Bayesian Computation with Application to Spatial Models and Neuroimaging

by

Ming Teng

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Biostatistics)
in The University of Michigan
2017

Doctoral Committee:

Professor Timothy D. Johnson, Co-chair
Associate Professor Farouk S. Nathoo, Co-chair, University of Victoria
Professor Stephen F. Taylor
Research Assistant Professor Lili Zhao
Associate Professor Sebastian Zöllner

Ming Teng
tengming@umich.edu
ORCID iD: 0000-0001-9921-8760

To my parents

ACKNOWLEDGEMENTS

First of all, I want to express my deep and sincere gratitude for my advisors Dr. Timothy D. Johnson and Dr. Farouk S. Nathoo. Dr. Johnson has introduced me to the fascinating world of statistics and continuously supported me all through these years. His patience, encouragement and outstanding guidance are extremely important for me to overcome many difficulties in my research. And I learned a lot from his broad knowledge of Bayesian statistics and computation. Dr. Nathoo has led me into the kingdom of Variational Bayes, and offered me excellent mentorship with his expertise in Bayesian methodologies and neuroimaging. To me, he is also like an elder friend, who helps and cares about many aspects of my life. I'm truly grateful for what I got from these two advisors.

I'd also like to express my sincere thanks to the committee members. I'd like to thank Dr. Sebastian Zöllner and Dr. Lili Zhao for spending extra time with me and offering their valuable suggestions. And I'd like to thank Dr. Stephan F. Taylor for his insightful opinions in the field of neuroimaging.

Thanks also goes to other members of our neuroimaging group: Dr. Jian Kang, Dr. Veronica Berrocal, Dr. Eunjee Lee, Cui Guo and Marco Benedetti. Thanks for your brilliant ideas and meaningful discussions.

I'd like to thank my friends and cousins. You have always supported me no matter when and where. I feel so happy to have you guys in my life.

Finally, I'd like to thank my beloved parents. You have given me such an unselfish love. In your eyes, I'm always the most precious one. Thus, I'll do my best to make you happy and proud. This dissertation is dedicated to you.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE

LIST OF TABLES

TABLE

LIST OF APPENDICES

APPENDIX

ABSTRACT

Analysis of Neuroimaging data has experienced great strides over the last few decades. Two key aspects of Neuroimaging data are its high-dimensionality and complex spatio-temporal autocorrelation. Classical approaches are somewhat limited in dealing with these two issues, as a result, Bayesian approaches are being utilized more frequently due to their flexibility. Despite their flexibility, there are several challenges for Bayesian approaches with respect to the required computation. First, the need for an efficient posterior computation method is paramount. Second, even in conjugate models, statistical accuracy in Bayesian computation may be hard to achieve. Since accuracy is of primary concern when studying the human brain, a careful and innovative exploration of Bayesian models and computation is necessary.

In this dissertation, we address some of these issues by looking at various Bayesian computational algorithms in terms of both accuracy and speed in the context of Neuroimaging data. The algorithms we study are the Hamiltonian Monte Carlo (HMC), Variational Bayes (VB), and integrated nested Laplace approximation (INLA) algorithms. HMC is a MCMC method that's particularly powerful for sampling in high-dimensional space with highly correlated parameters. It's robust and accurate, yet not as fast as some approximate Bayesian methods, for example, Variational Bayes (VB). However, since there is no theoretical guarantee that the resulting posterior derived from VB is accurate, its performance has to be analyzed on a case-by-case basis. INLA is another extremely fast method based on numerical integration with Laplace approximations but, like VB, there are no generally applicable theoretical guarantees of accuracy.

In Chapter II we focus on a particular spatial point process model, namely the log Gaussian Cox Process (LGCP), and consider applications to ecological and neuroimaging data. Inference for the LGCP is challenging due to its non-conjugacy and doubly stochastic property. We develop HMC and VB algorithms for the LGCP model and make comparisons with INLA. In Chapter III, we turn our focus to the general linear model with autoregressive errors (GLM-AR) which is widely used in analyzing fMRI single subject data. We derive an HMC algorithm and compare it with the VB algorithm and the mass univariate approach using the Statistical Parametric Mapping (SPM) software program. In Chapter IV, we extend the original GLM-AR model to a new model where the order of the AR coefficients can varying spatially across the brain and call it GLM with spatially varying autoregressive orders (SVARO). Using simulations and real data we compare our SVARO model with GLM-AR model implemented under both our MCMC sampler and the SPM VB algorithm.

Our results shed light on several important issues. While HMC almost always yields the most accurate results, the performance of VB is strongly model specific. INLA is a fast alternative to MCMC methods but we observe some limitations when examining its accuracy in certain settings. Furthermore, our new SVARO model performs better than the GLM-AR model in a number of ways. Not surprisingly, more accurate algorithms generally require more computational time. By systematically evaluating the pros and cons of each method, we believe our work to be practically useful for those researchers considering the use of these methods.

CHAPTER I

Introduction

We live in a $3D$ world. Our brain helps us to store and analyze the instantaneous events that happen all round us. Yet the power of an individual brain is limited. Spatial statistics, as a tool, can help us make use of these huge amounts of data and information. Through spatial or spatiotemporal models, we can analyze not only the events worldwide, but also in particular, our brain themselves, namely with neuroimaging. Thus, spatial modeling is a fascinating field and has attracted generations of statisticians for exploration.

One of the biggest characteristics of spatial data is its high-dimensionality in nature. For example, current technologies in neuroimaging, in particular functional magnetic resonance imaging (fMRI), allow us to acquire images at units of millimeters in space and seconds in time. This leads to over $50,000$ voxels, or voxel elements, for one single subject at a single time point. Each fMRI study lasting for several hundred seconds resulting in thousands of time series of length in the several hundreds. Typically, these data will have both spatial and temporal correlation. This high-dimensional problem is challenging from both a statistical and computational standpoint. Classical statistical approaches rely on models with strict assumptions that are not flexible enough to deal with the complex correlation found in these images. As a result, researchers have turned to Bayesian methods. Bayesian methods are quite flexible and allow modelling of the spatiotemporal correla-

tion found in these images. Nevertheless, traditionally, Bayesian methods tend to be very computationally intense. Thus the focus of this dissertation is on Bayesian computational methods for spatial and spatiotemporal models with applications to point processes and neuroimaging.

The remainder of this introduction is organized as follows: in Section 1.1, we introduce spatial point process models. Then in Section 1.2, we cover the topic of neuroimaging, with focus on magnetic resonance imaging (MRI) and fMRI. A brief review of existing Bayesian computational approaches are given in Section 1.3. Finally in Section 1.4, we provide an outline of this dissertation.

1.1 Spatial point processes

There are basically three types of spatial data (Banerjee et al., 2014). 1) Point reference data, where the spatial domain is assumed fixed; 2) areal data, where the spatial domain is fixed and partitioned into separate areal units, and 3) point pattern data, where the location itself is random. An example of point pattern data is trees in a rain forest, where each point represents the location of a particular tree. And we want to study the spatial clustering of these trees, in which case the location of the trees are considered random and of primary interest. Due to the randomness of the location of points, special models have to be developed to analyze point pattern data, and these are typically referred to as spatial point process models. Spatial point processes refer to a random pattern of countable points, say S, in a d dimensional Euclidean space $\mathcal{R}^d$ (Møller and Waagepetersen, 2003).

There are various types of spatial point processes. Among them, and perhaps the most fundamental, is the spatial Poisson point process. A point process is called a homogeneous Poisson point process on a spatial domain S with constant intensity λ if 1) for any $B \subseteq S$, $N(B)$ is Poisson distributed with mean $\mu(B) = \int_S \lambda(s)ds.$ and ii) conditional on $N(B)$,

the points are i.i.d distributed with density $\lambda(\cdot)/\mu(B)$ (Møller and Waagepetersen, 2003). The Poisson point process has many wide applications. For example, Wang et al. (2016) used it to model traffic conditions, where he assumes the arrival of cars has a homogeneous intensity. Kolmogorov also used it to study the formulation of crystals in metals (Chiu et al., 2013).

However, in many applications, the point pattern is such that the intensity function is not constant. This naturally brings up the question as whether one can model this inhomogeneity in the point pattern. Given that the non-constant intensity, $\lambda(s)$, is a known function, the process is called an inhomogeneous Poisson point process. When the intensity function is not known, we assume that the intensity function itself is a random field, or random spatial process. Perhaps the most widely used model is the Cox process (Cox and Isham, 1980). A Cox process is a "doubly stochastic" process in which not only the spatial point pattern is random, but the underlying intensity is random as well. Taking this one step further, if we assume the log of the intensity is a Gaussian Random Field (GRF, i.e. the generalization of Gaussian Process to multi-dimensional space), this leads to a log-Gaussian Cox process (LGCP) (Møller et al., 1998). Due to its mathematical convenience and many other appealing properties of the Gaussian process, log-Gaussian Cox processes have received a lot of attention in applications, ranging from ecology, disease mapping, brain imaging and finance (Basu and Dassios, 2002).

Despite its popularity, the intensity of a LGCP model is not easy to estimate in either the classical or Bayesian setting. This is due to two reasons: it is doubly stochastic, and the non-conjugacy resulting from a combination of a Poisson process and a Gaussian process. Although this hierarchical structure of a LGCP and the nonparametric nature of a GRF has given Bayesian methodology a natural advantage over classical statistical methods, it is still full of challenges. MCMC based samplers are typically too time consuming,

while deterministic based Bayesian approximations have not been fully explored and systematically verified in terms of accuracy for this model. Although one paper (Taylor and Diggle, 2013) has recently compared two Bayesian algorithms, there is still a need for a more broad comparison of the most up-to-date computational approaches. This need is the motivation for Chapter II, where we develop a Hamiltonian Monte Carlo (HMC) algorithm (Neal, 2011) and a variational Bayes (VB) algorithm (MacKay, 1997) for the LGCP and compare the statistical and computational efficiency of these two algorithms with the integrated nested Laplacian algorithm (INLA) (Rue et al., 2009).

1.2 Neuroimaging

Neuroimaging is a relatively young discipline that arose in the 20th century (Filler, 2010). Modern neuroimaging platforms include positron emission tomography (PET), magnetoencephalography (MEG), electroencephalogram (EEG), MRI, and fMRI, to name a few. In this dissertation, I only use data from MRI and fMRI.

MRI images internal structures of the body. In particular it is excellent at differentiating between structures that contain mostly fat or mostly water, that is, soft tissues. MRI relies on the complex physical property of atoms called nuclear magnetic moments and are outside the scope of this dissertation. We refer the interested reader to (Lazar, 2008).

On the other hand, task-based fMRI measures functional activity of the brain. It does so by measuring the blood oxygen level dependent (BOLD) signal. (Ogawa et al., 1990). The BOLD signal is a surrogate for neuronal activity. The interested reader is referred to (Lindquist et al., 2008).

Having acquired fMRI data, the neuroscientist is often interested in a statistical analysis of the data. Given task-based fMRI data, a primary goal is to determine which voxels and/or regions of the brain are activated during the task. Determination of brain activation

is traditionally obtained by statistical inference using the general linear model (Friston et al., 1994)

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{E} \tag{1.1}$$

Here $\mathbf{Y}$ is the response image, $\mathbf{X}$ is the task-related design matrix, $\mathbf{W}$ is the vector regression coefficients. In the Bayesian context, $\mathbf{W}$ is often assumed to arise from a sparse random field prior to account for spatial correlation (Penny et al., 2005). The last term, $\mathbf{E}$, is the vector of errors, and it is typical to assume these follow a multivariate normal distribution with a diagonal covariance matrix and common variance σ^2. However, with different sources of temporal noise and autocorrelation among successive time points, this ideal assumption is never achieved. Many researchers have proposed different ways to solve this issue. Friston et al. (1995) and Worsley and Friston (1995) proposed to pre-smooth the data, also known as "coloring", and correct for the autocorrelation by adjusting the degrees of freedom in the error term. Bullmore et al. (1996) proposed to de-correlated the data, namely "prewhitening", by using an autoregressive (AR) model as estimates of the error term. In the Bayesian framework, Penny et al. (2003, 2005) put an AR(p) prior on the error term, and later extend it to be AR process where the regression coefficients are allowed to vary spatially (Penny et al., 2007). The resulting model is therefore termed as "GLM-AR" model. Based on a comparison of model evidence for different orders, they claimed that a low order AR process (1-3) suffices as the optimal order is 1 in most of the voxels (Penny et al., 2003). This modelling assumption for the error term remained unchanged in various forms of later work (Woolrich et al., 2004b; Makni et al., 2006; Bezener et al., 2016). However, no one has investigated whether such an assumption is reasonable based on model accuracy and statistical efficiency. In fact, as we illustrate in Chapter IV, this assumption may not be optimal. Thus, we relax this assumption and extend the homogeneous low order AR assumption to an heterogeneous

high order AR process, allowing the AR order to vary spatial by adopting an underlying Ising prior. We call this model the general linear model with spatially-varying autoregressive orders (GLM-SVARO). A more detailed review of temporal modelling, as well as our extension is discussed in Chapter IV.

1.3 Bayesian computation

As one can see in Section 1.1 and 1.2, the excessive size of data, together with complex spatiotemporal correlation has put a great demand for highly efficient Bayesian computational algorithms. A brief review of today's Bayesian paradigm suggests that we can split these algorithms into two parts: fully Bayesian methods based on simulated Markov chain Monte Carlo samples, and approximate Bayesian methods that approximate the posterior distribution deterministically.

Fully Bayesian methods rely mainly on MCMC algorithms to sample from the posterior distribution of the model parameters. Gibbs sampling is probably the most widely used, assuming that the posterior can be sampled from all of its full conditional distributions. When the underlying distribution is not easy to sample from, Metropolis Hastings (MH), or MH within Gibss are natural choices since they are universally applicable. Neal (2003) proposed slice sampling. By introducing an auxiliary variable and sampling from a "slice" of the distribution, it not only avoids sampling directly from a potentially complex distribution, but also results in more efficient draws than random walk MH. Based on this Murray et al. (2010) proposed the elliptical slice sampling method where the step size of the proposal of a MH ((Bernardo et al., 1998)) is sampled using the slice sampler.

Since the above methods do not rely on evaluating the gradient of the log posterior, we refer to them as gradient-free samplers. There is another class of MCMC samplers that rely on evaluating the gradient of the log posterior to build more efficient proposals.

The Metropolis adjusted Langevin algorithm (MALA) (Roberts and Rosenthal, 1998) is a representative one. The proposal of MALA is drawn from Langevin dynamics, which makes use of the gradient of the log posterior distribution. The result is a more efficient sampler than common MH algorithms. More broadly, MALA is a special case of the Hybrid (or Hamiltonian) Monte Carlo sampler (HMC) (Neal, 2011). In HMC, the proposal of the move is taken by solving the Hamiltonian equations using numerical procedures. Thus, the movement is in direction of the gradient of the log posterior, thus avoiding the much of the random walk behavior by common MH algorithms.

Different from MCMC based methods, approximate Bayesian methods are relatively fast, and typically require fewer iterations. This gain in computational speed comes at a price: lack of statistical efficiency and lack of theoretical convergence guarantees. Variational Bayes (VB) (Jordan et al., 1999; Neal and Hinton, 1998) is such a method. In VB, the marginal likelihood is approximated using a family of distributions that is often more easy to handle. Then the Kullback-Leibler (KL) divergence between the true and approximate distribution is minimized using a series of optimization techniques. Most often, a mean field approximation (Bishop, 2006) assumption will be used to simplify the distribution for tractable inference. Another similar approach, named "Expectation Propagation" (EP) (Minka, 2001), tries to minimize the pseudo-distance between the approximate distribution and the true distribution.

More recently, Rue et al. (2009) proposed the integrated nested Laplace approximation (INLA) for latent Gaussian models. INLA is based on the numerical integration of Laplace approximations made to the latent field as well as to its hyper-parameters. It is based on sparse Gaussian Markov random fields and focuses on the marginal posterior distributions of the parameters instead of the joint posterior distribution.

Since fully Bayesian methods and approximate Bayesian methods both have their ad-

vantages as well as limitations, it is of importance to compare their performance with respect to large spatiotemporal data set such as those obtained via neuroimaging. This is a key focus of this dissertation.

1.4 Dissertation outline

The outline of this dissertation is as follows. In Chapter II, we develop a fast HMC algorithm and a mean-field approximation VB algorithm for estimating LGCPs and compare these two algorithms with two INLA algorithms. Motivated by these results, we turn our attention to fMRI data and derive a HMC algorithm for the widely used GLM-AR model(Penny et al., 2007) in Chapter III. The results are compared with those from VB and the mass univariate approach of the Statistical Parametric Mapping software (SPM, Ashburner et al. (2014)). In Chapter IV, we extend the original GLM-AR model to a new GLM-SVARO model. In the GLM-AR model, each time-series in the fMRI data are assumed to follow an autoregressive model of low order (1-3) and the same order is assumed across all time-series. In our extension of this model (GLM-SVARO), we relax both of these assumptions. The results are compared with the GLM-AR model using both VB (via an SPM software add-on) as well as MCMC to sample from the posterior distribution. Finally in the last chapter, we make conclusions and also discuss future directions worth exploring.

CHAPTER II

Bayesian Computation for Log-Gaussian Cox Processes: A Comparative Analysis of Methods

2.1 Introduction

Spatial point process models for point pattern data have many applications including those involving ecology, geology, seismology, and neuroimaging. The theory of point processes has its roots with the work of Poisson in the 19th century, while more modern treatments with a statistical focus include Daley and Vere-Jones (1988), Møller et al. (1998) and Illian et al. (2008). Among models for a spatial point process, the homogeneous Poisson process is the most fundamental but its use is limited in many applications due to its simplistic nature. A related but more flexible process is the Log-Gaussian Cox Process (LGCP), a process that is obtained by assuming a hierarchical structure, where at the first level the process is assumed Poisson conditional on the intensity function, and at the second level the log of the intensity function is assumed to be drawn from a Gaussian process. The flexibility of the model arises from the Gaussian process prior specified over the log-intensity function. Given this hierarchical structure with a Gaussian process at the second level, fitting this model to observed spatial point pattern data is a computational challenge (Murray et al., 2012).

A number of approaches have been developed to estimate the LGCP model in both the classical and Bayesian frameworks. In Diggle (1985) the authors propose an adaption of

Rosenblatt's kernel method (Rosenblatt et al., 1956) for the purpose of non-parametric estimation and then derive an expression for the mean squared error based on a stationarity assumption. A Bayesian framework is considered in Møller et al. (1998) where the authors propose the use of the Metropolis-Adjusted Langevin Algorithm (MALA) (Besag, 1994) for Monte Carlo sampling of the posterior distribution. These authors also introduce a discretization of the spatial domain in order to attain computational tractability. Adams et al. (2009) proposes an exact estimation method to deal with a modification of such a point process which they term the sigmoidal Gaussian Cox process. To avoid discretization of the spatial domain, Gonçalves and Gamerman (2015) proposed a Markov chain Monte Carlo algorithm that samples from the joint posterior distribution of the LGCP model. A particular choice of the dominating measure is used to obtain the likelihood function without discretization and this is shown to be crucial to devise a valid MCMC algorithm. Although avoiding the error from discretization, the authors describe potential identifiability problems that require careful choice of prior distributions. In addition the computational complexity of the proposed MCMC algorithm could limit the application of this approach when considering very large data sets.

With regards to MCMC algorithms for Bayesian approaches, MALA is a special case of the potentially more efficient Hamiltonian Monte Carlo (HMC) (Neal, 1995) algorithm that uses the notion of Hamiltonian dynamics to construct proposals for the Metropolis-Hastings algorithm and has been adopted recently for an increasing number of applications (Neal, 2011). Extending further, the Riemann Manifold Hamiltonian Monte Carlo algorithm proposed by Girolami and Calderhead (2011) is a generalization of both MALA and HMC that can lead to more efficient posterior sampling in some cases. One drawback of Riemann Manifold HMC is that it requires the inversion of a potentially large matrix (the expected Fisher information matrix plus the negative Hessian of the log prior) at each

iteration and this is not computationally feasible for very high-dimensional problems such as those typically involving the LGCP model.

An advantage associated with the use of MCMC algorithms for Bayesian computation is the underlying theory which guarantees simulation consistent estimation of various important characteristics of the posterior distribution. Thus the practitioner is assured of an accurate Monte Carlo representation of the posterior distribution given a sufficient amount of sampling effort. A drawback is that MCMC can be computationally intense, and this has motivated several alternative deterministic approaches for approximate Bayesian inference.

One such approach is Variational Bayes (VB) (MacKay, 1997), where the approximation to the posterior distribution is assumed to lie within some convenient family of distributions and then an optimization is carried out to minimize the Kullback-Leibler divergence measuring the discrepancy between the true posterior and the approximation. Often the family of distributions within which the approximation is assumed to lie is based on the notion of a mean field approximation, which corresponds to assuming posterior independence between certain model parameters. The idea in this case is to replace stochastic posterior dependence between parameters with deterministic dependence between the posterior moments in a manner that minimizes Kullback-Leibler divergence. Variational Bayes approximations have been applied successfully to the analysis of hidden Markov models in MacKay (1997) and to other mixture models Humphreys and Titterington (2000). Zammit-Mangion et al. (2012) used Variational Bayes for models of spatiotemporal systems represented by linear stochastic differential equations and demonstrated quick and efficient approximate inference both for continuous observations and point process data.

Mean field Variational Bayes is well suited for dealing with models within the conjugate exponential family where closed form solutions for the iterative steps of the optimiza-

tion algorithm are available. In general such closed form solutions may not be available and additional approximations are then required. This is the case with the LGCP model. Mean field Variational Bayes approximations for non-conjugate models can be obtained by incorporating further approximations based on the delta method or the Laplace method Wang and Blei (2013). These approximations are successfully applied to a correlated topic model and a Bayesian logistic regression model in Wang and Blei (2013), and to robust Bayesian models in Wang and Blei (2015). For the LGCP model, tractable variational approximations can be obtained following this approach, where a mean field approximation with further approximations based on the Laplace method are used to handle the non-conjugate structure of the model. The variational approach considered in this paper can be thought of as an application of the techniques in Wang and Blei (2013) to the case of the LGCP model.

In Nguyen and Bonilla (2014) the authors approximate the posterior distribution of models incorporating a latent Gaussian process (which includes the LGCP model) using a mixture of Gaussian distributions, and derive a fixed-form variational approach for implementing this approximation. However, the proposed approach appears limited to one-dimensional point processes and is not applied to spatial point processes which is the focus of our work. Lloyd et al. (2014) proposed a variational Bayes approach for fitting a Cox process assuming that the intensity function is the square of a Gaussian process. Unfortunately, this variational approach cannot be applied for estimation of the LGCP model, where the intensity is assumed to be the exponential of a Gaussian process, as the update steps for computing the proposed variational approximation to the posterior then become intractable.

Variational Bayes approximations can work well in some settings and the corresponding approximations can be computed relatively fast. A drawback is that there is no underly-

ing theory guaranteeing the accuracy of the approximation or characterizing its error, thus these approximations need to be evaluated on a case-by-case basis, and they may or may not achieve reasonable accuracy depending on the utility of the practitioner. One contribution of this paper is the derivation of a mean field VB approximation which incorporates the Laplace method for the LGCP model. As far as we are aware such approximations have not been considered previously for this model. Another contribution of this paper is to compare this VB approximation with HMC in terms of both statistical and computational efficiency.

An alternative approach for approximate Bayesian inference that has gained tremendous popularity in the statistical literature is the integrated nested Laplace approximation (INLA) (Rue et al., 2009). INLA is less generally applicable than MCMC or VB as it assumes the model has a latent Gaussian structure with only a moderate number of hyperparameters. For spatial modeling the approach makes use of the Gaussian Markov Random field (Rue and Held, 2005) and corresponding approximations which are known to be computationally efficient. The basis of INLA is the use of the Laplace approximation and numerical integration with latent Gaussian models to derive approximate posterior marginal distributions. INLA does not produce an approximation to the joint posterior which is a drawback of the approach in settings where the joint posterior (as opposed to the marginals) is of interest.

For spatial models incorporating a Gaussian Random field (GRF) with a Matérn correlation structure, Lindgren et al. (2011) develop an approximate approach based on stochastic partial differential equations (SPDE) and these approximations have been combined for use with INLA. The essence of the approach is to specify a SPDE that has as its solution the GRF and then the SPDE representation is used in conjunction with basis representations to approximate the process over the vertices of a 2-dimensional mesh covering

the spatial domain. The value of the process at any location is then obtained based on interpolation of the values at the mesh vertices. In recent work, Simpson et al. (2012) evaluated this approximation applied to the LGCP model with spatially varying covariates and demonstrated adequate performance for the settings and data considered there.

A comparison between INLA and MALA for models incorporating GMRF approximations is considered in Taylor and Diggle (2013). In our work, we compare for the LGCP model Bayesian computation based on HMC, VB with a Laplace approximation, INLA, and INLA with the SPDE approximation. The comparisons we make are with respect to computational time, properties of estimators, posterior variability, and goodness fitness checking based on posterior predictive methods. Our objective is to provide practical guidance for users of the LGCP model. In addition to these comparisons, there are two novel aspects to the work presented here. First, we develop a mean field variational Bayes approximation that incorporates the Laplace method to deal with the non-conjugacy of the LGCP model. To the best of our knowledge this is the first time such an approximation has been developed for approximate Bayesian inference with the LGCP model. Second, we apply HMC for fully Bayesian inference and a novel aspect of our implementation is that HMC is used to update the decay (correlation parameter) associated with the latent Gaussian process. A result is that the sampling algorithm mixes very well and to our knowledge the development of the HMC algorithm in this context is the first of its kind.

The remainder of the paper proceeds as follows. Section 2 discusses various approaches for conducting Bayesian inference for the LGCP model. Section 3 presents a comparison of approaches through simulation studies, while Section 4 makes comparisons using two real point pattern data sets, the first arising from an ecological application and the second arising from a neuroimaging study. The paper concludes with a discussion in Section 5.

2.2 Bayesian inference for log-Gaussian Cox processes

2.2.1 Model specification

Consider an inhomogeneous Poisson process $\mathcal{Z}(s)$ with intensity function $\lambda(s)$, $s \in S \subseteq R^2$. Without loss of generality we shall assume that S is the unit square. The density of a Poisson process does not exist with respect to Lebesgue measure, but the Radon-Nikodym derivative does exist with respect to a unit-rate Poisson process (Møller et al., 1998). We will call this derivative the density of the Poisson process. Given a set of K points $\{s_k\} = \{s_1, ..., s_K\} \subset S$, where both the number K and the locations s_k are random, the density is given by

$$\pi[\{s_k\} \mid \lambda(s)] = \exp\left\{\int_S [1 - \lambda(s)]\, ds\right\} \prod_{k=1}^{K} \lambda(s_k). \tag{2.1}$$

where $\lambda(s)$ is the intensity function. If we further assume that the log of the intensity function arises from a Gaussian random field (GRF) $\mathcal{Y}(s)$ so that $\lambda(s) = \exp(\mathcal{Y}(s))$, then this hierarchical process is called a log-Gaussian Cox process (LGCP) (Møller et al., 1998). The LGCP, assumed to be stationary and isotropic, is uniquely determined by the mean function $\mu(s)$ and the covariance function $\text{Cov}(s, s') = \sigma^2 r(||s - s'||)$ of the Gaussian process $\mathcal{Y}(s)$, where σ^2 is the marginal variance and $r(||s - s'||)$ denotes correlation as a function of the Euclidean distance $||s - s'||$. Two commonly used correlation functions are the power exponential function (Møller and Waagepetersen, 2003)

$$r_p(||s - s'||) = \exp(-\rho||s - s'||^{\delta})$$

where $\rho > 0$ is the decay parameter, $\delta \in (0, 2]$ is the power exponential term (which we will take as a known constant throughout this manuscript); and the Matérn correlation function (Matérn, 1960)

$$r_m(||s - s'||) = \left(\Gamma(\nu)2^{\nu-1}\right)^{-1}(||s - s'||/\phi)^{\nu} K_{\nu}(||s - s'||/\phi)$$

where $\phi > 0$ is the range parameter, $\nu > 0$ is the shape parameter, and K_ν is the modified Bessel function of the second kind.

To fit the model in a tractable way a common approach is to divide the spatial domain into an $n \times n$ uniform grid of equally spaced cells (Møller et al., 1998) and to make the simplifying assumption that the log-intensity is constant over each grid cell so that the log-intensity $\mathcal{Y}(s)$ within a given cell, say the i^{th} cell, is constant and characterized by its value at the corresponding centroid, c_i, of cell i, $i \in \{1...n^2\}$. The unique log-intensity values then comprise a vector $\mathbf{Y} = (\mathcal{Y}(c_1), \mathcal{Y}(c_2), ..., \mathcal{Y}(c_{n^2}))^{\mathrm{T}}$. To simplify notation we let $Y_i = \mathcal{Y}(c_i)$ and y_i is a realized value of Y_i. From the defining property of a GRF, $\mathbf{Y}$ follows a multivariate normal distribution $\mathbf{Y} \sim \mathrm{N}(\mu \mathbf{1}_{n^2}, \sigma^2 \mathbf{C})$, where $\mathbf{C}$ is the $n^2 \times n^2$ correlation matrix with elements $r(||c_i - c_j||)$. Let θ be the set of parameters determining the mean and covariance of the GRF (e.g. $\theta = (\mu, \sigma^2, \rho)$ for the power correlation and $\theta = (\mu, \sigma^2, \phi)$ for the Matérn), and let A denote the area of each cell in the uniform grid. Under this discretization, the log density (see Equation (2.1)) is

$$\log \pi(\{s_k\} \mid \theta, \mathbf{y}) = \text{constant} + \sum_i [y_i n_i - A \exp(y_i)]$$

where n_i is the number of points in $\{s_k\}$ occurring in the i^{th} grid cell. The log posterior can then be expressed as

$$\begin{aligned} \log \pi(\theta, \mathbf{y} \mid \{s_k\}) \quad &= \quad \text{constant} + \sum_i [y_i n_i - A \exp(y_i)] \\ &\quad -0.5(\mathbf{y} - \mu \mathbf{1}_{n^2})^{\mathrm{T}} \sigma^{-2} \mathbf{C}^{-1} (\mathbf{y} - \mu \mathbf{1}_{n^2}) \\ &\quad -0.5 n^2 \log(\sigma^2) - 0.5 \log(|\mathbf{C}|) + \log \pi(\theta) \end{aligned} \tag{2.2}$$

where $\pi(\theta)$ is the prior density of the parameter vector θ. The computational problem for Bayesian inference is the calculation of $\pi(\theta, \mathbf{y} \mid \{s_k\})$ and its associated marginals or properties of these distributions. This computation is nontrivial because the calculation of the normalizing constant is nontrivial, particularly when the dimension of the parameter

space is high. We now discuss three approaches for approximating the posterior distribution and/or its marginals.

2.2.2 Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) has its origins with the work of Alder and Wainwright (1959) and Duane et al. (1987) and was first introduced into the statistical literature by Neal (1995). It is a Metropolis-Hastings algorithm that can be used for sampling a high-dimensional target distribution more efficiently than an algorithm based on random-walk proposals, especially when the parameters are highly correlated. The algorithm uses the notion of the (separable) Hamiltonian $H(\mathbf{q}, \mathbf{p})$ from physics that is defined as the sum of potential energy $U(\mathbf{q})$ and kinetic energy $K(\mathbf{p})$, where $\mathbf{q}$ and $\mathbf{p}$ are random vectors that refer to position and momentum. The connection to Bayesian computation lies with relating $U(\mathbf{q})$ to the posterior distribution and hence $\mathbf{q}$ to the model parameters, and with introducing auxiliary Gaussian random variables to represent momentum $\mathbf{p}$, a vector having the same length as $\mathbf{q}$. The evolution of this system is then described by the Hamilton equations from statistical mechanics:

$$\begin{aligned} \frac{dq_i}{dt} &= \frac{\partial H}{\partial p_i} \\ \frac{dp_i}{dt} &= -\frac{\partial H}{\partial q_i} \end{aligned} \tag{2.3}$$

which, if an analytic solution exists, produces a draw from the posterior distribution. In practice this system is solved using numerical integration techniques (Neal, 2011) and the resulting approximate solution is accepted or rejected using a Metropolis-Hastings step.

