviour of the latent processes relating to any complex eigenvalues in the decomposition of a vector autoregressive process have similar behaviour to an AR(2) process, expanding on the details provided

in Prado (1998) which is, to our knowledge, the only other presentation of the derivation of this theory in the literature.

In Chapter 4 we provided a comparison of existing methods for Bayesian inference on the order of stationary univariate autoregressions, through a simulation study to compare computation speeds and mixing. This allowed us to make recommendations as to which method performs best. Additionally, we developed a new procedure for determining the order of stationary univariate autoregressions by adapting methods already in use for variable selection in regression. This procedure consisted of a new representation of a spike-and-slab prior that we found to be computationally faster than the existing representation of the spike-and-slab prior found in Barnett *et al.* (1996).

In Chapter 5 we presented the main contribution of the thesis by providing a solution to the problem of learning the order of the general class of stationary vector autoregressions in the Bayesian framework. Two increasing shrinkage priors were used to increasingly shrink the partial autocorrelations at higher lags towards zero. We defined the effective order of the model to be the highest lag for which the partial autocorrelation matrix is determined to be non-zero, allowing us to use the samples obtained from the increasing shrinkage priors to determine the model order. We first discussed the cumulative shrinkage process (Legramanti *et al.*, 2020) in the context of determining the order of stationary vector autoregressions. For model order determination, we provided a method for calculating a Rao-Blackwellised estimate of the posterior mass function of the effective order of the process. Additionally, we presented a method for choosing θ_∞, which determines the location of the spike in the prior, using classical time series theory. We then considered the multiplicative gamma process (Bhattacharya & Dunson, 2011) for order determination in stationary vector autoregressions. We suggested a suitable truncation criterion to determine whether the partial autocorrelation matrices were effectively zero, using classical time series theory on the estimators of the partial autocorrelation function. This allowed us to determine the effective order of the process. We explored the success of each prior in determining the true order of a stationary vector autoregression using a simulation experiment in which the true order was known.

In Chapter 6 we contributed to the literature on biological processes in the brain by applying our methods to EEG data. We found that the cumulative shrinkage process was very sensitive to the choice of hyperparameters in the presence of model misspecification, in particular regarding the choice of the hyperparameter θ_∞. However, we obtained useful results using the more robust multiplicative gamma process. We used this prior to investigate relationships between different regions of the brain and explore its ultradian rhythms.

7.3 Further work

There are a number of different directions that could be taken to extend the work in this thesis to cover a wider variety of modelling situations.

7.3.1 Dynamic models

An obvious limitation in the application to EEG data was the necessity to pick out contiguous segments of data where stationarity was a plausible assumption. For example, in Section 3.2 we fit a simple hidden Markov model to the data from the delta band in individual A. Both the hidden Markov model and a visual inspection of the data suggested that there were at least two regimes within the data, with different mean and variance. However, as remarked in Section 2.2.3, stationary autoregressions often serve as building blocks in the construction of more complex models. Motivated by applications involving EEG data where subjects make (reversible) transitions between states of wakefulness and sleep, or states of normal brain activity and seizure, an extension to the work in this thesis would be to explore a hidden Markov model in which a (locally) stationary vector autoregression describes the within-state dynamics. Such a model would be ideally suited to a wide variety of time-series where there are occasional reversible step-changes in a process which otherwise appears to be mean reverting.

In a hidden Markov model the observed data is assumed to depend on an unobserved process which moves between hidden states. The number of states is fixed and therefore it is not possible to model new, unexpected changes in the behaviour of the process through this constrained set of states which are based on expected behaviour. Furthermore, the probabilities of moving between different states are fixed. A more flexible modelling approach would be to model a multivariate time series as a mixture of VARs with the Dirichlet process used as a mixing measure (Kalli & Griffin, 2018). Each regime in the data could then be modelled by a different mixture component, with a potentially infinite number of components, with stationary VAR processes of potentially unknown model order used to model the transition densities in each component. Following the work in Kalli & Griffin (2018), the mixing weights determined by the Dirichlet process would also be more flexible than the transition probabilities in a hidden Markov model, as they would depend on the previous lags.

If the data do not display clear-cut regime changes, a more appropriate locally stationary state space model would employ a continuous state vector that evolved at every time step. This could take the form of a locally stationary time-varying vector autoregression. An m-variate time-varying vector autoregression of order p, denoted TV-VAR$_m(p)$, is defined as

$$\boldsymbol{y}_t = \phi_{t1}\boldsymbol{y}_{t-1} + \ldots + \phi_{tp}\boldsymbol{y}_{t-p} + \boldsymbol{\varepsilon}_t,$$

where $\boldsymbol{\varepsilon}_t \sim N_m(\mathbf{0}, \Sigma_t)$ and the parameters $\phi_{t1}, \ldots, \phi_{tp}$ and Σ_t change at every time step. Whilst a time-varying vector autoregression like this may be globally non-stationary, for slowly changing parameters, sections of the data could effectively be modelled as locally stationary, where the local constraint may have the advantage of reducing the variance of predictions from the model. In this case, to enforce local stationarity, the parameters $\phi_{t1}, \ldots, \phi_{tp}$ at every time t would be restricted to lie in the stationary region. This could be achieved by mapping $(\phi_{t1}, \ldots, \phi_{tp})$ to a set of transformed partial autocorrelation parameters $(A_{t1}, \ldots, A_{tp})$. These would then be assumed to evolve according to a second autoregressive process or a random walk; for example, for each $i = 1, \ldots, p$ we could take

$$\mathrm{vec}(A_{ti}) = \mathrm{vec}(A_{t-1,i}) + \boldsymbol{\eta}_t, \qquad \boldsymbol{\eta}_t \sim N_{m^2}(\mathbf{0}, \Omega).$$

Similar ideas, though without the reparameterisation have been considered in Francq & Zakoian (2001) and Stelzer (2009).

7.3.2 Learning the orders of VARMA processes

Although the work in this thesis focuses on vector autoregressions, the methods described can be extended to the general class of vector autoregressive moving average (VARMA) models. A vector autoregressive moving average model of order (p, q) has the form

$$\boldsymbol{y}_t = \phi_1\boldsymbol{y}_{t-1} + \ldots + \phi_p\boldsymbol{y}_{t-p} + \boldsymbol{\varepsilon}_t + \psi_1\boldsymbol{\varepsilon}_{t-1} + \ldots + \psi_q\boldsymbol{\varepsilon}_{t-q}$$

where $\boldsymbol{\varepsilon}_t \sim N(\mathbf{0}, \Sigma)$ for $t = q+1, \ldots, n$, the parameters $\psi_i \in M_{m\times m}(\mathbb{R})$, $i = 1, \ldots, q$, are $m \times m$ moving average coefficient matrices and the parameters $\phi_1, \ldots, \phi_p$ are the autoregressive coefficients as discussed throughout this thesis. In backshift notation this can be written as

$$\phi(B)\boldsymbol{y}_t = \psi(B)\boldsymbol{\varepsilon}_t$$

where $\psi(u) = 1 + \phi_1 u + \ldots + \phi_q u^q$, $u \in \mathbb{C}$, is the characteristic moving average polynomial. The methods discussed throughout this thesis can be used to enforce stationarity in a VARMA model and learn the model order p of the autoregressive component. The VARMA process is invertible if and only if all roots of the equation

$$\det\{\psi(u)\} = 0$$

lie outside the unit circle. The subset of matrices $\psi_1, \ldots, \psi_q$ which lie within the invertible region can be denoted $\mathcal{C}_{q,m}$. To constrain the parameters to the invertible region, the

algorithm discussed in Section 5.1 to map from the autoregressive coefficient matrices and variance to the partial autocorrelation matrices $P_1, \ldots, P_p$ can also be used to map from the moving average coefficient matrices and variance to a set of matrices $Q_1, \ldots, Q_q$. Then, the second mapping described in Section 5.1 can be used to map these matrices to a set of unconstrained matrices. To determine the model order q, the multiplicative gamma process can be used to increasingly shrink the unconstrained matrices at higher lags towards zero, in a direct equivalent to the method used for the unconstrained partial autocorrelation matrices. A truncation criterion could then be used to determine which of the matrices are effectively zero. Unfortunately, the matrices $Q_1, \ldots, Q_q$ do not have a clear interpretation, and so making a principled choice of truncation criterion for the reparameterised moving average coefficients would not be straightforward.

7.3.3 Sparsity

Throughout this thesis we focused on the benefits of choosing a prior which increasingly shrinks the partial autocorrelation matrices at higher lags towards zero, to allow inference on the model order. An additional use for shrinkage priors is to encourage sparsity in a model. The number of parameters in a $\text{VAR}_m(p)$ model is $O(m^2)$ which can become very large as m grows. This can make inference on the model parameters difficult, particularly if there are only a small number of observations in the data. The resulting posteriors for Φ and Σ may then be imprecise if there is not enough data to be informative, resulting in posterior densities with a large variance. As well as increasingly shrinking the transformed partial autocorrelation matrices at higher lags towards zero, as a global-local shrinkage prior the multiplicative gamma process encourages shrinkage of the individual elements of the transformed partial autocorrelation matrices towards zero. Shrinking individual elements of the transformed partial autocorrelation matrices can be helpful in reducing the overall number of parameters in the model. Ideally we would encourage sparsity in the elements of the autoregressive matrices Φ as this has a direct interpretation in terms of Granger causality, see Section 6.3.2. However, to enforce stationarity we place a prior on the unconstrained matrices $A_1, \ldots, A_{p_{\max}}$, rather than the autoregressive coefficient matrices. Whilst we can encourage sparsity in these unconstrained matrices, a zero in the (i, j)-th element of A_s does not map to a zero in the (i, j)-th position of P_s, which then in turn does not map to a zero in the (i, j)-th position of ϕ_s. Unfortunately there is no clear interpretation of a sparse A_s matrix in the same way there would be for a sparse ϕ_s matrix, and so it would be better if we could encourage sparsity in the ϕ_s matrices, rather than the unconstrained matrices. However, the problem of encouraging sparsity in the ϕ_s matrices whilst also enforcing stationarity would be very challenging.

7.3.4 EEG application

Another direction that could be taken to extend this work in the context of the EEG data is to place a greater focus on the spatial relationships between the different regions of the brain. In our work, we did not directly exploit the possibility that regions in the brain that are closer to each other may be more highly correlated than those that are further apart. Throughout this thesis we have placed a prior on the innovation variance Σ. Alternatively, a prior could be placed on the stationary variance matrix Γ_0 and this prior could be chosen to represent the belief that brain regions that are closer together are likely to be more highly correlated than those that are further apart. For example, we could construct a prior where the mean is proportional to an exponential correlation matrix with (i, j)-th entry $\exp(-\theta d_{ij})$, where d_{ij} is the distance between regions i and j.

Appendix A

Derivations

A.1 Sketch proof of stationarity condition

As discussed in Section 2.2.3, a vector autoregression is stationary if and only if the roots of the equation $\det\{\phi(u)\} = 0$ lie outside the unit circle. This section provides a sketch proof that this condition holds, with further details to be found in Luetkepohl (2005).

We can write the $\text{VAR}_m(p)$ process $\{\boldsymbol{y}_t\}$ as a $\text{VAR}_{mp}(1)$ process $\{\boldsymbol{Y}_t\}$ where

$$\boldsymbol{Y}_t = \phi \boldsymbol{Y}_{t-1} + \boldsymbol{u}_t$$

with

$$\boldsymbol{Y}_t = \begin{pmatrix} \boldsymbol{y}_t \\ \boldsymbol{y}_{t-1} \\ \vdots \\ \boldsymbol{y}_{t-p+1} \end{pmatrix}, \qquad \phi = \begin{pmatrix} \phi_1 & \phi_2 & \dots & \phi_p \\ I_m & {}_m0 & \dots & 0_m \\ \vdots & \vdots & \ddots & \vdots \\ 0_m & 0_m & \dots & I_m \end{pmatrix} \qquad \text{and} \qquad \boldsymbol{u}_t = \begin{pmatrix} \boldsymbol{\varepsilon}_t \\ \boldsymbol{0} \\ \vdots \\ \boldsymbol{0} \end{pmatrix}$$

where $E(\boldsymbol{u}_t) = \boldsymbol{0}$ and $\text{Var}(\boldsymbol{u}_t) = \Sigma'$. By substituting in the form of $\boldsymbol{Y}_{t-1}, \boldsymbol{Y}_{t-2}, \dots$, we

can write the $\text{VAR}_{mp}(1)$ process as

$$\begin{aligned}
\boldsymbol{Y}_t &= \phi \boldsymbol{Y}_{t-1} + \boldsymbol{u}_t \\
&= \phi(\phi \boldsymbol{Y}_{t-2} + \boldsymbol{u}_{t-1}) + \boldsymbol{u}_t \\
&= \phi^2 \boldsymbol{Y}_{t-2} + \phi \boldsymbol{u}_{t-1} + \boldsymbol{u}_t \\
&= \phi^2(\phi \boldsymbol{Y}_{t-3} + \boldsymbol{u}_{t-2}) + \phi \boldsymbol{u}_{t-1} + \boldsymbol{u}_t \\
&= \phi^3 \boldsymbol{Y}_{t-3} + \phi^2 \boldsymbol{u}_{t-2} + \phi \boldsymbol{u}_{t-1} + \boldsymbol{u}_t \\
&\vdots \\
\boldsymbol{Y}_t &= \phi^j \boldsymbol{Y}_{t-j} + \sum_{i=0}^{j-1} \phi^i \boldsymbol{u}_{t-i}.
\end{aligned}$$

Assume that the process was started in the infinite past and that all the eigenvalues of ϕ have modulus less than one. Then, as $j \to \infty$ the $\phi^j \to 0_m$ and so the term $\phi^j \boldsymbol{Y}_{t-j}$ can be ignored in the limit as $j \to \infty$. Furthermore, the sequence ϕ^i, $i = 0, 1, \ldots$, is absolutely summable and so the infinite sum

$$\sum_{i=0}^{\infty} \phi^i \boldsymbol{u}_{t-i}$$

exists in mean square (see Luetkepohl (2005) for further details). Therefore, assuming that all the eigenvalues of ϕ have modulus less than one and that the process was started in the infinite past, the process $\boldsymbol{Y}_t$ can be written as

$$\boldsymbol{Y}_t = \sum_{i=0}^{\infty} \phi^i \boldsymbol{u}_{t-i}$$

for $t = 0, \pm 1, \pm 2, \ldots$. The mean of this process is

$$\text{E}(\boldsymbol{Y}_t) = \sum_{i=0}^{\infty} \phi^i \text{E}()\boldsymbol{u}_{t-i}) = \boldsymbol{0}$$

for all t. The autocovariances of the $\text{VAR}_{mp}(1)$ process can be shown to be

$$\Gamma_s' = \sum_{i=0}^{\infty} \phi^{s+i} \Sigma' \phi^{i^{\text{T}}}$$

for all t (see Luetkepohl (2005) for further details). As the mean and autocovariances of the process are constant over time, this process is stationary. If we had not assumed that the eigenvalues of ϕ all had modulus of less than one then this would not hold as the term $\phi^j \boldsymbol{Y}_{t-j}$ could not have been ignored, leading to the mean and autocovariances depending on time t. Therefore, stationarity only holds if all eigenvalues of ϕ are less than one. This

condition is equivalent to saying stationarity holds if and only if

$$\det(I_m - \phi u) \neq 0 \quad \text{for } |u| \leq 1.$$

Equivalently, in terms of the original $\text{VAR}_m(p)$ process, this can be written as

$$\det(I_m - \phi_1 u - \ldots - \phi_p u^p) \neq 0 \quad \text{for } |u| \leq 1.$$

A.2 Reparameterising a scalar dynamic linear model

In Section 2.2.5 we followed Prado (1998) in reparameterising the scalar DLM found in Equations (2.4) and (2.5) to obtain the representation in Equations (2.6) and (2.7). The following derivation verifies that this reparameterisation holds.