To carry out the computations required for the LGCP model we use a combination of re-parametrization of the random field and numerical techniques based on the 2D Fast Fourier transform (FFT) as in Møller et al. (1998). Note that although Girolami and Calderhead (2011) has applied RM-HMC to LGCP model, their computation is slow due to inverting

the fisher information matrix and a Cholesky factorization of the correlation matrix. While here we can greatly speedup matrix multiplications involving the correlation matrix $\mathbf{C}$ (we note here that we use the power exponential correlation function) by using FFT. A second reason to use the re-parametrization is that we avoid inversion on the correlation matrix at each iteration, which in our simulations and data analyses below is a 4096×4096 matrix. Although this size of a matrix can be inverted on a computer, it is computationally expensive. Furthermore, this FFT trick can handle much larger matrices that would be too large to invert on most computers. To use this trick we require the matrix to be block-circulant as there is a direct relationship between the eigenvalue-eigenvector decomposition of a block-circulant matrix and the 2D discrete Fourier transform. However, the correlation matrix has a block-Toeplitz structure. A block-Toeplitz matrix can always be extended to a block-circulant matrix (Wood and Chan, 1994). To do so, we extend the original $n \times n$ grid to an $m \times m$ grid and wrap it on a torus, where $m = 2^g$ and g is an integer such that $m \geq 2(n-1)$. The metric of this toroidal space is then defined by the minimum distance between two points. It is easy to show that the new correlation matrix $\mathbf{E}$ (of which $\mathbf{C}$ is a submatrix), whose elements are based on the metric defined on the torus, is a block-circulant matrix (Møller et al., 1998). In extending the space we must also expand the vector of latent variables $\mathbf{Y}$ in a corresponding manner, and we refer to this new vector as $\mathbf{Y}^{ext}$ (of which $\mathbf{Y}$ is a subvector). Also, we set the number of points in cell i, m_i, still to be n_i if i is on the original grid and equal to 0 otherwise.

The block-circulant extended correlation matrix can be decomposed as $\mathbf{E} = \mathbf{F}\Lambda\mathbf{F}^H$, where $\mathbf{F}$ is the matrix of eigenvectors, Λ is the diagonal matrix containing the corresponding eigenvalues of $\mathbf{E}$, and H denotes the complex conjugate transpose. Given a random vector $\mathbf{v}$ of length m^2 the product $\mathbf{E}\mathbf{v}$ can be obtained by calculating $\mathbf{F}^H\mathbf{v}$, $\Lambda\mathbf{F}^H\mathbf{v}$ and then $\mathbf{F}\Lambda\mathbf{F}^H\mathbf{v}$ in order. Note that the first and last calculations amount to a discrete inverse

Fourier transform and a discrete Fourier transform (DFT), respectively. The middle calculation is simply element-wise multiplication of Λ and the vector $\mathbf{F}^H v$ (Rue and Held, 2005). As a result, the complexity of the required matrix operations can be reduced to $O(m^2 \log(m^2))$ using the FFT.

After extension of the grid we re-parametrize the latent variables $\mathbf{Y}^{ext}$ as $\mathbf{Y}^{ext} = \mu \mathbf{1}_{m^2} + \sigma \mathbf{E}^{\frac{1}{2}} \boldsymbol{\gamma}$ where 1_{m^2} denotes the m^2-dimensional vector of ones. And $\boldsymbol{\gamma} = (\gamma_1, ..., \gamma_{m^2})^{\mathrm{T}}$ with $\gamma_i \overset{iid}{\sim} \mathbf{N}(0, 1), i = 1, ..., m^2$. The gradients, used in the HMC algorithm, for all the parameters are straightforward to derive except for ρ, we give the expression for ρ here and refer the readers to Appendix A for more details.

$$\frac{\partial \log \pi(\rho \mid \cdot)}{\partial \rho} = -\frac{\sigma}{2}\Big[\mathbf{m} - A \exp\left(\mu \mathbf{1}_{m^2} + \sigma \mathbf{E}^{\frac{1}{2}} \boldsymbol{\gamma}\right)\Big]^T \mathbf{E}^{-\frac{1}{2}} \mathbf{E}^* \boldsymbol{\gamma}, \tag{2.4}$$

where $\pi(\mid \cdot)$ denotes the full conditional given the data and other parameters, $\mathbf{m}$, $\mathbf{E}^*$ are defined in the appendix. And as $\mathbf{E}^{\frac{1}{2}}$, $\mathbf{E}^{-\frac{1}{2}}$, $\mathbf{E}^*$ are all block-circulant matrices, FFT can be used.

With the stochastic representation as in Equation 2.3 the HMC algorithm is based on setting $\mathbf{U}(\mathbf{q}) = -\log\left[\pi(\{s_k\} \mid \theta, \boldsymbol{\gamma}) \pi(\theta, \boldsymbol{\gamma})\right]$ where $\mathbf{q} = (\boldsymbol{\gamma}^{\mathrm{T}}, \theta^{\mathrm{T}})^{\mathrm{T}}$, and the kinetic energy term is $K(\mathbf{p}) = \mathbf{p}^{\mathrm{T}} \mathbf{M}^{-1} \mathbf{p}/2$, where $\mathbf{M}$ is a symmetric, positive-definite 'mass matrix' and auxiliary momentum variables $\mathbf{p}$ (a vector of length $m^2 + 3$). In our work we set $\mathbf{M}$ to be a diagonal matrix with distinct diagonal components $m_{\boldsymbol{\gamma}}$, m_μ, $m_{\sigma^{-2}}$ and m_ρ corresponding to $\boldsymbol{\gamma}$ and θ.

Each iteration of the HMC algorithm involves a block update of $\boldsymbol{\gamma}$ and θ based on the Hamiltonian Monte Carlo scheme. Each such update requires $L + 1$ evaluations of the gradient vector $\nabla_{\theta, \boldsymbol{\gamma}} \log \pi(\theta, \boldsymbol{\gamma} \mid \{s_k\})$ for some $L \geq 1$. If $L = 1$, the HMC algorithm reduces to MALA, which typically mixes faster than the random walk Metropolis-Hastings algorithm (Roberts et al., 2001) but not as fast as the more general HMC algorithm. Letting

$\boldsymbol{\gamma}^*, \theta^*$ be the current value in the Markov chain for $\boldsymbol{\gamma}, \theta$, the HMC update, based on a step size $\varepsilon > 0$, proceeds as follows:

Algorithm 1 HMC algorithm

---

1. Simulate latent vector $\mathbf{p}^* \sim \mathrm{N}_{m^2+3}(\mathbf{0}, \mathbf{M})$. Set

$$\begin{aligned} \left(\boldsymbol{\gamma}^{(0)}, \theta^{0)}\right) &= (\boldsymbol{\gamma}^*, \theta^*) \\ \mathbf{p}^{(0)} &= \mathbf{p}^* + \frac{\varepsilon}{2} \nabla_{\theta^*, \boldsymbol{\gamma}^*} \left[\log \left\{\pi\left(\boldsymbol{\gamma}^*, \theta^* \mid \{s_k\}\right) \pi\left(\theta^*, \boldsymbol{\gamma}^*\right)\right\}\right]. \end{aligned}$$

2. For l=1,...,L,

$$\begin{aligned} \left(\boldsymbol{\gamma}^{(l)}, \theta^{(l)}\right)^{\mathrm{T}} &= \left(\boldsymbol{\gamma}^{(l-1)}, \theta^{(l-1)}\right)^{\mathrm{T}} + \varepsilon \mathbf{M}^{-1} \mathbf{p}^{(l-1)} \\ \mathbf{p}^{(l)} &= \mathbf{p}^{(l-1)} + \varepsilon_l \nabla_{\theta^{(l)}, \boldsymbol{\gamma}^{(l)}} \left[\log \left\{\pi\left(\boldsymbol{\gamma}^{(l)}, \theta^{(l)} \mid \{s_k\}\right) \pi\left(\theta^{(l)}, \boldsymbol{\gamma}^{(l)}\right)\right\}\right] \end{aligned}$$

   where $\varepsilon_l = \varepsilon$ for $l < L$ and $\varepsilon_L = \varepsilon/2$.

3. Accept $\left(\gamma^{(L)}, \theta^{(L)}\right)$ as the new state for (γ, θ) with probability

$$\alpha = \min\left(1, \exp\left\{-H\left(\mathbf{q}^{(L)}, \mathbf{p}^{(L)}\right) + H\left(\mathbf{q}^*, \mathbf{p}^*\right)\right\}\right)$$

   else remain in the current state γ^*, θ^* with probability $1 - \alpha$.

Repeat steps 1–3 for a sufficiently long time.

At each iteration, the number of steps in the numerical integration, L, is drawn from a Poisson distribution with mean 100, while the step size, ε, is initially chosen to be 0.005 and adjusted adaptively during burning so that the acceptance rate is approximately 0.65 (Beskos et al., 2013). Trace plots of the parameters are examined and based on these we adjust the values of $\mathbf{M}$ to improve the mixing.

2.2.3 Mean field Variational Bayes with Laplace Approximation

As an alternative to HMC (or MCMC) sampling of the posterior distribution, a deterministic approximation can be employed. Mean field variational Bayes (MFVB) is one such approximation that has been applied successfully to a number of problems, including spatial models for high-dimensional problems requiring fast computations (Nathoo et al.,

2014). For the MFVB algorithm, we return to the parametrization of the model in Equation (2.2). Let $q(\mathbf{y}, \theta)$ be an arbitrary density function having the same support as the posterior density $\pi(\theta, \mathbf{y} \mid \{s_k\})$. Letting $\log \pi(\{s_k\})$ denote the marginal likelihood of the model, we can express its logarithm as

$$\begin{aligned}
\log \pi(\{s_k\}) &= \int q(\mathbf{y}, \theta) \log \left\{ \frac{\pi(\{s_k\}, \mathbf{y}, \theta)}{q(\mathbf{y}, \theta)} \right\} \\
&\quad + \int q(\mathbf{y}, \theta) \log \left\{ \frac{q(\mathbf{y}, \theta)}{\pi(\theta, \mathbf{y} \mid \{s_k\})} \right\} \\
&\geq \int q(\mathbf{y}, \theta) \log \left\{ \frac{\pi(\{s_k\}, \theta, \mathbf{y})}{q(\mathbf{y}, \theta)} \right\} \equiv F(q)
\end{aligned}$$

such that the functional $F(q)$ is a lower bound for $\log \pi(\{s_k\})$ for any q. The approximation is obtained by restricting q to a manageable class of density functions, and maximizing F over that class. We develop the approximation under the assumption that the GRF has a power exponential correlation function, and for now, we will assume that ρ is known, so that C is assumed known in what follows.

We assume that the approximating density q can be factorized

$$q(\mathbf{y}, \theta) = \left[\prod_{i=1}^{n^2} q(y_i) \right] q(\mu) q(\sigma^2). \tag{2.5}$$

Under this assumption, a coordinate ascent algorithm is applied to maximize F which leads to a sequence of coordinate-wise updates taking the form

$$\begin{aligned}
q(y_i) &\propto \exp\{\mathrm{E}_{-q(y_i)}[\log \pi(\{s_k\} \mid \mathbf{y}) \pi(\mathbf{y} \mid \theta)]\} \qquad i = 1, ..., n^2 \\
q(\mu) &\propto \exp\{\mathrm{E}_{-q(\mu)}[\log \pi(\mathbf{y} \mid \theta) \pi(\mu)]\} \\
q(\sigma^2) &\propto \exp\{\mathrm{E}_{-q(\sigma^2)}[\log \pi(\mathbf{y} \mid \theta) \pi(\sigma^2)]\}
\end{aligned}$$

where $E_{-q(x)}[\cdot]$ denotes the expectation taken with respect to the set of random variables $\{\mathbf{y}, \theta\} \backslash x$ under the variational approximation q_{-x}, and the updates steps are iterated to the convergence of F. We describe the derivation of the update steps below. In what follows, $E[\cdot]$ will denote the expectation of its argument under the variational approximation q.

If conditionally conjugate Gaussian and inverse-gamma priors are chosen for $\pi(\mu)$ and $\pi(\sigma^2)$ respectively, $\mu \sim N(\mu_\mu, \sigma_\mu^2)$, $\sigma^2 \sim G^{-1}(\alpha, \beta)$ the distributions $q(\mu)$ and $q(\sigma^2)$ comprising the update steps will also be Gaussian and inverse-gamma. To derive the update step for μ we have

(2.6)

$$\begin{aligned}\mathrm{E}_{-q(\mu)} \log \pi(\mathbf{y} \mid \theta)\pi(\mu) &= c - \frac{1}{2}\left[(\mu_\mathbf{y} - \mu\mathbf{1})^T \mathrm{E}(\sigma^{-2})(C^{-1})(\mu_\mathbf{y} - \mu\mathbf{1})\right] - \frac{1}{2}(\mu - \mu_\mu)^2/\sigma_\mu^2 \\ &= c - \frac{1}{2}\left[(\mathrm{E}(\sigma^{-2})\mathbf{1}^T(C^{-1})\mathbf{1} + 1/\sigma_\mu^2)\mu^2 - 2\mathrm{E}(\sigma^{-2})\mu_\mathbf{y}{}^T(C^{-1})\mathbf{1}\mu + \mu_\mu/\sigma_\mu^2\right]\end{aligned}$$

where $\mu_\mathbf{y} = E[\mathbf{y}]$ and c denotes a constant not depending on μ. As (2.6) is a quadratic function of μ it follows that $q(\mu)$ is the density of a normal distribution $N(\mu_\mu^*, \sigma_\mu^{*2})$ where after some algebra we have

$$\mu_\mu^* = (\mathrm{E}(\sigma^{-2})\mu_\mathbf{y}{}^T(C^{-1})\mathbf{1} + \mu_\mu/\sigma_\mu^2)(\sigma_\mu^{*2})^{-1}, \;\; \sigma_\mu^{*2} = (\mathrm{E}(\sigma^{-2})\mathbf{1}^T(C^{-1})\mathbf{1} + 1/\sigma_\mu^2)^{-1}.$$

To derive the update step for σ^2 we have

$$\begin{aligned}\mathrm{E}_{-q(\sigma^2)} \log \pi(\mathbf{y} \mid \theta)\pi(\sigma^2) = c - \frac{n^2}{2}\log\sigma^2 - \frac{1}{2}\sigma^{-2}\mathrm{E}_\mu[(\mu_\mathbf{y} - \mu\mathbf{1})^T C^{-1} \\ (\mu_\mathbf{y} - \mu\mathbf{1}) + \mathrm{Tr}(C^{-1}\Sigma_\mathbf{y})] - (\alpha + 1)\log\sigma^2 - \beta/\sigma^2\end{aligned}$$

where $\Sigma_\mathbf{y}$ is the covariance matrix of $\mathbf{y}$ under $q(\mathbf{y})$ and c denotes a constant not depending on σ^2. Simplifying this expression yields

$$\begin{aligned}\mathrm{E}_{-q(\sigma^2)} \log \pi(\mathbf{y} \mid \theta)\pi(\sigma^2) = c - (\alpha + 1 + \frac{n^2}{2})\log\sigma^2 - (\beta + \frac{1}{2}[(\mu_\mathbf{y} - \mu_\mu^*\mathbf{1})^T(C^{-1}) \\ (\mu_\mathbf{y} - \mu_\mu^*\mathbf{1}) + \sigma_\mu^{*2}\mathbf{1}^T(C^{-1})\mathbf{1} + \mathrm{Tr}(C^{-1}\Sigma_\mathbf{y})])/\sigma^2\end{aligned}$$

and thus $q(\sigma^2)$ is the density of an Inverse-Gamma distribution $G^{-1}(\alpha + \frac{n^2}{2}, \beta^*)$ where $\beta^* = \beta + \frac{1}{2}\left[(\mu_\mathbf{y} - \mu_\mu^*\mathbf{1})^T(C^{-1})(\mu_\mathbf{y} - \mu_\mu^*\mathbf{1}) + \sigma_\mu^{*2}\mathbf{1}^T(C^{-1})\mathbf{1} + \mathrm{Tr}(C^{-1}\Sigma_\mathbf{y})\right]$.

As the Gaussian prior for y_i is not conditionally conjugate for the LGCP model, the variational Bayes update for $q(y_i)$ is not a standard distribution and is therefore not easy

to compute without some further approximation. We derive an update step for $q(y_i)$ by applying the Laplace method (Wang and Blei, 2013) within the variational Bayes update. We have

$$\begin{aligned}
&\mathrm{E}_{-q(y_i)} \log \pi(\{s_k\} \mid \mathbf{y})\pi(\mathbf{y} \mid \theta) = c + (y_i n_i - e^{y_i} A) \\
&\quad -\frac{1}{2}\left[(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1})^T \mathrm{E}(\sigma^{-2})(C^{-1})(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1}) + \mathrm{Var}(\mu)E(\sigma^{-2})\mathrm{Tr}((C^{-1})\mathbf{1}\mathbf{1}^T)\right] \\
&\quad = c + (y_i n_i - e^{y_i} A) - \frac{1}{2}\left[(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1})^T \mathrm{E}(\sigma^{-2})(C^{-1})(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1})\right] \qquad (2.7)
\end{aligned}$$

where $\tilde{\mathbf{y}}$ denotes $(\mathrm{E}(y_1), ..., \mathrm{E}(y_{i-1}), y_i, \mathrm{E}(y_{i+1}), ..., \mathrm{E}(y_{n^2}))^T$. Taking the derivative with respect to y_i yields,

$$\begin{aligned}
f(y_i) &= \partial \mathrm{E}_{-q(y_i)} \log \pi(\{s_k\} \mid \mathbf{y})\pi(\mathbf{y} \mid \theta)/\partial y_i \\
&= n_i - Ae^{y_i} - \mathrm{E}(\sigma^{-2})\left[(C^{-1})(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1})\right]_i \\
&= n_i - Ae^{y_i} - \mathrm{E}(\sigma^{-2})\sum_j (C^{-1})_{ij}(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1})_j \\
&= -Ae^{y_i} - \mathrm{E}(\sigma^{-2})(C^{-1})_{ii} y_i + n_i \qquad (2.8) \\
&\quad + \mathrm{E}(\sigma^{-2})(C^{-1})_{ii}\mathrm{E}(\mu) - \mathrm{E}(\sigma^{-2})\sum_{j \neq i} (C^{-1})_{ij}(\tilde{\mathbf{y}} - \mathrm{E}(\mu)\mathbf{1})_j
\end{aligned}$$

Given (2.8) we find $\hat{y}_i$ such that $f(\hat{y}_i) = 0$. We use Newton's method to obtain a numerical solution where the starting value for Newton's method is obtained by omitting the linear term in equation (2.8) and solving the resulting simplified equation exactly. We then take the second derivative with respect to y_i, $H(y_i) = \partial f(y_i)/\partial y_i = -Ae^{y_i} - \mathrm{E}(\sigma^{-2})(C^{-1})_{ii}$ and given this and the solution $\hat{y}_i$, the Laplace method yields a normal distribution $N(\hat{y}_i, -H(\hat{y}_i)^{-1})$ for $q(y_i)$ which approximates the VB update. The variational-Laplace approximation to the posterior of the latent field $\mathbf{y}$ is then $\mathbf{y} \sim N(\mu_{\mathbf{y}}, \Sigma_{\mathbf{y}})$ where $\mu_{\mathbf{y}} = (\hat{y}_1, ..., \hat{y}_{n^2})^T$ and $\Sigma_{\mathbf{y}}$ is a diagonal matrix with the i^{th} element being $-H(\hat{y}_i)^{-1}$.

Given the update steps derived above, the approximate posterior distribution under the variational-Laplace approximation takes the form (2.5) where the component densities are

standard distributions

$$\begin{aligned} \mathbf{y} &\sim \mathrm{N}(\mu_{\mathbf{y}}^{(q)}, \Sigma_{\mathbf{y}}^{(q)}) \\ \mu &\sim \mathrm{N}(\mu_{\mu}^{(q)}, \sigma_{\mu}^{2(q)}) \\ \sigma^2 &\sim G^{-1}(\alpha^{(q)}, \beta^{(q)}). \end{aligned}$$

The parameters determining these distributions are called the 'variational parameters'. These parameters are obtained through the sequence of update steps derived above which are used to determine equations expressing each variational parameter in terms of the remaining variational parameters; beginning with initial values for these parameters the equations are iterated to convergence of F.

With respect to the parameter ρ of the power exponential correlation function, we have found that its inclusion as an unknown parameter into the variational approach leads to convergence problems in a number of trial examples. To deal with this problem, we estimate this parameter prior to running the VB algorithm using the method of minimum contrast (Diggle and Gratton, 1984), i.e., to use non-linear least squares estimation to fit a non-parametric estimated covariance function. The mean field VB algorithm incorporating the Laplace method for the LGCP model is presented in detail in Algorithm 2.

2.2.4 INLA

The integrated nested Laplace approximation (INLA) is another approach for constructing a deterministic approximation to the posterior distribution that can be applied to the fairly broad class of latent Gaussian models. The details underlying the approach have been described in a number of recent papers including the seminal work of Rue et al. (2009). We provide here only a brief overview of aspects that are relevant for use with the LGCP model.

For spatial models, INLA makes extensive use of the Gaussian Markov random field

Algorithm 2 Mean Field VB Algorithm with Laplace Method

1. Initialize the priors $\mu_\mu, \sigma_\mu^2, \alpha, \beta$.

2. Initialize $\mu_{\mathbf{y}}^{(q)}$, $\Sigma_{\mathbf{y}}^{(q)}$, $\mu_\mu^{(q)}$, $\sigma_\mu^{2(q)}$, $\beta^{(q)}$ and $\mathrm{E}(\sigma^{-2}) = (\alpha + n^2/2)/(\beta^{(q)})$

3. Obtain ρ using the minimum contrast method, compute $\mathbf{C}^{-1}$ where $c_{kl} = \exp(-\rho||c_k - c_l||^\delta)$.

4. For $i = 1, ..., n^2$, compute $\mu_{y_i}^{(q)}$ such that

$$\begin{aligned} &-A\exp(\mu_{y_i}^{(q)}) - \mathrm{E}(\sigma^{-2})\left(\mathbf{C}^{-1}\right)_{ii}\mu_{y_i}^{(q)} + n_i \\ &\quad +\mu_\mu^{(q)}\mathrm{E}(\sigma^{-2})\left(\mathbf{C}^{-1}\right)_{ii} - \mathrm{E}(\sigma^{-2})\sum_{j\neq i}\left(\mathbf{C}^{-1}\right)_{ij}[\mu_{\mathbf{y}}^{(q)} - \mu_\mu^{(q)}\mathbf{1}]_j = 0 \end{aligned}$$

   where $[\cdot]_j$ denotes the j^{th} element of a vector.
   Compute $H(\mu_{y_i}^{(q)}) = -A\exp(\mu_{y_i}^{(q)}) - \mathrm{E}(\sigma^{-2})\left(\mathbf{C}^{-1}\right)_{ii}$.
   Obtain $\mu_{\mathbf{y}}^{(q)}$ and $\Sigma_{\mathbf{y}}^{(q)}$ where $\mu_{\mathbf{y}}^{(q)} = (\mu_{y_1}^{(q)}, ..., \mu_{y_{n^2}}^{(q)})^{\mathrm{T}}$ and $\Sigma_{\mathbf{y}}^{(q)}$ is a diagonal matrix with diagonal elements $-H(\mu_{y_i}^{(q)})^{-1}$.

5. Compute $\mu_\mu^{(q)} = (\mathrm{E}(\sigma^{-2})\mu_{\mathbf{y}}^{(q)^{\mathrm{T}}}\mathbf{C}^{-1}\mathbf{1} + \mu_\mu/\sigma_\mu^2)(\mathrm{E}(\sigma^{-2})\mathbf{1}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{1} + 1/\sigma_\mu^2)^{-1}$. Compute $\sigma_\mu^{2(q)} = (\mathrm{E}(\sigma^{-2})\mathbf{1}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{1} + 1/\sigma_\mu^2)^{-1}$.

6. Compute $\beta^{(q)} = \beta + 0.5[(\mu_{\mathbf{y}}^{(q)} - \mu_\mu^{(q)}\mathbf{1})^{\mathrm{T}}\mathbf{C}^{-1}(\mu_{\mathbf{y}}^{(q)} - \mu_\mu^{(q)}\mathbf{1}) + \sigma_\mu^{2(q)}\mathbf{1}^{\mathrm{T}}\mathbf{C}^{-1}\mathbf{1} + \mathrm{Tr}(\mathbf{C}^{-1}\Sigma_{\mathbf{y}}^{(q)})]$. Obtain $\mathrm{E}(\sigma^{-2}) = (\alpha + n^2/2)/(\beta^{(q)})$

7. Compute the lower bound

$$\begin{aligned} F(q) \quad = \quad & \sum_i(\mu_{y_i}^{(q)}n_i - A\exp(\mu_{y_i}^{(q)} - 0.5H(\mu_{y_i}^{(q)})^{-1})) - 0.5\mathrm{E}\log(|\mathbf{C}|) \\ & -0.5[(\mu_\mu^{(q)} - \mu_\mu)^2 + \sigma_\mu^{2(q)}]/\sigma_\mu^2 \\ & +\sum_i 0.5\log\left(-H(\mu_{y_i}^{(q)})^{-1}\right) + 0.5\log\sigma_\mu^{2(q)} \end{aligned}$$

Repeat 4–7 until the increase in $F(q)$ is negligible.

(GMRF) which is a Gaussian distribution having a sparse precision matrix. Algorithms for fitting models incorporating a GMRF can be made efficient through the use of numerical methods for sparse matrices. In the case of the LGCP model the latent GRF which is a spatially continuous process is approximated by a GMRF on a discrete lattice so that these numerical methods can be applied. We consider this approximation for the case where the GRF is a Matérn field with ν known, so that $\theta = (\mu, \sigma^2, \phi)$ and the log intensity is characterized by $\mathbf{Y}$ as before. An approximation $\tilde{\pi}(\theta \mid \{s_k\})$ to the marginal posterior distribution $\pi(\theta \mid \{s_k\})$ is first obtained using the Laplace method. An approximation $\tilde{\pi}(y_i \mid \theta, \{s_k\})$ to the density of the full conditional distribution of each component of the latent field is then obtained using one of the three methods: a Gaussian approximation, the Laplace approximation, or a simplified Laplace approximation. The latter option is based on a series expansion of the Laplace approximation which has a lower computational cost. The marginal posterior distributions for the log intensity values of the LGCP model are then approximated through numerical integration over a discrete grid for θ

$$\tilde{\pi}(y_i \mid \{s_k\}) = \sum_k \tilde{\pi}(y_i \mid \theta_k, \{s_k\}) \tilde{\pi}(\theta_k, \{s_k\}) \Delta_k$$

where $\{\Delta_k\}$ are a set of area weights associated with the grid.

Recently, Lindgren et al. (2011) develop an approximation to certain GRFs with Matérn correlation functions by specifying stochastic partial differential equations (SPDEs) that have certain Matérn processes as their solution. This SPDE representation provides an explicit link to GMRFs through a basis function representation of the solution where the corresponding weights comprise a GMRF with dependencies determined by a triangular mesh covering the spatial domain. This approximation can also be embedded within INLA and is implemented within the R-INLA package (obtained at www.r-inla.org) which allows for different mesh sizes. Increasing the size of the mesh will increase the accuracy of this approximation but will also increase the required time for computation.

We note, and express, here that the INLA package only allows for a Matérn correlation structure, whereas for the HMC and VB algorithms we use the power exponential family for computational purposes (e.g. it allows for easy calculation of the gradients).

2.3 Simulation Studies

Here the methods described in the previous section are compared using simulation studies. The discretized spatial domain is taken to be a 64×64 grid on the unit square, so that the simulated data are based on 4096 spatial locations. Each study is based on 1000 data sets simulated under the discretized LGCP model where we compare a total of six approaches: HMC incorporating FFT methods on the extended grid, VB incorporating the Laplace method, INLA with a simplified Laplace approximation (INLA I), INLA with a full Laplace approximation (INLA II), INLA with the SPDE model based on a mesh size of 436 (INLA III), and INLA with the SPDE model based on a mesh size of 4075 (INLA IV). The first mesh size was chosen purposely small and the second mesh size was chosen to be approximately the same as the number of cells in the discretized grid. In what follows, INLA I and INLA II are also referred to as INLA with the lattice method, INLA III and INLA IV are also referred to as INLA with SPDE.

Our HMC and VB algorithms were derived under the assumption of a power exponential correlation for the GRF; whereas, we use the implementation of INLA in the standard R-INLA package that assumes a Matérn correlation. When simulating data we assume the GRF has a Matérn correlation and we apply all six approaches to the resulting data. Thus, INLA is based on a correctly specified covariance function and therefore has an advantage over HMC and VB which have the correlation function mis-specified. We will also assume that the parameter ν (in the Matérn model) is fixed and known, so that using INLA we only estimate the decay parameter ϕ in the Matérn model. As we cannot directly compare ρ and

ϕ we are not able to compare directly the properties of the corresponding estimators. As an alternative we make comparisons with respect to the distance at which the correlation function drops to 0.5, denoted as $d_{0.5}$ and defined by the equations

$$r_p(d_{0.5}) = r_m(d_{0.5}) = 0.5 \tag{2.9}$$

where $r_p(\cdot)$ and $r_m(\cdot)$ denote the power exponential and Matérn correlation functions respectively.

2.3.1 Simulation One

We first simulate data sets from the LGCP model where the Matérn field has $\mu = 5$, $\sigma^2 = 3.5$, $\phi = 0.02$, and $\nu = 1$. Based on these parameters we simulate the GRF once and, based on this realization of the latent field, we simulate 1000 independent replicates of the data. Along with estimation of the log intensity values we also estimate μ, σ^2, $d_{0.5}$ and E(N), where E(N) is the expected total number of points of the process within the spatial domain. The estimators are evaluated with respect to bias, variance and MSE. For the HMC and VB algorithms we match the shape of the power correlation function with that of the Matérn correlation function based on nonlinear least squares to estimate, and fix, the value for δ in the power exponential model, and we obtain a value of $\delta = 1.312$.

In terms of priors, HMC assumes a flat prior $\pi(\mu) \propto 1$, $\sigma^2 \propto \text{I}(0, \infty)$ and $\rho \propto \text{I}(0, \infty)$. With VB we are constrained to use conditionally conjugate priors and set $\mu \sim \text{N}(0, 625)$, $\sigma^2 \sim G^{-1}(1, 1)$, while ρ is estimated by the method of minimum contrast and assumed known in the VB algorithm. For INLA with the lattice method, we assign diffuse priors $\sigma^{-1} \sim G(0.001, 0.001)$ and $d_I \sim G(0.001, 0.001)$ where $d_I = \sqrt{8\nu}\phi$, and for INLA with SPDE we use the default joint-normal prior. Additional discussion of the priors and related numerical issues associated with INLA SPDE are mentioned in Section 5. INLA III has a mesh size of 436 (based on a length 0.1 for the inner mesh and 0.5 for outer mesh)

and INLA IV has a mesh size of 4075 (based on a length 0.03 for the inner mesh and 0.5 for outer mesh). We acknowledge that the use of different priors for the HMC, VB, and INLA methods is not ideal as the differences we observe in the simulation results may, to some extent, be driven by differences in the priors. However, as the priors are taken to be fairly diffuse in all cases we do not expect the differences in the priors to play a significant role. As a practical matter the form of the prior may sometimes be driven by the choice of computational algorithm used for Bayesian computation. Indeed, convergence issues may also impact the prior used in some circumstances. While not ideal from a theoretical perspective these issues are unavoidable from a practical standpoint. For example, the use of VB typically calls for conditionally conjugate priors while for INLA we are constrained to use certain forms for the priors that are built into the R-INLA package. As the different computational algorithms are often implemented with different priors we feel that these comparisons offer useful practical guidance for users despite these differences.

Figure 2.1 shows the true value of the discretized latent field $\mathbf{Y}$ in comparison to the average marginal posterior mean obtained from each of the methods under consideration. The average marginal posterior mean is the average of the estimates obtained from each of the 1000 simulation replicates. In this case we see that HMC, VB, INLA I and INLA II all produce similar average reconstructions of $\mathbf{y}$; whereas, the results from INLA III and INLA IV appear more over-smoothed, with the degree of over-smoothing decreasing as the mesh size increases.

Taking HMC as the baseline, a plot of the log-relative mean squared error (MSE) associated with the posterior mean estimator of $\mathbf{Y}$ for VB and INLA is shown in Figure 2.2. Points above (below) the black line indicate larger (smaller) MSE relative to that obtained from HMC. Both VB and INLA I-IV tend to produce estimators that have a higher MSE than the corresponding estimator obtained from HMC when the value of the log-intensity

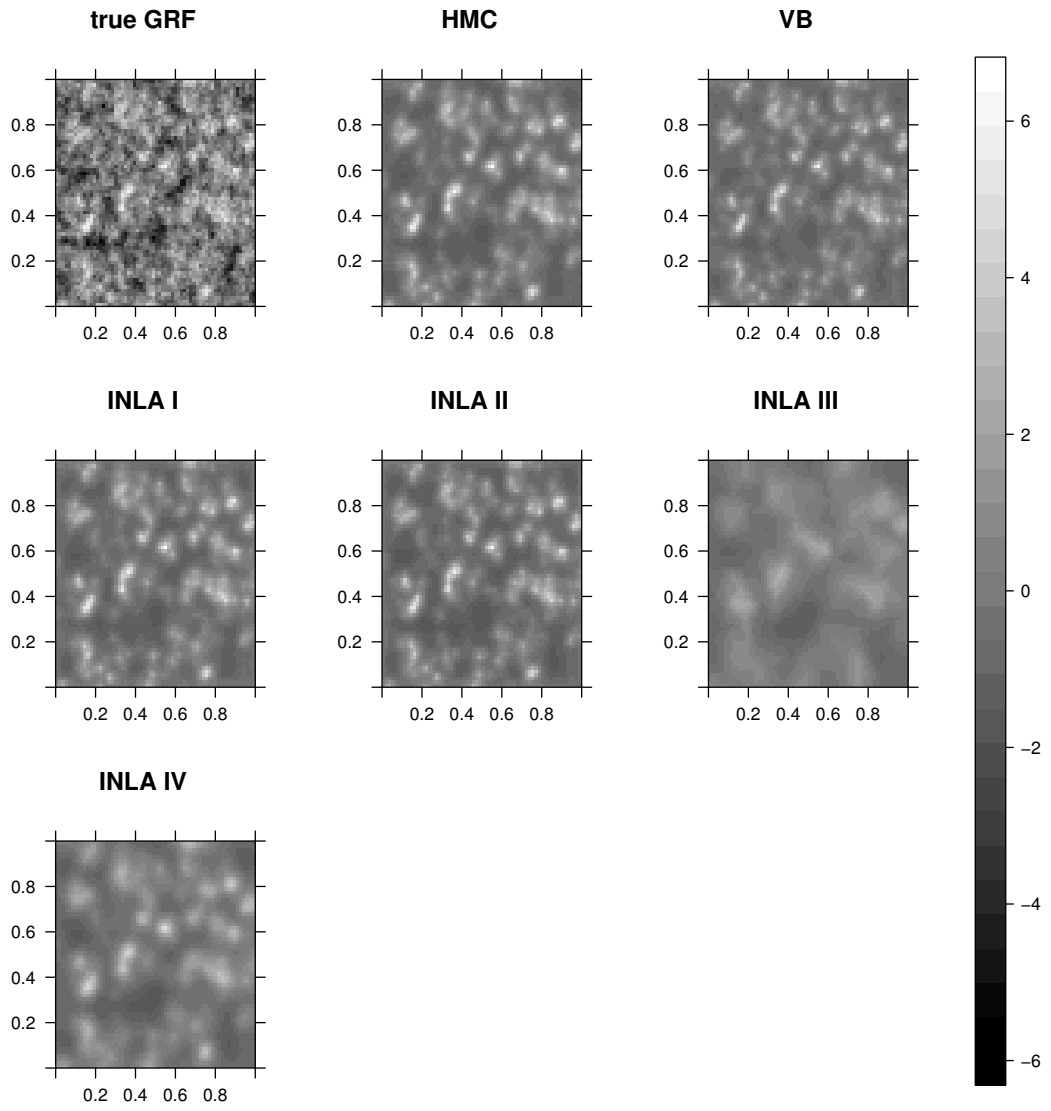


Figure 2.1: Average marginal posterior mean of the log-intensity over 1000 samples from the first simulation study. Upper left panel is the true GRF.

approaches either tail of the distribution of log-intensity values. Conversely, the methods appear to outperform HMC for values of the log-intensity around the median of this distribution. These differences appear to be more variable for INLA III and INLA IV compared with VB, INLA I and INLA II.

Table 2.1 displays the bias, variance, and MSE for the posterior mean estimators of μ, σ^{-2}, $d_{0.5}$ and $\mathrm{E}(N)$. In this and all subsequent tables we display, for HMC, the actual

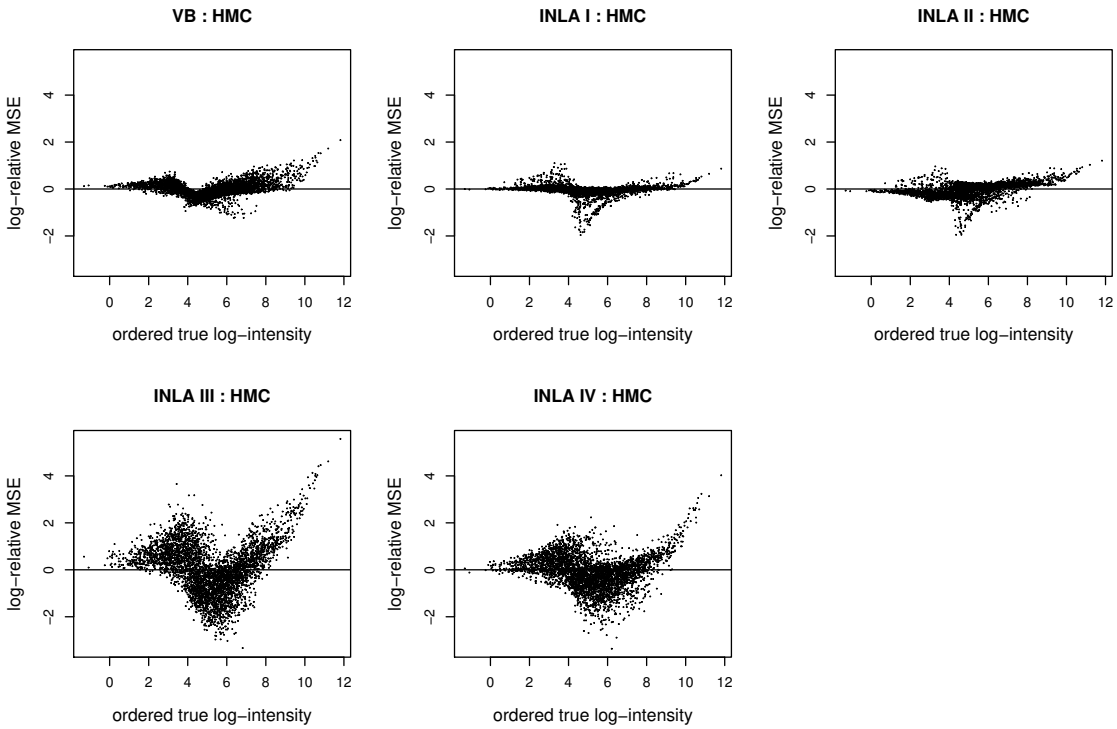


Figure 2.2: Log-relative MSE of estimated latent GRF from VB and INLA I–IV to that from HMC for the first simulation study. Each point represents the log-relative MSE of the discretized GRF. Points located above, on and below the horizontal line denotes bigger, equal and smaller MSE than that from HMC.

values of bias, variance and MSE, whereas for VB and INLA the values are relative to those obtained from HMC. For estimation of these parameters we see that VB and INLA generally have larger bias and MSE compared with HMC. Both INLA I and INLA II outperform VB slightly, while INLA III and INLA IV lag behind the alternatives rather significantly. By construction, INLA I and INLA II will provide identical estimates for σ^{-2} and $d_{0.5}$ and this is reflected in the table. For the estimation of $\mathrm{E}(N)$, HMC outperforms VB with respect to bias, variance, and MSE, while these measures are not reported for INLA as we only obtain marginal posterior distributions of the log-intensity values in this case and thus cannot estimate $\mathrm{E}(N)$.

While Table 2.1 displays the variance of the posterior mean estimators, we display in Table 2.2 the average (over simulation replicates) marginal posterior variance. The

Parm	Measure		Relative Measure				
		HMC	VB	INLA I	INLA II	INLA III	INLA IV
$\mu = 5$	Bias	0.046	12.681	5.469	8.471	25.805	15.435
	Var	0.010	0.476	0.903	0.663	8.453	0.632
	MSE	0.012	29.441	6.143	13.507	127.227	43.558
$\sigma^{-2} = 0.286$	Bias	-0.010	6.984	-3.727	-3.727	-72.906	-14.545
	Var	3.20E-04	0.454	1.338	1.338	36.531	4.810
	MSE	4.08E-04	11.436	4.191	4.191	1236.295	51.801
$d_{0.5} = 0.025$	Bias	3.30E-05	-122.467	122.245	122.245	1704.511	1115.598
	Var	2.20E-06	0.465	1.244	1.244	1199.323	4.574
	MSE	2.20E-06	7.895	8.588	8.588	2626.682	616.258
$\mathrm{E}(N) = 910.29$	Bias	-1.337	-252.070	-	-	-	-
	Var	918.282	1.420	-	-	-	-
	MSE	920.069	124.820	-	-	-	-

Table 2.1: Summary of the statistical properties for the hyper-parameters from the first simulation study. The values shown in table from VB and INLA I-IV are relative to that from HMC.

associated large sample theory guarantees that the posterior variance as obtained from HMC is simulation consistent. As such the posterior variance for VB and INLA are again listed relative to that obtained from HMC in order to determine the extent to which these approaches under-estimate or over-estimate posterior variability. For VB we see that the marginal posterior variance is under-estimated which is in line with expectations from the literature (Nathoo et al., 2013, 2014). INLA I and II provide measures of variability that are closer to that of HMC, while INLA III and IV tend to over-estimate the marginal posterior variance, with this over-estimation being substantial when the smaller mesh size is used. With respect to average computational time, HMC requires 679 seconds based on 1500 total iterations with the first 500 thrown away as burn-in; VB requires 453 seconds to run to convergence (about 300 iterations); INLA I runs for 46 seconds, INLA II requires 196 seconds, INLA III requires 10 seconds, and INLA IV requires 154 seconds. The algorithms are run on an iMac with a 3.2 GHz Intel Core i5 processor and 16GB memory.

Average Marginal Var		Relative Ave. Marg. Var				
Parm	HMC	VB	INLA I	INLA II	INLA III	INLA IV
μ	0.028	0.511	0.868	0.333	85.232	2.287
σ^{-2}	0.001	0.029	1.576	1.576	6.780	17.061
$d_{0.5}$	4.80E-06	0.022	1.667	1.667	87549.788	7.161

Table 2.2: Marginal variance estimates of the parameters for the first simulation study. VB and INLA I-IV are relative to HMC

2.3.2 Simulation Two

In order to make comparisons in a setting where there is a slower decay for the spatial correlation and smoother realizations of the random field our second study is based on setting $\phi = 0.05$ and $\nu = 3$ with all other settings remaining unchanged. Figure 2.3 presents the average posterior mean log-intensity over 1000 simulation replicates. Comparing the images the overall best reconstruction seems to arise from both VB and HMC, followed by INLA I, INLA II, and INLA IV all of which capture the general features of the true latent field, while INLA III seems subject to over-smoothing as before. Figure 2.4 displays the log-relative MSE of the five methods in comparison with HMC as in Figure 2.2. Interestingly, INLA IV seems to have the best performance in terms of MSE here, which taken together with Figure 2.3 suggests that the estimators from INLA IV may have lower variance while still achieving adequate bias.

With respect to the hyper-parameters, Table 2.3 displays the bias, variance, and MSE of the posterior mean estimators for μ, σ^{-2}, $d_{0.5}$ and $\mathrm{E}(N)$. INLA III and INLA IV have the smallest MSE for μ, but have large MSE for $d_{0.5}$. As before HMC attains the lowest MSE for estimation of σ^{-2} and $d_{0.5}$. In terms of $\mathrm{E}(N)$, HMC outperforms VB with respect to bias, variance and MSE. Table 2.4 shows the average marginal posterior variance for the hyper-parameters. INLA I and II generally under-estimate the marginal posterior variance. VB shows significant under-estimation of the marginal variance for σ^{-2} and $d_{0.5}$, and over-estimation of the posterior variance for μ. As with the previous study, INLA III and INLA IV over-estimate the marginal posterior variance for all three hyper-parameters. In terms of average timing, HMC requires 521 seconds for a total of 2000 iterations with the first 1000 iterations discarded as burn-in; VB requires 467 seconds and typically required a greater number of iterations to converge (about 1000) compared to simulation one; INLA I takes 134 seconds, INLA II takes 357 seconds, INLA III takes 11 seconds, INLA IV

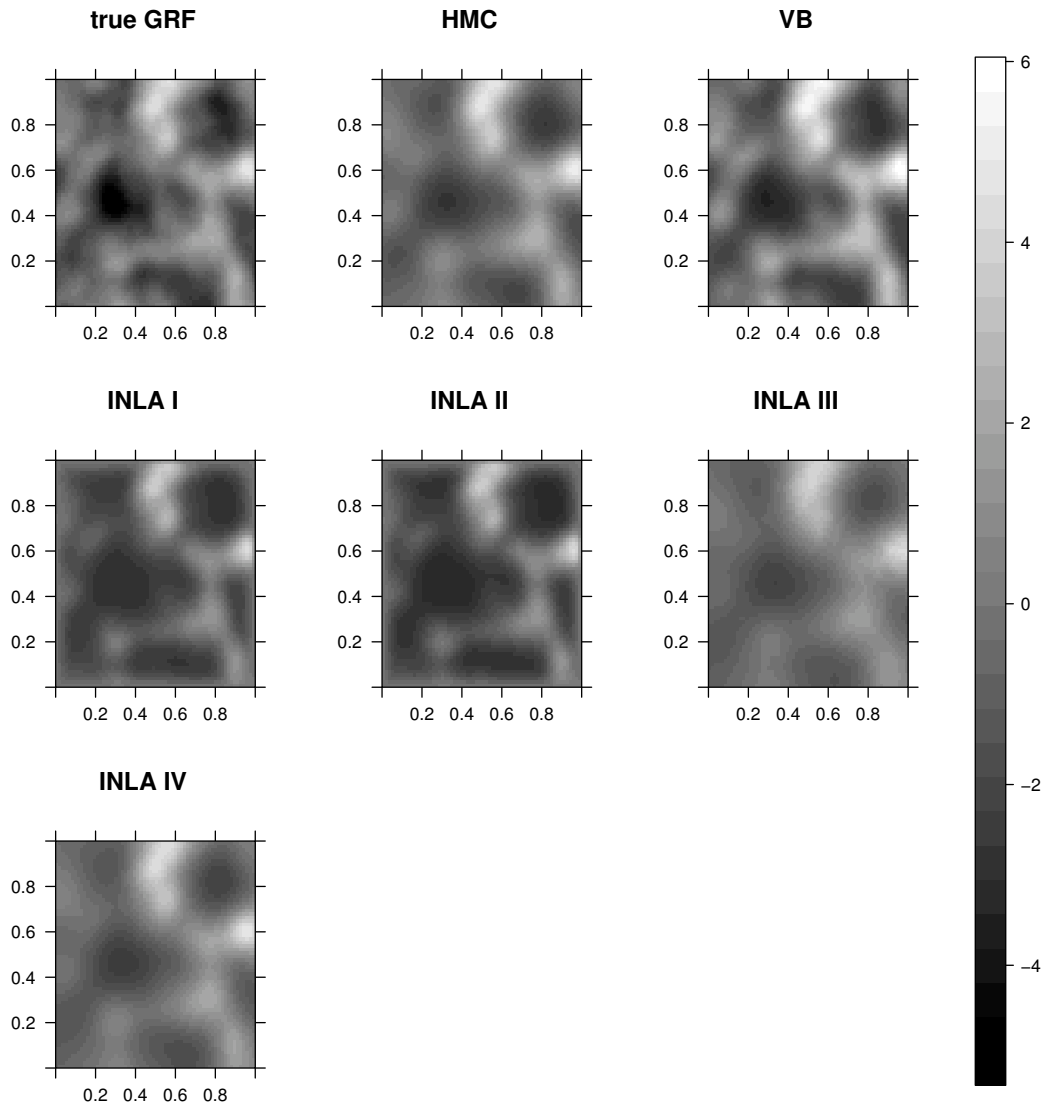


Figure 2.3: Average marginal posterior mean of the log-intensity over 1000 samples from the second simulation study. Upper left panel is the true GRF.

takes 227 seconds.

2.4 Application

We next compare the computational algorithms through an application to two data sets where the LGCP model is applied in both cases. In addition to comparing the methods with respect to posterior summaries of the parameters of interest, we also make compar-

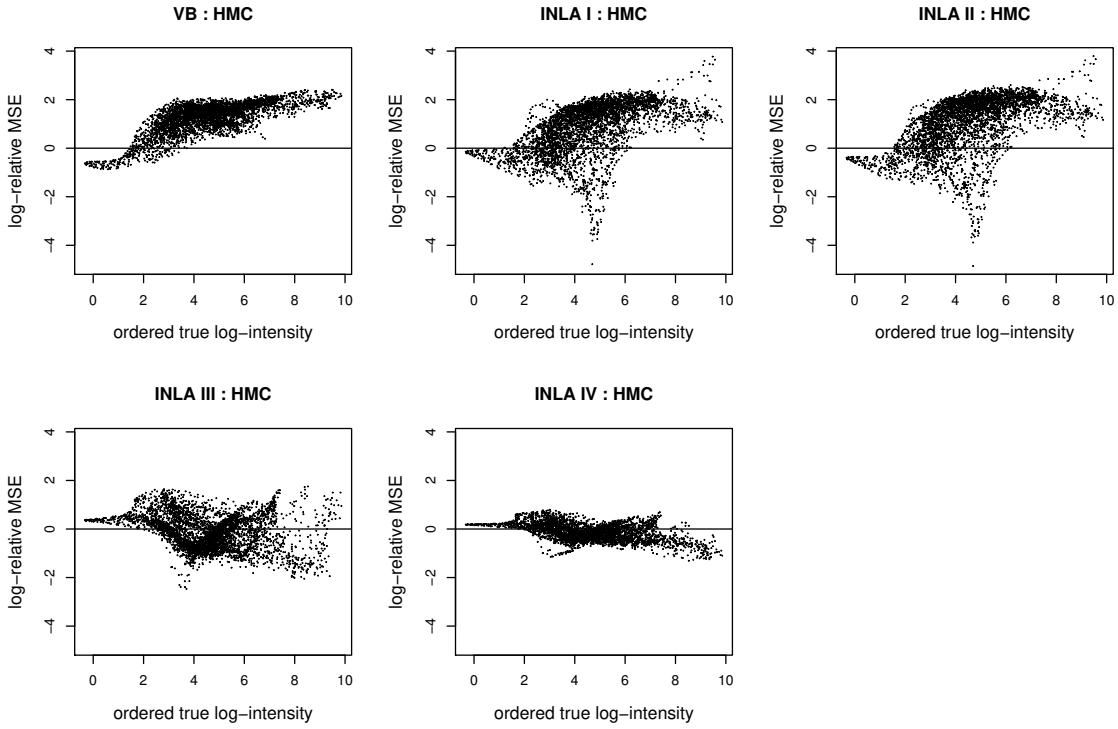


Figure 2.4: Log-relative MSE of estimated latent GRF from VB and INLA I–IV to that from HMC for the second simulation study. Each point represents the log-relative MSE of the discretized GRF. Points located above, on and below the horizontal line denotes bigger, equal and smaller MSE than that from HMC.

Parm	Measure		Relative Measure				
		HMC	VB	INLA I	INLA II	INLA III	INLA IV
$\mu = 5$	Bias	-0.458	2.752	-1.513	-1.488	0.047	0.458
	Var	0.029	1.169	4.682	4.617	0.932	1.232
	MSE	0.239	6.797	2.579	2.507	0.115	0.334
$\sigma^{-2} = 0.286$	Bias	-0.017	15.316	3.355	3.355	2.038	4.477
	Var	1.20E-03	0.003	3.75	3.75	1.131	0.849
	MSE	0.001	48.492	5.301	5.301	1.755	4.817
$d_{0.5} = 0.13$	Bias	-1.00E-02	4.364	6.893	6.893	-43.224	-31.924
	Var	7.90E-05	0.191	1.236	1.236	24.439	17.627
	MSE	1.80E-04	10.868	27.431	27.431	1068.207	584.571
$\mathrm{E}(N) = 494.12$	Bias	0.077	1201.419	-	-	-	-
	Var	496.455	1.199	-	-	-	-
	MSE	496.461	18.527	-	-	-	-

Table 2.3: Summary of the statistical properties for the hyper-parameters from the second simulation study. The values shown in table from VB and INLA I-IV are relative to that from HMC.

isons with respect to goodness-of-fit checking using the posterior predictive distribution (Gelman et al., 1996) which has been applied for checking hierarchical spatial models in a number of applications including disease ecology and neuroimaging (Nathoo, 2010; Kang et al., 2011, e. g.)). The posterior predictive checks are based on the L function (Illian

Average Marginal Var		Relative Ave. Marg. Var				
Parm	HMC	VB	INLA I	INLA II	INLA III	INLA IV
μ	0.255	6.084	0.301	0.299	5.952	4.531
σ^{-2}	0.006	3.36E-05	0.650	0.650	23.322	20.661
$d_{0.5}$	3.10E-04	0.001	0.164	0.164	12.023	5.935

Table 2.4: Marginal variance estimates of the parameters from the second simulation study. VB and INLA I-IV are relative to HMC

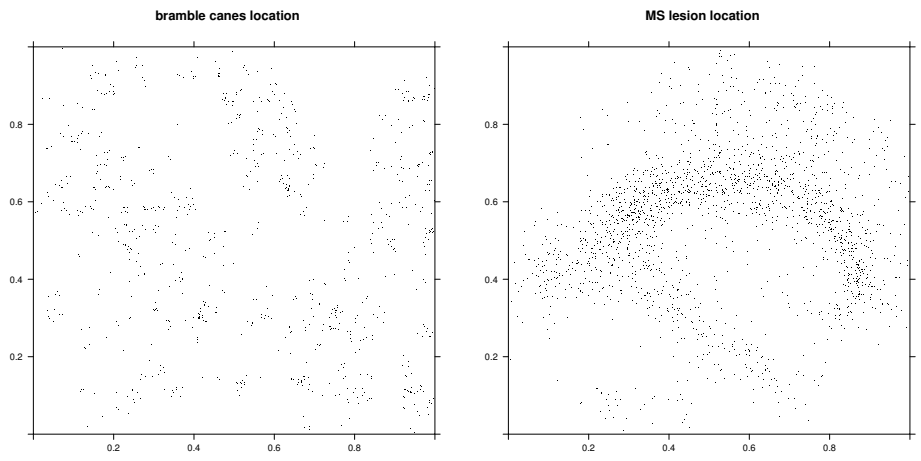
et al., 2009) where we simulate, based on the model, posterior predictive replicates of the discrepancy measure $\Delta(r) = L(r, \{s_k\}^{obs}, \mathbf{y}, \theta) - L(r, \{s_k\}^{rep}, \mathbf{y}, \theta)$ where r denotes the distance, $\{s_k\}^{obs}$ denotes the observed data, $\mathbf{y}$ and θ are drawn from the posterior distribution, and $\{s_k\}^{rep}$ denotes replicate data that is drawn from the posterior predictive distribution. For a given distance range r, if the value $\Delta(r) = 0$ is observed to lie as an extreme value in either tail of the posterior predictive distribution we may question the fit of the model as characterized by the L function at that distance. As INLA does not provide the joint posterior distribution we are unable to simulate predictive realizations and thus we make the predictive comparisons only between HMC and VB.

All priors are the same as in the simulation studies unless otherwise indicated. We also compare VB and the INLA methods with HMC as HMC is simulation consistent.

2.4.1 Bramble Canes data

The data record the (x, y) locations of 823 bramble canes in a field of $9\,\mathrm{m}^2$, rescaled to a unit square. The data are depicted in Figure 2.5(a) and were recorded and analyzed by Hutchings (1979) and further analyzed by Diggle (Diggle et al., 1983).

To determine the value of δ in the power-exponential correlation function and the value of ν in the Matérn correlation function, we use the method of minimum contrast which estimates the values as $\hat{\delta} = 0.51$, $\hat{\nu} = 0.02$. As the R-INLA package only offers three possible values $\nu = 1, 2, 3$, we select $\nu = 1$ as it is the closest of the three choices to the estimate obtained from the minimum contrast method. Figure 2.6 depicts the posterior



(a) (b)

Figure 2.5: (a) shows the Bramble canes locations; (b) shows the MS lesion locations. Both are represented by black dots, rescaled to the unit square

mean of the log-intensity as obtained by each of the six methods. HMC, INLA I and INLA II result in estimated images that appear fairly consistent; results obtained from VB are also consistent with HMC but less so than INLA I and II, while both INLA III and INLA IV appear to over-smooth the estimated latent field relative to the other methods.

Posterior summaries of the hyper-parameters μ, σ^{-2} and $d_{0.5}$ are presented in Table 2.5. Here we see that VB is under-estimating the posterior variance for all hyper-parameters while the point estimates for μ, σ^{-2} are larger than those obtained from HMC, but within somewhat reasonable bounds. The point estimates obtained from INLA with the lattice method are closer to those obtained from HMC than those obtained from INLA with SPDE. In Figure 2.7 we compare the marginal posterior variance of each element of the latent field as obtained from all of the methods to the posterior variance obtained from HMC. In this case all of the methods under-estimate the posterior variance and this under-estimation is most severe for INLA with SPDE.

Figure 2.8 compares the 95% posterior predictive intervals for $\Delta(r)$ as obtained from both HMC and VB. Comparing the figures indicates that the posterior predictive variability

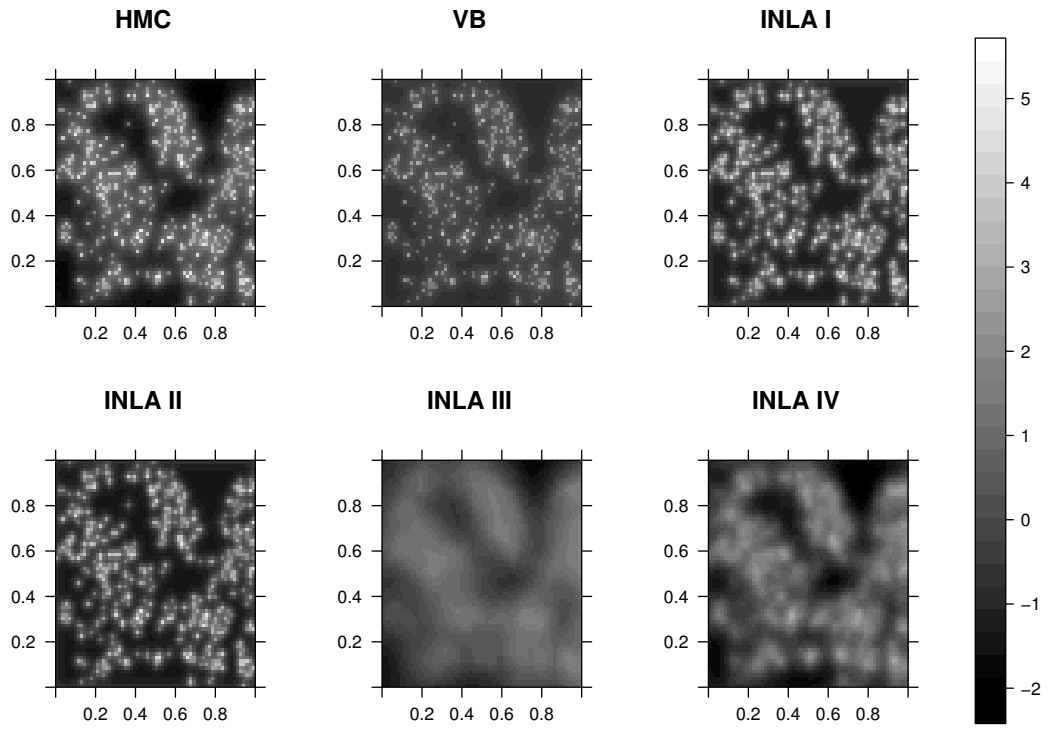


Figure 2.6: Posterior mean of the latent GRF for the bramble canes data set, estimated from HMC, VB and INLA I–IV.

Parm	Measure		Relative Measure				
		HMC	VB	INLA I	INLA II	INLA III	INLA IV
μ	Post. Mean	5.019	1.223	0.996	1.078	1.197	1.161
	Post. Var	0.016	0.52	1.503	0.535	32.469	4.744
σ^{-2}	Post. Mean	0.272	1.743	1.135	1.135	2.461	1.998
	Post. Var	0.001	0.194	1.581	1.581	114.182	26.765
$d_{0.5}$	Post. Mean	0.025	0.165	0.768	0.768	9.215	2.894
	Post. Var	8.00E-05	9.98E-06	0.025	0.025	46.779	0.945

Table 2.5: Summary of parameter estimation for the bramble canes data set, VB and INLA I-IV are relative to HMC.

is under-estimated for VB, primarily at the lower distance ranges r. The implication of this data analysis is that posterior predictive checks for VB under similar settings would be conservative. In terms of the data, the posterior predictive check as obtained from HMC does not reveal a lack-of-fit with respect to the chosen discrepancy measure. With respect to timing, HMC requires 597s for 1500 iterations and 500 burn-in iterations; VB actually requires more time than HMC in this example and takes 1012s requiring 137 iterations to

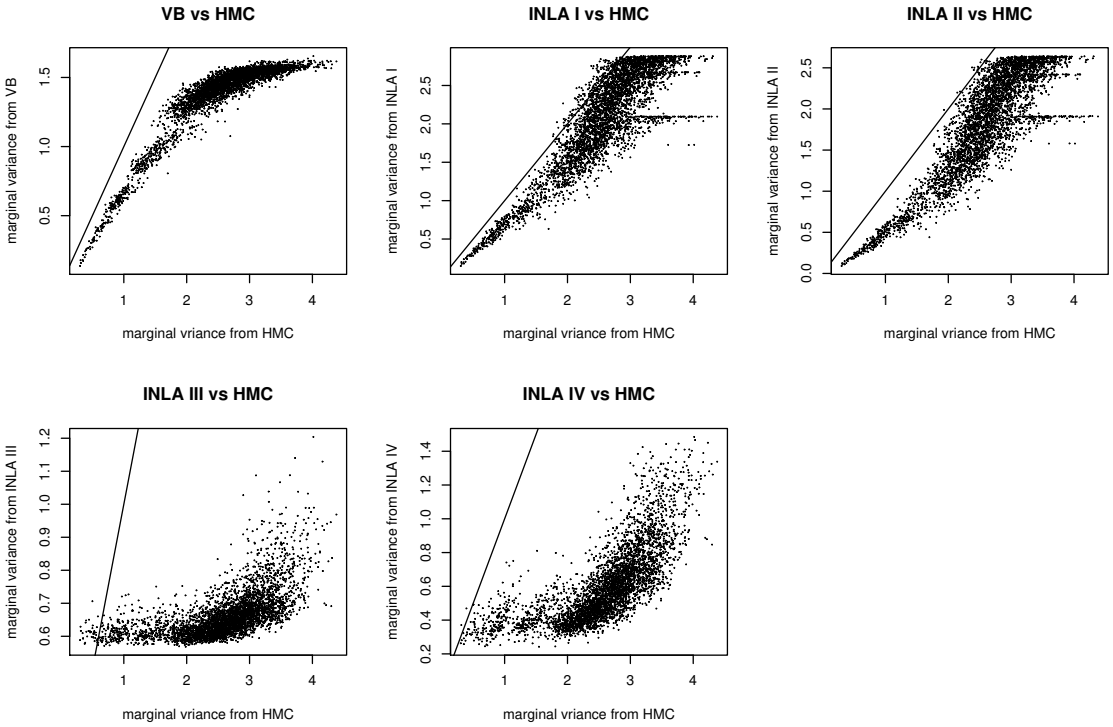
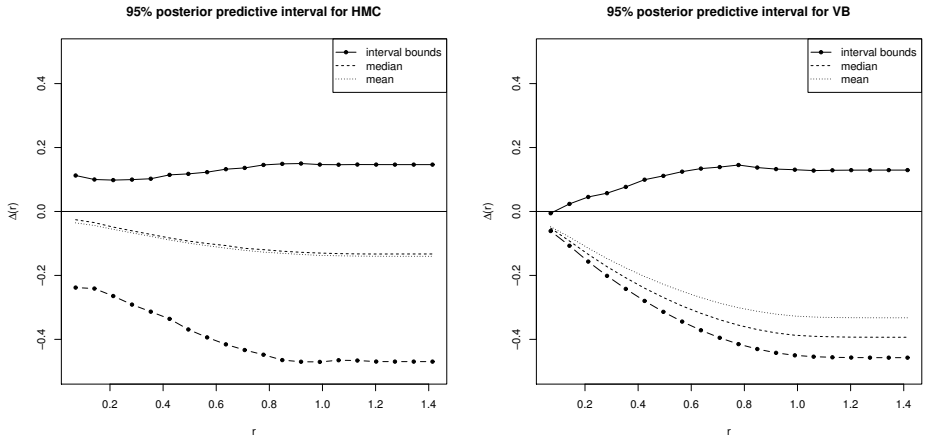


Figure 2.7: Scatter plot of the marginal posterior variance of the latent GRF from VB and INLA I-IV compared with those from HMC. Bramble canes data set.

convergence. INLA I require 55s, INLA II 172s, INLA III 11s, and INLA IV takes 227s.



(a) (b)

Figure 2.8: 95% posterior predictive interval for HMC (a) and VB (b) for the bramble canes data set. The bounds are denoted by solid lines while the mean and median are denoted by dashed lines. These are obtained at 20 distinct distances.

2.4.2 Multiple Sclerosis MRI Data

Our second application consists of a point pattern depicting the locations of Multiple Sclerosis (MS) lesions obtained from taking a slab of sagittal slices (10mm thick) obtained from magnetic resonance imaging from a cohort of MS patient and converting the spatial domain to the unit square. The point pattern consists of 1950 locations and is depicted in Figure 2.5(b). Aside from the application this data set differs from the first in that the observed level of aggregation is higher and the points are more unevenly distributed. The method of minimum contrast is used to select values of $\hat{\delta} = 1.165$ and $\hat{\nu} = 1$ for the covariance functions. The posterior mean of the log-intensity values are depicted in Figure 2.9. In this case all methods seem to capture the same general features of the image.