The eigendecomposition of the state evolution matrix is such that $G = BAB^{-1}$ where A is a diagonal matrix containing the eigenvalues of G and B is the matrix containing the corresponding eigenvectors as columns. Additionally, F_i is defined to be the i-th column of the observation matrix F. Prado (1998) define $\boldsymbol{\gamma}_{ti} = H_i\boldsymbol{\theta}_t$ and $\boldsymbol{\delta}_{ti} = H_i\boldsymbol{\omega}_t$ where $H_i = \text{diag}(B^{\text{T}}F_i)B^{-1}$. Multiplying both sides of Equation (2.5) by H_i gives

$$\begin{aligned}
H_i\boldsymbol{\theta}_t &= H_i(G\boldsymbol{\theta}_{t-1} + \boldsymbol{\omega}_t) \\
\boldsymbol{\gamma}_{ti} &= H_iG\boldsymbol{\theta}_{t-1} + H_i\boldsymbol{\omega}_t \\
\boldsymbol{\gamma}_{ti} &= H_iBAB^{-1}\boldsymbol{\theta}_{t-1} + \boldsymbol{\delta}_{ti} \\
\boldsymbol{\gamma}_{ti} &= \text{diag}(B^{\text{T}}F_i)B^{-1}BAB^{-1}\boldsymbol{\theta}_{t-1} + \boldsymbol{\delta}_{ti} \\
\boldsymbol{\gamma}_{ti} &= \text{diag}(B^{\text{T}}F_i)AB^{-1}\boldsymbol{\theta}_{t-1} + \boldsymbol{\delta}_{ti} \\
\boldsymbol{\gamma}_{ti} &= A\text{diag}(B^{\text{T}}F_i)B^{-1}\boldsymbol{\theta}_{t-1} + \boldsymbol{\delta}_{ti} \\
\boldsymbol{\gamma}_{ti} &= AH_i\boldsymbol{\theta}_{t-1} + \boldsymbol{\delta}_{ti} \\
\boldsymbol{\gamma}_{ti} &= A\boldsymbol{\gamma}_{t-1,i} + \boldsymbol{\delta}_{ti}
\end{aligned}$$

giving the reparameterisation in Equation (2.7). Reparameterising Equation (2.4) in terms of $\boldsymbol{\gamma}_{ti}$ we have

$$
\begin{aligned}
y_{ti} &= F_i^{\mathrm{T}}\boldsymbol{\theta}_t \\
y_{ti} &= F_i^{\mathrm{T}}H_i^{-1}\boldsymbol{\gamma}_{ti} \\
y_{ti} &= F_i^{\mathrm{T}}[\mathrm{diag}(B^{\mathrm{T}}F_i)B^{-1}]^{-1}\boldsymbol{\gamma}_{ti} \\
y_{ti} &= F_i^{\mathrm{T}}B[\mathrm{diag}(B^{\mathrm{T}}F_i)]^{-1}\boldsymbol{\gamma}_{ti} \\
y_{ti} &= F_i^{\mathrm{T}}B\{\mathrm{diag}[(B^{\mathrm{T}}F_i)^{\mathrm{T}}]\}^{-1}\boldsymbol{\gamma}_{ti} \\
y_{ti} &= [(F_i^{\mathrm{T}}B)_1, (F_i^{\mathrm{T}}B)_2, \ldots, (F_i^{\mathrm{T}}B)_{mp}]
\begin{bmatrix}
\frac{1}{(F_i^{\mathrm{T}}B)_1} & 0 & \ldots & 0 \\
0 & \frac{1}{(F_i^{\mathrm{T}}B)_2} & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & \frac{1}{(F_i^{\mathrm{T}}B)_{mp}}
\end{bmatrix}\boldsymbol{\gamma}_{ti} \\
y_{ti} &= \mathbf{1}^{\mathrm{T}}\boldsymbol{\gamma}_{ti}
\end{aligned}
$$

which is the parameterisation given in Equation (2.6). In the derivation, $(F_i^{\mathrm{T}}B)_j$ denotes the j-th element of the vector $F_i^{\mathrm{T}}B$.

A.3 Univariate mapping between autoregressive parameters and partial autocorrelations

A.3.1 Initial definitions and results

For each $s = 1, \ldots, p$ define the following series of autoregressions:

$$y_{t+1} = \sum_{i=1}^{s} \phi_{si} y_{t-i+1} + \epsilon_{s,t+1}$$

where ϕ_{si} is the coefficient of the ith term y_{t-i+1} in the conditional expectation $\mathrm{E}(y_{t+1}|y_t, \ldots, y_{t-s+1})$ and $\sigma_s^2 = \mathrm{var}(\epsilon_{s,t+1}) = \mathrm{var}(y_{t+1}|y_t, \ldots, y_{t-s+1})$. This results in $\phi_{pi} = \phi_i$, $i = 1, \ldots, p$, and $\sigma_p^2 = \sigma^2$. Note that in the proof of the vector autoregressive mapping, provided in the Supplementary Materials of Heaps (2023), we also need to consider the backward time series, but as univariate stationary autoregressions are time reversible the results obtained from the forward and backward time series are the same, which simplifies our derivations.

For $s = 1, \ldots, p$, let G_s and g_s be an $s \times s$ matrix and an $s \times 1$ vector respectively such that

$$G_s = \begin{pmatrix} \gamma_0 & \gamma_1 & \cdots & \gamma_{s-1} \\ \gamma_1 & \gamma_0 & \cdots & \gamma_{s-2} \\ \vdots & \vdots & \ddots & \vdots \\ \gamma_{s-1} & \gamma_{s-2} & \cdots & \gamma_0 \end{pmatrix} \qquad \text{and} \qquad g_s = \begin{pmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_s \end{pmatrix}$$

where $\gamma_i = \text{Cov}(y_t, y_{t+i})$ for $i = 0, \ldots, s-1$ are the autocovariances. Additionally let $\tilde{g}_s$ be the reversed vector such that

$$\tilde{g}_s = Qg_s = \begin{pmatrix} \gamma_s \\ \gamma_{s-1} \\ \vdots \\ \gamma_1 \end{pmatrix}$$

where Q is an involutory and symmetric $s \times s$ matrix defined as

$$Q = \begin{pmatrix} 0 & \cdots & 0 & 1 \\ 0 & \cdots & 1 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 1 & \cdots & 0 & 0 \end{pmatrix}.$$

Let $y_{i:j} = (y_i, \ldots, y_j)^T$, $\boldsymbol{\phi}_s = (\phi_{s1}, \ldots, \phi_{ss})^T$ and $\boldsymbol{\phi}_{s,-s} = (\phi_{s1}, \ldots, \phi_{s,s-1})^T$. Using standard normal theory the conditional mean of y_{t+1} given its s predecessors $y_{t:t-s+1}$ is

$$\begin{aligned} \text{E}(y_{t+1}|y_{t:(t-s+1)}) &= \text{E}(y_{t+1}) + \text{Cov}(y_{t+1}, y_{t:(t-s+1)})\text{Var}(y_{t:(t-s+1)})^{-1}\{y_{t:(t-s+1)} - E(y_{t:(t-s+1)})\} \\ &= 0 + g_s^T G_s^{-1}(y_{t:(t-s+1)} - 0) \\ &= g_s^T G_s^{-1} y_{t:(t-s+1)} \\ &= \boldsymbol{\phi}_s^T y_{t:(t-s+1)} \end{aligned}$$

by definition for $s = 1, \ldots, p$. This leads to

$$\boldsymbol{\phi}_s = G_s^{-1} g_s \iff g_s = G_s \boldsymbol{\phi}_s \qquad s = 1, \ldots, p. \tag{A.1}$$

The conditional variance is

$$\begin{aligned} \text{Var}(y_{t+1}|y_{t:(t-s+1)}) &= \text{Var}(y_{t+1}) - \text{Cov}(y_{t+1}, y_{t:(t-s+1)})\text{Var}(y_{t:(t-s+1)})^{-1}\text{Cov}(y_{t:(t-s+1)}, y_{t+1}) \\ &= \gamma_0 - g_s^T G_s^{-1} g_s \\ &= \gamma_0 - g_s^T \boldsymbol{\phi}_s \\ &= \sigma_s^2 \end{aligned}$$

by definition for $s = 1, \ldots, p$. By taking transposes this gives

$$\sigma_s^2 = \gamma_0 - \boldsymbol{\phi}_s^T g_s. \tag{A.2}$$

Using the properties of Q we will also derive the following result:

$$Q\boldsymbol{\phi}_s = QG_s^{-1}g_s = Q^{-1}G_s^{-1}Q^{-1}Qg_s = (QG_sQ)^{-1}Qg_s = G_s^{-1}\tilde{g}_s \tag{A.3}$$

for $s = 1, \ldots, p$.

A.3.2 Proof of forward mapping algorithm

For $s = 1$, the right hand side of (A.1) simplifies to

$$g_1 = G_1\boldsymbol{\phi}_1 \iff \gamma_1 = \gamma_0\phi_{11}$$

and so

$$\phi_{11} = \frac{\gamma_1}{\gamma_0} = \frac{\gamma_1}{\sigma_0^2}. \tag{A.4}$$

For $s = 2, \ldots, p$, the vectors and matrices in (A.1) can be partitioned to give

$$\begin{pmatrix} G_{s-1} & \tilde{g}_{s-1} \\ \tilde{g}_{s-1}^T & \gamma_0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi}_{s,-s} \\ \phi_{ss} \end{pmatrix} = \begin{pmatrix} g_{s-1} \\ \gamma_s \end{pmatrix}$$

which is equivalent to

$$G_{s-1}\boldsymbol{\phi}_{s,-s} + \tilde{g}_{s-1}\phi_{ss} = g_{s-1}, \tag{A.5}$$

$$\tilde{g}_{s-1}^T\phi_{s,-s} + \gamma_0\phi_{ss} = \gamma_s. \tag{A.6}$$

Rearranging Equation (A.5) gives

$$\boldsymbol{\phi}_{s,-s} = G_{s-1}^{-1}(g_{s-1} - \tilde{g}_{s-1}\phi_{ss}) = G_{s-1}^{-1}g_{s-1} - G_{s-1}^{-1}\tilde{g}_{s-1}\phi_{ss}$$

and then using (A.1) and (A.3) this becomes

$$\boldsymbol{\phi}_{s,-s} = \boldsymbol{\phi}_{s-1} - Q\boldsymbol{\phi}_{s-1}\phi_{ss}. \tag{A.7}$$

Therefore, for $s = 2, \ldots, p$ and $i = 1, \ldots, s-1$, this reduces to

$$\phi_{si} = \phi_{s-1,i} - \phi_{ss}\phi_{s-1,s-i}. \tag{A.8}$$

Equation (A.8) is the result used in step 2b during the forward mapping algorithm. Now,

using Equations (A.6) and (A.7) with the properties of Q we obtain

$$\begin{aligned}
\gamma_0\phi_{ss} &= \gamma_s - \tilde{g}_{s-1}^T\boldsymbol{\phi}_{s,-s} \\
&= \gamma_s - \tilde{g}_{s-1}^T(\boldsymbol{\phi}_{s-1} - Q\boldsymbol{\phi}_{s-1}\phi_{ss}) \\
&= \gamma_s - \tilde{g}_{s-1}^T\boldsymbol{\phi}_{s-1} + \tilde{g}_{s-1}^TQ\boldsymbol{\phi}_{s-1}\phi_{ss} \\
&= \gamma_s - \tilde{g}_{s-1}^T\boldsymbol{\phi}_{s-1} + (Qg_{s-1})^TQ\boldsymbol{\phi}_{s-1}\phi_{ss} \\
&= \gamma_s - \tilde{g}_{s-1}^T\boldsymbol{\phi}_{s-1} + g_{s-1}^T\boldsymbol{\phi}_{s-1}\phi_{ss},
\end{aligned}$$

and so

$$\phi_{ss}(\gamma_0 - \boldsymbol{\phi}_{s-1}^Tg_{s-1}) = \gamma_s - \boldsymbol{\phi}_{s-1}^T\tilde{g}_{s-1}. \tag{A.9}$$

Therefore, we can obtain

$$\phi_{ss} = \frac{(\gamma_s - \boldsymbol{\phi}_{s-1}^T\tilde{g}_{s-1})}{(\gamma_0 - \boldsymbol{\phi}_{s-1}^Tg_{s-1})} = \frac{\gamma_s - \phi_{s-1,1}\gamma_{s-1} - \cdots - \phi_{s-1,s-1}\gamma_1}{\gamma_0 - \phi_{s-1,1}\gamma_1 - \cdots - \phi_{s-1,s-1}\gamma_{s-1}} \tag{A.10}$$

for $s = 1, \ldots, p$. This incorporates the special case given in Equation (A.4) where $\boldsymbol{\phi}_0^T\tilde{g}_0 = \boldsymbol{\phi}_0^Tg_0 = 0$. Equation (A.10) is the result used in step 2a of the forward mapping algorithm.

For univariate autoregressions, the partial autocorrelations are unambiguously defined as $\rho_{s+1} = \phi_{s+1,s+1}$; see for example, Chapter 3 of Shumway & Stoffer (2017). This yields the equation in step 2c in the forward mapping and so concludes our proof.

A.3.3 Proof of backward mapping algorithm

For the backward mapping algorithm, step 2(b)i is equivalent to step 2c in the forward mapping algorithm and step 2(b)ii is the same as step 2b in the forward mapping algorithm. Additionally, step 2(b)iv is a rearrangement of step 2a in the forward mapping using (A.2).

Partitioning the vectors in the expression for the conditional variance, $\sigma_{s+1}^2 = \gamma_0 - \boldsymbol{\Phi}_{s+1}^{\mathrm{T}}g_{s+1}$ $(s = 0, \ldots, p-1)$ as follows

$$\begin{aligned}
\sigma_{s+1}^2 &= \gamma_0 - (\boldsymbol{\Phi}_{s+1,-(s+1)}^{\mathrm{T}}, \phi_{s+1,s+1})\begin{pmatrix} g_s \\ \gamma_{s+1} \end{pmatrix} \\
&= \gamma_0 - \boldsymbol{\Phi}_{s+1,-(s+1)}^{\mathrm{T}}g_s - \phi_{s+1,s+1}\gamma_{s+1},
\end{aligned}$$

we can use (A.7) to show

$$\begin{aligned}
\sigma_{s+1}^2 &= \gamma_0 - (\boldsymbol{\Phi}_s^{\mathrm{T}} - \phi_{s+1,s+1}\boldsymbol{\Phi}_s^{\mathrm{T}}Q)g_s - \phi_{s+1,s+1}\gamma_{s+1} \\
&= \gamma_0 - \boldsymbol{\Phi}_s^{\mathrm{T}}g_s - \phi_{s+1,s+1}(\gamma_{s+1} - \boldsymbol{\Phi}_s^{\mathrm{T}}g_s).
\end{aligned}$$

Using (A.2) and (A.10), this gives

$$\begin{aligned}\sigma_{s+1}^2 &= \sigma_s^2 - \phi_{s+1,s+1}^2\sigma_s^2 \\ &= \sigma_s^2(1-\phi_{s+1,s+1}^2).\end{aligned} \tag{A.11}$$

Equation (A.11) is the result used in step 2(b)iii. Finally, rearranging (A.11) and using the fact $\phi_{s+1,s+1} = \rho_{s+1}$ gives the result in step 1b of the backward mapping.