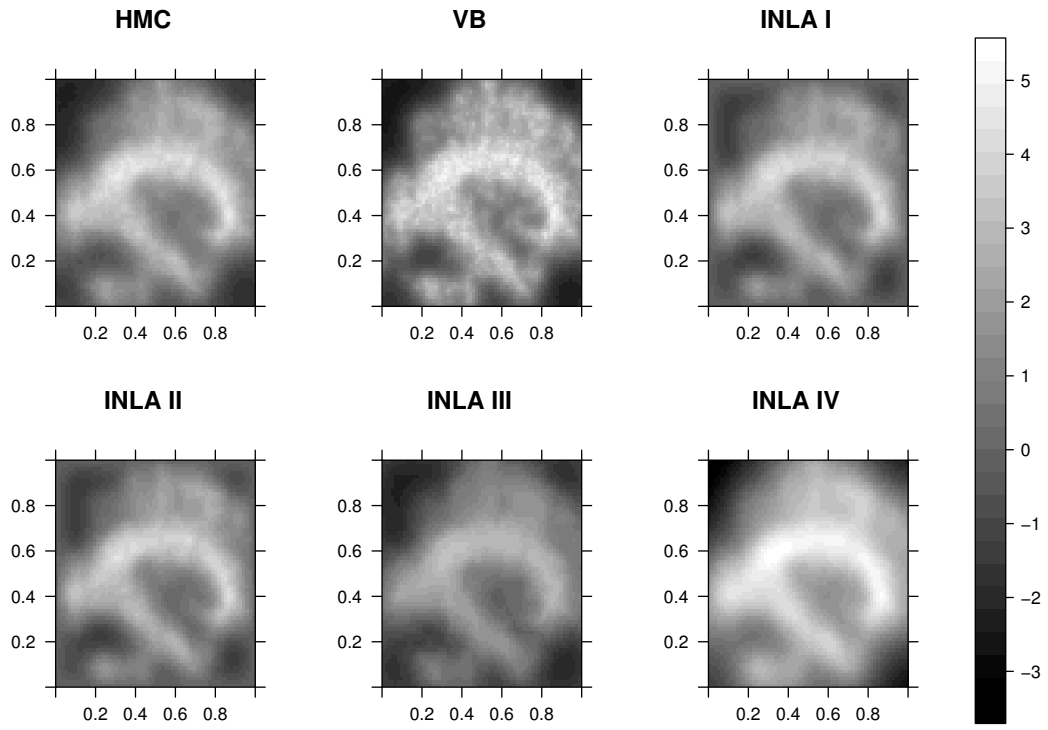


Figure 2.9: Posterior mean of the latent GRF, estimated from HMC, VB and INLA I–IV. MS data set.

Table 2.6 presents posterior summaries of the hyper-parameters. The point estimates for μ obtained from all approaches are fairly consistent while, relative to HMC, posterior

variance is not well estimated by either VB or INLA I-IV. Relative to HMC the precision σ^{-2} is not well estimated by any of the methods and similarly for the posterior variance of this parameter. As $d_{0.5}$ is estimated rather precisely by HMC, all of VB and INLA I-III either severely under-estimate or over-estimate the posterior variability though this is not the case for INLA IV. Figure 2.10 compares the marginal posterior variance for each cell of the discretized latent field obtained from all of the methods with the posterior variance obtained from HMC. Interestingly, in this case we find that VB over-estimates the posterior variability while INLA I-III all under-estimate the posterior variability to different degrees relative to HMC. The posterior variability arising from INLA IV gives extremely large values (up to 10 times larger than those obtained from HMC) which likely indicates a numerical problem, though we note again that INLA IV does give an adequate representation of the posterior mean for this data set.

Parm	Measure		Relative Measure				
		HMC	VB	INLA I	INLA II	INLA III	INLA IV
μ	Post. Mean	5.098	0.959	1.074	1.072	1.193	0.814
	Post. Var	0.301	2.638	0.081	0.084	0.197	32.731
σ^{-2}	Post. Mean	0.468	0.282	0.1	0.1	1.729	0.192
	Post. Var	0.005	0.002	31.83	31.83	1.19E-36	0.404
$d_{0.5}$	Post. Mean	0.18	0.886	14.379	14.379	0.503	2.789
	Post. Var	1.30E-04	2.41E-10	31653.155	31653.155	2.91E-37	0.959

Table 2.6: Summary of parameter estimation for the MS data set. VB and INLA I-IV are relative to HMC.

Turning to posterior predictive checks which are depicted in Figure 2.11 we see that VB has much wider 95% posterior predictive intervals than HMC. Although neither algorithm shows a lack of fit, using HMC, the model appears to fit the data better as the mean and median are much closer to zero for all ranges of r and the 95% predictive intervals are much tighter at each value of r. The greater posterior predictive variability arising from VB may be in part a result of the posterior variance of μ being over-estimated by VB. In terms of timing, HMC takes 1456s with 2000 iterations and 1000 burn-in. Again, VB actually requires more computation time 1608s with 2763 iterations required for convergence.

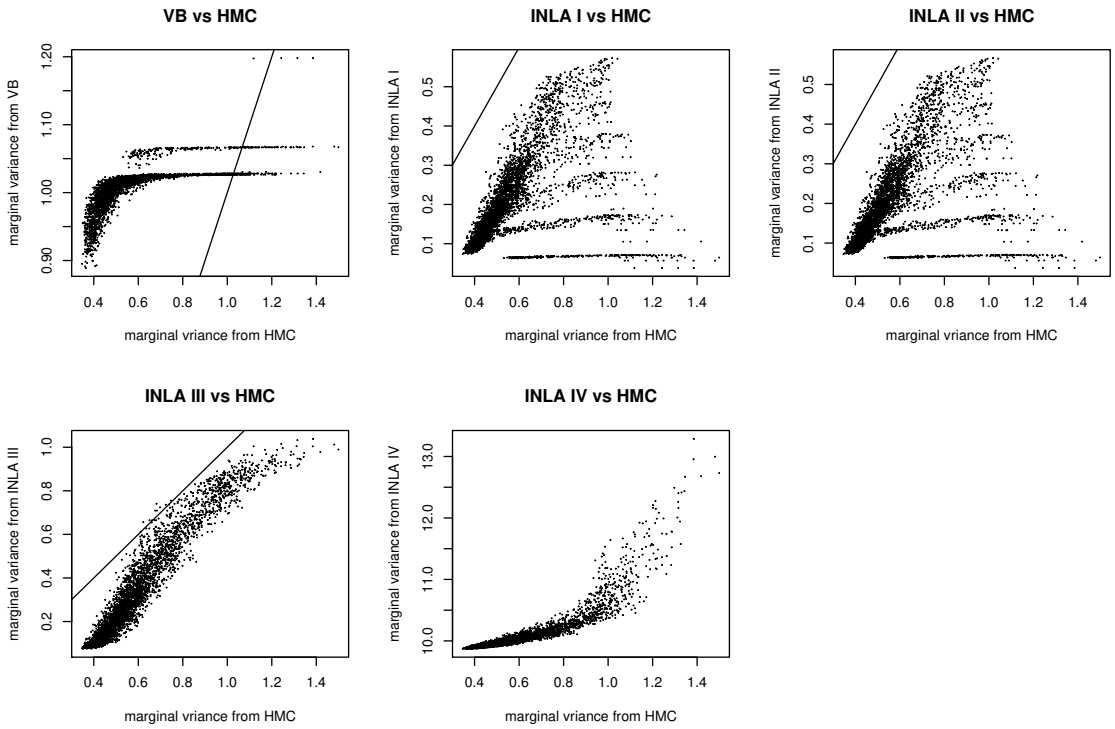
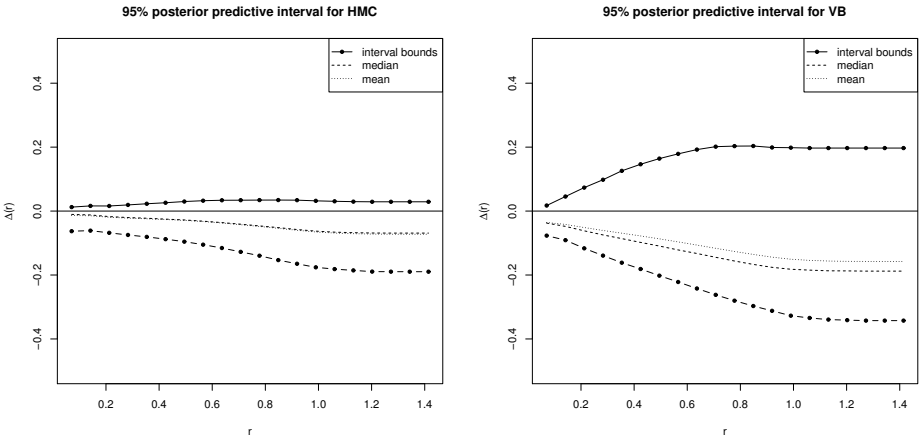


Figure 2.10: Scatter plot of the marginal posterior variance of the latent GRF from VB and INLA I-IV compared with those from HMC for the MS data set.

INLA I takes 47s, INLA II takes 166s, INLA III takes 57s, INLA IV takes 384s.



(a) (b)

Figure 2.11: 95% posterior predictive interval for HMC (a) and VB (b) for the MS data set. The bounds are denoted by solid lines while the mean and median are denoted by dashed lines. These are obtained at 20 distinct distances.

2.5 Discussion

We have compared HMC incorporating FFT matrix methods on an extended grid, VB incorporating the Laplace method, and four versions of INLA for Bayesian computation and estimation associated with the LGCP model. One would not expect to get back the true latent field following inference via any of these methods, rather, the gold standard is the posterior density of the field given the data. Of the methods considered, only HMC has theoretical guarantees in terms of (simulation) consistency in attaining the gold standard so the empirical comparisons considered here are of practical value. A number of settings for both simulated and real data have been adopted for these comparisons. Overall, in terms of point estimation of the latent field we do observe some differences in some settings; however, generally, all of HMC, VB, INLA I, and INLA II perform reasonably well, while INLA with SPDE has a tendency to over-smooth the field, though this tendency is reduced as the size of the underlying mesh is increased. Thus if point estimation of the log-intensity is the only objective we recommend the use of INLA I based on the required computation time. If, in addition, inference on hyper-parameters is of importance then it seems clear that HMC and the additional required computation time is necessary. As expected from the literature VB has a tendency to under-estimate posterior variability although on occasion it also seems to over-estimate posterior variability (see e.g. Daunizeau et al. (2009) for discussion of the latter issue and how it may also arise with VB). We also find that posterior predictive checking based on VB may not be representative of the true posterior predictive distribution. While VB has been applied successfully in a wide range of applications and is often the method of choice in machine learning the required computation time for the LGCP model and for the settings considered here suggest that it is not as accurate as HMC and not as computationally efficient as INLA I. Of course, our imple-

mentation of VB did not incorporate the FFT methods for matrix multiplication as this is not straightforward to implement within the VB framework. We acknowledge that fixing ρ in the VB algorithm based on the minimum contrast estimate is not ideal; however, this was the only practical possibility given the convergence problems we see for VB when ρ is treated as unknown. We suspect that the convergence problem is somehow related to the mean-field approximation and its assumption of posterior independence between ρ and the other model parameters, which likely conflicts with the data. Further examination of this issue, perhaps more generally for mean field approximations is a potentially interesting avenue for further work. To improve the variational approximation one approach that may be worth considering is the use of fixed-form multivariate Gaussian variational approximations which may have improved performance over mean field approximations.

To compare within the four flavors of INLA, the INLA with simplified Laplace method and the full Laplace method are quite similar in terms of accuracy, for the settings considered here; whereas, we see genuine gains in computation with the use of the simplified Laplace version of INLA. In addition to a tendency to exhibit over-smoothing, we have also found that INLA with SPDE can be numerically unstable in some situations and an inappropriate choice of step size in the Newton-Raphson algorithm can lead to convergence problems. Thus the choice of mesh is an important consideration. We acknowledge that the different priors and different covariance functions used with INLA relative to the competing approaches does make the interpretation of the simulation results comparing to VB and MCMC more difficult than had the same priors and covariance functions been used in all cases; however, as a practical matter, practitioners will use the built-in priors and the Matérn covariance function available in INLA, so we feel that these comparisons are of direct interest and practical value.

CHAPTER III

A Comparison of Variational Bayes and Hamiltonian Monte Carlo for Bayesian fMRI Time Series Analysis with Spatial Priors

3.1 Introduction

It is well known that fMRI data exhibit both spatial and temporal autocorrelation. A widely used approach for the analysis of such data is the general linear model with autoregressive errors and spatial smoothing priors for the regression coefficients (GLM-AR). Models of this sort have been developed in the Bayesian framework (Penny et al. (2005); Penny et al. (2007)) with approximate Bayesian inference based on mean field variational Bayes (VB). The VB approximation is used to handle the very large parameter space across voxels in the brain while maintaining computational tractability. While this approach often leads to computational efficiency, there are potential concerns with its accuracy. Nathoo et al. (2013) have discussed this issue and demonstrated examples with neuroimaging data where the mean field variational Bayes approximation can severely underestimate posterior variability and produce biased estimates of model hyper-parameters.

Simulation-based approaches for Bayesian computation such as importance sampling and Markov chain Monte Carlo (MCMC) have an underlying large sample theory guaranteeing simulation-consistent approximation (Robert and Casella, 2013) of various aspects of the posterior distribution, such as posterior moments and quantiles. Unfortunately, there is currently no such theory guaranteeing or characterizing the accuracy for VB approxi-

mations. As a result these approximations need to be checked on a case-by-case basis, typically against the output from properly tuned MCMC algorithms. In some cases, the quality of the VB approximation will be very good and in other cases the VB approximation can be quite poor. For a given model where the VB approximation is used, it is of practical importance for users to have some general understanding of the quality of this approximation, and if computational resources are available, to be able to check this for certain test cases (e.g. using the fMRI data from a select few subjects in a study). The contribution of this paper is to address this issue for the model developed by Penny et al. (2007) and the corresponding variational Bayes implementation in the SPM software.

In making comparisons with MCMC techniques, it is important that the particular MCMC algorithm being employed achieves adequate mixing and thus is able to traverse the parameter space fairly rapidly. This is a particularly important issue when dealing with spatial models for fMRI data as the number of parameters in the model and their potentially high posterior correlations can result in poor performance of standard MCMC algorithms such as the Gibbs sampler and the random walk Metropolis-Hastings algorithm, as well as algorithms that combine Gibbs and random walk Metropolis-Hastings moves. MCMC algorithms of this sort for spatio-temporal fMRI time series models have been developed by Woolrich et al. (2004b) where Gibbs sampling and single-component Metropolis-Hastings jumps are employed for posterior simulation. An alternative MCMC algorithm that is better suited for large parameter spaces with high posterior correlations is the HMC algorithm (Duane et al. (1987); Neal (1995)). For neuroimaging data and dynamic causal modeling, the HMC algorithm has been recently explored by Sengupta et al. (2016) where it is found that HMC and Langevin Monte Carlo are far superior to the random walk Metropolis algorithm when applied for the estimation of neural mass models. As far as we are aware, the derivation of HMC and its comparison to the mean field VB

approximation for the spatial model of Penny et al. (2007) currently implemented in the SPM software has not been considered previously.

In Section 2 we review briefly the spatial fMRI model and the VB algorithm for approximating the posterior distribution. We then derive HMC and discuss the tuning of this algorithm. In Section 3 we present two simulation studies as well as a comparison on the face repetition fMRI dataset considered in Henson et al. (2002). Section 4 concludes with a brief discussion.

3.2 Methods

We begin by briefly discussing the fMRI spatial model. We then describe the Variational Bayes (VB) and Hamiltonian Monte Carlo (HMC) algorithms that can be used to fit this model. We put a greater emphasis on the HMC algorithm as the VB algorithm has been discussed in Penny et al. (2005). The VB algorithm is implemented in the SPM12 software and for computations in this paper is run on MATLAB 2014a, on an iMac with 3.2 GHz and 16GB memory. The HMC algorithm code is written in C++, and implemented on the same machine in the case of our analysis of the face repetition data. For the simulation studies we run the HMC algorithm on a high-performance computing cluster (a Linux cluster powered by 12 dual quad-core Intel Xeon SMP compute nodes running at 2.33GHz per CPU). In all cases the HMC algorithm is run for 3000 iterations with first 2000 iterations discarded as burn-in, and the remaining 1000 iterations used to estimate features of the posterior distribution.

3.2.1 The fMRI spatial model

We let T denote the length of each time series, N the number of voxels, K the number of regressors in the linear model, and P the order of the temporal autoregressive process used to model the temporal correlation at each voxel. Throughout this paper, a matrix is

indicated with bold capital letters, while vectors are indicated with bold lower-case letters, and scalars are denoted by lower-case letters. The linear model at the n^{th} voxel, n=1,. . . ,N, is specified as

$$\mathbf{y}_{P+1:T,n} = \mathbf{X}\mathbf{w}_n + \mathbf{e}_n \tag{3.1}$$

where $\mathbf{y}_n = (y_{1n}, ..., y_{Tn})^T$ denotes the time series of length T recorded at the n^{th} voxel with last $(T - P)$ components denoted as $\mathbf{y}_{P+1:T,n}$, and where we condition on the first P components $\mathbf{y}_{1:P,n}$ for simplicity. $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_K)$ denotes the K columns of regressors each having length $T - P$; $\mathbf{w}_n$ is the corresponding vector of regression coefficients specific to voxel n. The regressors are typically stimulus indicators convolved with the hemodynamic response function (HRF), $x_{tk} = (v_k * h)(t)$, that is, the k^{th} regressor at time t, is the k^{th} stimulus v_k convolved with the HRF $h(\cdot)$ at time t. Details are described in Lindquist et al. (2008). The autoregressive process for the model errors is specified as

$$\mathbf{e}_n = \tilde{\mathbf{E}}_n\mathbf{a}_n + \mathbf{z}_n \tag{3.2}$$

where $\tilde{\mathbf{E}}_n = (\tilde{\mathbf{e}}_{P+1,n}, ..., \tilde{\mathbf{e}}_{Tn})^T$ is a $(T - P) \times P$ lagged prediction matrix with t^{th} row $\tilde{\mathbf{e}}_{tn} = (e_{t-1,n}, ..., e_{t-P,n})$; $\mathbf{a}_n = (a_{1n}, ..., a_{pn})^T$ is the corresponding vector of autoregressive coefficients for voxel n; $\mathbf{z}_n = (z_{P+1,n}, ..., z_{Tn})^T$ is the Gaussian noise for voxel n, with z_{tn} i.i.d with mean 0 and precision λ_n $(t = P + 1, ...T)$. The contribution to the log-likelihood for voxel n, is then:

$$l_n = -\frac{\lambda_n}{2}\sum_{t=P+1}^{T}\left[(y_{tn} - \mathbf{x}_t\mathbf{w}_n) - \tilde{\mathbf{e}}_{tn}\mathbf{a}_n\right]^2 + \frac{T-P}{2}\log\lambda_n + const \tag{3.3}$$

where $const$ denotes a constant that does not depend on the model parameters, and $\mathbf{x}_t$ is the t^{th} row of $\mathbf{X}$. We note that this formulation conditions on the data observed at the first P time points, and this conditioning, while not strictly necessary, simplifies the treatment of the model. As T is typically large compared with P, this conditioning may have little

effect on the resulting inference (Penny et al. (2003)). The overall log-likelihood is then obtained by summing l_n across all voxels $l = \sum_{n=1}^{N} l_n$.

Regarding priors for the model parameters, let $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_N)$ denote the set of regression coefficients across all of the voxels, so that $\mathbf{W}$ is $K \times N$. The rows of $\mathbf{W}$ are assumed a priori independent, but the model adopts a prior that incorporates spatial dependence across voxels (across the columns of $\mathbf{W}$ within each row). Let $\mathbf{w}_k$ be the k^{th} row of $\mathbf{W}$, a vector of length N, and let $\pi(\mathbf{W}|\boldsymbol{\alpha})$ denote the prior density which takes the form

$$\begin{aligned} \pi(\mathbf{W}|\boldsymbol{\alpha}) &= \prod_{k=1}^{K} \pi(\mathbf{w}_k^T|\alpha_k) \\ \mathbf{w}_k^T \mid \alpha_k &\sim \mathrm{N}(\mathbf{0}, \alpha_k^{-1}(\mathbf{S}^T\mathbf{S})^{-1}). \end{aligned} \tag{3.4}$$

where $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)^T$ are hyper-parameters. Here $\mathbf{S}$ is a spatial kernel and takes the form of a non-singular Laplacian matrix (Pascual-Marqui et al. (1994)) with elements:

$$s_{ij} = \begin{cases} deg, & \text{if } i = j \\ -1, & \text{if } i \neq j \text{ and i is adjacent to j} \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

where $deg = 4$ for a two dimensional model and $deg = 6$ for a three dimensional model. By formulating the spatial kernel matrix in this way, smoothing is achieved and it is easy to show that the precision matrix $\mathbf{S}^T\mathbf{S}$ is a sparse matrix with 13 non-zero elements on each row and each column for a two dimensional model, and 25 non-zero elements on each row and each column for a three dimensional model. Similarly, a spatial prior is used for the autoregressive coefficients $\mathbf{A} = (\mathbf{a}_1, ..., \mathbf{a}_N)$ across all voxels. Let $\mathbf{a}_p$ denote the p^{th} row

of $\mathbf{A}$, the prior for $\mathbf{A}$ is

$$\pi(\mathbf{A}|\boldsymbol{\beta}) = \prod_{p=1}^{P} \pi(\mathbf{a}_p^T|\beta_p)$$

$$\mathbf{a}_p^T \mid \beta_p \sim \mathrm{N}(\mathbf{0}, \beta_p^{-1}(\mathbf{D}^T\mathbf{D})^{-1}) \tag{3.6}$$

where $\boldsymbol{\beta} = (\beta_1, ..., \beta_P)^T$ are hyper-parameters; $\mathbf{D}$ is a spatial kernel matrix similar to $\mathbf{S}$, for simplicity we will assume that $\mathbf{D} = \mathbf{S}$.

For the hyper-parameters $\boldsymbol{\alpha} = (\alpha_1, ..., \alpha_K)^T$, $\boldsymbol{\beta} = (\beta_1, ..., \beta_P)^T$, and precision parameters $\boldsymbol{\lambda} = (\lambda_1, ..., \lambda_N)^T$, the model assumes that these parameters are conditionally independent with each following a Gamma distribution a priori:

$$\pi(\boldsymbol{\alpha} \mid q_1, q_2) = \prod_{k=1}^{K} \pi(\alpha_k \mid q_1, q_2) \tag{3.7}$$

$$\alpha_k \mid q_1, q_2 \sim G(q1, q2) \tag{3.8}$$

$$\pi(\boldsymbol{\beta} \mid r_1, r_2) = \prod_{p=1}^{P} \pi(\beta_p \mid r_1, r_2) \tag{3.9}$$

$$\beta_p \mid r_1, r_2 \sim G(r1, r2) \tag{3.10}$$

$$\pi(\boldsymbol{\lambda} \mid u_1, u_2) = \prod_{n=1}^{N} \pi(\lambda_n \mid u_1, u_2) \tag{3.11}$$

$$\lambda_n \mid u_1, u_2 \sim G(u1, u2) \tag{3.12}$$

where $G(q_1, q_2)$ denotes the density of the Gamma distribution with mean $q_1 q_2$ and variance $q_1 q_2^2$ and $q_1, q_2, r_1, r_2, u_1, u_2$ are fixed known values. In what follows we assume that $q_1 = r_1 = u_1 = 0.01$ and $q_2 = r_2 = u_2 = 100$.

Let $\boldsymbol{\theta} = (\mathbf{w}_1, ..., \mathbf{w}_K, \mathbf{a}_1, ..., \mathbf{a}_P, \boldsymbol{\alpha}^T, \boldsymbol{\beta}^T, \boldsymbol{\lambda}^T)^T$ denote the set of all parameters stacked in row-major order, we have $\dim(\boldsymbol{\theta}) = R$ where $R = (K + P + 1)N + K + P$, and the

log of the posterior density is

$$\log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = \sum_{n=1}^{N} \left\{ -\frac{\lambda_n}{2} \sum_{t=P+1}^{T} \left[(y_{tn} - \mathbf{x}_t \mathbf{w}_n) - \tilde{\mathbf{e}}_{tn} \mathbf{a}_n \right]^2 \right\}$$
$$+ \frac{T-P}{2} \sum_{n=1}^{N} \log \lambda_n + \sum_{k=1}^{K} \left[-\frac{1}{2} \mathbf{w}_k (\alpha_k (\mathbf{S}^T S)) \mathbf{w}_k^T + \frac{1}{2} \log(|\alpha_k (\mathbf{S}^T \mathbf{S})|) \right]$$
$$+ \sum_{p=1}^{P} \left[-\frac{1}{2} \mathbf{a}_p (\beta_p (\mathbf{D}^T \mathbf{D})) \mathbf{a}_p^T + \frac{1}{2} \log |\beta_p (\mathbf{D}^T \mathbf{D})| \right] + \sum_{k=1}^{K} \left[(q_1 - 1) \log \alpha_k - \alpha_k / q_2 \right]$$
$$+ \sum_{p=1}^{P} \left[(r_1 - 1) \log \beta_p - \beta_p / r_2 \right] + \sum_{n=1}^{N} \left[(u_1 - 1) \log \lambda_n - \lambda_n / u_2 \right] + const \tag{3.13}$$

where $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_N)$ is the fMRI response data. Bayesian inference for the various components of $\boldsymbol{\theta}$ requires computation of the corresponding appropriately normalized posterior marginal distributions. Strategies for this Bayesian computation are described in what follows.

3.2.2 Algorithm A: Variational Bayes

Variational Bayes is an optimization approach for constructing a deterministic approximation to the posterior distribution. Let $q(\boldsymbol{\theta})$ be a density function having the same support as the posterior density $p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X})$, and let $\log p(\mathbf{Y}|\mathbf{X})$ denote the logarithm of the marginal likelihood associated with the model and the response $\mathbf{Y}$, which depends on the known design $\mathbf{X}$. We can express the logarithm of the marginal likelihood as

$$\begin{aligned} \log p(\mathbf{Y}|\mathbf{X}) &= \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\mathbf{Y}, \boldsymbol{\theta}|\mathbf{X})}{q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta} \\ &\quad + \int q(\boldsymbol{\theta}) \log \left\{ \frac{q(\boldsymbol{\theta})}{p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X})} \right\} d\boldsymbol{\theta} \\ &\geq \int q(\boldsymbol{\theta}) \log \left\{ \frac{p(\mathbf{Y}, \boldsymbol{\theta}|\mathbf{X})}{q(\boldsymbol{\theta})} \right\} d\boldsymbol{\theta} \equiv F(q) \end{aligned}$$

such that the functional $F(q)$ is a lower bound for $\log p(\mathbf{Y}|\mathbf{X})$ for any q. The approximation is obtained by restricting q to a manageable class of density functions, and maximizing F over that class. In this case the class of density functions over which the optimization is

carried out is characterized by densities that can be factored as follows:

$$q(\boldsymbol{\theta}) = \prod_{n=1}^{N} q(\mathbf{w}_n) \prod_{n=1}^{N} q(\mathbf{a}_n) \prod_{k=1}^{K} q(\alpha_k) \prod_{p=1}^{P} q(\beta_p) \prod_{n=1}^{N} q(\lambda_n). \tag{3.14}$$

Let $\mathrm{E}_{-q_i}[\cdot]$ denote the expectation under q for all parameters excluding the i^{th} parameter. A coordinate ascent algorithm is applied to locally maximize F based on update steps of the form

$$q(\theta_i) \propto \exp \mathrm{E}_{-q_i}[\log p(\mathbf{Y}, \boldsymbol{\theta}|\mathbf{X})] \tag{3.15}$$

which are iterated to convergence. Details can be found in Penny et al. (2003) and Jordan et al. (1999). As mentioned in Section 1, the resulting approximate posterior distribution can be a very good approximation or conversely a very poor approximation of the true posterior density. While there are a number of factors that govern the quality of the approximation, as far as we are aware, there is currently no theory characterizing the error associated with mean field VB. A simple approach is to compare the VB approximation with an appropriately implemented MCMC algorithm which has an associated large sample theory.

3.2.3 Algorithm B: Hamiltonian Monte Carlo

Hamiltonian Monte Carlo (HMC) has its origins with the work of Alder and Wainwright (1959) and Duane et al. (1987) and was popularized in the statistical literature by Neal (1995). It is a Metropolis-Hastings algorithm that can be used to sample high-dimensional target distributions far more efficiently than algorithms based on random walk proposals, where the proposals for HMC are based on Hamiltonian dynamics. The algorithm works by introducing a Hamiltonian $H(\boldsymbol{\theta}, \boldsymbol{\xi})$ defined as the sum of potential energy $U(\boldsymbol{\theta})$ and kinetic energy $K(\boldsymbol{\xi})$, and the dynamics are written as follows:

$$\begin{aligned}\frac{d\theta_i}{dt} &= \frac{\partial H(\boldsymbol{\theta},\boldsymbol{\xi})}{\partial \xi_i} = \frac{\partial K(\boldsymbol{\xi})}{\partial \xi_i} \\ \frac{d\xi_i}{dt} &= -\frac{\partial H(\boldsymbol{\xi},\boldsymbol{\theta})}{\partial \theta_i} = -\frac{\partial U(\boldsymbol{\theta})}{\partial \theta_i}\end{aligned}$$

The continuous variable t here denotes the time evolution of the dynamic system, i ($i = 1, ..., R$) denotes the i^{th} index of the corresponding random vector. $U(\boldsymbol{\theta}) = -p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X})$ is the negative log probability density function of the distribution for $\boldsymbol{\theta}$ that we wish to sample from, and $K(\boldsymbol{\xi})$ is defined as $K(\boldsymbol{\xi}) = \boldsymbol{\xi}^T \mathbf{M}^{-1} \boldsymbol{\xi}/2$ where $\boldsymbol{\xi}$ is an auxiliary random vector having the same dimension as $\boldsymbol{\theta}$. Here $\mathbf{M}$ is referred to as the 'mass matrix' and is typically assumed diagonal. In practice this system is solved using numerical integration techniques (Neal, 2011), most commonly the leapfrog method. For fixed $\delta > 0$ one step of the method is comprised of the following updates:

$$\boldsymbol{\xi}(t + \delta/2) = \boldsymbol{\xi}(t) - \delta/2 \frac{\partial U}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}(t)) \tag{3.16}$$

$$\boldsymbol{\theta}(t + \delta) = \boldsymbol{\theta} + \delta \mathbf{M}^{-1} \boldsymbol{\xi}(t + \delta/2) \tag{3.17}$$

$$\boldsymbol{\xi}(t + \delta) = \boldsymbol{\xi}(t + \delta/2) - (\delta/2) \frac{\partial U}{\partial \boldsymbol{\theta}}(\boldsymbol{\theta}(t + \delta)) \tag{3.18}$$

The leapfrog method iterates through a total of L such steps, and the resulting approximate solution is used as a proposed value for the next state of the Markov chain in the Metropolis-Hastings (MH) algorithm.

The algorithm requires repeated calculation of the unnormalized log-posterior density and its gradient. A fast way to calculate the log-likelihood components is thus crucial. Previous MCMC methods for models similar to the one considered here (e.g. Woolrich et al. (2004b)) compute the log-likelihood by directly summing across voxels n and time points t. As a more efficient alternative we propose a calculation of the log-likelihood that can omit the summation across t. Let $\mathbf{a}_n^* = (-\mathbf{1}, \mathbf{a}_n^T)^T$, so $a_{pn}^* = a_{pn}$ if $p \geq 1$ and

$a^*_{pn} = -1$ if $p = 0$. The log-likelihood contribution for voxel n can be expressed as:

$$l_n = -\frac{\lambda_n}{2}\mathbf{a}_n^{*T}\mathbf{F}\mathbf{a}_n^* + \frac{T-P}{2}\log\lambda_n + const. \tag{3.19}$$

where the specific form of $\mathbf{F}$ and its derivation is given in Appendix B. Under this formulation, the sum across t can be pre-computed rather than computed at every iteration of the algorithm. This changes the computational complexity of the likelihood evaluation from $O(TNKP)$ to $O(NK^2P^2)$. Since $K \times P$ is typically smaller than T, this can make the computation faster, in our experience 10 to 20 times faster for datasets of the size considered in Section 3. Based on this form of the log-likelihood the gradient of the log-posterior density is derived as:

$$\nabla w_{kn} \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = \lambda_n \mathbf{a}_n^{*T}\mathbf{G}\mathbf{a}_n^* - \alpha_k(\mathbf{S}^T\mathbf{S})_n\mathbf{w}_k^T \tag{3.20}$$

$$\nabla a_{pn} \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = \lambda_n \mathbf{f}_p\mathbf{a}_n^* - \beta_p(\mathbf{D}^T\mathbf{D})_n\mathbf{a}_p^T \tag{3.21}$$

$$\nabla \alpha_k \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = -\frac{1}{2}\mathbf{w}_k(\mathbf{S}^T\mathbf{S})\mathbf{w}_k^T + (\frac{N}{2} + q_1 - 1)/\alpha_k - \frac{1}{q_2} \tag{3.22}$$

$$\nabla \beta_p \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = -\frac{1}{2}\mathbf{a}_p(\mathbf{D}^T\mathbf{D})\mathbf{a}_p^T + (\frac{N}{2} + r_1 - 1)/\beta_p - \frac{1}{r_2} \tag{3.23}$$

$$\nabla \lambda_n \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = -\frac{1}{2}\mathbf{a}_n^{*T}\mathbf{F}\mathbf{a}_n^* + \frac{(T-P)/2 + u_1 - 1}{\lambda_n} - \frac{1}{u_2} \tag{3.24}$$

where $(\mathbf{S}^T\mathbf{S})_n$ and $(\mathbf{D}^T\mathbf{D})_n$ denotes the n^{th} row of $\mathbf{S}^T\mathbf{S}$ and $\mathbf{D}^T\mathbf{D}$ respectively. Specific derivations including the form of $\mathbf{G}$ and $\mathbf{f}_p$ are given in Appendix B.

There are a variety of block updating schemes that can be employed when updating the parameters in the MCMC algorithm. For simplicity, we have tried various component-wise updates and have found that component-wise updates lead to very poor mixing of the sampling chain. On the other hand, updating the entire parameter vector $\boldsymbol{\theta}$ as a single high-dimensional block works well and produces adequate mixing when HMC is applied to this model. Letting $*$ indicate the current state of the sampling chain, the HMC algorithm proceeds as in Algorithm 1. Software written in C++ implementing the HMC algorithm is

Algorithm 3 HMC for GLM-AR

1. Initialize the parameters $\boldsymbol{\theta}$, mass matrix $\mathbf{M}$, and Leapfrog step size δ and step number L.
2. Update $\boldsymbol{\theta}$:

   (a) Simulate latent vector $\boldsymbol{\xi}^* \sim \mathrm{N}(\mathbf{0}, \mathbf{I})$. Let $\boldsymbol{\theta}^{(0)} = \boldsymbol{\theta}^*$, $\boldsymbol{\xi}^{(0)} = \boldsymbol{\xi}^* + \frac{\delta}{2}\nabla_{\boldsymbol{\theta}} \log p(\boldsymbol{\theta}^* \mid \mathbf{Y})$

   (b) For $l = 1, ..., L$, let

   $$\boldsymbol{\theta}^{(l)} = \boldsymbol{\theta}^{(l-1)} + \delta/\mathbf{M}\boldsymbol{\xi}^{(l-1)}$$

   $$\boldsymbol{\xi}^{(l)} = \boldsymbol{\xi}^{(l-1)} + \delta^{(l)}\nabla \log p(\boldsymbol{\theta}^{(l)} \mid \mathbf{Y}, \mathbf{X})$$

   where $\delta^{(l)} = \delta$ for $l < L$ and $\delta^{(L)} = \delta/2$

   (c) Accept $\boldsymbol{\theta}^{(L)}$ as the new state for $\boldsymbol{\theta}$ with probability

   $$\alpha_a = \min(1, e^{-H(\boldsymbol{\theta}^{(L)}) + H(\boldsymbol{\theta}^*)})$$

   where $H(\boldsymbol{\theta}) = -\log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) + \boldsymbol{\xi}^T\mathbf{M}^{-1}\boldsymbol{\xi}/2$
   Else remain in the current state $\boldsymbol{\theta}^*$ with probability $1 - \alpha_a$.

3. Repeat step 2 for the desired number of samples.

available at: *http://www.math.uvic.ca/∼nathoo/publications.html*.

Tuning the HMC algorithm requires appropriate choice of $\mathbf{M} = \mathrm{diag}\{m_1, \ldots, m_R\}$, δ, and L. We choose $\delta = 0.00002$ as the initial value and adaptively adjust its value to obtain an optimal acceptance rate of around 0.65 (Beskos et al., 2013) for a given value of L. Larger values for L are useful in suppressing random walk behaviour of the chain, and we use $L = 250$ in this work. Aside from examining the acceptance rate, mixing is judged from the output based on looking at the traceplots of some parameters specific to randomly chosen voxels, and we typically examine the traceplots of hyper-parameters as these components of the chain often will mix slower than components corresponding to parameters higher up in the model hierarchy. Mixing is also judged based on estimation of the batch means Monte Carlo standard error (BMSE) (Fishman and Yarberry (1997)), a measure that is easy to implement and is widely used in practice.

As different parameters tend to have different scales, setting m_i can also be important,

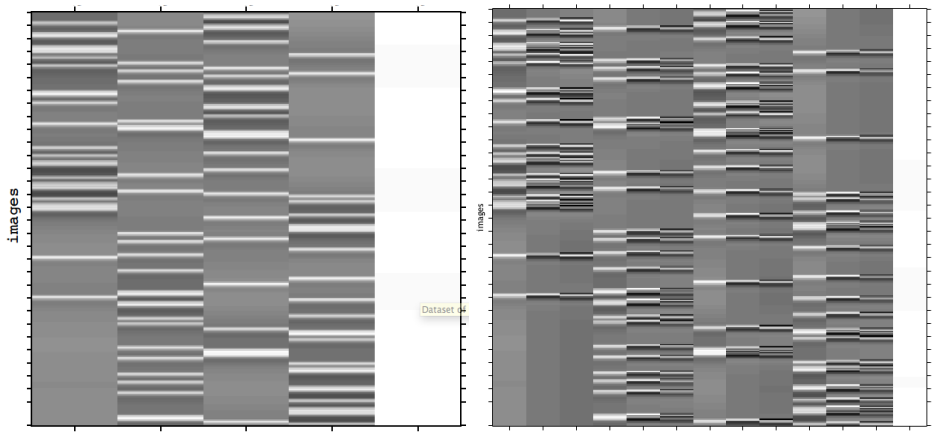
and this issue is discussed extensively in Neal (2011). In practice, we have found that for problems having moderate dimension and complexity, setting all $m_i = 1$ $(i = 1, ..., R)$ is sufficient (e.g, Simulation 3.1). As the model complexity and dimensionality increases, we set the m_i to be roughly proportional to the reciprocal of the posterior variance of the i^{th} parameter for i=1,...,R. This variance, of course, is unknown so it is estimated based on a preliminary run of HMC with $m_i = 1$ $(i = 1, ..., R)$. This process is iterated a few times until adequate mixing of the chain is observed based on its output and the measures described above. We use this approach to tune the values of M in the application considered in Section 3.3.

3.3 Results

We conduct two simulation studies to compare features of the posterior distributions obtained from HMC and VB. This is followed by a real data analysis where we compare the results obtained from HMC, VB, and the traditional mass univariate approach. The simulation studies and application are based on the face-repetition dataset discussed in Henson et al. (2002). A detailed description of this dataset can be found online at *http://www.fil.ion.ucl.ac.uk/spm/data/*. The data are collected as part of an event-related fMRI study in which greyscale images of faces were repeatedly presented to a subject for 500 ms replacing the baseline, an oval chequerboard, that was present throughout the inter stimulus interval. Each of the faces were presented twice; some were familiar to the subject while others were not. This setup leads to four experimental conditions $U1$, $U2$, $F1$, $F2$, representing familiar or unfamiliar(F/U) faces observed for the first or second(1/2) time.

The fMRI signal is measured at $T = 351$ time points during the experiment. The design matrix used in the analysis has $(T - P)$ rows and K columns. In our first simulation study we set $K = 5$ corresponding to the four experimental conditions convolved with the

canonical HRF, plus a constant term. The design matrix is depicted in Figure 3.1a. In the second simulation study we consider a larger design matrix where each of the four study conditions is convolved with the canonical HRF, its dispersion derivative and its temporal derivative, respectively, resulting in $K = 13$ columns (the last column corresponding to a constant term). The design matrix for the second simulation study is depicted in Figure 3.1b.



(a) (b)

Figure 3.1: Design matrix for simulation study one (a) and simulation study two (b). In panel (a), the first four columns correspond to stimuli U1, U2, F1, F2 convolved with the canonical HRF respectively. In panel (b), the 1st, 4th, 7th, and 10th columns are convolved with the canonical HRF, the 2nd, 5th, 8th, and 11th columns are convolved with its temporal derivative, the 3rd, 6th, 9th, and 12th columns are convolved with its dispersion derivative. The last blank column in both panels (a) and (b) represents the constant term.

We set the spatial domain to be a 2-dimensional lattice divided into a 53×63 grid, and then a brain-shaped mask is applied to this lattice, resulting in $N = 2087$ voxels for the domain that our simulation studies are carried out on. The true values of the parameters $\mathbf{W}$, $\mathbf{A}$, and noise variables $\mathbf{z}_1, ..., \mathbf{z}_N$ are simulated based on model assumptions and fixed values of $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\lambda}$ discussed below. Given the parameter values, the data $\mathbf{Y}$ are simulated from the model and 100 replicate datasets are simulated in each study.

To compare VB and HMC with respect to point estimation, we use the simulation replicates and the known true values of the model parameters to estimate the average squared

bias (ASBIAS) and the average mean squared error (AMSE) of estimators based on the posterior mean, where the average is taken across voxels. To compare the two approaches with respect to posterior variability we use the average marginal variance (AVAR). Letting $\hat{w}_{knj}$ denote the posterior mean estimate of w_{kn} obtained from the j^{th} $(j = 1, ..., J)$ simulation replicate, and $\sigma^2(\hat{w}_{knj})$ denote the corresponding posterior variance, the three measures above for $\mathbf{w}_k$ (where k corresponds to the k^{th} regressor) are computed as:

$$\text{ASBIAS}(\mathbf{w}_k) = \frac{1}{N}\sum_{n=1}^{N}(\sum_{j=1}^{J}\hat{w}_{knj}/J - w_{kn})^2 \tag{3.25}$$

$$\text{AMSE}(\mathbf{w}_k) = \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}(\hat{w}_{knj} - w_{kn})^2 \tag{3.26}$$

$$\text{AVAR}(\mathbf{w}_k) = \frac{1}{NJ}\sum_{n=1}^{N}\sum_{j=1}^{J}\sigma^2(\hat{w}_{knj}) \tag{3.27}$$

These same measures are applied to the autoregressive coefficients $\mathbf{a}_p$. We also compute the correlation of each estimated $\mathbf{w}_k$ and $\mathbf{a}_p$ vectors with the truth, and average these correlations across simulation replicates. To compare VB and HMC with respect to the spatial smoothness of the estimated images we use Moran's I (Moran, 1950). Negative values indicate negative spatial autocorrelation and positive values indicate positive spatial autocorrelation, a zero value corresponds to no spatial dependence. We compute Moran's I for each image of estimated parameters and then average these values (AMoran) across the J simulation replicates. For $\mathbf{w}_k$ this measure takes the form

$$\text{AMoran} = \frac{1}{J}\sum_{j=1}^{J}\frac{N}{\sum_{n_1}\sum_{n_2}\phi_{n_1n_2}}\frac{\sum_{n_1}\sum_{n_2}\phi_{n_1n_2}(w_{kn_1j} - \bar{w}_{kj})(w_{kn_2j} - \bar{w}_{kj})}{\sum_{n_1}(w_{kn_1j} - \bar{w}_{kj})^2} \tag{3.28}$$

where $\bar{w}_{kj} = \sum_{n=1}^{N} w_{knj}$, $\phi_{n_1n_2}$ is the weight for voxel pair (n_1, n_2) $(n_1 = 1, ..., N, n_2 = 1, ...N)$, and here this is chosen as the reciprocal of the distance between the centroids of n_1 and n_2.

3.3.1 Simulation Study I

We assume in this case that the data generating mechanism corresponds to a first-order autoregressive process. In simulating the true values of the regression coefficients and autoregressive coefficients we assign equal values to the precision of the regression coefficients, $\alpha_k = 1$ $(k = 1, ...5)$ and we set $\beta_1 = 1000$ which will result in auto-regressive coefficients having much smaller values than the regression coefficients. For the precision of the noise we simulate these values from a Gamma distribution $\lambda_n \overset{\text{i.i.d}}{\sim} G(10, 10)$ $(n = 1, ..., N)$.

Both VB and HMC are applied to the simulated datasets and images depicting the average (over simulation replicates) posterior mean estimates obtained from both methods and the true values are shown in Figure 3.2, where we show the images corresponding to $\mathbf{w}_1$ and $\mathbf{a}_1$. Figures depicting comparisons for the full set of parameters are shown in Figures 1-2 of the Supplementary Material. In this case the results obtained from HMC and VB are very similar and both correspond well with the truth.

The summary statistics discussed above are computed and their values are listed in Table 3.1. As the VB implementation in SPM does not provide the posterior variance of the auto-regressive coefficients as part of its output, we leave these cells blank in the table (including those for HMC since comparisons are of interest) . The statistics corresponding to HMC in the table are the actual values while those for VB are expressed as the percentage of the corresponding values obtained from HMC. From the table, we can see that VB tends to produce smaller squared bias than HMC, but the MSE is roughly equivalent. The posterior variance statistics obtained from VB are also fairly close to those obtained from HMC, with slightly larger values for the former. Thus the over-confidence problem sometimes associated with VB (Bishop, 2006), (Nathoo et al., 2013) does not seem to be an issue in this case. Both algorithms are performing well in terms of point estimation as they

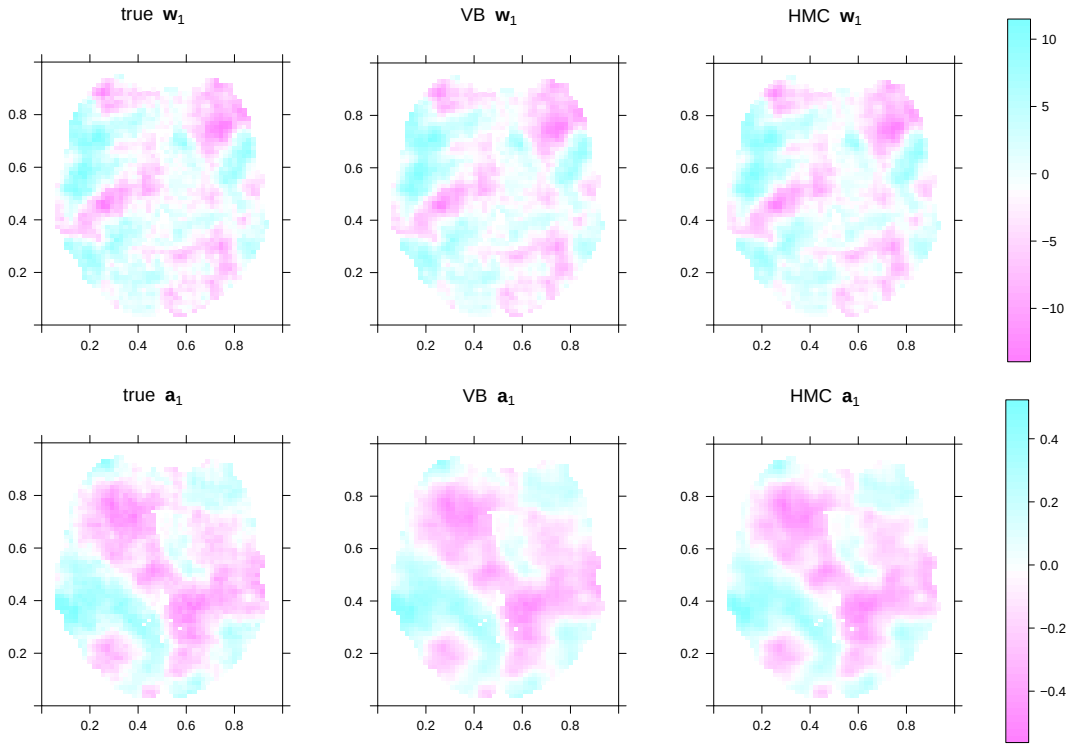


Figure 3.2: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_1$ and $\mathbf{a}_1$ from HMC and VB. The estimates are compared with true image in each row.

achieve a high level of correlation (around 0.99) with the true values. In terms of Moran's I, the images estimated using VB and HMC have approximately the same amount of spatial autocorrelation in their posterior estimates, and both are similar to the true Moran's I. In summary, VB and HMC both perform adequately well in this study.

methods	measure	W1	W2	W3	W4	W5	A1
true	Moran's I	0.121	0.169	0.136	0.187	0.122	0.179
HMC	ASBIAS	0.123	0.120	0.105	0.110	0.001	4.52E-04
	AMSE	0.405	0.452	0.412	0.420	0.007	1.19E-03
	AVAR	0.411	0.468	0.425	0.435	0.008	
	Correlation	0.997	0.999	0.998	0.998	1.000	0.995
	Moran's I	0.123	0.171	0.137	0.189	0.122	0.182
VB	ASBIAS	67%	58%	65%	72%	77%	91%
	AMSE	107%	104%	103%	105%	103%	104%
	AVAR	112%	109%	108%	108%	105%	
	Correlation	100%	100%	100%	100%	100%	100%
	Moran's I	100%	100%	100%	100%	100%	102%

Table 3.1: Summary statistics for Simulation Study I. The results from VB are presented as a percentage of those obtained HMC. The true value of Moran's I is listed for each regressor in the first row as a reference.

Comparing the two algorithms with respect to computation time on a standard iMac with 3.2 GHz Intel Core i5. HMC (coded in C++) takes 23min for 3000 iterations while VB takes 1min per simulated dataset.

3.3.2 Simulation Study II

In the second simulation study we aim to further compare the performance of the two algorithms in a harder and more complex situation, by including more coefficients with these coefficients having unequal variance in the sense described below. Specifically, we extend the design matrix to include the canonical HRF, its temporal derivative, and its dispersion derivative. By convolving these functions with the four stimuli we get 13 regressors (with the last corresponding to the constant term). We also increase the order of the auto-regressive process from $P = 1$ to $P = 3$. The precision parameters are set as follows: $\alpha_1 = \alpha_2 = \alpha_3 = 0.1$, $\alpha_4 = \alpha_5 = \alpha_6 = 0.5$, $\alpha_7 = \alpha_8 = \alpha_9 = 1.0$, $\alpha_{10} = \alpha_{11} = \alpha_{12} = 2.0$, $\alpha_{13} = 1.0$. $\beta_1 = 1000$, $\beta_2 = 2000$, $\beta_3 = 5000$. The values for the noise precision are again generated as $\lambda_n \overset{\text{i.i.d}}{\sim} Gamma(10, 10)$ $(n = 1, ..., N)$.

Figure 3.3 shows the image of the average (over simulation replicates) posterior mean estimates from HMC and VB for $\mathbf{w}_1$ and $\mathbf{a}_1$. Similar Figures for the remaining parameters are shown in the Supplementary Material, Figures 3-8. Both HMC and VB appear to provide similar estimates which correspond well with the truth.

The summary statistics are computed as before and these are presented in Table 3.2. Generally, the observations made in Simulation Study I seem to carry over in that VB tends to produce smaller bias in point estimation but roughly equivalent MSE. Examining the average marginal posterior variance again indicates that VB does not exhibit an over-confidence problem in this case. The average correlation between the estimates and the truth obtained from HMC and VB are nearly the same, as seen in Study I. The measures of spatial correlation based on Moran's I are also again roughly equivalent for the two

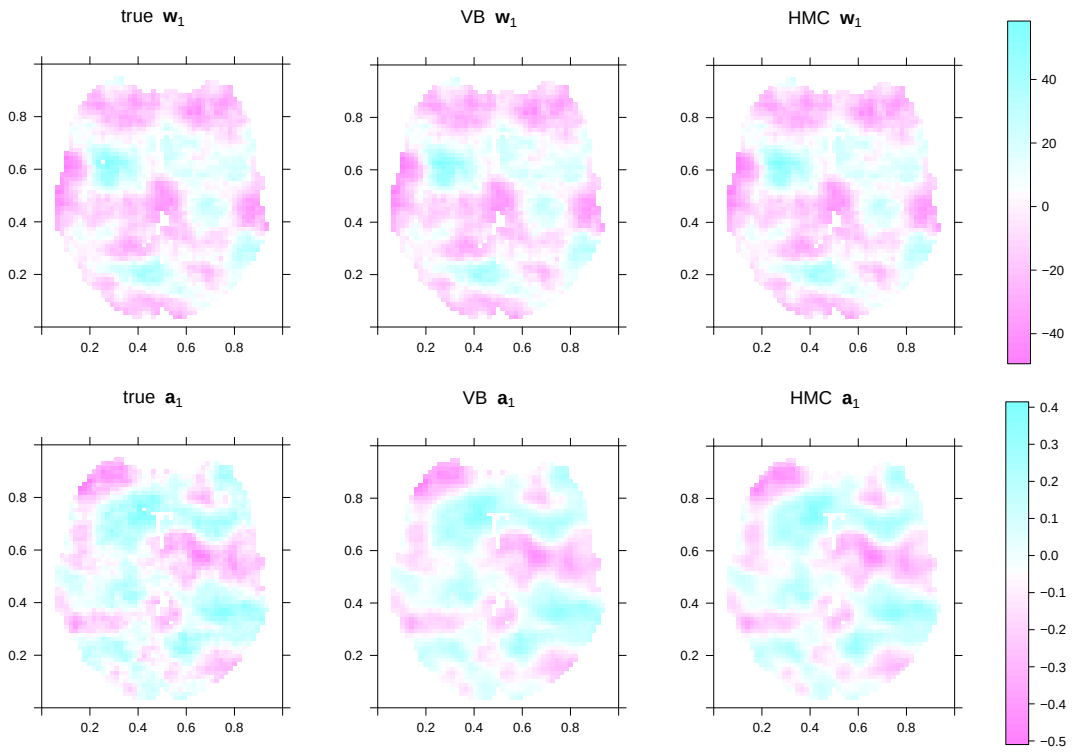


Figure 3.3: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_1$ and $\mathbf{a}_1$ from HMC and VB. The estimates are compared with true image in each row.

approaches.

In terms of timing, HMC takes 6.6 hours for 3000 iterations while VB takes 1 minute for a single simulation replicate.

3.3.3 Real Application

In this section, we will compare the estimation results from HMC, VB and the classical mass univariate approach (MUA) to examine possible differences in a real dataset obtained from a single subject. The dataset we focus on is again the face-repetition dataset; however, we now use the actual data and fit the model over the entire 3-dimensional brain volume based on a 3-dimensional grid having dimensions $53 \times 63 \times 52$ with a total of 56526 voxels.

Pre-processing steps are conducted in SPM12: All functional images are aligned to the first image using a six-parameter rigid-body transformation. All the time series are

methods	measure	W1	W2	W3	W4	W5	W6	W7	W8
true	Moran's I	0.111	0.137	0.151	0.144	0.125	0.128	0.121	0.109
HMC	ASBIAS	0.054	1.097	0.711	0.115	0.973	0.840	0.108	0.860
	AMSE	0.610	4.336	3.566	0.549	2.317	2.160	0.444	1.840
	AVAR	0.617	4.181	3.466	0.561	2.244	2.112	0.459	1.807
	Correlation	1.000	0.998	0.999	0.999	0.991	0.992	0.998	0.982
	Moran's I	0.111	0.139	0.152	0.145	0.128	0.132	0.123	0.112
VB	ASBIAS	73%	61%	57%	63%	89%	90%	78%	102%
	AMSE	102%	101%	100%	104%	98%	98%	102%	97%
	AVAR	101%	114%	110%	106%	107%	105%	104%	100%
	Correlation	100%	100%	100%	100%	100%	100%	100%	100%
	Moran's I	100%	100%	100%	100%	102%	102%	101%	106%
		W9	W10	W11	W12	W13	A1	A2	A3
true	Moran's I	0.104	0.148	0.189	0.130	0.128	0.108	0.127	0.174
HMC	ASBIAS	0.761	0.123	0.639	0.576	0.002	4.71E-04	3.72E-04	3.13E-04
	AMSE	1.639	0.369	1.197	1.203	0.009	1.19E-03	8.95E-04	5.58E-04
	AVAR	1.607	0.384	1.126	1.211	0.009			
	Correlation	0.980	0.996	0.977	0.983	1.000	0.992	0.988	0.975
	Moran's I	0.108	0.151	0.198	0.133	0.128	0.111	0.129	0.182
VB	ASBIAS	116%	88%	103%	94%	99%	96%	123%	99%
	AMSE	102%	103%	102%	96%	102%	105%	101%	98%
	AVAR	97%	104%	109%	104%	100%			
	Correlation	100%	100%	100%	100%	100%	100%	100%	100%
	Moran's I	108%	101%	107%	104%	100%	102%	105%	105%

Table 3.2: Summary statistics for Simulation Study I. The results from VB are presented as a percentage of those obtained HMC. The true value of Moran's I is listed for each regressor in the first row as a reference.

interpolated to the acquisition time of the 12^{th} slice. Images are also spatially normalized to a standard EPI template using a non-linear warping method. For MUA, the data are also pre-smoothed using a Gaussian kernel with FWHM of 8mm. We computed the global mean g of all time series and scaled each time series by $100/g$; to remove low frequency drift each time series was also high pass filtered using a default cutoff of $128s$. The design matrix is the same as that considered in Simulation Study I, shown in Figure 3.1a. We fit the model with an autoregressive order of $P = 1$ as in Penny et al. (2005).

Both HMC and VB are initialized with starting values obtained from applying ordinary least squares regression (OLS) at each voxel. The hyper-parameters of the prior for the two algorithms are the same as those used previously, which corresponds to the default in the SPM software. For the mass matrix $\mathbf{M}$ in HMC, we use the tuning method described in Section 2.3. The trace plots for select parameters are displayed in the Supplementary Material, Figures 9-12, and these indicate adequate mixing of the sampling chain.

We note that the SPM implementation of VB when applied to analyze data over the whole brain volume uses a graph-partitioning algorithm (Harrison et al. (2008b)). This works by dividing the whole brain into several disjoint regions and in each region the VB estimation is carried out independently. For this particular dataset, the graph partitioning algorithm divided the brain into 38 regions. Although this has the advantage of saving computational time, we find that this produces some artifacts as indicated below.

To compare the three methods with respect to point estimation we compute the correlation (across voxels) of the estimates, and these values are presented in Table 3.3 which displays the correlation for each of the five regression coefficients $\mathbf{w}_1$ to $\mathbf{w}_5$ comparing VB and MUA to HMC. We see that HMC and VB have estimation (posterior mean) results that are highly correlated. The correlation between HMC and MUA for the intercept is only 0.66; we suspect that pre-smoothing of the data (MUA) might be causing this relatively low value.

Correlation	$\mathbf{w}_1$	$\mathbf{w}_2$	$\mathbf{w}_3$	$\mathbf{w}_4$	$\mathbf{w}_5$
(VB, HMC)	0.91	0.93	0.92	0.91	1.00
(MUA, HMC)	0.87	0.84	0.84	0.83	0.66

Table 3.3: Correlation (across voxels) in the estimated regression coefficients obtained from HMC and VB, and HMC and MUA.

Images depicting the estimated coefficients are shown in Figures 3.4 and 3.5. Due to space restrictions we only display the estimates of $\mathbf{w}_1$ and $\mathbf{a}_1$ on the 26^{th} plane out of 52 planes along the z-axis. Additional figures displaying estimates for the other regression coefficients are presented in the Supplementary Material, Figures 13-14. As seen in the simulation studies, HMC and VB yield very similar posterior mean estimates in terms of auto-regressive coefficients. In terms of regression coefficients the estimates from HMC seem to be a bit smoother than those from VB, but still similar in general. Estimates from MUA seem to exhibit a greater degree of spatial smoothing.

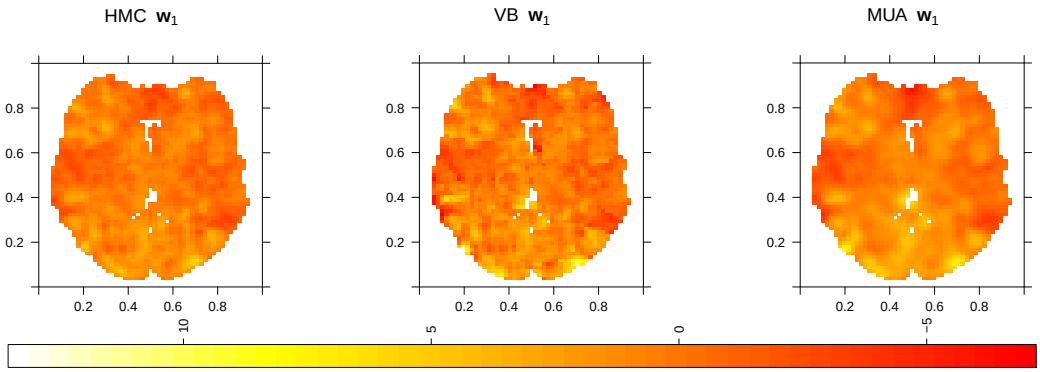


Figure 3.4: Posterior mean estimates of $\mathbf{w}_1$ on the 26^{th} plane out of 52 planes along the z-axis.

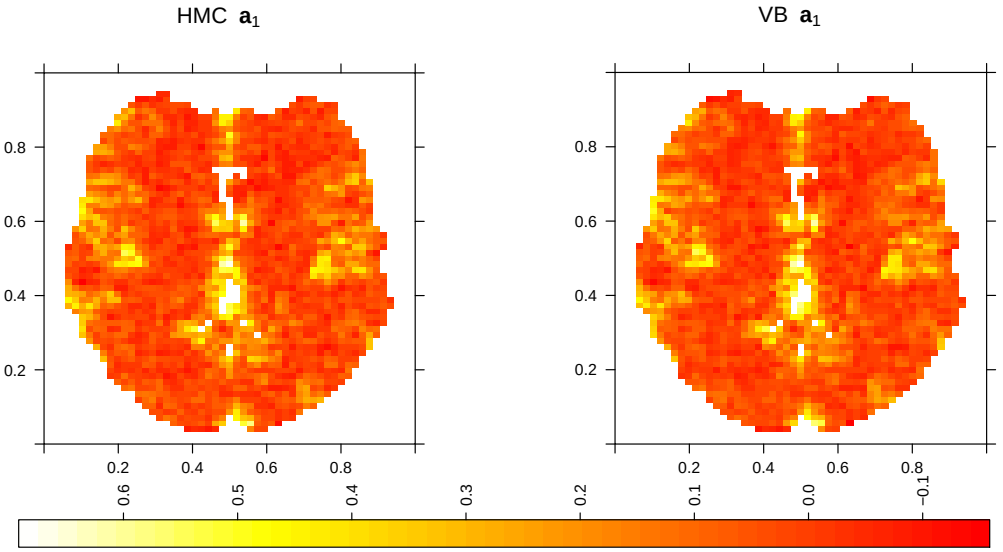


Figure 3.5: Posterior mean estimates of $\mathbf{a}_1$ on the 26^{th} plane out of 52 planes along the z-axis.

To compare VB and HMC with respect to the posterior marginal variance of the regression coefficients, we take the log-ratio of the posterior marginal variance obtained from VB over that obtained from HMC at each voxel, and examine these log-ratio values across all voxels. Doing so we find that for a great proportion of voxels, VB is actually over-estimating the posterior marginal variance relative to HMC. This is unexpected as it is more often the case that VB tends to underestimate posterior variance. After closer examination we suspect that this overestimation may be arising as a result of the graph-partitioning algorithm used in the SPM implementation of VB. This is demonstrated in

Figure 3.6 which depicts an image of the log-ratio marginal-variance values for a single slice for $\mathbf{w}_1$ alongside the graph-partitioned regions, and also in the Supplementary Material, Figure 15, which shows similar images for all of the regression coefficients. From the figures we see that the locations where the posterior marginal variance obtained from VB is higher than that obtained from HMC tend to align with the boundaries of the graph-partitioned regions. We further note that HMC and VB tended to produce similar values of the posterior marginal variance in our simulation studies, and that the graph partitioning algorithm is not used in the 2-dimensional case. It appears that the graph partitioning leads to the over-estimation of the posterior variance in this case, as there would be no spatial smoothing across the boundaries of the graph-partitioned regions.