A.4 Marginal distribution for $\boldsymbol{a}_s|\pi_s$ in CUSP prior allowing prior dependence

Let

$$\boldsymbol{a}_s = \text{vec}(A_s) = (a_{s,11},\ldots,a_{s,mm})|\theta_s \sim \text{N}_{m^2}(\boldsymbol{0}, V\theta_s)$$

for $s = 1,\ldots,H$, where V is an $m \times m$ positive definite matrix, then

$$\begin{aligned}\pi(\boldsymbol{a}_s,\theta_s|\pi_s) =& \pi(\boldsymbol{a}_s|\theta_s)\pi(\theta_s|\pi_s) \\ =& |2\pi\theta_s V|^{-1/2}\exp\left\{-\frac{1}{2}\boldsymbol{a}_s^{\text{T}}(\theta_s V)^{-1}\boldsymbol{a}_s\right\} \times \left\{(1-\pi_s)\frac{b^a\theta_s^{-a-1}}{\Gamma(a)}\exp\left(-\frac{b}{\theta_s}\right) + \pi_s\delta_{\theta_\infty}\right\} \\ =& (1-\pi_s)\frac{b^a}{\Gamma(a)(2\pi)^{m/2}}\theta_s^{-a-m/2-1}|V|^{-1/2}\exp\left\{-\frac{1}{\theta_s}\left(\frac{\boldsymbol{a}_s^{\text{T}}V^{-1}\boldsymbol{a}_s}{2}+b\right)\right\} \\ &+ \pi_s|2\pi\theta_s V|^{-1/2}\exp\left\{-\frac{1}{2}\boldsymbol{a}_s^{\text{T}}(\theta_s V)^{-1}\boldsymbol{a}_s\right\}\delta_{\theta_\infty}.\end{aligned}$$

Taking the integral of each half of the sum separately:

$$\begin{aligned}
&\int_0^\infty (1-\pi_s)\frac{b^a}{\Gamma(a)(2\pi)^{m/2}}\theta_s^{-a-m/2-1}|V|^{-1/2}\exp\left\{-\frac{1}{\theta_s}\left(\frac{\boldsymbol{a}_s^{\mathrm{T}}V^{-1}\boldsymbol{a}_s}{2}+b\right)\right\}d\theta_s \\
&\quad=(1-\pi_s)\frac{b^a}{\Gamma(a)(2\pi)^{m/2}}|V|^{-1/2}2^{a+m/2}\Gamma\left(a+\frac{m}{2}\right)\left(\boldsymbol{a}_s^{\mathrm{T}}V^{-1}\boldsymbol{a}_s+2b\right)^{-a-m/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{m}{2}\right)}{\Gamma(a)(2\pi)^{m/2}}|V|^{-1/2}b^a b^{-m/2}b^{m/2}2^{a+m/2}\left(\boldsymbol{a}_s^{\mathrm{T}}V^{-1}\boldsymbol{a}_s+2b\right)^{-a-m/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{m}{2}\right)}{\Gamma(a)(2\pi)^{m/2}}b^{-m/2}|V|^{-1/2}(2b)^{a+m/2}\left(\boldsymbol{a}_s^{\mathrm{T}}V^{-1}\boldsymbol{a}_s+2b\right)^{-a-m/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{m}{2}\right)}{\Gamma(a)(2\pi)^{m/2}}b^{-m/2}|V|^{-1/2}\left(\frac{\boldsymbol{a}_s^{\mathrm{T}}V^{-1}\boldsymbol{a}_s+2b}{2b}\right)^{-a-m/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{m}{2}\right)}{\Gamma(a)(2\pi)^{m/2}}a^{-m/2}\frac{b^{-m/2}}{a^{-m/2}}|V|^{-1/2}\left(1+\frac{a\boldsymbol{a}_s^{\mathrm{T}}V^{-1}\boldsymbol{a}_s}{2ab}\right)^{-a-m/2} \\
&\quad=(1-\pi_s)\frac{\Gamma\left(a+\frac{m}{2}\right)}{\Gamma(a)(2a\pi)^{m/2}}|b/aV|^{-1/2}\left\{1+\frac{\boldsymbol{a}_s^{\mathrm{T}}(b/aV)^{-1}\boldsymbol{a}_s}{2a}\right\}^{-a-m/2} \\
&\quad=(1-\pi_s)\mathrm{t}_{m^2,2a}(A_s;\mathbf{0},b/aV).
\end{aligned}$$

Then

$$\begin{aligned}
&\int_0^\infty \pi_s|2\pi\theta_s V|^{-1/2}\exp\left\{-\frac{1}{2}\boldsymbol{a}_s^{\mathrm{T}}(\theta_s V)^{-1}\boldsymbol{a}_s\right\}\delta_{\theta_\infty}(\theta_s)d\theta_s \\
&\qquad=\pi_s|2\pi\theta_\infty V|^{-1/2}\exp\left\{-\frac{1}{2}\boldsymbol{a}_s^{\mathrm{T}}(\theta_\infty V)^{-1}\boldsymbol{a}_s\right\} \\
&\qquad=\pi_s\mathrm{N}_{m^2}(A_s;\mathbf{0},\theta_\infty V).
\end{aligned}$$

As such,

$$\pi(\boldsymbol{a}_s|\pi_s)=(1-\pi_s)\mathrm{t}_{m^2,2a}(\mathbf{0},b/aV)+\pi_s\mathrm{N}_{m^2}(\mathbf{0},\theta_\infty V).$$

A.5 Derivation of $\mathrm{E}(p^*|\boldsymbol{y})$ in CUSP prior

Legramanti *et al.* (2020) define a set of auxiliary indicator variables $z_s, s=1,\ldots,H$, with probabilities $\Pr(z_s=l|\omega_l)=\omega_l$ to enable the use of full conditional distributions in Gibbs sampling, discussed further in Section 5.3.3. Using these indicator variables, as discussed in Section 5.3.3, we can write

$$p^*=H-\sum_{s=1}^{H}\mathbb{1}(z_s\leq s).$$

Then

$$\begin{aligned}
\mathrm{E}(p^*|\boldsymbol{y}) &= H - \sum_{s=1}^{H} \mathrm{E}\{\mathbb{1}(z_s \leq s|\boldsymbol{y})\} \\
&= H - \sum_{s=1}^{H} \Pr(z_s \leq s|\boldsymbol{y}) \\
&= H - \sum_{s=1}^{H} \sum_{k=1}^{s} \Pr(z_s = k|\boldsymbol{y}). \qquad \text{(A.12)}
\end{aligned}$$

Using Stan we obtain a sample

$$(A_1^{(m)}, \ldots, A_H^{(m)}, \boldsymbol{\omega}^{(m)}, \Sigma^{(m)}|\boldsymbol{y}), \quad m = 1, \ldots, M$$

of size M from the posterior distribution

$$p(A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma|\boldsymbol{y}).$$

We can then calculate the Rao-Blackwellised estimate of each $\Pr(z_s = k|\boldsymbol{y})$, $s = 1, \ldots, H$, $k = 1, \ldots, s$ as

$$\begin{aligned}
\Pr(z_s = k|\boldsymbol{y}) &= \int p(z_s = k, A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma|\boldsymbol{y})(dA_1)\ldots(dA_H)(d\boldsymbol{\omega})(d\Sigma) \\
&= \int \Pr(z_s = k|A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma) \\
&\times p(A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma|\boldsymbol{y})(dA_1)\ldots(dA_H)(d\boldsymbol{\omega})(d\Sigma) \\
&\approx \frac{1}{M} \sum_{m=1}^{M} \Pr(z_s = k|A_1^{(m)}, \ldots, A_H^{(m)}, \boldsymbol{\omega}^{(m)}, \Sigma^{(m)})
\end{aligned}$$

where

$$\begin{aligned}
\Pr(z_s = k|A_1, \ldots, A_H, \boldsymbol{\omega}, \Sigma) &= \frac{\Pr(A_s|z_s = k, \boldsymbol{\omega})\Pr(z_s = k|\boldsymbol{\omega})}{\Pr(A_s|\boldsymbol{\omega})} \\
&\propto \Pr(A_s|z_s = k, \boldsymbol{\omega})\Pr(z_s = k|\boldsymbol{\omega}) \\
&\propto \Pr(A_s|z_s = k, \boldsymbol{\omega})\omega_k \\
&\propto \begin{cases} \omega_k \mathrm{N}_{m^2}(\mathbf{0}, \theta_\infty V) & \text{for } k = 1, \ldots, s \\ \omega_k \mathrm{t}_{m^2, 2a}(\mathbf{0}, b/aV) & \text{for } k = s+1, \ldots, H. \end{cases}
\end{aligned}$$

This can be used in combination with (A.12) to calculate $\mathrm{E}(p^*|\boldsymbol{y})$.

Appendix B

Simple hidden Markov model

In Section 3.2 we fit a simple hidden Markov model to the multivariate EEG data from the delta band in individual A. In this section we detail the hidden Markov model used and the associated priors used for Bayesian inference.

In a multivariate hidden Markov model, we have a sequence of m-variate observations $\{\boldsymbol{y}_t\}$ in which the distribution of the variable $\boldsymbol{y}_t$ is dependent on the value of an unobserved state variable x_t. Further details on hidden Markov models can be found in Frühwirth-Schnatter (2006). Suppose we have K hidden states. In the hidden Markov model fitted in Section 3.2 we assumed the observations $\{\boldsymbol{y}_t\}$ were normally distributed such that

$$\boldsymbol{y}_t \sim N_m(\boldsymbol{\mu}_k, \Sigma_k)$$

where $\boldsymbol{\mu}_k$ and Σ_k are the mean and variance respectively if $x_t = k$, with possible values of $k \in \{1, \ldots, K\}$. In a hidden Markov model the parameter vector $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_K)$ contains the probability of being in states $k = 1, \ldots, K$ at the initial observation and the matrix A is a $K \times K$ transition matrix where the (i, j)-th element represents the probability of transitioning from state i to state j during a time step. The joint probability distribution for the observed data $\{\boldsymbol{y}_t\}$ and the unobserved states $\{x_t\}$ then becomes

$$p(\boldsymbol{y}_{1:N}, \boldsymbol{x}_{1:N}) = p(x_1)p(\boldsymbol{y}_1|x_1)\prod_{t=2}^{N} p(x_t|x_{t-1})p(\boldsymbol{y}_t|x_t)$$

where $\boldsymbol{y}_{1:N}$ and $x_{1:N}$ refer to the sequence of observations $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N$ and hidden states $x_1, \ldots, x_N$ respectively.

When fitting the hidden Markov model to the electroencephalography data in Chapter 3, we used Bayesian inference to infer the values of the parameters $\boldsymbol{\pi}$, A, $\boldsymbol{\mu}_k$, $k = 1, \ldots, K$, and Σ_k, $k = 1, \ldots, K$. The vector $\boldsymbol{\pi}$ containing the initial state probabilities

was assigned a uniform Dirichlet prior such that

$$\boldsymbol{\pi} \sim \text{Dir}(\mathbf{1})$$

where $\mathbf{1}$ is a K-vector of ones. Similarly, each row of the transition matrix A was assigned a uniform Dirichlet prior such that

$$A_{k,} \sim \text{Dir}(\mathbf{1})$$

where $A_{k,}$ denotes the k-th row of A. The means $\boldsymbol{\mu}_k$ for each state, $k = 1, \ldots, K$ are given multivariate normal distributions such that

$$\boldsymbol{\mu}_k \sim N_m(\mathbf{0}, I_m)$$

where $\mathbf{0}$ is an m-vector of zeros and I_m is the $m \times m$ identity matrix. Finally, the variance matrices for each state Σ_k, $k = 1, \ldots, K$, are given inverse Wishart distributions

$$\Sigma_k \sim \text{W}^{-1}(S, \nu)$$

where S is an $m \times m$ scale matrix and ν is the degrees of freedom. In our analysis we take S to be the identity matrix I_m and $\nu = m + 4$ to ensure the variance is finite. We use Stan to sample from the posterior distribution of the parameters. The means $\boldsymbol{\mu}_k$ and variances Σ_k are not identifiable in the posterior, and so we order the sampled means and variances by the first element of the means $\boldsymbol{\mu}_k$. In addition to sampling from the posterior distributions of the unknown model parameters, we use Stan's inbuilt function `hmm_latent_rng` to generate posterior draws of the hidden states $x_{1:N}$.

Appendix C

Stan programmes

C.1 Hidden Markov model

This section contains the Stan programme used for inference on a simple hidden Markov model.

```
data {
  int<lower=1> N; // number of observations
  int<lower=1> K; // number of hidden states
  int<lower=1> m; // number of variates
  array[N] vector[m] y; //data
  int<lower=0> n_miss; //Number of missing data points
  array[n_miss] int<lower=1> ind_miss; //Indices of missing data points
  int<lower=m+3> df; //degrees of freedom in prior for variance
  matrix[m, m] S;  //scale matrix in prior for variance
}

parameters {
  // Discrete state model
  simplex[K] pi; // initial state probabilities
  array[K] simplex[K] A; // transition probabilities
  // Continuous observation model
  array[K] vector[m] mu; // observation means
  array[K] cov_matrix[m] Sigma; //observation variances
  array[n_miss] vector[m] y_miss; //Missing data points
}

transformed parameters {
```

```
  matrix[K, K] tpm;
  matrix[K,N] log_omega;
  array[N] vector[m] y_complete; //Data with missing data inferred

  for(k in 1:K){
    tpm[k] = A[k]';
  }

  y_complete = y;
  if(n_miss>0) {
    for(i in 1:n_miss){
      y_complete[ind_miss[i]] = y_miss[i];
    }
  }
  for (n in 1 : N) {
    for(k in 1:K){
      // The observation model could change with n, or vary in a number of
    //  different ways (which is why log_omega is passed in as an argument)
    log_omega[k, n] = multi_normal_lpdf(y[n] | mu[k], Sigma[k]);
    }

  }
}

model {
  pi ~ dirichlet(rep_vector(1, K)); //prior for initial state probabilities
  for (k in 1:K){
    //prior for observation means
    mu[k] ~ multi_normal(rep_vector(0,m), diag_matrix(rep_vector(1,m)));
    Sigma[k] ~ inv_wishart(df,S); //prior for observation variances
    A[k] ~ dirichlet(rep_vector(1, K)); //prior for transition probabilities
  }

  target += hmm_marginal(log_omega, tpm, pi);
}

generated quantities {
  //probability of being in hidden state
```

```
  matrix[K, N] hidden_probs = hmm_hidden_state_prob(log_omega, tpm, pi);
  //posterior modal state
  array[N] int states = hmm_latent_rng(log_omega, tpm, pi);
}
```

C.2 Prior for AR(p) process reparameterised in terms of partial autocorrelations, when p is known

This section contains the Stan programme used for inference on an autoregressive process with a known order, reparameterised in terms of its partial autocorrelations.

```
functions{
  //function to map between partial autocorrelations and autoregressive
  //parameters
  //also outputs the theoretical autocovariances
  vector PA_to_AR(vector pa,real sigmasquared){
    //vector to store sigma^2_s in mapping algorithm
    vector[num_elements(pa)+1] sigmasquared_s;
    //vector to store theoretical autocovariances
    vector[num_elements(pa)+1] gamma;
    // initialise
    real phisplusone_init = pa[1];
    sigmasquared_s[num_elements(pa)+1] = sigmasquared;

    //find gamma_0 using mapping algorithm
    for(s in 0:(num_elements(pa)-1)){
      sigmasquared_s[num_elements(pa)-s]=sigmasquared_s[num_elements(pa)-s+
                                                        1]./(1-pa[num_elements(pa)-s]^2);
    }
    gamma[1]=sigmasquared_s[1];

    //return initialisation if p=1
    if(num_elements(pa)==1){
      vector[2] out;
      out[1]=phisplusone_init;
      out[2]=gamma[1];
      return out;
```

```
}
//proceed with mapping algorithm if p>1
else{
  //store theoretical autocovariance for i=0,...,p-1
  vector[(num_elements(pa))] acv;
  //store previous iterations values
  vector[(num_elements(pa))] store;
  //to store output
  vector[2*num_elements(pa)] out;
  //algorithm when s=1
  sigmasquared_s[2]=sigmasquared_s[1].*(1-pa[1]^2);
  gamma[2]=pa[1]*sigmasquared_s[1];
  //rest of algorithm
  for(splusone in 2:(num_elements(pa))){
    //phi_{s+1} in algorithm
    vector[splusone] phisplusone;
    //phi_s in algorithm
    vector[splusone-1] phis;
    //used to store gamma in reverse order
    vector[splusone-1] gammas;
    //Initialise phi_s
    if(splusone==2){
      phis=[phisplusone_init]';

    }
    else{
      //phi_s is phi_{s+1} from previous iteration
      phis=store[1:(splusone-1)];
    }
    //from algorithm
    phisplusone[splusone]=pa[splusone];
    for(i in 1:(splusone-1)){
      //From algorithm
      phisplusone[i]=phis[i]-phisplusone[splusone].*phis[splusone-i];
    }
    //stores phi_{s+1} so it can be taken to next iteration
    store[1:splusone]=phisplusone;
    //from algorithm
```

```
        sigmasquared_s[splusone+1]=sigmasquared_s[splusone].*
                                                (1-pa[splusone]^2);
        //reverse s gamma values
        for(i in 1:(splusone-1)){
          gammas[i]=gamma[splusone-i+1];
        }
        //from algorithm
        gamma[splusone+1]=pa[splusone]*sigmasquared_s[splusone]+
                                              dot_product(phis,gammas);
      }
      //output is vector containing p autoregressive parameters followed
      //by gamma 0,...,p-1
      acv=gamma[1:num_elements(pa)];
      out[1:num_elements(pa)]=store;
      out[(num_elements(pa)+1):2*num_elements(pa)]=acv;
      return out;
    }
  }
  //function to create a toeplitz matrix from a vector
  matrix toeplitz(vector vec){
    matrix[num_elements(vec), num_elements(vec)] mat;
    for(i in 1:num_elements(vec)){
      for(j in 1:num_elements(vec)){
        mat[i,j]=vec[abs(i-j)+1];
      }
    }
    return mat;
  }
}



data {
  int<lower=0> N;  //Length of time series
  int<lower=1> p; //Number of partial autocorrelations
  vector[N] y;     //Time series
  real<lower=0> alpha_pa; //a in partial autocorrelations prior
  real<lower=0> beta_pa;  //b in partial autocorrelations prior
```

```
  real<lower=0> alpha_sigmasq; //a in sigma squared prior
  real<lower=0> beta_sigmasq;  //b in sigma squared prior
}

transformed data {
  real mu; //Mean of process
  mu = 0; //Zero mean
}

parameters {
  vector<lower=0,upper=1>[p] scaled_pa;  //partial autocorrelations scaled
                                         //to be in (0,1)
  real<lower=0> sigmasq;
}

transformed parameters {
  vector<lower=-1,upper=1>[p] pa; //Transformed from scaled_pa to
                                  //be in (-1,1)
  vector[p] AR; //Autoregressive parameters
  vector[p] acv; //theoretical autocovariance
  vector[2*p] func; //vector to store output of function
  cov_matrix[p] G; //autocovariance matrix
  pa = scaled_pa*2-1; //transform scaled parameters back to correct range
  func=PA_to_AR(pa,sigmasq); //store output of mapping function
  AR=func[1:p];
  acv=func[(p+1):2*p];
  G=toeplitz(acv); //create autocovariance matrix from vector
}


model {
  y[1:p]~multi_normal(rep_vector(0,p),G); //initial distribution

  //for distribution when t>p
  for(t in (p+1):N){
    real AR_mean; //mean
    vector[p] ys; //to store previous p values in time series
    for(i in 1:p){
```

```
      ys[i]=y[t-i];
    }
    AR_mean = mu + dot_product(AR, ys-mu);  //calculate mean
    y[t]~normal(AR_mean, sqrt(sigmasq)); //Distribution when t>p
  }

  //beta prior for scaled partial autocorrelations
  scaled_pa ~ beta(alpha_pa, beta_pa);
  //inverse gamma prior for sigma squared
  sigmasq ~ inv_gamma(alpha_sigmasq, beta_sigmasq);
}
```

C.3 Exchangeable prior for A_s when p is known

This section contains the Stan programme used for inference of a vector autoregressive process with a known order using an exchangeable prior for the elements of the matrices $A_1, \ldots, A_p$.

```
functions{
  //function to calculate the symmetric matrix-square-root of a matrix
  matrix sqrtm(matrix A) {
    int m = rows(A);
    tuple(matrix[m, m], vector[m]) eigen = eigendecompose_sym(A);
    matrix[m, m] eprod = diag_post_multiply(eigen.1, sqrt(sqrt(eigen.2)));
    return tcrossprod(eprod);
  }


  //function to map from the transformed partial autocorrelation
  //matrices A to the partial autocorrelation matrices P
  matrix A_to_P(matrix A){
    int m = cols(A);
    matrix[m,m] C;
    matrix[m,m] B;
    matrix[m,m] P;
    C=diag_matrix(rep_vector(1,m))+tcrossprod(A);
    B=sqrtm(C);
    P=mdivide_left_spd(B,A);
```

```
  return P;
}

//function to map from the partial autocorrelation matrices
//P to the autogressive matrices phi
//also outputs the theoretical autocovariances
array[,] matrix P_to_AR(array[] matrix P,matrix Sigma){
  int p = size(P);
  int m = cols(Sigma);
  array[p+1] matrix[m,m] Sigma_s;
  array[p+1] matrix[m,m] S_s;
  array[p+1] matrix[m,m] Sigma_star_s;
  array[p+1] matrix[m,m] S_star_s;
  array[p+1] matrix[m,m] Gamma;
  array[2,p] matrix[m,m] phiGamma;
  matrix[m,m] Y;
  matrix[m,m] sqY;
  array[p] matrix[m,m] phi_splusone;
  array[p] matrix[m,m] phi_splusone_star;
  // initialise
  Sigma_s[p+1] = Sigma;
  S_s[p+1]=sqrtm(Sigma_s[p+1]);


  //find gamma_0 using mapping algorithm
  for(s in 0:(p-1)){
    Y=diag_matrix(rep_vector(1,m))-tcrossprod(P[p-s]);
    sqY=sqrtm(Y);
    S_s[p-s] = mdivide_right_spd(mdivide_left_spd(sqY,
                                 sqrtm(quad_form_sym(Sigma_s[p-s+1], sqY))),
                                                                    sqY);
    Sigma_s[p-s] = tcrossprod(S_s[p-s]);
  }
  Gamma[1]=Sigma_s[1];
  Sigma_star_s[1]=Gamma[1];
  S_star_s[1]=S_s[1];
  phi_splusone[1]=mdivide_right(S_s[1]*P[1],S_star_s[1]);
  phi_splusone_star[1]=mdivide_right(S_star_s[1]*P[1]',S_s[1]);
```

```
Sigma_s[2]=Sigma_s[1]-quad_form_sym(Sigma_star_s[1],phi_splusone[1]');
Sigma_star_s[2]=Sigma_star_s[1]-
                        quad_form_sym(Sigma_s[1],phi_splusone_star[1]');
S_s[2]=sqrtm(Sigma_s[2]);
S_star_s[2]=sqrtm(Sigma_star_s[2]);
Gamma[2]=(phi_splusone[1]*Sigma_star_s[1])';
//return initialisation if p=1
if(p==1){

  phiGamma[1,p]=phi_splusone[1];
  phiGamma[2,p]=Gamma[1];
  return phiGamma;
}
//proceed with mapping algorithm if p>1
else{
  //store theoretical autocovariance for i=0,...,p-1
  array[p] matrix[m,m] store;
  array[p] matrix[m,m] store_star;

  store[1]=phi_splusone[1];
  store_star[1]=phi_splusone_star[1];

  //rest of algorithm
  for(splusone in 2:p){
    //phi_s in algorithm
    array[splusone-1] matrix[m,m] phi_s;
    array[splusone-1] matrix[m,m] phi_s_star;
    //used to store gamma in reverse order
    matrix[m,m] Gamma_temp;
    matrix[m,m] matrix_sum;
    //Initialise phi_s
    phi_s=store[1:(splusone-1)];
    phi_s_star=store_star[1:(splusone-1)];
    //from algorithm
    phi_splusone[splusone]=mdivide_right(S_s[splusone]*P[splusone],
                                         S_star_s[splusone]);
    phi_splusone_star[splusone]=mdivide_right(S_star_s[splusone]*
                                        P[splusone]',S_s[splusone]);
```

```
        for(i in 1:(splusone-1)){
          //From algorithm
          phi_splusone[i]=phi_s[i]-
                          phi_splusone[splusone]*phi_s_star[splusone-i];
          phi_splusone_star[i]=phi_s_star[i]-
                          phi_splusone_star[splusone]*phi_s[splusone-i];
        }
        //stores phi_{s+1}
        store[1:splusone]=phi_splusone[1:splusone];
        store_star[1:splusone]=phi_splusone_star[1:splusone];
        //from algorithm
        Sigma_s[splusone+1]=Sigma_s[splusone]-
            quad_form_sym(Sigma_star_s[splusone],phi_splusone[splusone]');
        S_s[splusone+1]=sqrtm(Sigma_s[splusone+1]);
        Sigma_star_s[splusone+1]=Sigma_star_s[splusone]-
            quad_form_sym(Sigma_s[splusone],phi_splusone_star[splusone]');
        S_star_s[splusone+1]=sqrtm(Sigma_star_s[splusone+1]);
        //reverse s gamma values
        matrix_sum = phi_splusone[splusone]*Sigma_star_s[splusone];
        for(i in 1:(splusone-1)){
          Gamma_temp=Gamma[splusone-i+1];
          matrix_sum += phi_s[i]*Gamma_temp';
        }
        //from algorithm

        Gamma[splusone+1]=matrix_sum';
      }
      //output is vector containing p autoregressive parameters followed
      //by Gamma 0,...,p-1
      phiGamma[1,]=phi_splusone;
      phiGamma[2,]=Gamma[1:p];
      return phiGamma;
    }
  }


}
```

```
data {
  int<lower=1> m; //Dimension of VAR process
  int<lower=0> N; //Length of time series
  int<lower=1> p; //Order of process
  array[N] vector[m] y; //Time series

  //Hyperparameters in inverse Wishart prior for Sigma
  int<lower=m+3> nu; // Degrees of freedom
                     //(limit ensures variance is finite)
  real<lower=0> scale_diag;       // Diagonal element in scale matrix
  real<lower=-scale_diag/(m-1)> scale_offdiag; // Off-diagonal element in
                                               // scale matrix

  //Hyperparameters in exchangeable prior for elements of A matrices
  vector[2] es;
  vector<lower=0>[2] fs;
  vector<lower=0>[2] gs;
  vector<lower=0>[2] hs;
}

transformed data {
  vector[p*m] y_init;
  vector[m] mu;
  matrix[m, m] scale_mat;  // Scale-matrix in prior for Sigma
  mu = rep_vector(0.0, m); //Zero mean of VAR process
  for(i in 1:m) {
    for(j in 1:m) {
      if(i==j) scale_mat[i, j] = scale_diag;
      else scale_mat[i, j] = scale_offdiag;
    }
  }
  for(i in 1:p){
    y_init[(1+(i-1)*m):i*m]=y[i];
  }
}


parameters {
```

```
  array[p] matrix[m,m] A; //transformed partial autocorrelation matrices
  cov_matrix[m] Sigma; //error variance
  array[2] vector[p] mus; //hyperparameters mu_s in prior
  array[2] vector<lower=0>[p] omega; //hyperparameters omega_s in prior
 }

transformed parameters {
  array[p] matrix[m,m] P; //Partial autocorrelation matrices, P_s
  array[p] matrix[m,m] phi; //The phi_i
  array[2,p] matrix[m,m] phiGamma; //place to store output
                                   //of mapping function
  array[p] matrix[m,m] Gamma; //Autocovariances
  cov_matrix[m*p] G; //Stationary variance of (y_1, ..., y_p)
  vector[p*m] mu_long; //mp-vector with mu repeated p times

  for(i in 1:p){
    P[i]=A_to_P(A[i]);
  }

  for(i in 1:p){
    mu_long[(1+(i-1)*m):i*m] = mu;
  }

  phiGamma=P_to_AR(P,Sigma);
  phi=phiGamma[1,];
  Gamma=phiGamma[2,];

  for(i in 1:p){
      for(j in 1:p){
        if(i<=j){
          G[(1+m*(i-1)):m*i,(1+m*(j-1)):m*j]=Gamma[j-i+1];
        }
        else{
          G[(1+m*(i-1)):m*i,(1+m*(j-1)):m*j]=Gamma[i-j+1]';
        }

      }
    }
```

```
}


model {
  y_init~multi_normal(mu_long,G); //initial distribution

  //for distribution when t>p
  for(t in (p+1):N){
    vector[m] AR_mean; //conditional mean of y_t
    array[p] vector[m] ys; //to store previous p values in time series
    for(i in 1:p){
      ys[i]=y[t-i];
    }
    AR_mean=mu;
    for(i in 1:p){
      AR_mean=AR_mean+phi[i]*(ys[i] - mu);
    }
    y[t]~multi_normal(AR_mean,Sigma); //Distribution when t>p
  }

  //Hierarchical prior for elements of A matrices
  for(s in 1:p){
    for(i in 1:m){
      for(j in 1:m){
        if(i==j){
          A[s,i,i]~normal(mus[1,s],1/sqrt(omega[1,s]));
        }
        else{
          A[s,i,j]~normal(mus[2,s],1/sqrt(omega[2,s]));
        }
      }
    }
  }
  for(i in 1:2){
    mus[i]~normal(es[i],fs[i]);
    omega[i]~gamma(gs[i],hs[i]);
    }
```

```
  //Inverse Wishart prior for Sigma
  Sigma ~ inv_wishart(nu,scale_mat);


}
```

C.4 Cumulative shrinkage process

This section contains the Stan programme used for inference on the order of a vector autoregressive process using the cumulative shrinkage process prior described in Section 5.3.3. Note that the full content of the `sqrtm`, `A_to_P` and `P_to_AR` functions are omitted as they are identical to those provided in the Stan programme in Appendix C.3.

```
functions{

  //function to calculate the symmetric matrix-square-root of a matrix
  matrix sqrtm(matrix A) {
    ...
  }


  //function to map from the transformed partial autocorrelation
  //matrices A to the partial autocorrelation matrices P
  matrix A_to_P(matrix A){
    ...
  }

  //function to map from the partial autocorrelation matrices
  //P to the autogressive matrices phi
  //also outputs the theoretical autocovariances
  array[,] matrix P_to_AR(array[] matrix P,matrix Sigma){
    ...
  }


  //function to map from the nu parameters to omega in the CUSP prior
  vector nu_to_omega(vector nu){
    vector[num_elements(nu)] log_omega;
```

```
    real log_omega_temp;
    vector[num_elements(nu)] omega;
    log_omega[1]=log(nu[1]);
      for(i in 2:num_elements(nu)){
        log_omega_temp=log(nu[i]);
        for(j in 1:(i-1)){
          log_omega_temp=log_omega_temp+log(1-nu[j]);
        }
        log_omega[i]=log_omega_temp;

    }
    omega=exp(log_omega);
    //return output;
    return omega;
    }


  //function to map from the omega parameters to pi in the cusp prior
  vector omega_to_pi(vector omega){
    vector[num_elements(omega)] pi;
    if(num_elements(omega)==1){
      pi=omega;
      return pi;
    }
    else{
         for(i in 1:(num_elements(omega)-1)){
          pi[i]=sum(omega[1:i]);
    }
    }

    pi[num_elements(omega)]=1;
    //return output;
    return pi;
    }


}
```

```
data {
  int<lower=1> m; //Dimension of VAR process
  int<lower=0> N; //Length of time series
  int<lower=1> p; //Maximum order
  int<lower=0> n_miss; //Number of missing data points
  array[N] vector[m] y;     //Time series
  array[n_miss] int<lower=1> ind_miss; //Indices of missing data points

  //Hyperparameters in inverse Wishart prior for Sigma
  int<lower=m+3> df; // Degrees of freedom
                     //(limit ensures variance is finite)
  real<lower=0> scale_diag;           // Diagonal element in scale matrix
  real<lower=-scale_diag/(m-1)> scale_offdiag; /* Off-diagonal element in
                                                  scale matrix */

  //Hyperparameters in cumulative shirnkage process prior for elements
  // of A matrices
  real<lower=0> alpha; //prior expectation of order
  real<lower=0> a;    //parameters for inv-gamma for slab
  real<lower=0> b;  //parameters for inv-gamma for slab
  real<lower=0> theta_inf; //value of spike
}


transformed data {
  vector[m] mu;
  matrix[m, m] scale_mat;  // Scale-matrix in prior for Sigma
  mu = rep_vector(0.0, m); //Zero mean of VAR process
  for(i in 1:m) {
    for(j in 1:m) {
      if(i==j) scale_mat[i, j] = scale_diag;
      else scale_mat[i, j] = scale_offdiag;
    }
  }
}
```

```
parameters {
  array[p] matrix[m,m] A;
  cov_matrix[m] Sigma;
  vector<lower=0,upper=1>[p-1] nu_start;  //all elements of nu
                                //except for last which is set as 1
  array[n_miss] vector[m] y_miss;
}



transformed parameters {
  array[p] matrix[m,m] P; //Partial autocorrelation matrices, P_s
  array[p] matrix[m,m] phi; //The phi_i
  array[2,p] matrix[m,m] phiGamma; //place to store output of
                                   // mapping function
  array[p] matrix[m,m] Gamma; //Autocovariances
  cov_matrix[m*p] G; //Stationary variance of (y_1, ..., y_p)
  vector[p*m] y_init; //Initial values
  vector[p*m] mu_long; //mp-vector with mu repeated p times
  vector<lower=0,upper=1>[p] nu; //nu to set up probabilities for pi
  vector<lower=0,upper=1>[p] pi; //probabilities of being 0
  vector<lower=0,upper=1>[p] omega;
  array[N] vector[m] y_complete; //Data with missing data inferred
  y_complete = y;
  if(n_miss>0) {
    for(i in 1:n_miss){
      y_complete[ind_miss[i]] = y_miss[i];
    }
  }

  for(i in 1:p){
    y_init[(1+(i-1)*m):i*m]=y_complete[i];
  }

  for(i in 1:p){
    P[i]=A_to_P(A[i]);
  }
```

```
  for(i in 1:p){
    mu_long[(1+(i-1)*m):i*m] = mu;
  }

  phiGamma=P_to_AR(P,Sigma);
  phi=phiGamma[1,];
  Gamma=phiGamma[2,];

  for(i in 1:p){
      for(j in 1:p){
        if(i<=j){
          G[(1+m*(i-1)):m*i,(1+m*(j-1)):m*j]=Gamma[j-i+1];
        }
        else{
          G[(1+m*(i-1)):m*i,(1+m*(j-1)):m*j]=Gamma[i-j+1]';
        }

      }
    }

  nu[1:(p-1)]=nu_start;
  nu[p]=1;
  omega=nu_to_omega(nu);
  pi=omega_to_pi(omega);

}



model {
  vector[2] tmp;
  y_init~multi_normal(mu_long,G); //initial distribution

  //for distribution when t>p
  for(t in (p+1):N){
    vector[m] AR_mean; //conditional mean of y_t
    array[p] vector[m] ys; //to store previous p values in time series
    for(i in 1:p){
```

```
      ys[i]=y_complete[t-i];
    }
    AR_mean=mu;
    for(i in 1:p){
      AR_mean=AR_mean+phi[i]*(ys[i] - mu);
    }
    y_complete[t]~multi_normal(AR_mean,Sigma); //Distribution when t>p
  }

  //Cumulative shrinkage process prior for A matrices
  for(s in 1:p){
    tmp[1]=log(1-pi[s])+multi_student_t_lpdf(to_vector(A[s])|2*a,
                        rep_vector(0,m*m),diag_matrix(rep_vector(b/a,m*m)));
    tmp[2]=log(pi[s])+multi_normal_lpdf(to_vector(A[s])|rep_vector(0,m*m),
                                        diag_matrix(rep_vector(theta_inf,m*m)));
    target +=log_sum_exp(tmp);
  }
  nu_start ~ beta(1,alpha);

  //Inverse Wishart prior for Sigma
  Sigma ~ inv_wishart(df,scale_mat);


}
```