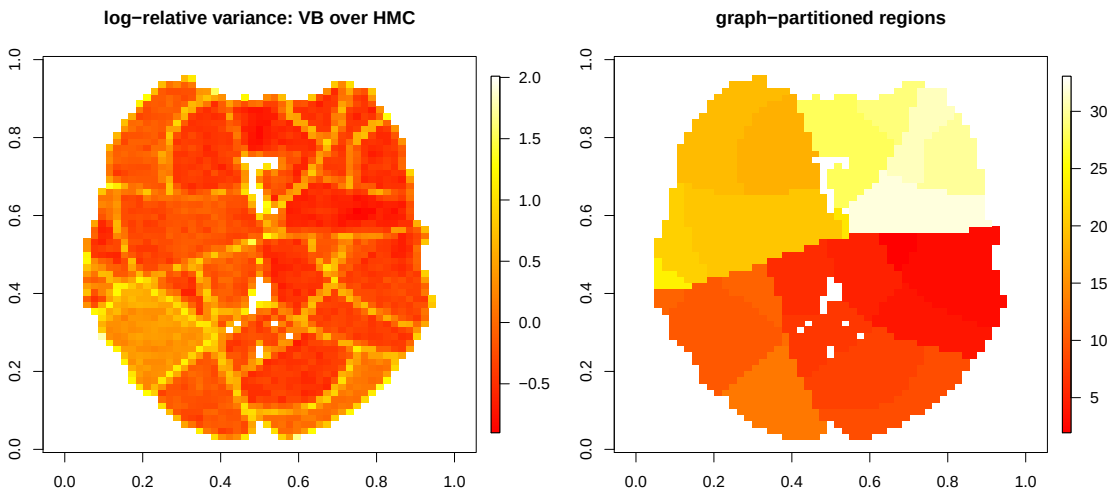


Figure 3.6: Log-relative ratio of the marginal posterior variance of the regression coefficient obtained from VB over that obtained from HMC. The yellow regions in the left image indicate locations where VB results in greater posterior variance relative to HMC for $\mathbf{w}_1$, the right image shows the graph-partitioned regions. Both are from the 26^{th} plane out of 52 planes along the z-axis.

We next examine and make comparisons with respect to activations. We do this by first defining a contrast vector $\mathbf{c} = (1, 1, 1, 1, 0)^T/4$. We multiply this vector by $\mathbf{w}$, where $\mathbf{w}$ denotes the vector of regression coefficients at a given voxel, to get a contrast (or effect

size) $\mathbf{c}^T\mathbf{w}$. We note that this contrast measures the effect of faces in the experiment at a given voxel. The posterior distribution of the contrast is then shown across voxels using a posterior probability map (PPM). This map is based on two thresholds, the first being an effect size threshold γ_e and the other being a probability threshold γ_p. The value of γ_e is set to be 1% greater than the global mean (across voxels) of $\mathbf{c}^T\mathbf{w}$ (Ashburner et al. (2014)). The value of the probability threshold is set to be $\gamma_p = 0.95$. At each voxel we then compute, using the posterior distribution,

$$Pr(\mathbf{c}^T\mathbf{w} > \gamma_e) \tag{3.29}$$

and we highlight those voxels where the posterior probability is greater than $\gamma_p = 0.95$. The PPM's obtained from HMC and VB are depicted in Figure 3.7.

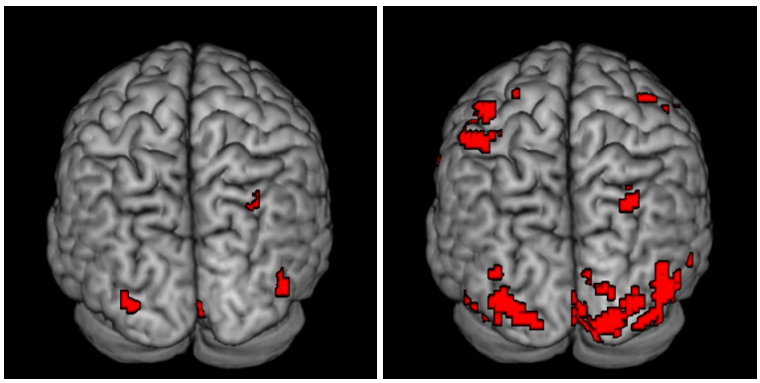


Figure 3.7: PPM showing the activated voxels, with an effect size threshold of 1% greater than the global mean and a probability threshold of 95%. The left map is obtained from HMC and right map is obtained from VB. The activations are displayed as red dots on a 3-d surface from the posterior view.

The PPM's obtained from the two approaches are generally similar, though with more voxels indicated as activated by VB in this particular case. In terms of timing, HMC takes 8.42 hours for 3000 iterations, VB takes 36 minutes, MUA takes 36 seconds with all computations performed on a standard iMac with with 3.2 GHz Intel Core i5.

3.4 Discussion

In this paper we have compared HMC and mean field VB for Bayesian inference in the spatial GLM-AR model. Comparisons were made in two simulation studies with a 2-dimensional grid and an actual single subject fMRI dataset based on a 3-dimensional grid. We found that for this particular model, under the settings considered, that HMC and VB provide similar estimates of the posterior distribution, both in terms of point estimation and also somewhat surprisingly in terms of posterior variability. In Section 3.3 we found visible differences when comparing the classical and Bayesian approaches. The classical approach does not assume spatial priors and the data are pre-smoothed so this is not unexpected. Differences seen when comparing HMC to VB in Section 3.3 seem largely due to the graph-partitioning algorithm used in the SPM implementation of VB, where VB tends to over-estimate the posterior marginal variance along the edges of the graph-partitioned regions. In terms of timing, HMC is considerably slower than VB as expected. This is based on running the HMC algorithm for 3000 iterations with the final 1000 iterations used to estimate features of the posterior distribution. We have also run a test case with a much larger Monte Carlo sample of 30000 iterations with the final 15000 iterations used to estimate features of the posterior distribution and have found the results to be very similar to those obtained with the smaller Monte Carlo sample size. Overall, for this particular model and for the settings considered here, our work justifies the use of mean field VB and its implementation in SPM based on our comparisons with the results obtained from HMC. Our work also speaks more generally to the issue of variational Bayes inference and the importance of checking the accuracy of variational Bayes approximations as there is currently no theory that we are aware of guaranteeing the accuracy of these approximations.

CHAPTER IV

Bayesian Analysis of fMRI Time Series with Spatially-Varying Autoregressive Orders

4.1 Introduction

In the analysis of functional magnetic resonance imaging (fMRI) data a key challenge is dealing with spatial and temporal correlation. The temporal correlation can arise from many sources, including scanner drift at very low frequencies, slow vascular/metabolic oscillations that are typically of moderate to low frequency, and some other sources of noise such as breathing and heartbeat. Simply ignoring these autocorrelations is dangerous and may lead to increase false positive discoveries (Makni et al., 2006). To deal with these issues, a variety of approaches have been proposed. One commonly used approach, namely "prewhitening", works by estimating the autocorrealtion in the errors and then de-correlate the noise using the estimates. Representative includes the AR process by Bullmore et al. (1996) and autoregressive-moving average (ARMA) model by Locascio et al. (1997). Besides these stationary time series models, a non-stationary $1/f$ models are also proposed (Zarahn et al., 1997; Bullmore et al., 2004). According to Friston et al. (2000), prewhitening can produce an extraneous source of bias. Alternatively, a band-pass filtering known as coloring can be applied to the data first, followed by the application of some models to deal with the autocorrelation in the colored data. For a review and discussion of these approaches the reader is referred to Woolrich et al. (2001). While

high-pass filtering has proven to be beneficial in increasing the power of the statistical analysis, the low-pass filtering involved in coloring is considered controversial in that it tends to add autocorrelation into the data (Skudlarski et al., 1999; Della-Maggiore et al., 2002).

While accurate temporal modeling is important for estimation of the fMRI signal in the presence of noise, traditional approaches for data analysis do this while ignoring the spatial correlation and apply the temporal model at each voxel independently. More specifically, this mass univariate approach, considered to be the classical approach to the analysis of fMRI data, includes a smoothing step involving a spatial Gaussian filter that is applied to the data first (Friston et al., 1995), followed by model estimation at each voxel, and then statical inference is based on random field theory (Worsley and Friston, 1995) which is applied to adjust for multiplicity in the spatial domain. While this approach remains the most common approach for analyzing fMRI data it has been criticised on a number of grounds. For example, the Gaussian kernel that is used to smooth the data has to be pre-specified and introduces artificial correlation into the data. In addition, this approach does not directly account for spatial correlation in the model.

Partly as a result of these criticisms, Bayesian models with spatial structured priors have been proposed which allow for the calculation of posterior probability maps (PPM) for activation, where this inference is based on an explicit spatial modelling and does not require smoothing the data with a Gaussian kernel nor does it require the use of random-field theory-based adjustments for multiplicity. A variety of spatial-temporal Bayesian models have been proposed. One model that is widely used and implemented in the SPM software is the GLM-AR model (Penny et al., 2003, 2005, 2007), which assumes that the data can be decomposed into two sources of variability. The first source is the product of a design matrix for the fMRI experiment convolved with a haemodynamic response

function (HRF) and experimental factors, and the second source represents temporally correlated noise which can be modeled using a low-order AR structure. In addition, the regression coefficients and the autoregressive coefficients vary across voxels and are assigned spatial smoothing priors. Gössl et al. (2001) has proposed a model where the data are decomposed into three sources, a spatial stimulus, a deterministic trend and a white noise process. However, this modelling approach may not account for some higher frequency stochastic noise components. Woolrich et al. (2004b) assumed that the temporal noise arises from both large scale and small scale variation, and built a space-time simultaneously auto-regressive model that accounts for both scales of variation. Methods focusing on spatial variable selection have also been proposed (see, e.g., Bezener et al. (2016), Lee et al. (2014),Musgrove et al. (2016)); while Kim et al. (2010) proposed a mixture of experts model to represent spatial activation clusters. While these models have a number of different characteristics which make the approaches unique, most of them commonly assume a homogeneous, low order AR or ARMA process for the temporal noise. By homogeneous, we mean that the order of the AR or ARMA process is assumed constant across all voxels of the brain. This assumption is also made in Penny et al. (2003); however, as we demonstrate using a simple empirical example in the next section, this homogeneous AR assumption may not be appropriate with real fMRI data.

Instead of formulating the model at each voxel and then adopting spatial smoothing priors for parameters across voxels, another branch of research is based on vector autoregressive (VAR) processes, see Harrison et al. (2003). This approach allows for time-lagged dependence across voxels and spatial-temporal interaction but fitting these models across a large number of voxels is computationally intractable and low-rank approximations have to be used. These models are also useful for studying effective brain connectivity, where one time course is used to predict the other (Castruccio et al., 2016; Chang and Glover, 2010).

Another line of work chooses to model the temporal noise as a $1/f$ long memory process, and applies discrete wavelet transforms (DWT) towards fitting the model (see, e.g., Jeong et al. (2013); Bullmore et al. (2004); Fadili and Bullmore (2002); Meyer (2003)). While this approach seems promising, our focus in this paper will be with modeling the short term memory using the classical AR process and spatial priors. The reason we choose to work with the AR process is because of its mathematical amenability and simplicity, and its wide use in different areas of science. A novel aspect of our work is that we allow the data to determine the order of the AR process at each voxel using ideas from Bayesian spatial variable selection.

Computation is an important issue when considering Bayesian spatial-temporal models for fMRI data. While the main focus of this paper lies with the development of a new model, another aspect of this work is the comparison of fully Bayes and approximate Bayesian computation methods. Due to the computational burden associated with fitting models to high-dimensional brain imaging data, approximate Bayesian methods have received considerable attention in the neuroimaging literature. One such method is the variational Bayes (VB) inference (Penny et al., 2003, 2007; Woolrich et al., 2004a). As there are currently no theoretical results quantifying the accuracy of VB methods (in contrast to MCMC which is justified by the large sample theory of stationary Markov chains), the evaluation of VB has to be performed on a case-by-case basis. In some cases, the performance of VB can be quite good and in other cases it can be quite poor. In this paper, besides the implementation of our new model based on a suitably designed MCMC sampler, we have implemented an MCMC algorithm to fit the original GLM-AR model which we compare to our new model as well as to the VB implementation in SPM. Our studies indicate that under a low signal-to-noise (SNR) ratio the accuracy of MCMC will outperform VB according to several criteria. This finding can be considered an extension

to our previous work in Chapter III.

4.1.1 Motivating example

Our motivating example comes from a single subject in a fMRI experiment examining a face-repetition stimulus. The experiment involves the presentation of either famous faces (F) or non-famous faces (N) with each type of face being presented two times. After convolving each of them with three types of haemodynamic response functions (HRFs), this leads to a design matrix having twelve columns plus one extra column for an intercept term in the regression model. After performing the necessary pre-processing steps as described in Penny et al. (2005), we fit a simple linear regression at each of the voxels. After obtaining the residuals from each voxel-specific fit, we fit an AR process up to order 12 for each voxel using the "ar" function in R. We then selected the optimal AR orders based on the AIC criteria, with the output figure shown in Figure 4.1.

The figure shows considerable variablity in the estimated AR order across voxels. While most of the AR orders are smaller than 4, higher orders up to 12 do exist in some of the voxels. Furthermore, these estimated AR orders tend to show some extent of spatial clustering. If, as is often done, we simply model the data using a homogeneous low-order AR process, then the voxels with higher order estimated AR orders would be incorrectly modelled, and this inaccuracy in the modeling of temporal noise might in turn have an impact on the inference on the covariates of interest, resulting in potentially false inferences about brain activation. To address this issue, we propose a spatially varying autoregressive order (SVARO) model, where the AR orders at each voxel are assumed to be heterogeneous across the brain. This is made possible by adopting a spike-and-slab prior with a stochastic search variable selection scheme. The spatial clustering of AR orders are incorporated by imposing an Ising prior (Ising, 1925) as the latent indicator for the spike and slab indicator variables across voxels. We update the latent indicators using the

Swendsen-Wang algorithm (Swendsen and Wang, 1987) alternating with Gibbs sampling in our MCMC algorithm. To prevent the phase transition problem associated with the Ising model, we derive theoretical bounds as in Li et al. (2015) and use these bounds to prevent the problem. We compare our model with the GLM-AR model of Penny et al. (2007) (implemented under two schemes: our self written MCMC sampler and the VB algorithm available in the SPM software) in terms of bias, variance, MSE, sensitivity as well as the log-pseudo marginal likelihood (Geisser and Eddy, 1979). We conduct these comparisons using two simulation studies and also on the real motivating data set.

The rest of the paper is organized as follows: In Chapter 4.2 we discuss the model formulation and the MCMC sampling method. Chapter 4.3 presents the simulation studies. Chapter 4.4 presents an analysis of the face-repetition data set. Finally, in Chapter 4.5 we provide a discussion and outline some possible directions for future work.

4.2 Methods

4.2.1 The model

We let P denote the maximum possible order of the AR process at each voxel while K denotes the number of regression coefficients representing the mean in the model at each voxel. Using similar notation as in Chapter III, for voxel n $(n = 1, ...N)$, we let $\mathbf{y}_n$ denote the observed time series of length T. For simplicity, our model is specified conditional on the first P observations at each voxel so that the likelihood function is constructed based on the model of the remaining $T - P$ observations in the time series. We let $\mathbf{X}$ denote the $(T-P)\times K$ design matrix, $\mathbf{w}_n$ denote the K-dimensional vector of regression coefficients at voxel n, and $\mathbf{e}_n$ denotes the corresponding error term. Define the vector $\mathbf{y}_n \equiv \mathbf{y}_{1:T,n}$, the entire time series observed at voxel n. The hierarchical model is specified in several

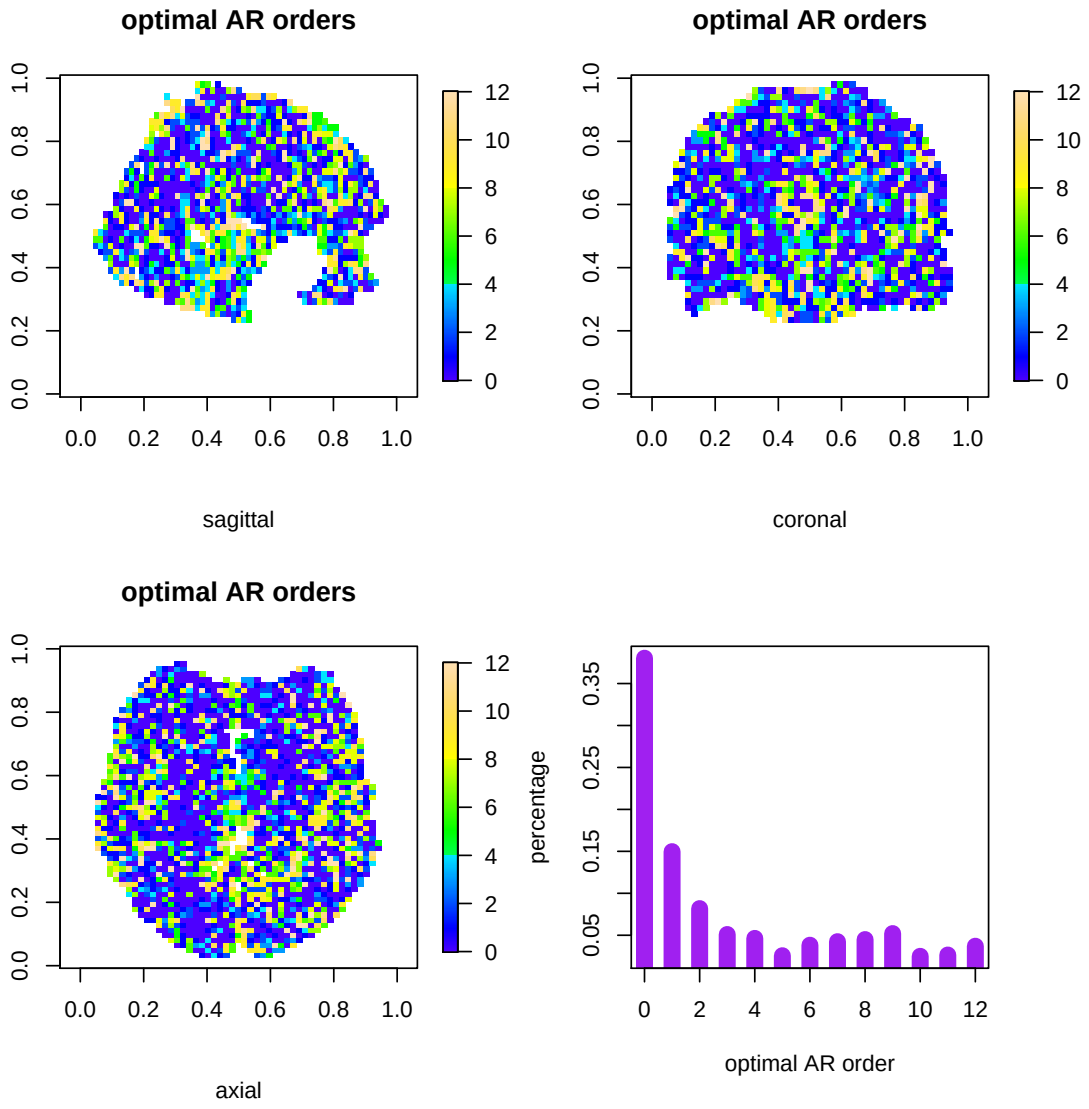


Figure 4.1: Optimal maximum AR orders selected based on MLE. The 1st, 2nd and 3rd image denotes the sagittal, coronal and axial view of the brain. The 4th figure is a histogram of the distribution of optimal orders in each voxel. The upper bound is set to 12 (default threshold) when doing this experiment.

stages. The first stage is a general linear model:

$$\mathbf{y}_{P+1:T,n} = \mathbf{X}\mathbf{w}_n + \mathbf{e}_n, \tag{4.1}$$

where we emphasize again the implicit conditioning on $y_{1:P,n}$ $(n = 1, ...N)$. Let $\tilde{\mathbf{E}}_n$ denote the embedded error (or lagged prediction) matrix of dimension $(T - P) \times P$, with t, p element $(\mathbf{y}_{P+1:T,n} - \mathbf{X}\mathbf{w}_n)_{[t-p]}$ where the notation $[i]$ denotes the i^{th} index of the vector.

Further, $\mathbf{z}_n \equiv z_{P+1:T,n}$ denotes a vector of i.i.d mean-zero Gaussian random variables with precision λ_n. The second stage is then an AR model at each voxel:

$$\mathbf{e}_n = \tilde{\mathbf{E}}_n \mathbf{a}_n + \mathbf{z}_n \tag{4.2}$$

where $\mathbf{a}_n$ is a vector of autoregressive coefficients for the time series at voxel n.

Letting $const$ denote a constant term, the log-likelihood for voxel n, is

$$l_n = -\frac{\lambda_n}{2} \sum_{t=P+1}^{T} \left[(y_{tn} - \mathbf{x}_t \mathbf{w}_n) - \tilde{\mathbf{e}}_{tn} \mathbf{a}_n \right]^2 + \frac{T-P}{2} \log \lambda_n + const. \tag{4.3}$$

Summing this likelihood over n, we can get the overall likelihood:

$$l = \sum_{n=1}^{N} \left\{ -\frac{\lambda_n}{2} \sum_{t=P+1}^{T} \left[(y_{tn} - \mathbf{x}_t \mathbf{w}_n) - \tilde{\mathbf{e}}_{tn} \mathbf{a}_n \right]^2 + \frac{T-P}{2} \log \lambda_n + const. \right\} \tag{4.4}$$

4.2.2 Spatial modelling

At the next level of the model we specify a spatial smoothing prior for the regression coefficients $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_N)$. Following Penny et al. (2005), we assume that the prior for $\mathbf{W}$ takes the form

$$\pi(\mathbf{W}) = \prod_{k=1}^{K} \pi(\mathbf{w}_k) \tag{4.5}$$

$$\mathbf{w}_k \sim \mathrm{N}(\mathbf{0}, \alpha_k^{-1} (\mathbf{S}^T \mathbf{S})^{-1}) \tag{4.6}$$

where the model assumes independence in the coefficients associated with the different columns of the design matrix while the multivariate normal distribution is assigned as the prior for $\mathbf{w}_k$, the vector of coefficients corresponding to the k^{th} column of the design matrix, the different elements corresponding to different voxels.

Here $\mathbf{S}$ is known as a Laplacian matrix with diagonal term equal to the corresponding number of first order neighbors of a given voxel and -1 on the off-diagonal in positions corresponding to neighbors of a given voxel. This form for the prior accommodates spatial

smoothing while also being sparse and convenient to work with computationally. In the SPM12 software this prior is referred to as the "LORETA" prior. Ultimately, what is of primary interest in studies of brain activation is a posterior probability of some function of these regression coefficients, and this posterior probability is computed at each voxel to produce posterior probability maps (PPM, see Penny et al. (2005)). α_k is assigned a conditionally conjugate hyper-prior

$$\alpha_k \overset{iid}{\sim} \text{Gamma}(q_1, q_2)\ (k = 1, ...K). \tag{4.7}$$

4.2.3 Temporal modelling

The key difference between our model and the model of Penny et al. (2007) lies in our modeling of the temporal noise. Rather than assuming AR orders are homogeneous throughout all of the voxels (we refer the readers to Chapter III and Penny et al. (2007) for model details), we allow for variability in the order of the AR processes across voxels. In addition, we adopt a spatial prior for this variability under the assumption that the AR orders of neighboring voxels will be similar. Specifically, for each voxel n and order p, $p = 1, \ldots, P$, we assign the latent indicator variable γ_{pn} to the p^{th} AR coefficient a_{pn}, such that given γ_{pn} $(p = 1...P, n = 1...N)$, a_{pn} will be conditionally independent. γ_{pn} will take value 1 if order p is present for voxel n and 0 otherwise. Conditional on γ_{pn}, a_{pn} will either have a normal distribution or unit mass at 0. This is commonly referred to as the spike-and-slab prior (George and McCulloch, 1993; Mitchell and Beauchamp, 1988), though we note that our formulation is a spatial spike-and-slab prior.

$$\pi(\mathbf{a} \mid \boldsymbol{\gamma}) = \prod_n \prod_p \pi(\mathbf{a}_{pn} \mid \gamma_{pn}) \tag{4.8}$$

$$\pi(a_{pn} \mid \gamma_{pn}) = \gamma_{pn}\phi(a_{pn}; 0, \tau_p^2) + (1 - \gamma_{pn})I_0(a_{pn}) \tag{4.9}$$

Here, $\phi(\cdot; a, b)$ is the pdf of a normal distribution with mean a and variance b and $I_0(\cdot)$

is the indicator function that it's argument equals 0, and where γ_{pn} is the binary indicator. τ_p is the precision of the slab and is again given a Gamma prior $\tau_p \sim \text{Gamma}(r_1, r_2)$.

The advantage of introducing such a prior is three fold: First, the orders in the AR process at each voxel that lack support from the data can be effectively removed from the model as the corresponding AR coefficients can be shrunk exactly to 0. This allows us to infer which orders are present in which voxels. Second, the number of voxels with high AR orders is non-zero but expected to be small, which is an aspect of this prior that can be controlled by tuning the hyper-parameters. Third, for some of the voxels there might be vacancies in some of the middle orders while there are some non-zero coefficients for higher orders. The proposed model is flexible enough to allow for this behaviour, since we have a total of P independent Ising process, one for each possible order $p \in \{1, \ldots, P\}$.

There are of course other model selection techniques that could have been considered. For example a type of Bayesian lasso could have bee used as an alternative to the spike-and-slab prior. Wang et al. (2007) has applied the lasso to the selection of AR processes, and for Bayesian lasso we refer to Schmidt and Makalic (2013). A recent alternative prior known as the "non-local" prior for variable selection has been proposed by Johnson and Rossell (2012) and has been demonstrated to have desirable consistency properties and yield smaller prediction errors in large sample settings. A review of Bayesian priors that can be employed for model selection is presented in O'Hara et al. (2009).

We assume that the indicator processes are independent across different orders, $\pi(\boldsymbol{\gamma}) = \prod_p \boldsymbol{\gamma}_p$, where $\boldsymbol{\gamma}_p = (\gamma_{p1}, ..., \gamma_{pN})^T$. The simplest variable selection model would assume γ_{pn} follows a Bernouli distribution (George and McCulloch, 1993). Here, in order to allow for the borrowing of information across neighbors as well as to model the spatial clustering effect of AR orders, we choose to use the Ising model (Smith and Fahrmeir,

2007) independently for each $p = 1...P$.

$$P(\boldsymbol{\gamma}_p) \propto \exp\left(\beta_{0p}\sum_{n}\gamma_{pn} + \beta_{1p}\sum_{n_1\sim n_2} I(\gamma_{pn_1} = \gamma_{pn_2})\right), \tag{4.10}$$

where β_{0p} and β_{1p} are two hyper-parameters controlling the sparsity and smoothness of the binary latent field respectively. Typically, a higher value of β_{0p} results in less sparsity and a higher value of β_{1p} indicates more smoothness. One issue with the Ising model that requires some care is the choice of hyper-parameters. When these parameters take values near what is known as the "phase transition" boundary, the mixing of an MCMC sampler will suffer from critical slow down (Stanley et al., 1987). To avoid the phase transition boundary, we adopt an analytical approach similar to Li et al. (2015) to quantify the value for the bounds for β_{0p} and β_{1p}. Derivations are given in Subsection 4.2.5 and Appendix C.

4.2.4 MCMC updating scheme

As shown in Appendix C, most of the updates related to posterior sampling of our model can be accomplished via Gibbs sampling. One exception is to update latent indicator $\boldsymbol{\gamma}$, and for these we use a Swendsen-Wang algorithm, alternating Swendson-Wang updates with Gibbs updates (Johnson et al., 2013). This strategy has proven successful in improving the mixing of the Markov chain sampler and results in faster block updates in various studies (Higdon, 1998).

To implement a Swendsen-Wang update, we first find the full conditional density of $\boldsymbol{\gamma}_p$ to be

$$P(\boldsymbol{\gamma}_p \mid \cdot) \propto L(\boldsymbol{\gamma}_p)\exp\left(\beta_{0p}\sum_{n}\gamma_{pn} + \beta_{1p}\sum_{n_1\sim n_2}\mathrm{I}\{\gamma_{pn_1} = \gamma_{pn_2}\}\right) \tag{4.11}$$

where $L(\boldsymbol{\gamma}_p)$ denotes the likelihood term associated with $\boldsymbol{\gamma}_p$. We next define what is known as a "bond variable" (an auxiliary variable) $\phi_{pn_1n_2}$, for each first-order neighboring pair $n_1 \sim n_2$ ($\{n_1, n2\} \in N$). Let $\boldsymbol{\phi}_p = \{\phi_{pn_1n_2} : n_1 \sim n_2\}$, with

$$\phi_{pn_1n_2} \mid \boldsymbol{\gamma}_p \sim \text{Unif}(0, \exp(\beta_{1p}\mathrm{I}\{\gamma_{pn_1} = \gamma_{pn_2}\})) \tag{4.12}$$

where Unif$(\cdot)$ denotes uniform distribution, then we have

$$P(\boldsymbol{\gamma}_p \mid \boldsymbol{\phi}_p, \cdot) \propto L(\boldsymbol{\gamma}_p) \exp(\beta_{0p} \sum_n \gamma_{pn}) \prod_{n_1 \sim n_2} \mathrm{I}\{0 \leq \phi_{pn_1n_2} \leq \exp(\beta_{1p} \mathrm{I}\{\gamma_{pn_1} = \gamma_{pn_2}\})\} \tag{4.13}$$

From Equation 4.12 we know that

$$P(\phi_{pn_1n_2} > 1 \mid \boldsymbol{\gamma}) = \int_1^{\exp\left(\beta_{1p}\mathrm{I}\{\gamma_{pn_1}=\gamma_{pn_2}\}\right)} d\phi_{pn_1n_2} > 0 \Leftrightarrow \gamma_{pn_1} = \gamma_{pn_2} \tag{4.14}$$

The meaning behind this is that, if $\phi_{pn_1n_2} > 1$, then γ_{pn_1} and γ_{pn_2} can be considered as "bonded" with probability $1 - \exp(-\beta_{1p})$. Thus, $\boldsymbol{\phi}_p$ will partition the voxels into S_p different clusters, where all the latent indicators in a given cluster share the same value (i.e. either 1 or 0). Let $\{n\}$ denote the cluster containing voxel n, then the full conditional of $\gamma_{p\{n\}}$ takes the following form

$$P(\boldsymbol{\gamma}_{p\{n\}} = \mathbf{1} \mid \cdot) \propto L(\boldsymbol{\gamma}_{p\{n\}} = \mathbf{1}) \exp\left(\beta_{0p} \sum_{n \in \{n\}} \gamma_{pn}\right), \tag{4.15}$$

and a draw from this distribution is easily made after normalization. Additional details are provided in Appendix C.

To evaluate the performance of our model, we make comparisons with the standard GLM-AR spatial model. One implementation of this model that we make comparisons to is the Variational Bayes (VB) method in SPM12 software. Another implementation is our self-written MCMC sampler for the same model. Although the accuracy of VB has been verified in a setting with high signal-to-noise ratio (SNR) by Chapter III, we have noticed here that under a low SNR, MCMC will outperform VB according to certain metrics. This will be illustrated in the simulation studies and our motivating application. Thus, besides the SVARO model we have developed here, we will name the VB version and MCMC version of GLM-AR model as PVB and PMCMC respectively.

4.2.5 Bound construction

The hyper-parameters in the Ising model play a vital role in posterior estimation. Without careful selection, we might face potential mixing problems associated with "phase-transition" (Stanley et al., 1987). There are various approaches to sampling such hyper-parameters, Johnson et al. (2013) estimated them using path sampling (Gelman and Meng, 1998), Shu et al. (2015) proposed a Monte Carlo EM algorithm to obtain a point estimate of the hyper-parameters, but these procedures would be too time consuming for our model, considering that we have over 10 independent Ising fields. Smith and Fahrmeir (2007) proposed to update the hyper-parameters and binary indicators together, but this approach still suffers from potential possibility of sampling over the phase transition boundary. Here, we adopt a similar approach as in Li et al. (2015) and construct some theoretical bounds to prevent the possible phase transition, the resulting hyperparameters values are then chosen as fixed in that bound. This procedure turns out to work well in our analysis and studies.

To construct the bounds, we first write out the posterior conditional density w.r.t. $\boldsymbol{\gamma}_p$,

$$
\begin{aligned}
P(\boldsymbol{\gamma}_p \mid \cdot) \propto \quad & \exp\Big(\beta_{0p}\sum_n \gamma_{pn} + \beta_{1p}\sum_{n_1 \propto n_2} \mathrm{I}\{\gamma_{pn_1} = \gamma_{pn_2}\} \\
& + \sum_n \sum_t \frac{-\lambda_n}{2}(e_{tn} - \sum_p \tilde{e}_{tnp} a_{pn})^2\Big)
\end{aligned} \tag{4.16}
$$

In our model where multiple orders exists across space, it is natural to assume that: 1) For high AR orders, we assume there are relatively few voxels; and 2) For low AR orders, the posterior density when low AR orders exist is greater than that when low AR orders do not exist, meaning $P(\boldsymbol{\gamma}_p \mid \cdot)$ is greater than $P(\mathbf{0} \mid \cdot)$.