C.5 Multiplicative gamma process

This section contains the Stan programme used for inference on the order of a vector autoregressive process using the multiplicative gamma process prior described in Section 5.3.4. Note that the full content of the `sqrtm`, `A_to_P` and `P_to_AR` functions are omitted as they are identical to those provided in the Stan programme in Appendix C.3.

```
functions{
  //function to calculate the symmetric matrix-square-root of a matrix
  matrix sqrtm(matrix A) {
    ...
  }
```

```
  //function to map from the transformed partial autocorrelation
  //matrices A to the partial autocorrelation matrices P
  matrix A_to_P(matrix A){
    ...
  }

  //function to map from the partial autocorrelation matrices
  //P to the autogressive matrices phi
  //also outputs the theoretical autocovariances
  array[,] matrix P_to_AR(array[] matrix P, matrix Sigma){
    ...
  }

}



data {
  int<lower=1> m; //Dimension of VAR process
  int<lower=0> N; //Length of time series
  int<lower=1> p; //Maximum order
  int<lower=0> n_miss; //Number of missing data points
  array[N] vector[m] y;     //Time series
  array[n_miss] int<lower=1> ind_miss; //Indices of missing data points

  //Hyperparameters in inverse Wishart prior for Sigma
  int<lower=m+3> df; // Degrees of freedom
                     //(limit ensures variance is finite)
  real<lower=0> scale_diag;  // Diagonal element in scale matrix
  real<lower=-scale_diag/(m-1)> scale_offdiag; /* Off-diagonal element in
                                                  scale matrix */
  //Hyperparameters in multiplicative gamma process prior for
  //elements of A matrices
  real<lower=0> a;
  real<lower=0> a1;
  real<lower=0> a2;
}
```

```
transformed data {
  vector[m] mu;
  matrix[m, m] scale_mat;  // Scale-matrix in prior for Sigma
  mu = rep_vector(0.0, m); //Zero mean of VAR process
  for(i in 1:m) {
    for(j in 1:m) {
      if(i==j) scale_mat[i, j] = scale_diag;
      else scale_mat[i, j] = scale_offdiag;
    }
  }
}


parameters {
  array[p] matrix[m,m] A; //The A matrices
  cov_matrix[m] Sigma; //Error variance, Sigma
  array[n_miss] vector[m] y_miss; //Missing data points
  //Parameters in multiplicative gamma process prior for the elements
  //of the A matrices
  vector<lower=0>[p] delta;
  array[p] matrix<lower=0>[m,m] lambda;
}



transformed parameters {
  array[p] matrix[m,m] P; //Partial autocorrelation matrices, P_s
  array[p] matrix[m,m] phi; //The phi_i
  array[2,p] matrix[m,m] phiGamma; //place to store output
                                   //of mapping function
  array[p] matrix[m,m] Gamma; //Autocovariances
  cov_matrix[m*p] G; //Stationary variance of (y_1, ..., y_p)
  vector<lower=0>[p] tau; //Parameter in multiplicative gamma process prior
  vector[p*m] y_init; //Initial values
  vector[p*m] mu_long; //mp-vector with mu repeated p times
```

```
array[N] vector[m] y_complete; //Data with missing data inferred
y_complete = y;
if(n_miss>0) {
  for(i in 1:n_miss){
    y_complete[ind_miss[i]] = y_miss[i];
  }
}

for(i in 1:p){
  y_init[(1+(i-1)*m):i*m]=y_complete[i];
}
for(i in 1:p){
  P[i]=A_to_P(A[i]);
}
for(i in 1:p){
  mu_long[(1+(i-1)*m):i*m] = mu;
}
phiGamma=P_to_AR(P,Sigma);
phi=phiGamma[1,];
Gamma=phiGamma[2,];

for(i in 1:p){
    for(j in 1:p){
      if(i<=j){
        G[(1+m*(i-1)):m*i,(1+m*(j-1)):m*j]=Gamma[j-i+1];
      }
      else{
        G[(1+m*(i-1)):m*i,(1+m*(j-1)):m*j]=Gamma[i-j+1]';
      }

    }
  }
  {
    vector[p] logtau;
    logtau[1]=log(delta[1]);
    for(j in 2:p){
      logtau[j]=logtau[j-1]+log(delta[j]);
    }
```

```
    tau = exp(logtau);
  }

}



model {
  y_init~multi_normal(mu_long,G); //initial distribution

  //for distribution when t>p
  for(t in (p+1):N){
    vector[m] AR_mean; //conditional mean of y_t
    array[p] vector[m] ys; //to store previous p values in time series
    for(i in 1:p){
      ys[i]=y_complete[t-i];
    }
    AR_mean=mu;
    for(i in 1:p){
      AR_mean=AR_mean+phi[i]*(ys[i] - mu);
    }
    y_complete[t]~multi_normal(AR_mean,Sigma); //Distribution when t>p
  }

  // Multiplicative gamma process prior for A matrices
  for(s in 1:p){
    for(i in 1:m){
      for(j in 1:m){
        lambda[s,i,j]~gamma(a/2,a/2);
        A[s,i,j]~normal(0,1/sqrt(lambda[s,i,j]*tau[s]));
      }
    }

  }
  delta[1]~gamma(a1,1);
  for(i in 2:p){
    delta[i]~gamma(a2,1);
  }
```

```
  //Inverse Wishart prior for Sigma
  Sigma ~ inv_wishart(df,scale_mat);
}
```

Appendix D

Results of Box-Jenkins approach to model fitting

In Section 3.3 we used classical methods for determining the order of univariate ARMA models to explore the dependence structure in the data. In the main thesis we included results from the delta band of individual A, with the results from the remaining data sets included here. Tables D.1 to D.7 contain the models chosen for each data set using an iterative approach to the Box-Jenkins method, as discussed in Section 3.3. Figures D.1 to D.7 contain pairs plots of the residuals obtained from fitting the chosen models to each region, for each data set. In all data sets, the residuals from the different regions are correlated suggesting we should model the data as multivariate time series, as discussed in Section 3.3.

Region	Chosen model
1	AR(7)
2	ARMA(1,1)
3	AR(4)
4	AR(3)
5	AR(2)
6	AR(7)
7	AR(5)
8	AR(5)
9	AR(6)

Table D.1: Models chosen for each region for the beta band in individual A using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

Region	Chosen model
1	ARMA(2,1)
2	white noise
3	white noise
4	AR(3)
5	ARMA(1,1)
6	AR(4)
7	ARMA(1,1)
8	ARMA(2,1)

Table D.2: Models chosen for each region for the delta band in individual B using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

Region	Chosen model
1	ARMA(7,1)
2	white noise
3	white noise
4	white noise
5	white noise
6	white noise
7	white noise
8	AR(2)

Table D.3: Models chosen for each region for the beta band in individual B using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

Region	Chosen model
1	ARMA(5,2)
2	ARMA(4,1)
3	ARMA(4,3)
4	ARMA(7,2)
5	ARMA(6,1)
6	AR(7)
7	ARMA(5,2)
8	ARMA(4,1)

Table D.4: Models chosen for each region for the delta band in individual C using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

Region	Chosen model
1	ARMA(5,1)
2	AR(5)
3	AR(4)
4	AR(5)
5	ARMA(1,1)
6	ARMA(5,2)
7	ARMA(4,1)
8	ARMA(2,1)

Table D.5: Models chosen for each region for the beta band in individual C using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

Region	Chosen model
1	AR(2)
2	AR(2)
3	ARMA(1,1)
4	AR(1)
5	ARMA(1,1)
6	AR(2)
7	AR(3)
8	AR(1)
9	AR(1)
10	AR(2)
11	ARMA(1,1)
12	AR(2)
13	AR(1)

Table D.6: Models chosen for each region for the delta band in individual D using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

Region	Chosen model
1	ARMA(5,2)
2	ARMA(2,1)
3	ARMA(2,1)
4	AR(1)
5	ARMA(2,1)
6	ARMA(2,1)
7	ARMA(2,1)
8	ARMA(2,1)
9	AR(1)
10	AR(5)
11	ARMA(2,1)
12	ARMA(2,1)
13	AR(1)

Table D.7: Models chosen for each region for the beta band in individual D using an iterative approach to the Box-Jenkins method. Region names are detailed in 3.1 and the regions are depicted in 3.1.

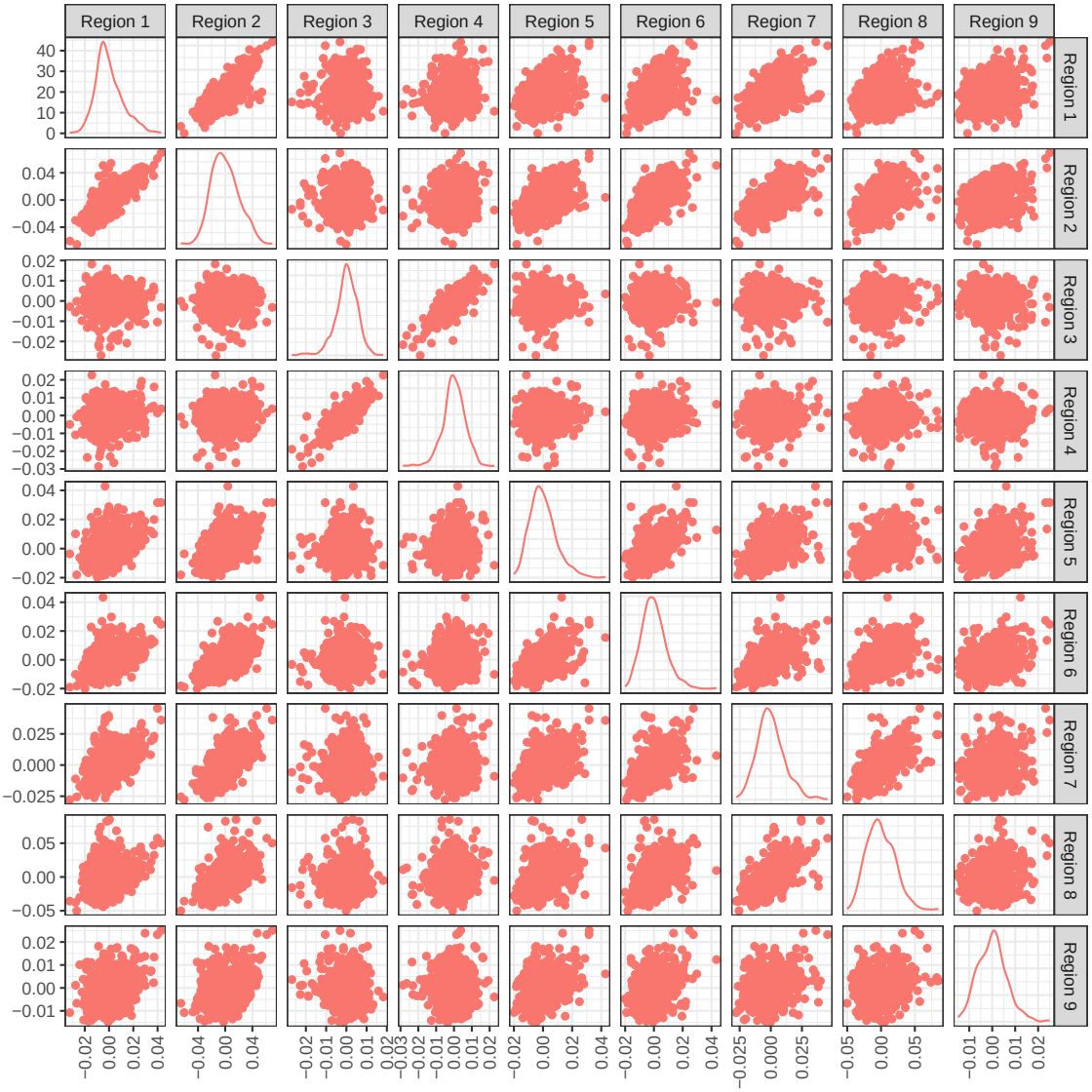


Figure D.1: Pairs plot of the residuals obtained from each region after fitting the models in Table D.1 for the beta band in individual A. Region names are detailed in 3.1 and the regions are depicted in 3.1.