Let π_p denote the candidate voxels selected for order p, then we know that the maximum number of neighbors they can achieve is when all the candidate voxels form a cube. Let $V_p = (\pi_p N)^{1/3}$ denote the length of an edge of this cube, then from Appendix C, we know

that there are $3V_p^2(V_p-1)$ neighboring pairs. Based on this it is easy to see that

$$\beta_{0p}\sum_n \gamma_{pn} + \beta_{1p}\sum_{n_1\sim n_2} \mathbf{I}(\gamma_{pn_1}=\gamma_{pn_2}) = \beta_{0p}V_p^3 + 3\beta_{1p}V_p^2(V_p-1) \tag{4.17}$$

According to 1), we know that for high AR orders (typically $P > 8$) $\beta_{0p} + 3\beta_{1p} < 0$. According to 2), we know that for low AR orders (typically $P < 4$),

$$\begin{aligned}\sum_n\sum_t \frac{-\lambda_n}{2}(e_{tn} - \sum_{p_0\neq p}\tilde{e}_{tnp_0}a_{p_0n})^2 \leq \sum_n\sum_t \frac{-\lambda_n}{2}(e_{tn} \\ -\sum_{p_0}\tilde{e}_{tnp_0}a_{p_0n})^2 + \Big[\beta_{0p}\sum_n\gamma_{pn} + \beta_{1p}\mathbf{I}(\gamma_{pn_1}=\gamma_{pn_2})\Big]\end{aligned} \tag{4.18}$$

Reorganizing this by moving the first term on right-hand side to left produces:

$$\begin{aligned}\sum_n\sum_t \frac{-\lambda_n}{2}\left[(e_{tn} - \sum_{p_0\neq p}\tilde{e}_{tnp_0}a_{p_0n})^2 - (e_{tn} - \sum_{p_0}\tilde{e}_{tnp_0}a_{p_0n})^2\right] \\ \leq \Big[\beta_{0p}\sum_n\gamma_{pn} + \beta_{1p}\mathbf{I}(\gamma_{pn_1}=\gamma_{pn_2})\Big]\end{aligned} \tag{4.19}$$

The two terms in the bracket on the left side can be considered as one with and without $\tilde{e}_{tnp}a_{pn}$. Thus, it can be roughly considered as the residual sum of squares of a common linear regression when a_{pn} is included in the model or not. Let R_{pn}^2 denote the coefficient of determination for voxel n and order p, then we have

$$\beta_{0p}\sum_n\gamma_{pn} + \beta_{1p}\sum_{n_1\sim n_2}(\gamma_{pn_1}=\gamma_{pn_2}) \geq -\frac{1}{2}\sum_n\sum_t\frac{R_{pn}^2}{1-R_{pn}^2} \tag{4.20}$$

Combined with Equation 4.17, we have

$$\beta_{0p}V_p^3 + 3\beta_{1p}V_p^2(V_p-1) \geq -\frac{1}{2}\pi_p NT\frac{R_{pn}^2}{1-R_{pn}^2} \tag{4.21}$$

For a 3-dimensional grid we assume $N = 56526$ as the number pf voxels. Among them, a proportion of $\pi_p = 0.1$ are selected for order p. So $V_p = (\pi_p N)^{1/3} = 17.8$. We assume that 5% of the variation can be explained as a result of order p, so $R_{pn}^2 = 0.05$. We then have $\beta_{0p} + 2.83\beta_{1p} \geq -9.26$.

Note that the inequality above just gives a range values for the hyper-parameters, rather than providing the values directly. In practice, the exact values of hyper-parameters are largely determined by the researcher, which should be combined with one's prior experience and an initial analysis of the data. We suggest obtaining such values based on some exploratory ad-hoc approaches, e.g., a linear regression at each voxel followed by fitting an AR process. Then the the estimated optimal orders can be used as a reference when determining the hyper-parameters in the Ising model. This method has turned out to work well empirically as demonstrated in Section 5.

4.2.6 Log pseudomarginal likelihood

To compare the model performance between SVARO and PMCMC, we use the log pseudo marginal likelihood (LPML). This criteria for model selection is proposed by Geisser and Eddy (1979) and enjoys wide application due to its ease of computation based on MCMC sampling output.

To begin with, let M denote the model, and $\boldsymbol{\theta} = \{\mathbf{W}, \mathbf{A}, \boldsymbol{\alpha}, \boldsymbol{\tau}, \boldsymbol{\lambda}\}$ denote the parameters. The time series $\mathbf{y}_n$ in each voxel n will be conditionally independent of the time series at other voxels given the model and parameters, $p(\mathbf{Y} \mid \boldsymbol{\theta}, M) = \prod_{n=1}^{N} p_i(\mathbf{y}_n \mid \boldsymbol{\theta}, M)$. In this spirit, the LPML is an approximation to the marginal likelihood under the simplifying assumption

$$p(\mathbf{Y} \mid M) \approx \prod_{n=1}^{N} p_n(\mathbf{y}_n \mid \mathbf{y}_{-n}, M) \tag{4.22}$$

where $p_n(\mathbf{y}_n \mid \mathbf{y}_{-n}, M)$ is the predictive distribution of $\mathbf{y}_n$ under the model, which when evaluated at the observed $\mathbf{y}_n$ is referred to as n^{th} conditional predictive ordinate (CPO_n). Then the LPML is defined as $LPML = \sum_{n=1}^{N} \log(CPO_n)$. The CPO_n can be estimated from MCMC output by

$$CPO_n^{-1} \approx \frac{1}{L} \sum_{l=1}^{L} \frac{1}{p_n(\mathbf{y}_n \mid \boldsymbol{\theta}_l, M)} \tag{4.23}$$

where l denotes the l^{th} posterior draw from an MCMC sampler where the total number of draws is L, excluding the burn-in. Interested readers may refer to Gelfand and Dey (1994) for more details of this estimator.

4.2.7 Posterior probability maps

A primary emphasis on fMRI data analysis is inference for activation, so we provide some basic background on contrast and posterior probability maps (PPM). A contrast for a certain voxel n is the inner product of a contrast vector $\mathbf{c}$ with the regression coefficient in that voxel $\mathbf{w}_n$. The contrast vector $\mathbf{c}$ is typically a weighted vector with elements consisting of 1 and -1 representing an effect of interest. For example, to study the effect of A versus B when these are the only two conditions would lead to a contrast vector of $\mathbf{c} = (1, -1)^T$.

Having defined contrast, A PPM is a map that shows the posterior probability of activation for each voxel: $P(\mathbf{c}^T\mathbf{w}_n > \delta_e)$. Here δ_e is a pre-specified "activation threshold", for example, a value that corresponds to 1% of the global mean value. Thus, PPM looks at the probability of the contrast $\mathbf{c}^T\mathbf{w}_n$ being greater than activation threshold δ_e, given the data.

To formally determine activation in the brain, one can look at a thresholded PPM. This is obtained by exerting a second threshold, namely a "probability threshold" δ_p, onto the original PPM. Thus, a voxel is "activated" if $P(\mathbf{c}^T\mathbf{w}_n > \delta_e) > \delta_p$. This δ_p reflects the confidence of the inference and usually takes a value above 0.9 (e.g. 0.95 or 0.99). This process discretizes the PPM into "non-activated" and "activated" voxels and is commonly used in summarizing a Bayesian analysis for brain activation.

4.3 Simulations

4.3.1 Simulation design

In this simulation, we aim to compare our model (SVARO), with PMCMC, and PVB. The simulation is based on a real single-subject face repetition dataset (Henson et al., 2002). Complete information on the dataset can be found online at *http://www.fil.ion.ucl.ac.uk/spm/data/*. In this experiment, famous faces and non-famous faces are presented two times, resulting in four types of stimulus (F1,F2, U1, U2). These stimuli are convolved with a haemodanymic response function (HRF) to be formally used as regressors in the statistical model. In terms of voxels, we exert a 2D brain mask on the z-axis into the brain consisting of $53 \times 63 \times 52$ voxels, and this gave us 2087 voxels.

To simplify the computation we assume there are only two columns in the design matrix ($K = 2$). The first column, or slope, is set to be the first stimulus (F1) convolved with the HRF, and the second column is the intercept (a vector of 1). The slope is generated under a mean zero multivariate normal distribution $\mathbf{w}_1 \sim N(0, (10\mathbf{S}^T\mathbf{S})^{-1/2})$, while the intercept is generated under a mean 100 multivaraite normal distribution $\mathbf{w}_2 \sim N(100, (10\mathbf{S}^T\mathbf{S}^T)^{-1/2})$. The white noise will have a precision of $\lambda_n = 0.1$ $(n = 1...N)$. This corresponds to a low signal-to-noise (SNR) ratio, where the temporal noise will play a greater role in the datay. In the following, we will carry out two simulations. The first will be simulated under our model and the second will be generated under the standard spatial GLM-AR model. In these two simulations, we aim to look at the estimation accuracy of the slope ($\mathbf{w}_1$), intercept ($\mathbf{w}_2$), and autoregressive coefficients ($\mathbf{a}_p$ $p = 1...P$), and also whether the difference in inference for these coefficients will lead to a possible difference in the final inference on activation. All the simulations are replicated 100 times.

4.3.2 Simulation I

Here we simulate under the SVARO model. We assume that the maximum order is $P = 8$. The precisions are set as $\tau_p = 20$ $(p = 1...P)$. For simplicity, we assume that all AR orders are generated spatially according to the same hyper-parameter of the Ising model, i.e., $\beta_{0p} = -0.2$ and $\beta_{1p} = 0.3$. The AR order in PMCMC and PVB are set to $P = 1$ as is fairly standard practice.

Figures 4.2 shows the true AR orders, estimated maximum orders using SVARO, and the difference of the two. The estimated maximum orders are obtained by averaging the posterior mean of different orders in each voxel over their simulation replicates, and further rounded to be 1 or 0 using a threshold of 0.5. The highest orders that have probability greater than 0.5 are considered as the maximum order. We can see that most of the orders match between the two figures indicating good performance. There are some negative values in the difference map.

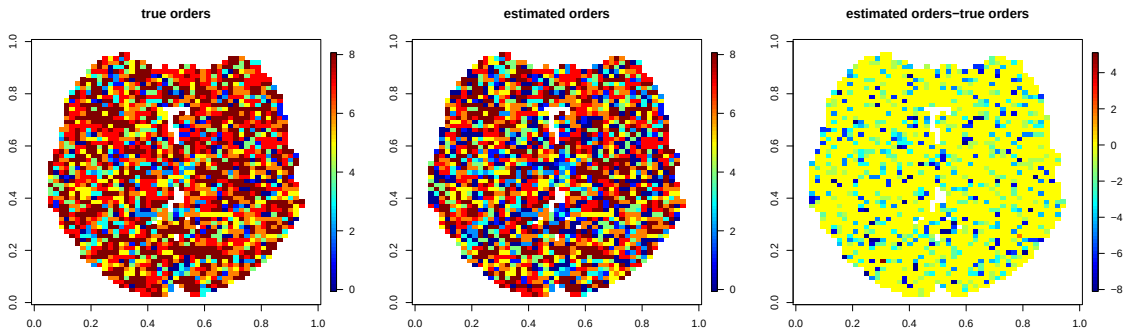


Figure 4.2: Maximum orders of AR coefficients in each voxel. The left one denotes the true generated maximum orders, ranging from 0 to 8. The middle one shows the posterior estimates of the maximum orders. The right one denotes the difference between the two.

We next look at how the three methods compare in estimating the 1^{st} AR coefficient. As shown in Figure 4.3, SVARO shows little error compared with the truth, indicating that our model has captured the autoregressive parameter quite well. In contrast, PMCMC

and PVB exhibit more bias, indicating a lack of fit for the temporal noise. Note that we are only displaying the SVARO estimates for the 1^{st} order for simplicity, the other orders exhibit the same trend, we refer interested readers to supplementary materials for details.

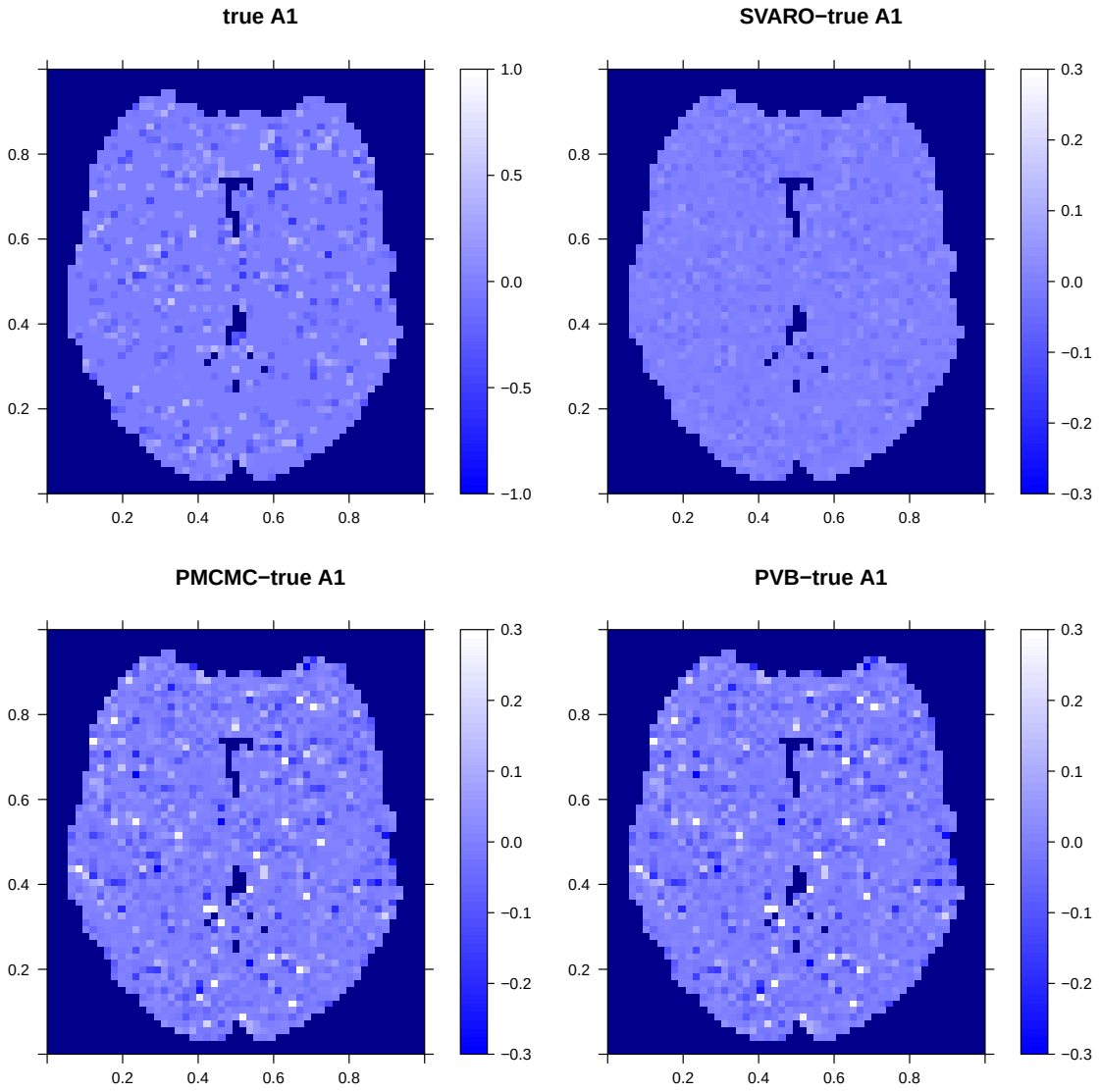


Figure 4.3: The top left image (with scale -1 to 1) denotes the true AR coefficients for $1st$ order. The 2nd, 3rd, and 4th image are the corresponding difference between truth and posterior mean of SVARO, PMCMC and PVB respectively. The color scale for the rest of the three images are truncated from -0.3 to 0.3 so that the error in SVARO is more visible. The posterior means are all averaged over 100 replicates.

Table 4.1 summarizes the average MSE for various parameters. These summaries are obtained by averaging the MSE of the corresponding parameters across all the voxels

		MSE		LPML	Timing
	$\mathbf{w}1$	$\mathbf{w}2$	$\mathbf{a}1$		
SVARO	0.478	0.030	0.001	-1842902	108min
PMCMC	113%	135%	509%	-1926620	11min
PVB	199%	138%	510%	-	1min

Table 4.1: Table of MSE, LPML and Timing for the three models. MSE is calculated by averaging MSE in each voxel and over simulation replicates. The MSE values for PMCMC and PVB are relative to those in SVARO.

and over simulation replicates. It is clear that SVARO has the smallest MSE for all of the parameters. In addition, PMCMC outperforms PVB in terms of slope, which is the primary parameter that inference is based on. This finding is different than that found in Chapter III because here we are using a low SNR, and this appears to be one setting where MCMC outperforms VB for this particular model. Table 4.1 also gives the LPML and the timing. SVARO has higher LPML than PMCMC, indicating a better model fit. Notice that the VB implementation can not be used to obtain the LPML. In terms of timing, SVARO takes 108min with 10,000 iterations following 10,000 burn-in iterations, PMCMC takes 11min with the same number of iterations, PVB is the fastest, and takes only 1min.

We next investigate how the differences observed for the individual parameters will impact the overall inference of interest. A sensitivity plot is presented in Figure 4.4. This figure is obtained by plotting the average sensitivity against a range of marginal posterior probability threshold from 0.9 to 1. We choose this range because it covers those values most often used in practice.

In terms of the underlying activation threshold, we use two thresholds: the true value of the contrast that corresponds to top 10% and top 5% of all the voxels. Thus, corresponding to a certain activation threshold and a certain probability threshold, the higher sensitivity is, the better the model is in terms of capturing activation. Again, a notable difference is observed when comparing the three methods, with SVARO giving the uniformly highest sensitivity across entire range of probability thresholds and PVB giving the lowest

sensitivity. PMCMC is better than PVB but still under-performs relative to SVARO.

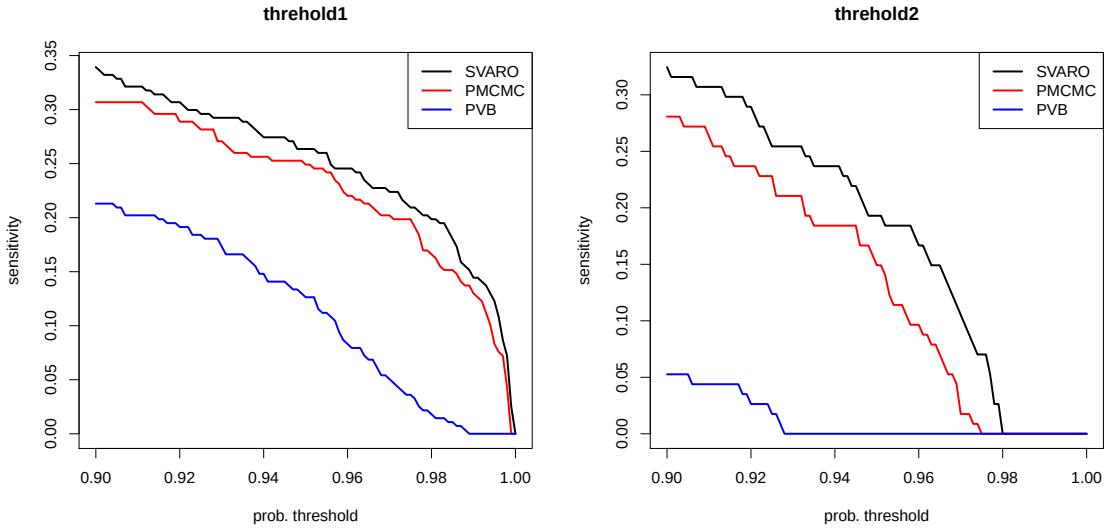


Figure 4.4: Thresholded sensitivity curve for the three methods, with two effect size thresholds. The left image has an effect size threshold corresponding to the top 10% of the values, while the right has an effect size threshold corresponding to the top 5% of the values. The x-axis denotes the probability threshold values and y-axis denotes the corresponding sensitivity.

To look at where the inferences might differ we plot the posterior probability maps (PPM) in Figure 4.5. The figure depicts the locations of the true activations and the posterior probability maps from SVARO. In addition, differences in the probability maps comparing SVARO with PMCMC and PVB are also depicted. Again SVARO appears to perform the best in producing the highest posterior probabilities for regions that are truly activated. PMCMC is similar to SVARO but its probability on those activated regions are slightly lower than those from SVARO, especially on the boundary. PVB under performs compared with the other two approaches by providing greater posterior probability on non-activated locations while providing smaller posterior probability on active locations.

4.3.3 Simulation II

Although the real data examined earlier suggest the existence of heterogeneous AR orders we want to see the performance of SVARO under a homogeneous AR order assump-

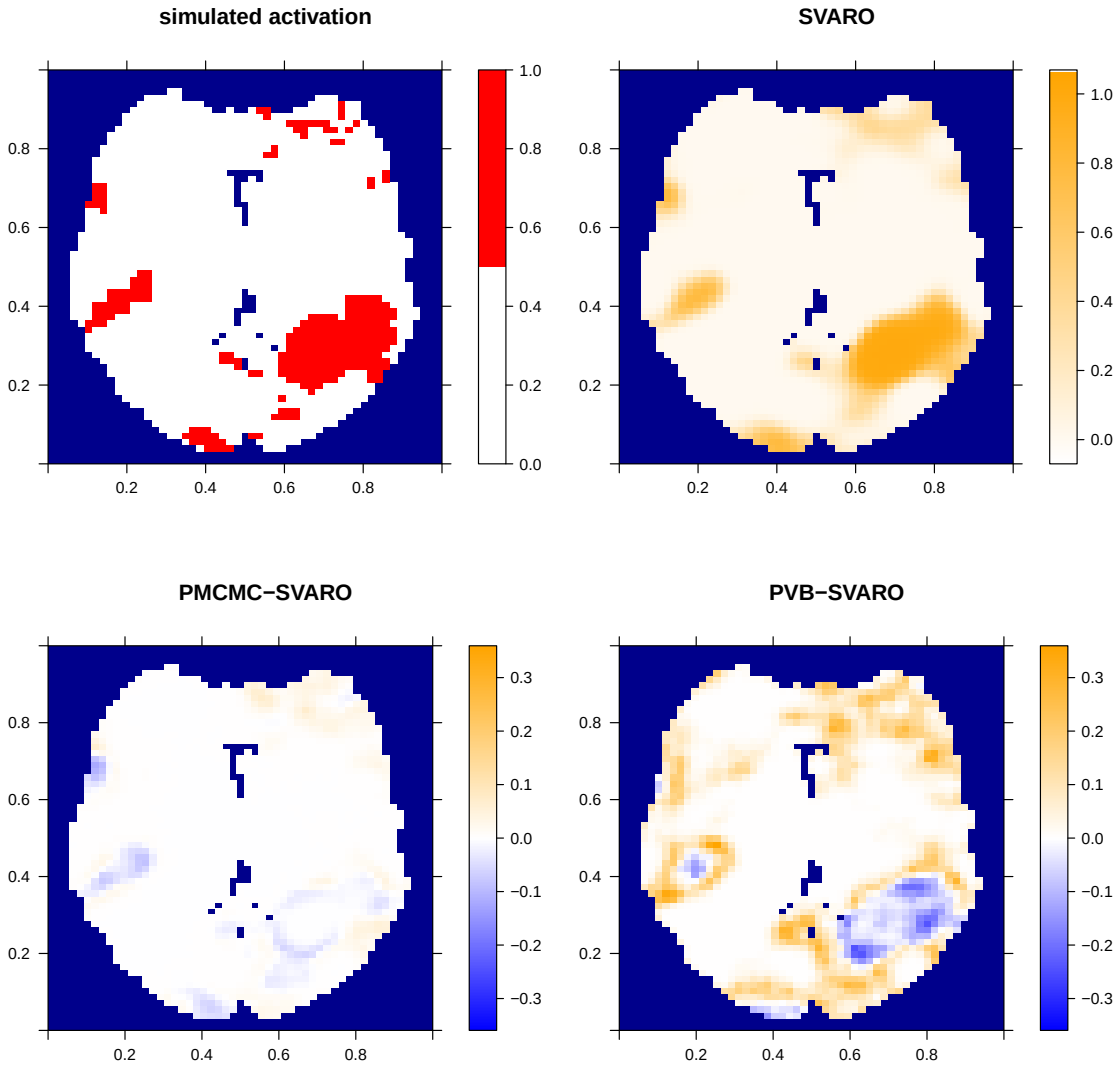


Figure 4.5: Topleft depicts the true activation map (red dots denote activation). The remaining panels are posterior probability maps (PPM) of activation obtained using SVARO, (SVARO-PMCMC) and (SVARO-PVB). The latter two reflect the difference of the two alternative approaches relative to SVARO.

tion. To do this we simulate under the competing GLM-AR model. The AR coefficients are simulated under the LORETA prior and the AR order is set to 1 for every voxel, with prior precision $\tau_p = 400$. We set the maximum order as $P = 12$ when applying SVARO. Thus, PMCMC and PVB are working under the true model while SVARO is working under a more general model one. Some comparisons are presented here while additional comparisons are presented in the Supplementary Material.

Table 4.2 shows the MSE summaries of the estimators. Under the competing model, SVARO still gives good performance in slope and intercept. Its MSE are slightly higher as expected. It is worth mentioning that PVB again under performs relative to PMCMC in terms of slope.

	MSE			LPML	Timing
	$\mathbf{w}1$	$\mathbf{w}2$	$\mathbf{a}1$		
SVARO	0.502	0.031	0.003	-1817287.93	206min
PMCMC	99%	97%	54%	-1875900.645	11min
PVB	167%	98%	49%	-	1min

Table 4.2: Table of MSE, LPML and Timing for the three models. MSE is calculated by averaging MSE in each voxel and over simulation replicates. The MSE values for PMCMC and PVB are relative to those in SVARO.

Figure 4.6 presents the sensitivity curves. Despite the data being simulated under a constant order AR assumption SVARO demonstrates similar sensitivity to that of PMCMC in both figures. The sensitivity curve of VB is uniformly lower than the other two because of the inaccurate estimation of slope parameter.

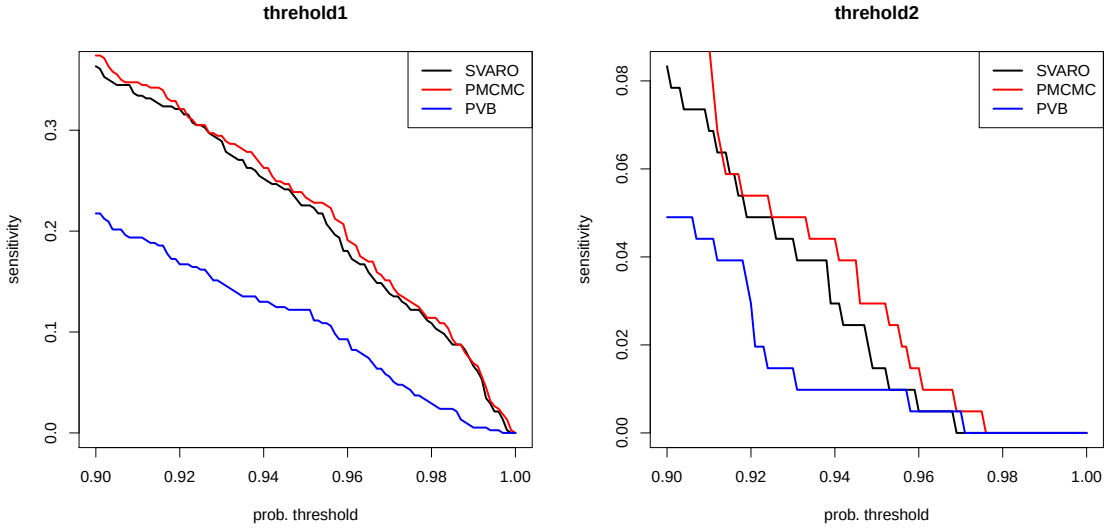


Figure 4.6: Thresholded sensitivity curve for the three methods, with two effect size thresholds. The left image has effect size threshold corresponding to the top 10% of the values, while the right has an effect size threshold corresponding to the top 5% of the values. The x axis denotes the probability threshold values and y axis denotes the corresponding sensitivity.

4.4 Real application

We turn our focus back to the face repetition data set that originally motivated our model development and look at the results from the three methods. In this analysis we use a complete set of five regression coefficients (F1,F2,U1,U2,intercept). We assume an AR order up to a maximum order of $P = 12$ when fitting the SVARO model, and an $AR(3)$ for PMCMC and PVB. One might think that the choice of an $AR(3)$ for the latter two approaches seems arbitrary, and this is exactly justification for the use of the SVARO model where such an arbitrary assumption need not be made. We apply the models to data for the whole brain in 3D with $56,526$ voxels.

The following pre-processing steps are applied to the data prior to fitting the Bayesian models: all functional images are aligned to the first image using a six-parameter rigid-body transformation. Then slice-timing correction is performed to set the standard acquisition time as the 12th slice. Images are spatially normalized to a standard EPI image. Global mean g is computed so that each time series was divided by $100/g$ to represent a percentage of g. Finally, a high-pass filter with cut-off frequency of $1/128$Hz is used to remove low frequency signal that would likely arise through scanner drift.

Table 4.3 presents the distribution of optimal orders from SVARO across voxels. Not surprisingly, the most frequent order is the 0 and 1^{st} order, taking about 35% and 23%, then the second order (about 12%). As order increases, its percentage generally decreases. However, it is interesting to note that the distribution is not strictly monotone decreasing with order, and that even as high an order as 8 is chosen for 10 percent of the total number of voxels. The existence of these higher orders and the variability in the orders is in correspondence with our exploratory analysis in the motivating example and indicates the necessity of a model fitting with high AR order structure.

Examining the marginal posterior estimates of parameters, since there are 5 covariates, we look at a contrast corresponding to the effect of "fame" (famous face vs non-famous face), $\mathbf{c}_f^T \mathbf{w}_n$, where $\mathbf{c}_f = (-1, -1, 1, 1, 0)^T/2$. The posterior mean and standard deviation (SD) for fame, as well as the posterior mean for the 1^{st} order autoregressive coefficient are shown in Figure 4.7. While the posterior SD of SVARO and PMCMC are very close, the posterior means show some differences between the two approaches. Also, the SD from PVB shows apparent discretization. This is due to a graph-partitioning that is incorporated in the algorithm for the sake of speeding up the computation. It is clear that the boundaries of these graph-partitioned regions have substantially higher SD than the interior locations. This finding is in accordance with Chapter III. The posterior mean of PVB also seems to suffer effects from this partitioning algorithm, thought its effect is not as pronounced as with the SD. Finally, in terms of the AR coefficients, PMCMC and PVB show a similar pattern of difference in comparison with SVARO. This is a natural result of the model assumption. Since PMCMC and PVB are both based on GLM-AR model. In terms of formal model comparison, SVARO gives an LPML value of -46589281.02 and PMCMC has a lower LPML of -48158447.77, from which appears that SVARO is the preferred model according to this model selection criterion.

Finally, we look at the effect of fame using thresholded PPMs. The activation threshold is set as 0.2% of the global mean value, and the probability threshold is set as 0.95. Figure 4.8 takes the middle slice from the sagittal, coronal and axial view. We can see that there is a match in terms of a majority of activation regions inferred from SVARO and PMCMC. A closer look reveals that PMCMC tends to make more scattered predictions across the back part of the brain, which is likely to have more false positives than SVARO. The LPML from the two models also supports this point. The number of activation regions from PVB are far greater than the number from the other two, and are apparently more scattered.

From the results of our our simulation studies, we suspect that these scattered activated regions are due to false positives from the algorithm. Figure 4.9 indicates how activations are distributed on the surface of the brain. The trend is the same as with Figure 4.8: while PMCMC is a little more liberal than PSVARO, PVB is far more liberal than the other two.

In terms of timing, PVB took about 1h to finish, PMCMC took 1 day, while SVARO took about a week of computation. To speed up SVARO, we suggest the use of parallel programming, which could make the algorithm run faster.

orders	0	1	2	3	4	5
percentage	35.29%	23.36%	14.19%	7.13%	9.65%	4.83%
6	7	8	9	10	11	12
13.37%	11.89%	11.84%	7.82%	3.04%	4.39%	6.24%

Table 4.3: Percentage of optimal orders from order 0 up to order 12, for all the $56,526$ voxels.

4.5 Discussion

In this paper, we have developed a new Bayesian hierarchical model, GLM-SVARO, that allows different AR orders across the brain, with the orders themselves displaying a certain level of spatial clustering based on an Ising model. We have compared it with a self-written MCMC sampler for the standard GLM-AR model and a VB algorithm for the same model. The results are interesting, under a low SNR ratio, VB seems to suffer from variance overestimation, leading to a much bigger MSE than the other two methods. It is likely that as temporal noise increases, there is a more vital role played by the AR correlation that increases the posterior correlation between different parameters and this makes the mean field assumption of VB less accurate.