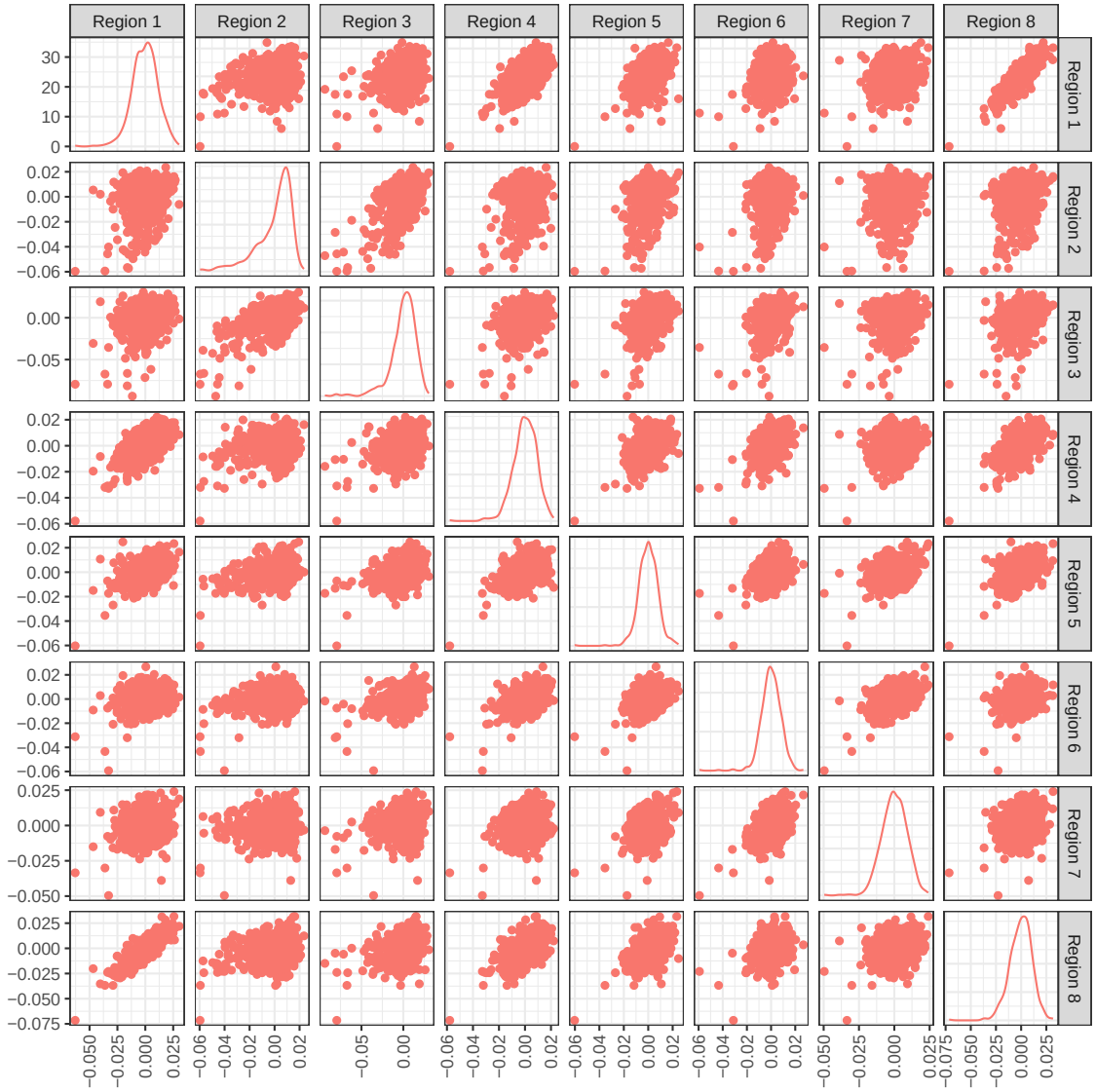


Figure D.2: Pairs plot of the residuals obtained from each region after fitting the models in Table D.2 for the delta band in individual B. Region names are detailed in 3.1 and the regions are depicted in 3.1.

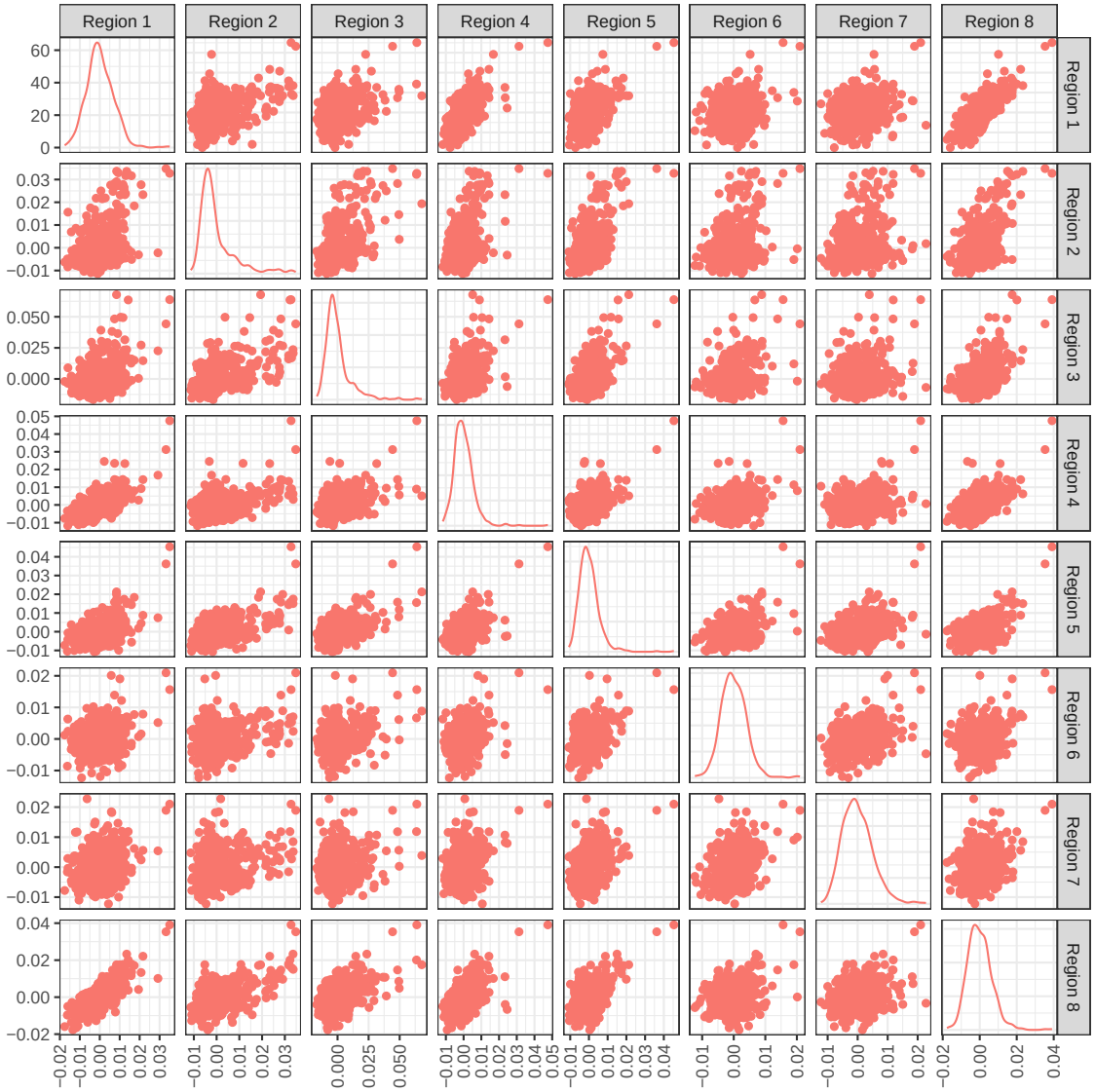


Figure D.3: Pairs plot of the residuals obtained from each region after fitting the models in Table D.3 for the beta band in individual B. Region names are detailed in 3.1 and the regions are depicted in 3.1.

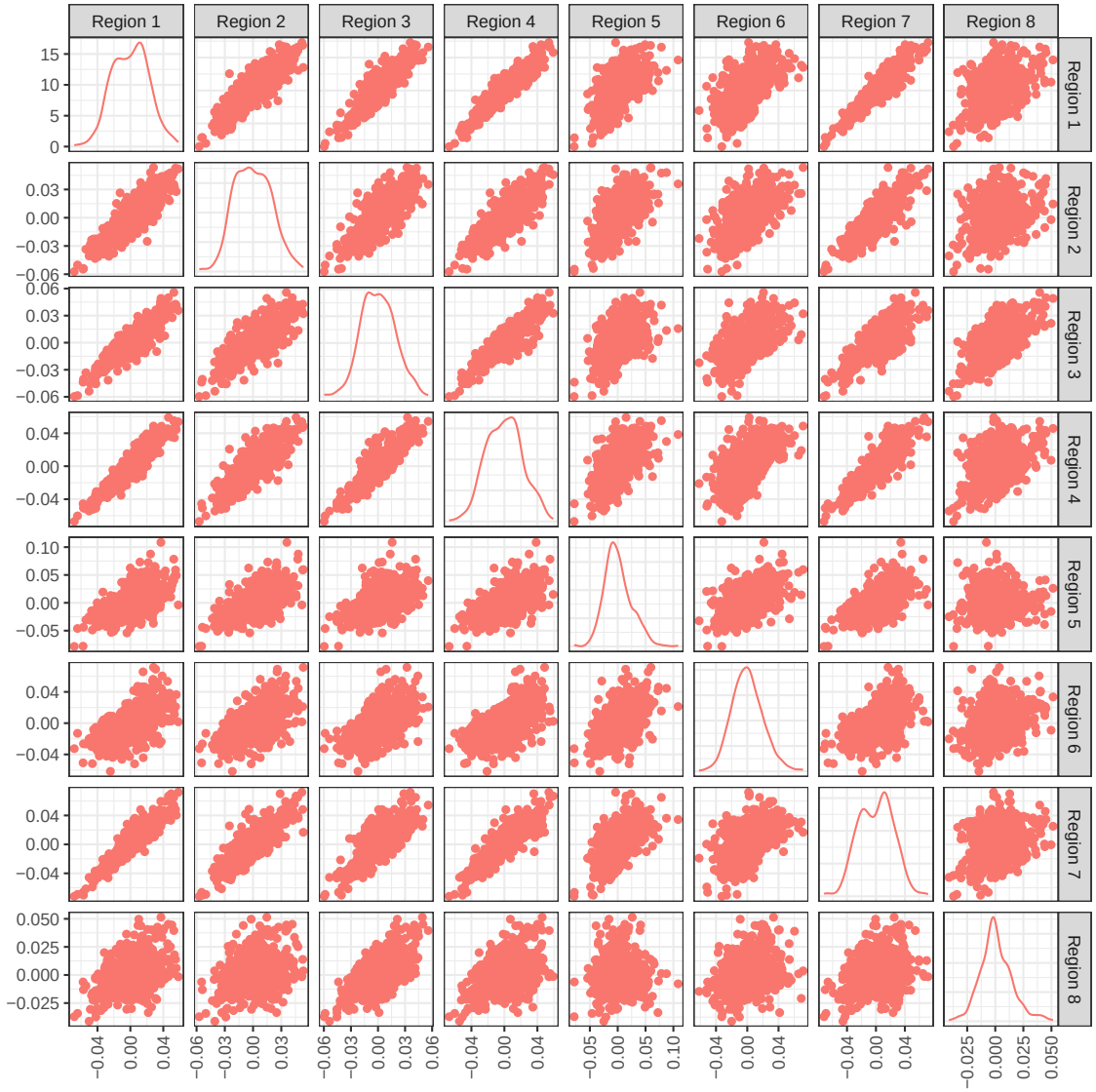


Figure D.4: Pairs plot of the residuals obtained from each region after fitting the models in Table D.4 for the delta band in individual C. Region names are detailed in 3.1 and the regions are depicted in 3.1.

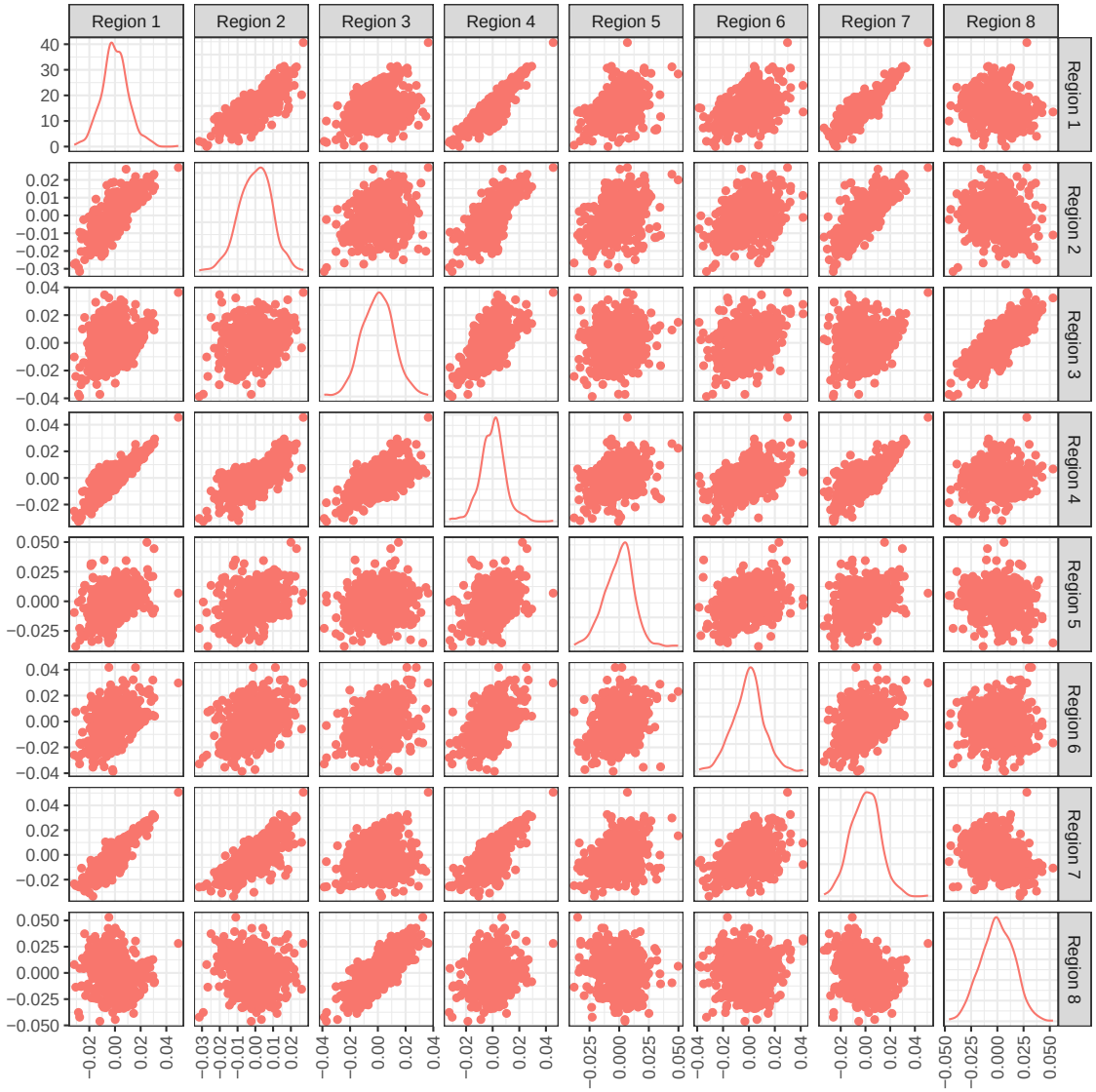


Figure D.5: Pairs plot of the residuals obtained from each region after fitting the models in Table D.5 for the beta band in individual C. Region names are detailed in 3.1 and the regions are depicted in 3.1.

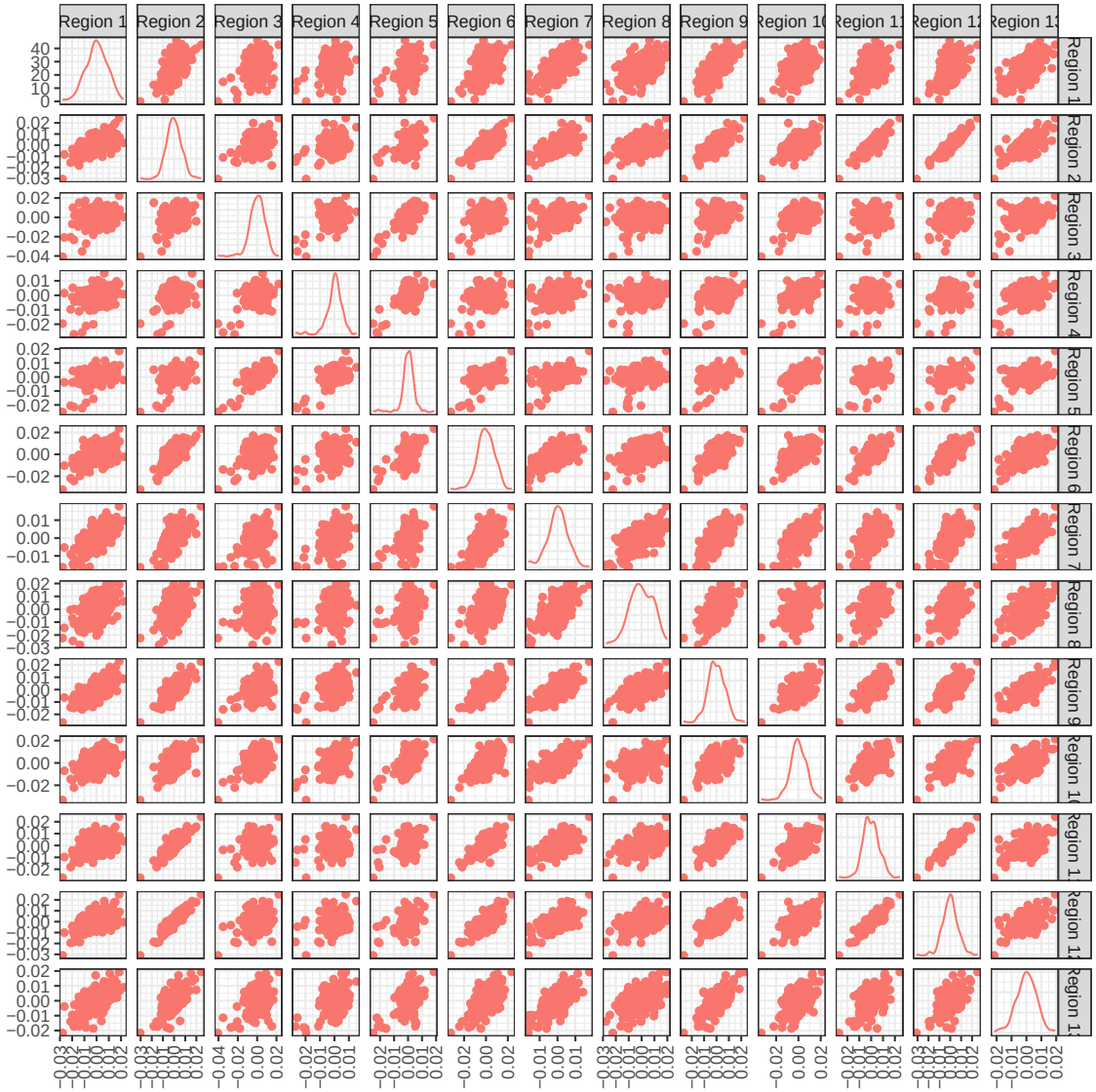


Figure D.6: Pairs plot of the residuals obtained from each region after fitting the models in Table D.6 for the delta band in individual D. Region names are detailed in 3.1 and the regions are depicted in 3.1.

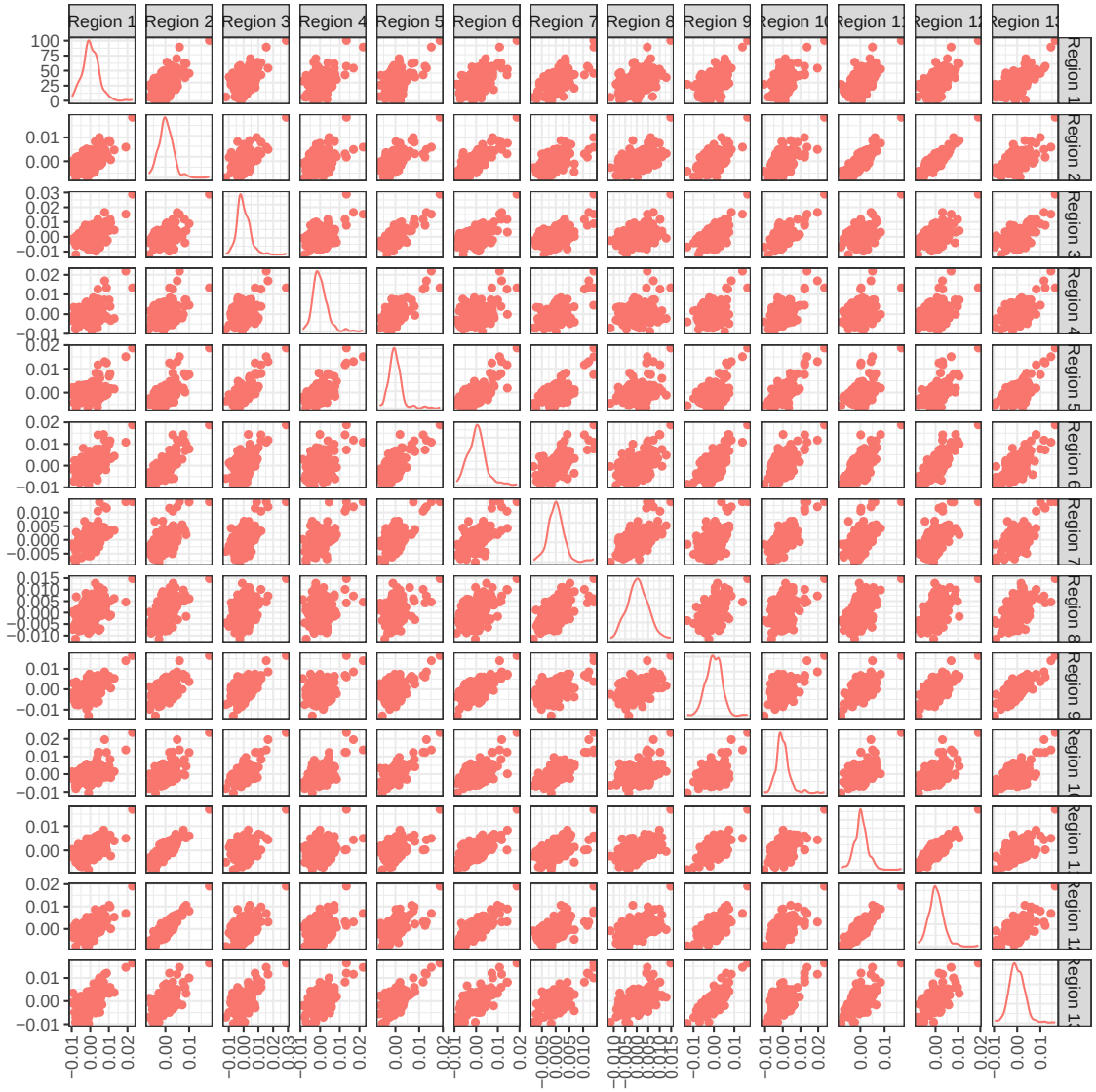
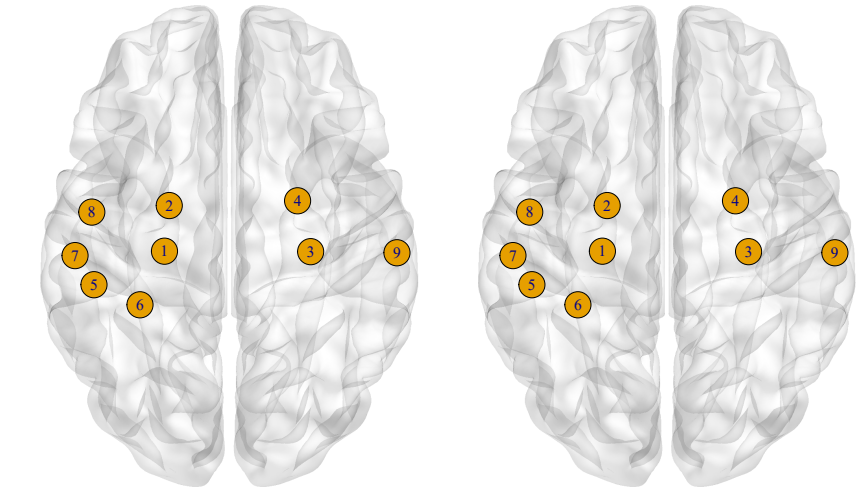


Figure D.7: Pairs plot of the residuals obtained from each region after fitting the models in Table D.7 for the beta band in individual D. Region names are detailed in 3.1 and the regions are depicted in 3.1.

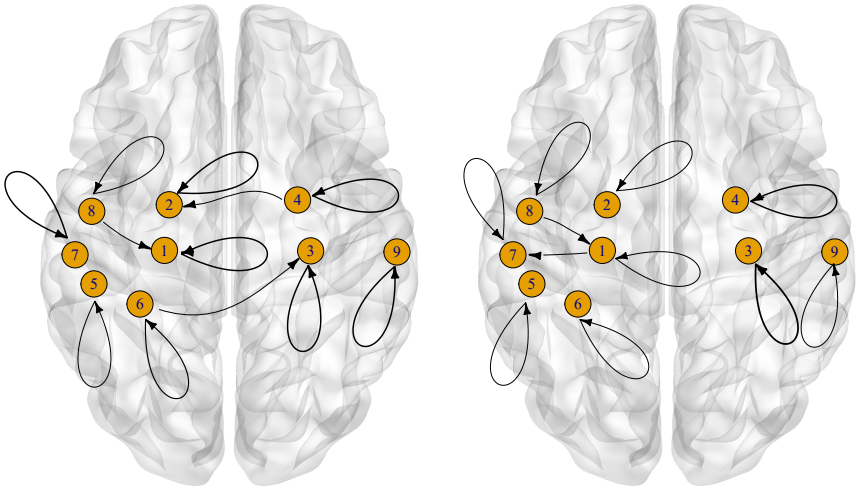
Appendix E

GraphicalVAR Granger causality plots

The following plots demonstrate the Granger networks obtained when using the graphicalVAR package (Epskamp, 2024) to estimate the autocoefficient matrices when fitting VAR(2) processes to the EEG data sets. The graphicalVAR package implements regularised-likelihood estimation with a lasso penalty on the individual autoregressive coefficients.

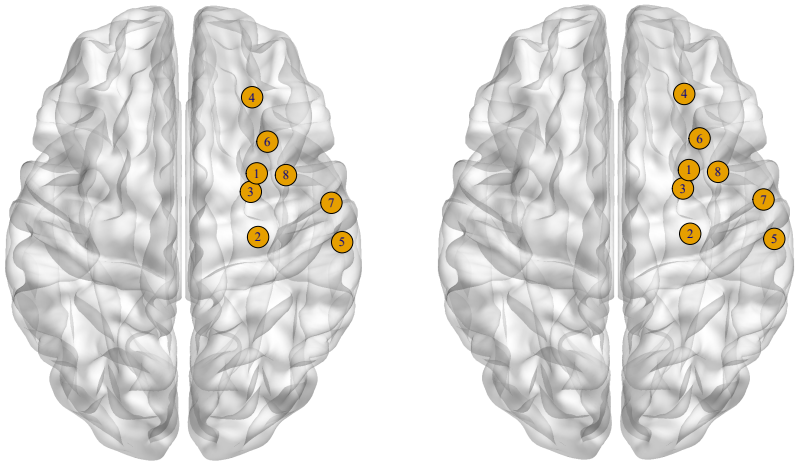


(a)

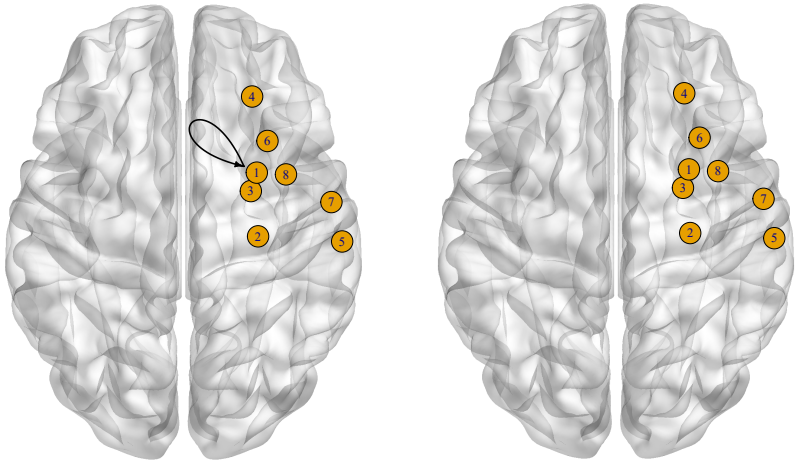


(b)

Figure E.1: Granger causality plots of the autoregressive coefficient matrices obtained using the graphicalVAR package overlaid on glass brains showing the locations of the regions, for the VAR process of individual A in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

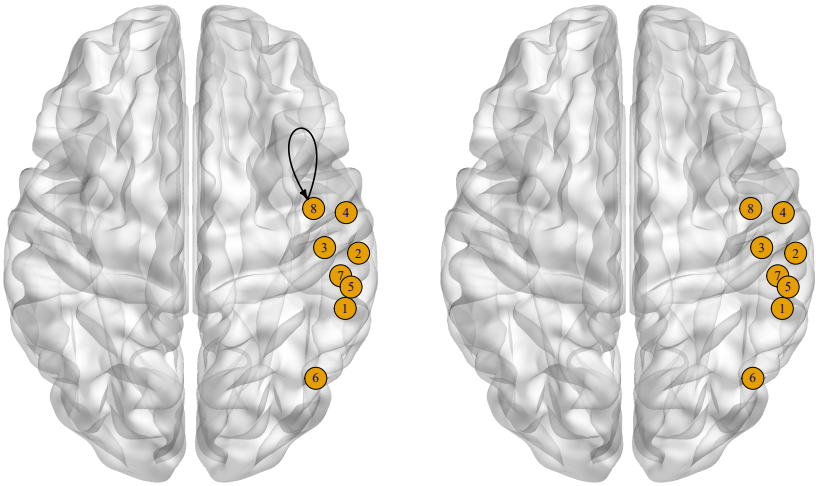


(a)

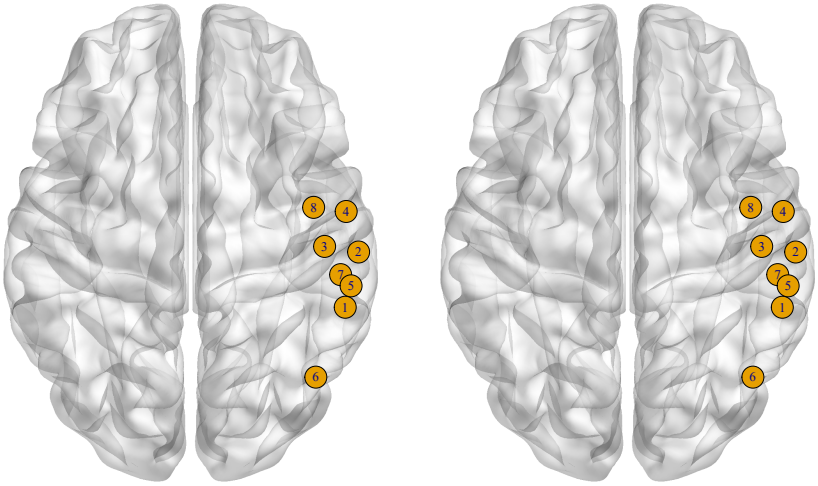


(b)

Figure E.2: Granger causality plots of the autoregressive coefficient matrices obtained using the graphicalVAR package overlaid on glass brains showing the locations of the regions, for the VAR process of individual B in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

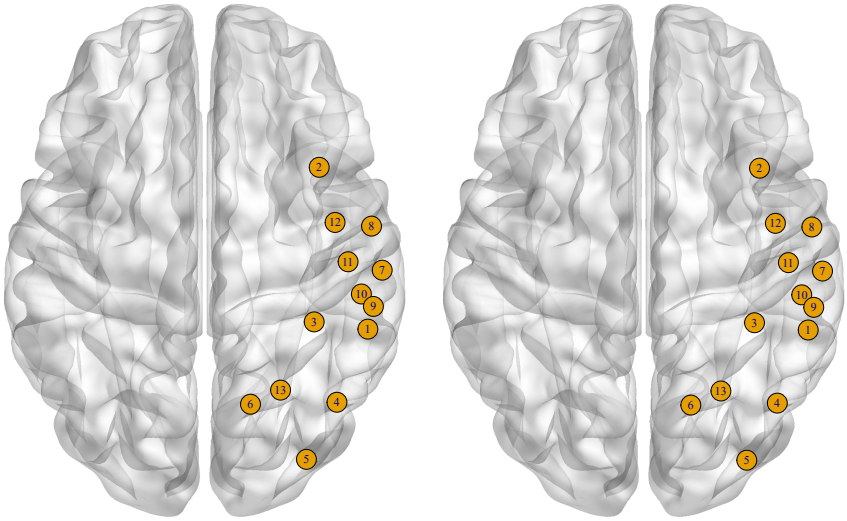


(a)

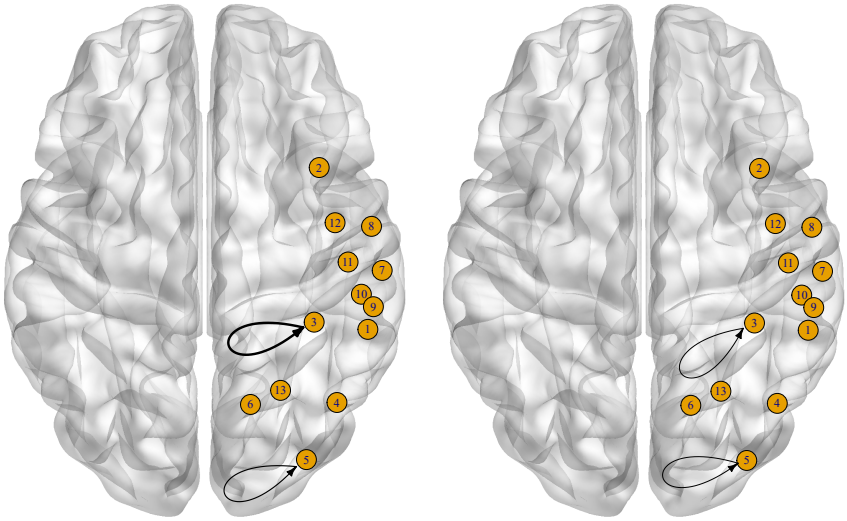


(b)

Figure E.3: Granger causality plots of the autoregressive coefficient matrices obtained using the graphicalVAR package overlaid on glass brains showing the locations of the regions, for the VAR process of individual C in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.



(a)



(b)

Figure E.4: Granger causality plots of the autoregressive coefficient matrices obtained using the graphicalVAR package overlaid on glass brains showing the locations of the regions, for the VAR process of individual D in (a) the beta band at lag 1 (left) and lag 2 (right), and (b) the delta band at lag 1 (left) and lag 2 (right). The region names for each individual are detailed in Table 3.1.

Bibliography

Akaike, H. 1974 A new look at the statistical model identification. *IEEE Transactions on Automatic Control* **19** (6), 716–723.

Andrieu, C. & Thoms, J. 2008 A tutorial on adaptive MCMC. *Statistics and computing* **18**, 343–373.

Ansley, C. F. & Kohn, R. 1986 A note on reparameterizing a vector autoregressive moving average model to enforce stationarity. *Journal of Statistical Computation and Simulation* **24** (2), 99–106.

Arnold, V. 1989 *Mathematical Methods of Classical Mechanics*, 2nd edn.

Atchadé, Y. F. & Rosenthal, J. S. 2005 On adaptive Markov chain Mhaonte Carlo algorithms. *Bernoulli* **11** (5), 815–828.

Barndorff-Nielsen, O. & Schou, G. 1973 On the parametrization of autoregressive models by partial autocorrelations. *Journal of Multivariate Analysis* **3** (4), 408–419.

Barnett, G., Kohn, R. & Sheather, S. 1996 Bayesian estimation of an autoregressive model using Markov chain Monte Carlo. *Journal of Econometrics* **74**, 237–254.

Bhattacharya, A. & Dunson, D. B. 2011 Sparse Bayesian infinite factor models. *Biometrika* **98** (2), 291–306.

Binks, R. L., Heaps, S. E., Panagiotopoulou, M., Wang, Y. & Wilkinson, D. J. 2024 Bayesian inference on the order of stationary vector autoregressions. *Bayesian Analysis* **1** (1), 1–22.

Box, G. & Jenkins, G. M. 1976 *Time Series Analysis: Forecasting and Control*. Holden-Day.

Brooks, S. P., Giudici, P. & Roberts, G. O. 2003 Efficient construction of reversible jump Markov chain Monte Carlo proposal distributions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **65** (1), 3–39.

Buse, A. 1982 The likelihood ratio, Wald, and Lagrange multiplier tests: An expository note. *The American Statistician* **36** (3), 153–157.

Canova, F. 2007 *VAR Models*, pp. 111–164. Princeton University Press.

Carpenter, B., Gelman, A., Hoffman, M., Lee, D., Goodrich, B., Betancourt, M., Brubaker, M., Guo, J., Li, P. & Riddell, A. 2017 Stan : A probabilistic programming language. *Journal of Statistical Software* **76**.

Carriero, A., Clark, T. E. & Marcellino, M. 2015 Bayesian vars: specification choices and forecast accuracy. *Journal of Applied Econometrics* **30** (1), 46–73.

Chiang, S., Guindani, M., Yeh, H. J., Haneef, Z., Stern, J. M. & Vannucci, M. 2016 Bayesian vector autoregressive model for multi-subject effective connectivity inference using multi-modal neuroimaging data. *Human Brain Mapping* **38** (3), 1311–1332.

Chiu, C.-A., Lu, M.-C., Zhong, Y.-L., Tsai, T.-Y., Liu, C.-J. & Ho, M.-C. 2023 Quantifying and analyzing brainwave electroencephalography with welch's method. *Sensors and Materials* **35** (5), 1579–1586.

Congdon, P. 2006 *Bayesian statistical modelling*, 2nd edn. Chichester, England ; Hoboken, NJ: John Wiley & Sons.

Desikan, R. S., Ségonne, F., Fischl, B., Quinn, B. T., Dickerson, B. C., Blacker, D., Buckner, R. L., Dale, A. M., Maguire, R. P., Hyman, B. T., Albert, M. S. & Killiany, R. J. 2006 An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *NeuroImage* **31** (3), 968–980.

Duane, S., Kennedy, A., Pendleton, B. J. & Roweth, D. 1987 Hybrid Monte carlo. *Physics Letters B* **195** (2), 216–222.

Durante, D. 2017 A note on the multiplicative gamma process. *Statistics & Probability Letters* **122**, 198–204.

Epskamp, S. 2024 *graphicalVAR: Graphical VAR for Experience Sampling Data*. R package version 0.3.4.

Fan, J., Sitek, K., Chandrasekaran, B. & Sarkar, A. 2022 Bayesian tensor factorized vector autoregressive models for inferring Granger causality patterns from high-dimensional multi-subject panel neuroimaging data. *arXiv:2206.10757* .

Francq, C. & Zakoian, J. M. 2001 Stationarity of multivariate Markov–switching ARMA models. *Journal of Econometrics* **102** (2), 339–364.

Frühwirth-Schnatter, S. 2006 *Finite mixture and Markov switching models*. New York: Springer.

Gabry, J. & Cesnovar, R. 2021 *cmdstanr: R Interface to 'CmdStan'*. https://mc-stan.org/cmdstanr.

Gelfand, A. E. & Smith, A. F. 1990 Sampling-based approaches to calculating marginal densities. *Journal of the American statistical association* **85** (410), 398–409.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A. & Donald, R. 2014 *Bayesian data analysis*, 3rd edn.

Gelman, A. & Rubin, D. B. 1992 Inference from iterative simulation using multiple sequences. *Statistical Science* **7** (4), 457–472, publisher: Institute of Mathematical Statistics.

Geman, S. & Geman, D. 1984 Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on pattern analysis and machine intelligence* (6), 721–741.

Goh, G. H., Maloney, S. K., Mark, P. J. & Blache, D. 2019 Episodic ultradian events—ultradian rhythms. *Biology* **8** (1), 15.

Goodman, J. & Weare, J. 2010 Ensemble samplers with affine invariance. *Communications in applied mathematics and computational science* **5** (1), 65–80.

Goyal, A. & Garg, R. 2020 Effective eeg connectivity by sparse vector autoregressive model. In *Proceedings of the 7th ACM IKDD CoDS and 25th COMAD*, p. 37–45. New York, NY, USA: Association for Computing Machinery.

Granger, C. W. 1969 Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: journal of the Econometric Society* pp. 424–438.

Haario, H., Saksman, E. & Tamminen, J. 2001 An adaptive Metropolis algorithm. *Bernoulli* **7** (2), 223–242, publisher: Bernoulli Society for Mathematical Statistics and Probability.

Han, C., Phillips, P. C. & Sul, D. 2017 Lag length selection in panel autoregression. *Econometric Reviews* **36** (1-3), 225–240.

Hannaford, N. E., Heaps, S. E., Nye, T. M., Curtis, T. P., Allen, B., Golightly, A. & Wilkinson, D. J. 2023 A sparse Bayesian hierarchical vector autoregressive model for microbial dynamics in a wastewater treatment plant. *Computational Statistics & Data Analysis* **179**, 107659.

Hastings, W. K. 1970 Monte Carlo sampling methods using Markov chains and their applications. *Biometrika* **57** (1), 97–109.

Hayashi, M., Sato, K. & Hori, T. 1994 Ultradian rhythms in task performance, self-evaluation, and EEG activity. *Perceptual and Motor Skills* **79** (2), 791–800.

Heaps, S. E. 2023 Enforcing stationarity through the prior in vector autoregressions. *Journal of Computational and Graphical Statistics* **32** (1), 74–83.

Heaps, S. E., Farrow, M. & Wilson, K. J. 2020 Identifying the effect of public holidays on daily demand for gas. *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **183** (2), 471–492.

Heaps, S. E. & Jermyn, I. H. 2023 Structured prior distributions for the covariance matrix in latent factor models ArXiv:2208.07831.

Herrera, R., Sun, M., Dahl, R., Ryan, N. & Sclabassi, R. 1997 Vector autoregressive model selection in multichannel eeg. In *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society.'Magnificent Milestones and Emerging Opportunities in Medical Engineering'(Cat. No. 97CH36136)*, , vol. 3, pp. 1211–1214. IEEE.

Hoffman, M. D., Gelman, A. *et al.* 2014 The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.* **15** (1), 1593–1623.

Huerta, G. & Prado, R. 2006 Structured priors for multivariate time series. *Journal of Statistical Planning and Inference* **136** (11), 3802–3821.

Huerta, G. & West, M. 1999 Priors and component structures in autoregressive time series models. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* **61** (4), 881–899.

Hurvich, C. M. & Tsai, C.-L. 1993 A corrected akaike information criterion for vector autoregressive model selection. *Journal of time series analysis* **14** (3), 271–279.

Ishwaran, H. & James, L. F. 2001 Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association* **96** (453), 161–173.

Jiang, X., Hu, X., Xu, W., Li, G. & Wang, Y. 2013 Inference of microbial interactions from time series data using vector autoregression model. In *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pp. 82–85.

Kalli, M. & Griffin, J. E. 2018 Bayesian nonparametric vector autoregressive models. *Journal of Econometrics* **203** (2), 267–282.

Kilian, L. & Ivanov, V. 2001 A practitioner's guide to lag-order selection for vector autoregressions. *Tech. Rep.*. CEPR Discussion Papers.

Koop, G. & Korobilis, D. 2010 Bayesian multivariate time series methods for empirical macroeconomics. *Foundations and Trends in Econometrics* **3** (4), 267–358.

Kuo, L. & Mallick, B. 1998 Variable selection for regression models. *Sankhyā: The Indian Journal of Statistics, Series B (1960-2002)* **60** (1), 65–81, publisher: Springer.

Legramanti, S., Durante, D. & Dunson, D. B. 2020 Bayesian cumulative shrinkage for infinite factorizations. *Biometrika* **107** (3), 745–752.

Lloyd, D. & Stupfel, M. 1991 The occurrence and functions of ultradian rhythms. *Biological Reviews of the Cambridge Philosophical Society* **66** (3), 275–299.

Luetkepohl, H. 2005 *The New Introduction to Multiple Time Series Analysis*.

Malinovskaia, A. 2022 Mixed effects spectral vector autoregressive model: With application to brain connectivity. *arXiv preprint arXiv:2210.03017* .

Manomaisaowapak, P., Nartkulpat, A. & Songsiri, J. 2022 Granger Causality inference in EEG source connectivity analysis: A state-space approach. *IEEE Transactions on Neural Networks and Learning Systems* **33** (7), 3146–3156.

Marriott, J., Ravishanker, N. & Gelfand, A. E. 1996 Bayesian analysis of ARMA processes: complete sampling based inference under full likelihoods .

Monahan, J. F. 1983 Fully bayesian analysis of arma time series models. *Journal of Econometrics* **21** (3), 307–331.

Neal, R. M. 2011 MCMC using Hamiltonian dynamics. In *Handbook of Markov Chain Monte Carlo* (ed. S. Brooks, A. Gelman, G. Jones & X.-L. Meng), pp. 113–162. Chapman & Hall/CRC.

Nielsen, B. 2006 Order determination in general vector autoregressions. *Lecture Notes-Monograph Series* pp. 93–112.

Panagiotopoulou, M., Papasavvas, C. A., Schroeder, G. M., Thomas, R. H., Taylor, P. N. & Wang, Y. 2022 Fluctuations in EEG band power at subject-specific timescales over minutes to days explain changes in seizure evolutions. *Human Brain Mapping* **43** (8), 2460–2477.

Plummer, M., Best, N., Cowles, K. & Vines, K. 2006 CODA: Convergence diagnosis and output analysis for MCMC. *R News* **6** (1), 7–11.

Prado, R. 1998 Latent structure in non-stationary time series. PhD thesis, Duke University.

Prado, R. & Huerta, G. 2002 Time-varying autoregressions with model order uncertainty. *Journal of Time Series Analysis* **23** (5), 599–618.

Prado, R. & West, M. 1997 Exploratory modelling of multiple non-stationary time series: Latent process structure and decompositions. In *Modelling longitudinal and spatially correlated data*, pp. 349–361. Springer.

Richardson, S. & Green, P. J. 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, _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/1467-9868.00095.

Roberts, G. O. & Rosenthal, J. S. 2009 Examples of adaptive MCMC. *Journal of computational and graphical statistics* **18** (2), 349–367.

Rousseau, J. & Mengersen, K. 2011 Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **73** (5), 689–710.

Schwarz, G. 1978 Estimating the dimension of a model. *The Annals of Statistics* **6** (2), 461 – 464.

Sen, D., Mishra, B. B. & Pattnaik, P. K. 2023 A review of the filtering techniques used in EEG signal processing. In *2023 7th International Conference on Trends in Electronics and Informatics (ICOEI)*, pp. 270–277.

Sethuraman, J. 1994 A Constructive Definition of Dirichlet Priors. *Statistica Sinica* **4** (2), 639–650, publisher: Institute of Statistical Science, Academia Sinica.

Shumway, R. H. & Stoffer, D. S. 2017 *Time Series Analysis and Its Applications With R Examples*, 4th edn.

Sims, C. A. 1980 Macroeconomics and reality. *Econometrica* **48** (1), 1–48.

Stan Development Team 2024 Stan modeling language users guide and reference manual, version 2.34.

Stelzer, R. 2009 On Markov-switching ARMA processes—stationarity, existence of moments, and geometric ergodicity. *Econometric Theory* **25** (1), 43–62.

Tibdewal, M. N., Mahadevappa, M., Ray, A. K., Malokar, M. & Dey, H. R. 2016 Power line and ocular artifact denoising from EEG using notch filter and wavelet transform. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, pp. 1654–1659.

Vehtari, A., Gelman, A., Simpson, D., Carpenter, B. & Bürkner, P.-C. 2021 Rank-normalization, folding, and localization: An improved R^ for assessing convergence of MCMC (with discussion). *Bayesian Analysis* **16** (2).

Vermaak, J., Andrieu, C., Doucet, A. & Godsill, S. J. 2004 Reversible jump Markov chain Monte Carlo strategies for Bayesian model selection in autoregressive processes. *Journal of Time Series Analysis* **25** (6), 785–809.

Wang, Y., Schroeder, G. M., Horsley, J. J., Panagiotopoulou, M., Chowdhury, F. A., Diehl, B., Duncan, J. S., McEvoy, A. W., Miserocchi, A., de Tisi, J. & Taylor, P. N. 2023 Temporal stability of intracranial electroencephalographic abnormality maps for localizing epileptogenic tissue. *Epilepsia* **00**, 1–11.

West, M. & Harrison, J. 1997 *Bayesian Forecasting and Dynamic Models*, 2nd edn.

Zhang, W., Cribben, I., Petrone, S. & Guindani, M. 2021 Bayesian time-varying tensor vector autoregressive models for dynamic effective connectivity. *arXiv:2106.14083* .