A further look at SVARO and PMCMC reveals that due to the flexibility of order assumption, SVARO is better that PMCMC in terms of not only accuracy and sensitivity, but also formal model selection using LPML. Through AR images and an exploratory analy-

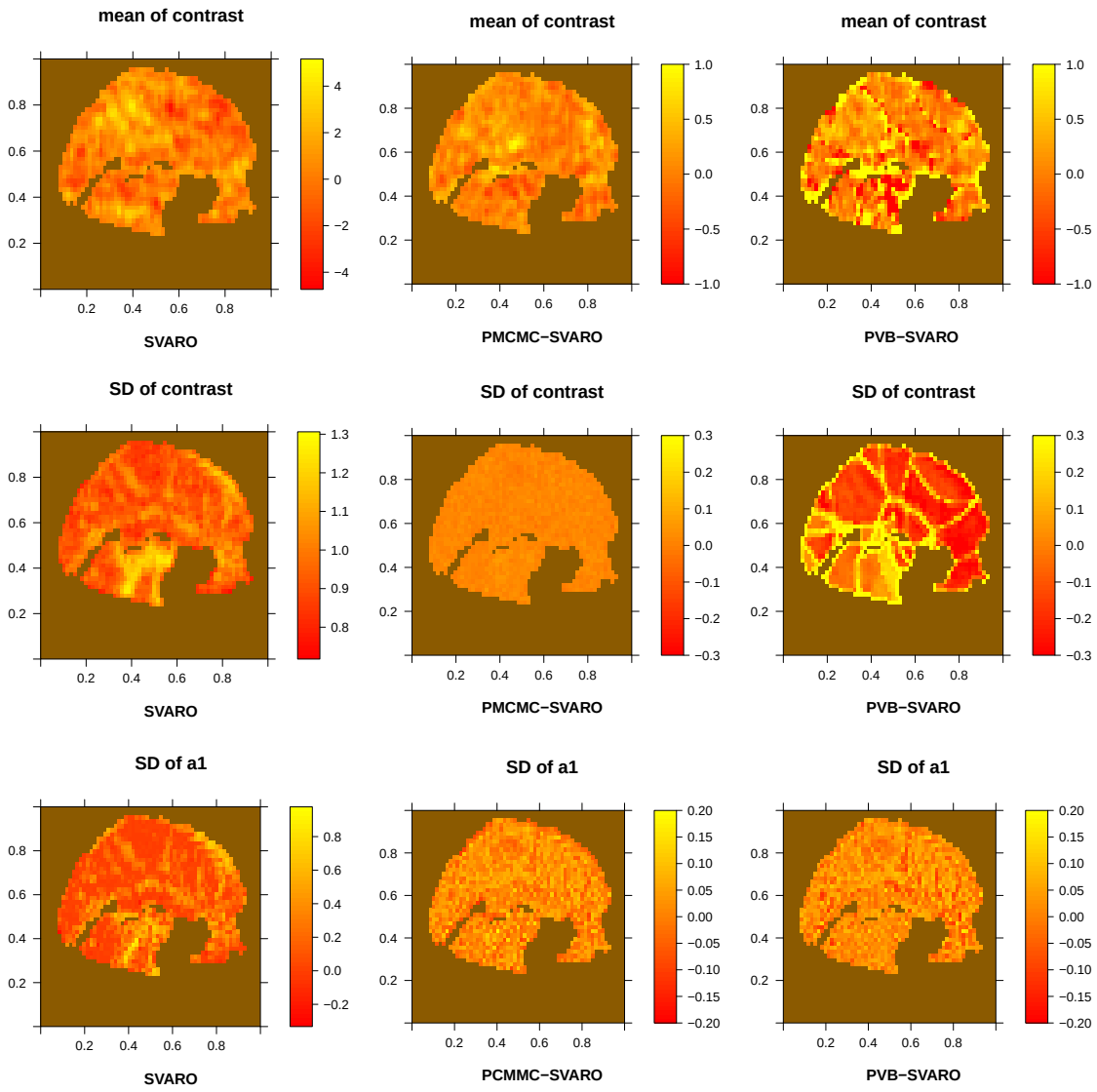


Figure 4.7: Posterior estimates from the middle (27/53) slice of the brain on sagittal view. From top row to bottom are: posterior mean of fame, posterior standard deviation of fame, and posterior mean of $\mathbf{a}_1$, respectively. From the left column to right are SVARO, SVARO−PMCMC and SVARO−PVB.

sis, we showed that a constant low-order AR assumption can be violated with real fMRI data. It is very likely that this issue is not unique to the face repetition data set.

There is a computational price to be paid for gaining the flexibility we have proposed in our model. Our model takes a longer time to run than PMCMC and PVB, mainly due to the calculation of the multiple orders. Noticing that the binary indicators of orders can be updated independently, we can in fact update these indicators using parallel programming

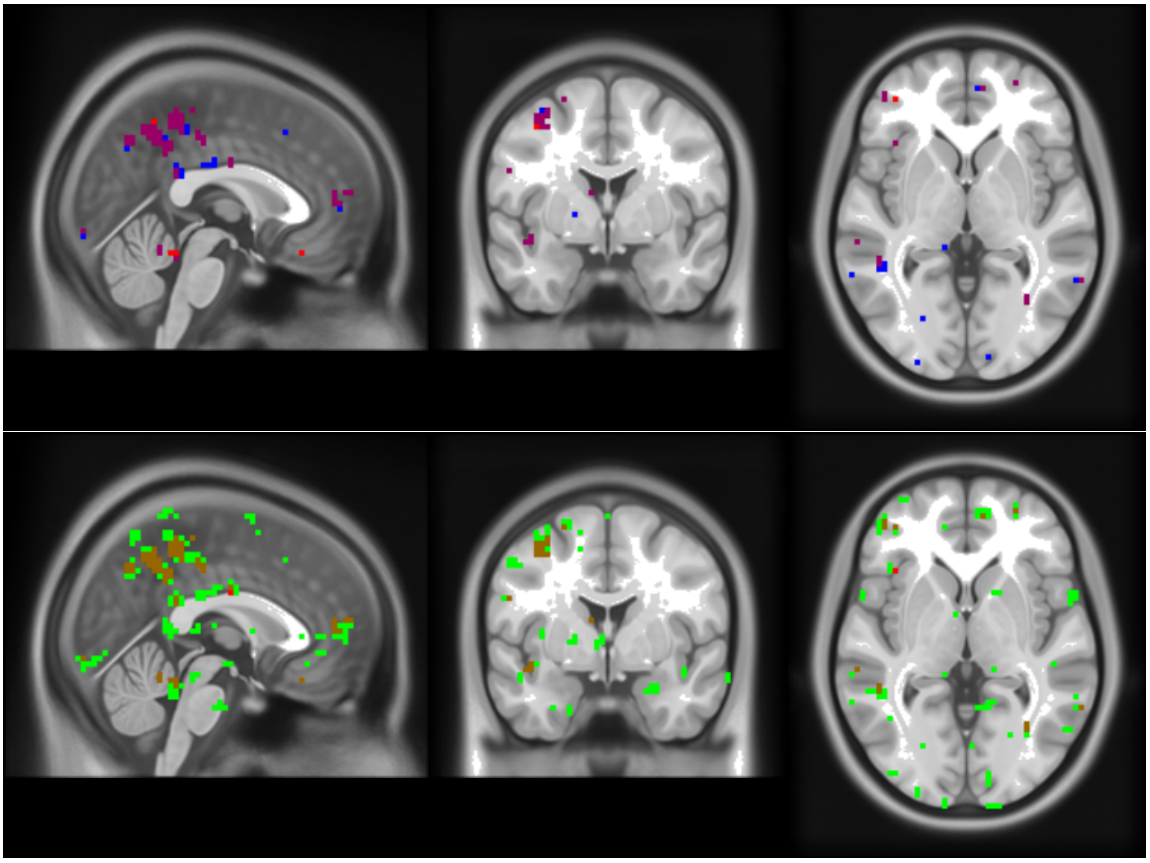


Figure 4.8: Activation maps for effect of fame on the three middle slices. From left to right are sagittal, coronal and transverse slice. Top row shows the activation from SVARO (red) and PMCMC (blue), with a joint region indicated by purple dots. The bottom row shows the activation from SVARO (red) and PVB (green), with the joint region denoted by brown dots.

techniques, thereby making the algorithm 10 to 20 times faster. This will be investigated in future work.

Another applicable, and perhaps more simple and straightforward idea is to assume a Potts model for the orders of AR coefficients. A Potts model, combined with a Dirichlet process prior for parameters has been investigated for selecting covariates of interest in brain imaging (Johnson et al., 2013). Here we can also apply it to the selection of autoregressive orders to yield a still flexible but more parsimonious model. Investigation of hyper-parameter estimation in the Ising model and the use of alternative spatial models is also of interest, as is increasing the scope of our comparison of methods to include wavelet

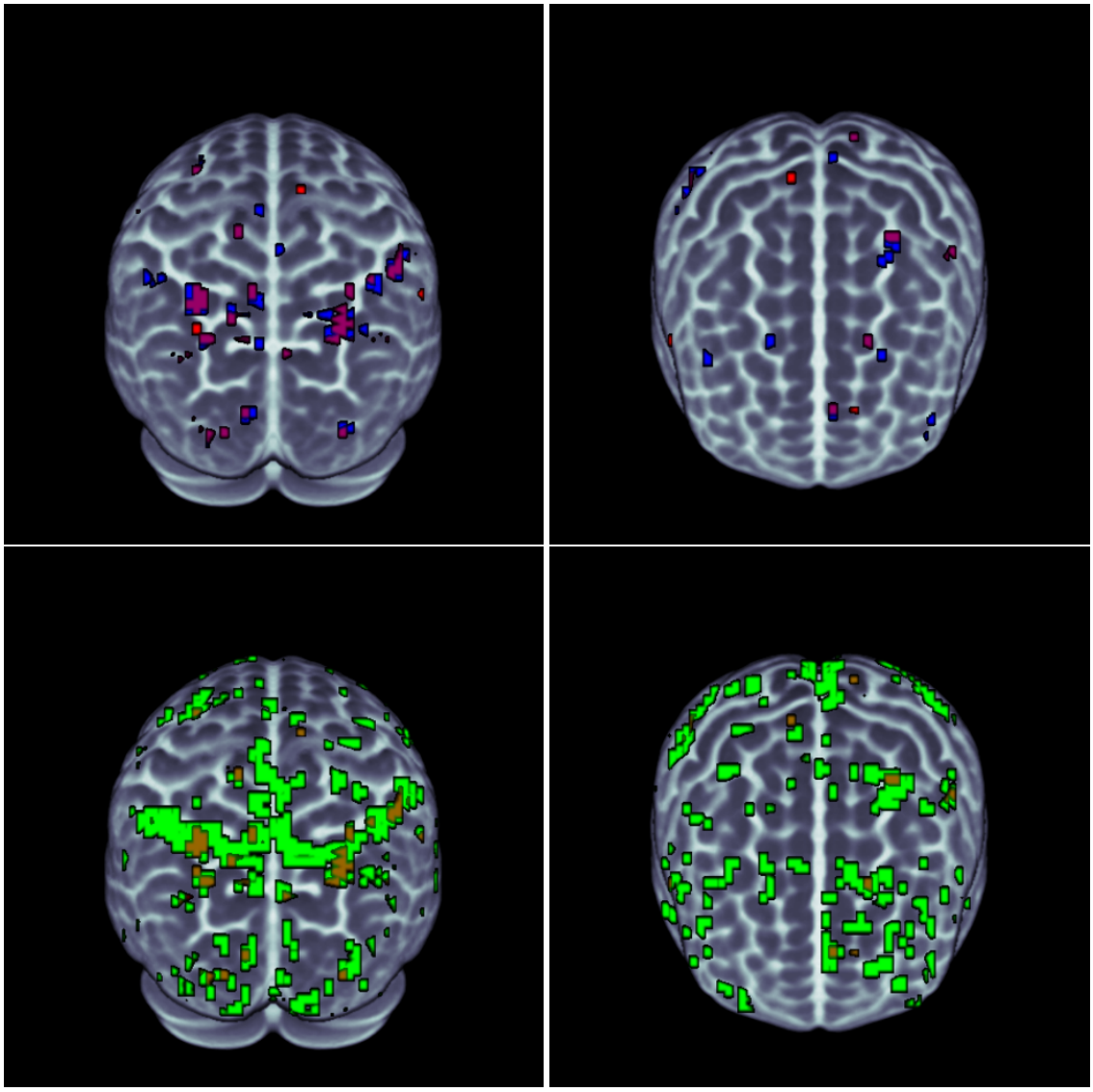


Figure 4.9: Activation maps for effect of fame in a 3D view. The left column is the posterior view while right column presents the anterior view. The top row shows the activation from SVARO (red) and PMCMC (blue), with joint region indicated by purple dots. The bottom row shows the activation from SVARO (red) and PVB (green), with joint region indicated with brown dots.

approaches that focus on long memory errors, or VAR models (Harrison et al., 2003).

CHAPTER V

Conclusion and Future Work

In this dissertation, we developed, implemented, and compared a series of fully and approximate Bayesian computational algorithms in the context of spatial point process models and neuroimaging. We also proposed a novel Bayesian hierarchical model, GLM-SVARO, based on evidence that the AR order varies spatially in some fMRI data sets.

In Chapter II, we derived an HMC algorithm for the LGCP model and combine it with the FFT algorithm for fast computation of large covariance matrices. To be able to use FFT we extend the grid so that the correlation matrix forms a circulant matrix (Møller et al., 1998). We also adopt a re-parametrization of the latent field so as to make the calculation of the gradient in HMC more mathematically convenient. We also developed a VB algorithm with mean-field approximation. To deal with the non-conjugacy of the latent field in the VB algorithm we used a Laplacian approximation, resulting in closed-form updates. We estimated the decay parameter in the correlation function by the method of minimum contrast to avoid convergence issues in the VB algorithm.

We also made use of the R INLA software package posted on the R-INLA web page. Among the three options: Gaussian, Simplified Laplace, and full Laplace, we chose the latter two for this project as they have better accuracy. We also tried two versions of INLA combined with the SPDE approach: one with mesh size of 436 and the other with mesh

size of 4075. We showed that the result are sensitive to the choice of mesh size and in some cases numerical stability is a concern for such methods.

We found, not surprisingly, that HMC is the most reliable approach in terms of accuracy. This is especially true if one is also interested in the hyper-parameters of the latent field. The accuracy of VB was not as good as its performance in other models (e.g. mixture models as Blei et al. (2006)). This is due to several reasons. First, through the mean field factorization spatial correlation is ignored. Second, with a combination of a Poisson-type likelihood and a Gaussian density, the resulting density of the discretized LGCP tends to depart from Gaussianity. This causes issues with the Laplacian approximation. Third, estimation of the decay parameter using the method of minimum contrast potentially introduces bias in the inference of the latent field.

Despite the accuracy loss compared to HMC, INLA was shown to be a promising alternative when speed is the primary concern. Its ability to compute complex point process models quickly is due to two reasons: 1) The Laplace approximation combined with numerical integration; and 2) Its GMRF approximation to the full GRF. However, INLA only models the full marginal distributions of the parameters. If interest is in the joint posterior, then INLA is not appropriate. For example, posterior predictive checks require an estimate of the joint posterior. Another known limitation is when the number of hyper-parameters is greater than a handful where the INLA approximation may become suspect.

Several avenues of future work are possible. To preserve the correlation of the latent field, a fixed-form VB algorithm is worth exploring. This would require one to solve some optimization problems related to the approximate family of distributions. Some research has already been done on this front. For example, Nguyen and Bonilla (2014) and Lloyd et al. (2014). For a nice discussion on fixed-form VB we refer the reader to Opper and Archambeau (2009). In order to avoid possible bias from the Laplacian approximation

and minimum contrast estimates of the decay parameter in the LGCP intensity function, Monte Carlo sampling can be adopted (Paisley et al., 2012). However, to obtain a small Monte Carlo error computational efficiency may be sacrificed for greater accuracy. To further speed up VB, and reduce storage, one can assume a sparse Gaussian field rather than a full GRF, this is an interesting future research direction when dealing with 3D brain imaging data sets with spatially varying covariates, as the full GRF assumption is usually prohibitive for such large data sets. As for INLA, since the full joint posterior distribution is estimated, combining INLA with a copula model may be a viable solution (Ferkingstad et al., 2015).

In Chapter III, we developed an HMC algorithm for the GLM-AR model and compared it with VB and the mass univariate approach. Our findings show that significant differences exist between Bayesian methods and classical approaches. In the mass univariate approach we not only pre-smooth the data but also account for the correlated multiple testing problem using random field theory; while in Bayesian approaches spatiotemporal correlation directly incorporated into the model.

More interestingly, we found that under the 2d simulation studies where the SNR is high, there is very good agreement between HMC and VB in terms of accuracy, variability, and spatial smoothness. This suggests that estimation under the VB algorithm is accurate. For 3d data, SPM VB software uses a graph-partitioned algorithm to speed up computation. However, we find that the use of such an algorithm results in larger variance along the partition boundaries. Since variance is a key component of inference, it is worth considering whether such an algorithm is necessary for the sake of speed.

Since the comparisons are under only one type of spatial prior for the signal, it would be interesting in future work to compare other priors. For example, some GMRF prior (Woolrich et al., 2004a) or a non-stationary diffusion based prior (Harrison et al., 2008a)—

as the use of different priors can lead to different posterior results. Also, as Sidén et al. (2017) points out, there are differences between 2d and 3d simulations. Hence, it would be interesting to conduct full 3d simulations for further exploration.

In Chapter IV, we proposed a new GLM-SVARO model based on empirical evidence that the face-repetition data set in Chapter III has spatially varying AR orders. Rather than fixing the AR orders throughout the brain as in the GLM-AR model, we assume that the orders vary spatially across the brain. Spatial clustering of the AR orders are induced through an underlying Ising prior. With an Ising prior, a potentially troublesome problem is the phase-transition. Therefore we developed theoretical boundaries to restrict the range of the Ising prior hyper-parameters. We update the binary indicators of the Ising prior using the Swendsen-Wang algorithm alternating with Gibbs sampling. We compared GLM-SVARO with GLM-AR where GLM-AR is estimated via both an MCMC algorithm and a VB algorithm.

When simulating data under the prior model, our model outperforms GLM-AR. With GLM-SVARO we get smaller MSE, uniformly higher sensitivity in capturing simulated activations, and higher LPML. When simulating under the competing GLM-AR, our model has slightly higher MSE than GLM-AR using an MCMC algorithm, but lower MSE than GLM-AR using Penny's VB algorithm. This illustrates two things. First, under model mis-specification, GLM-SVARO performs adequately well. Second, PVB tends to give poorer performance as the SNR decreases. The uniformly higher LPML suggests that GLM-SVARO is a better model than GLM-AR in terms of accuracy and posterior inference.

The greatest limitation of GLM-SVARO is computational time. Current version takes about 10 hours on a slice of data and about one week for a whole brain analysis. There are several ways to reduce this time. The most direct way is to use parallel programming.

As different AR orders are assumed independent at each voxel, this parallel computing is fairly feasible. Another possible alternative is to consider modeling the orders using a Potts model. And instead of updating multiple independent Ising fields, one only needs to update one state of the Potts model.

In conclusion, based on the work in this dissertation, we find that given the underlying theory of MCMC, a fully Bayesian approach, via MCMC or HMC, is the most reliable approach in terms of accuracy. Approximate Bayesian methods, when properly used, can serve as powerful tools for fast computation. However, the user should be aware of their limitations and proceed with caution. When and where these approximation methods are likely to be "proper" and "powerful", however, depend on the specific problem setting, and thus one should proceed carefully.

APPENDICES

APPENDIX A

Derivations in Chapter II

A.1 Gradient derivation for ρ

For the circulant matrix $\mathbf{E}$ with base $\mathbf{e} = (e_0, ..., e_{m^2-1})$, the i^{th} eigenvalue is given by:

$$\lambda_i = \sum_{j=0}^{m^2-1} \mathbf{e}_j \exp(\iota 2\pi ji/m^2) \tag{A.1}$$

where $\iota = \sqrt{-1}$. For the power exponential family of correlations $e_j = \exp(-\rho d_j^\delta)$ where d_j is the distance from origin. So we have

$$\lambda_i = \sum_{j=0}^{m^2-1} \exp(-\rho d_j^\delta) \exp(\iota 2\pi ji/m^2) \tag{A.2}$$

Thus, we have $\mathbf{E} = \mathbf{F}\Lambda\mathbf{F}^H$ where $\mathbf{F}$ is the matrix of eigenvectors, and Λ is a diagonal matrix of eigenvalues with i^{th} value to be λ_i.

To derive the partial derivative of $\log \pi(\rho \mid \cdot)$, we first derive the partial derivative of $\mathbf{E}^{\frac{1}{2}}\gamma$

$$\frac{\partial}{\partial\rho}(\mathbf{E}^{\frac{1}{2}}\gamma)_i = (\mathbf{F}\frac{\partial}{\partial\rho}\Lambda^{\frac{1}{2}}\mathbf{F}^H\gamma)_i \tag{A.3}$$

So we need the partial derivative of each diagonal element of $\Lambda^{1/2}$ w.r.t ρ.

$$\frac{\partial \lambda_i^{\frac{1}{2}}}{\partial \rho} = \frac{\partial}{\partial \rho}\Big(\sum_{j=0}^{m^2-1} \exp(-\rho d_j^\delta) \exp(\iota 2\pi j i/m^2)\Big)^{\frac{1}{2}} \tag{A.4}$$

$$= -\frac{1}{2}\lambda_i^{-\frac{1}{2}} \sum_{j=0}^{m^2-1} d_j^\delta e_j \exp(\iota 2\pi j i/m^2) \tag{A.5}$$

The summand in the last line turns out to be the base of a matrix with base $\mathbf{e}^* = (d_0^\delta e_0, ..., d_{m^2-1}^\delta e_{m^2-1})$. Consider the circulant matrix $\mathbf{D}$ with base $\mathbf{d} = (d_j^\delta, ..., d_{m^2-1}^\delta)$. Then it is easy to show that $\mathbf{E}^* = \mathbf{D} \odot \mathbf{E}$ is a circulant matrix with base $\mathbf{e}^* = \mathbf{d} \odot \mathbf{e}$ where $\odot$ represents element wise multiplication. And $\sum_{j=0}^{m^2-1} d_j^\delta e_j \exp(\iota 2\pi j i/m^2)$ is the i^{th} eigenvalue of $\mathbf{E}^*$. Call it ψ_i. Thus,

$$\frac{\partial \lambda_i^{\frac{1}{2}}}{\partial \rho} = -\frac{1}{2}\lambda_i^{-\frac{1}{2}}\psi_i \tag{A.6}$$

Putting this all together we have

$$\frac{\partial}{\partial \rho}(\mathbf{E}^{\frac{1}{2}}\gamma)_i = \frac{\partial}{\partial \rho} - \frac{1}{2}(\mathbf{F}\Lambda^{-\frac{1}{2}}\Psi\mathbf{F}^H\gamma)_i \tag{A.7}$$

$$= -\frac{1}{2}(\mathbf{F}\Lambda^{-\frac{1}{2}}\mathbf{F}^H\mathbf{F}\Psi\mathbf{F}^H\gamma)_i \tag{A.8}$$

$$= -\frac{1}{2}(\mathbf{E}^{-\frac{1}{2}}\mathbf{E}^*\gamma)_i \tag{A.9}$$

Now we can derive the gradient for ρ:

$$\frac{\partial}{\partial \rho}\log \pi(\rho \mid \cdot) = \frac{\partial}{\partial \rho}\Big\{ \sum_i \big[y_i m_i - A\exp(y_i)\big] + \log \pi(\rho)\Big\} \tag{A.10}$$

$$= \sum_i \frac{\partial}{\partial \rho}\Big\{\mu m_i + \sigma(\mathbf{E}^{\frac{1}{2}}\gamma)_i m_i - A\exp\big[\mu + \sigma(\mathbf{E}^{\frac{1}{2}}\gamma)_i\big]\Big\} + \frac{\pi'(\rho)}{\pi(\rho)} \tag{A.11}$$

$$= \sum_i \Big\{\sigma\frac{\partial}{\partial \rho}(\mathbf{E}^{\frac{1}{2}}\gamma)_i m_i - A\frac{\partial}{\partial \rho}\exp\big[\mu + \sigma(\mathbf{E}^{\frac{1}{2}}\gamma)_i\big]\Big\} \tag{A.12}$$

$$= -\frac{1}{2}\sigma\sum_i \Big\{m_i - \frac{A\sigma}{2}\exp\big[\mu + \sigma(\mathbf{E}^{\frac{1}{2}}\gamma)_i\big]\Big\}(\mathbf{E}^{-\frac{1}{2}}\mathbf{E}^*\gamma)_i \tag{A.13}$$

$$= -\frac{\sigma}{2}\Big[\mathbf{m} - A\exp\big(\mu\mathbf{1}_{m^2} + \sigma\mathbf{E}^{\frac{1}{2}}\gamma\big)\Big]^T\mathbf{E}^{-\frac{1}{2}}\mathbf{E}^*\gamma \tag{A.14}$$

where $\mathbf{m} = (m_1, ..., m_{m^2})$ and we use the fact that $\pi'(\rho) = 0$ for flat prior of ρ.

Because $\mathbf{E}^{\frac{1}{2}}\mathbf{E}^{*}\gamma = \mathbf{F}\Lambda^{-\frac{1}{2}}\mathbf{F}^{H}\mathbf{F}\Psi\mathbf{F}^{H}\gamma = \mathbf{F}\Lambda^{-\frac{1}{2}}\Psi\mathbf{F}^{H}\gamma$, we can use the DFT to compute all the matrix operations in the equation above.

APPENDIX B

Derivations in Chapter III

B.1 Re-expression of the log-likelihood

By elaborating the vector multiplication in Equation 3.3, we have

$$\text{(B.1)} \quad l_n = -\frac{\lambda_n}{2}\sum_{t=P+1}^{T}\left[y_{tn} - \sum_k x_{tk}w_{kn} - \sum_{p=1}^{P}(y_{t-p,n} - \sum_k x_{t-p,k}w_{kn})a_{pn}\right]^2 + \frac{T-P}{2}\log\lambda_n + const$$

Let $\mathbf{a}_n^* = (-1, \mathbf{a}_n^T)^T$, so $a_{pn}^* = a_{pn}$ if $p \geq 1$ and $a_{pn}^* = -1$ if $p = 0$, then equation (B.1) can be written as

$$
\begin{aligned}
l_n \quad &= -\frac{\lambda_n}{2}\sum_{t=P+1}^{T}\left[\sum_{p=0}^{P} y_{t-p,n}a^*_{pn} - \sum_{p=0}^{P}\sum_{k} x_{t-p,k}w_{kn}a^*_{pn}\right]^2 \\
&+\frac{T-P}{2}\log\lambda_n + const \\
&= -\frac{\lambda_n}{2}\sum_{t=P+1}^{T}\Big(\sum_{p_1=0}^{P}\sum_{p_2=0}^{P} y_{t-p_1,n}y_{t-p_2,n}a^*_{p_1n}a^*_{p_2n} - 2\sum_{p_1=0}^{P}\sum_{p_2=0}^{P}\sum_{k=1}^{K} y_{t-p_1,n} \\
&x_{t-p_2,k}w_{kn}a_{p_1n}a_{p_2n} + \sum_{p_1=1}^{P}\sum_{p_2=1}^{P}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K} x_{t-p_1,k_1}x_{t-p_2,k_2}w_{k_1n}w_{k_2n}a_{p_1n}a_{p_2n}\Big) \\
&+\frac{T-P}{2}\log\lambda_n + const \\
&= -\frac{\lambda_n}{2}\Big(\sum_{p_1=0}^{P}\sum_{p_2=0}^{P} yy_{p_1p_2n}a_{p_1n}a_{p_2n} - 2\sum_{p_1=0}^{P}\sum_{p_2=0}^{P}\sum_{k=1}^{K} yx_{p_1np_2k}w_{kn}a_{p_1n}a_{p_2n} \\
&+\sum_{p_1=0}^{P}\sum_{p_2=0}^{P}\sum_{k_1=1}^{K}\sum_{k_2=1}^{K} xx_{p_1k_1p_2k_2}w_{k_1n}w_{k_2n}a_{p_1n}a_{p_2n}\Big) + \frac{T-P}{2}\log\lambda_n + const
\end{aligned}
$$

where

$$
\begin{aligned}
yy_{p_1p_2n} &= \sum_{t=P+1}^{T} y_{t-p_1,n}y_{t-p_2,n}, \\
yx_{p_1np_2k} &= \sum_{t=P+1}^{T} y_{t-p_1,n}x_{t-p_2,k}, \\
xx_{p_1k_1p_2k_2} &= \sum_{t=P+1}^{T} x_{t-p_1,k_1}x_{t-p_2,k_2}.
\end{aligned}
$$

In this way, the sum across t can be pre-computed instead of computing at every iteration in the algorithm.

Define $\mathbf{F}$ to be a $P \times P$ matrix with (p_1, p_2) entry

$$
f_{p_1p_2} = yy_{p_1p_2n} - 2\sum_{k=1}^{K} yx_{p_1np_2k}w_{kn} + \sum_{k_1=1}^{K}\sum_{k_2=1}^{K} xx_{p_1k_1p_2k_2}w_{k_1n}w_{k_2n}
$$

Then the derivation above is just

$$
l_n = -\frac{\lambda_n}{2}\mathbf{a}^{*T}_n\mathbf{F}\mathbf{a}^*_n + \frac{T-P}{2}\log\lambda_n + const. \tag{B.2}
$$

which is Equation (19).

B.2 Derivation of the gradients

Based on the re-expression of the likelihood, the gradients are derived as follows:

$$\nabla w_{kn} \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = \lambda_n \Big(\sum_{p_1=0}^{P} \sum_{p_2=0}^{P} y x_{p_1 n p_2 k} a_{p_1 n} a_{p_2 n} - \sum_{p_1=0}^{P} \sum_{p_2=0}^{P} \sum_{k_2=1}^{K} x x_{p_1 k p_2 k_2} w_{k_2 n} a_{p_1 n} a_{p_2 n} \Big) - \alpha_k (\mathbf{S}^T \mathbf{S})_n \mathbf{w}_k^T$$
$$= \lambda_n \mathbf{a}_n^{*T} \mathbf{G} \mathbf{a}_n^* - \alpha_k (\mathbf{S}^T \mathbf{S})_n \mathbf{w}_k^T$$

where $\mathbf{G}$ is a $P \times P$ matrix with (p_1, p_2) entry $g_{p_1 p_2} = y x_{p_1 n p_2 k} - \sum_{k_2=1}^{K} x x_{p_1 k p_2 k_2} w_{k_2 n}$.

$$\nabla a_{pn} \log p(\boldsymbol{\theta} \mid \mathbf{Y}, \mathbf{X}) = \lambda_n \Big(- \sum_{p_2=0}^{P} y y_{p p_2 n} a_{p_2 n} + \sum_{p_2=0}^{P} \sum_{k=1}^{K} y x_{p n p_2 k} w_{kn} a_{p_2 n}$$
$$+ \sum_{p_1=0}^{P} \sum_{k=1}^{K} y x_{p_1 n p k} w_{kn} a_{p_1 n} - \sum_{p_2=0}^{P} \sum_{k_1=1}^{K} \sum_{k_2=1}^{K} x x_{p k_1 p_2 k_2} w_{k_1 n} w_{k_2 n} a_{p_2 n} \Big) - \beta_p (\mathbf{D}^T \mathbf{D})_n \mathbf{a}_p^T$$
$$= \lambda_n \mathbf{f}_p \mathbf{a}_n^* - \beta_p (\mathbf{D}^T \mathbf{D})_n \mathbf{a}_p^T$$

where $\mathbf{f}_p$ is just the p^{th} row of $\mathbf{F}$.

APPENDIX C

Derivations in Chapter IV

C.1 Log-likelihood

Let c denote the normalizing constant, the log-likelihood l can be expressed as:

$$\begin{aligned} l &= \sum_n \sum_t -\frac{\lambda_n}{2}(e_{tn} - \sum_p \tilde{e}_{ntp} a_{pn})^2 + \frac{T-P}{2}\sum_n \log \lambda_n + c \\ &= \sum_n \sum_t -\frac{\lambda_n}{2}\Big[(y_{tn} - \sum_k \mathbf{x}_{tk} w_{kn}) - \sum_p (y_{t-p,n} - \sum_k x_{t-p,k} w_{kn}) a_{pn}\Big]^2 \\ &+ \frac{T-P}{2}\sum_n \log \lambda_n + c \\ &= \sum_n \sum_t -\frac{\lambda_n}{2}\Big[\sum_p (y_{t-p,n} - \sum_k x_{t-p,k} w_{kn}) a^*_{pn}\Big]^2 + \frac{T-P}{2}\sum_n \log \lambda_n + c \end{aligned}$$

where $a^*_{pn} = a_{pn}$ if $p \neq 0$ and $a^*_{pn} = -1$ otherwise.

C.2 Priors

All priors are explicitly given in Chapter IV, here we add a supplement for $\mathbf{a}_n$. In practice, when we update $\mathbf{a}_n$, the spike-and-slab prior are parametrized as follows:

$$a_{pn} \mid \gamma_{pn} = \gamma_{pn}\mathrm{N}(0, \tau_p^{-1}) + (1-\gamma_{pn})\mathrm{N}(0, (\epsilon\tau_p)^{-1})$$

where ϵ is a very large constant so that the spike part has a extremely low variance that approximates point mass at zero. This parametrization can help with the mixing than a

pure spike which is an exact point mass at 0. The log-prior is therefore:

$$\log \pi(a_{pn} \mid \gamma_{pn}) = -\frac{\tau_p}{2} a_{pn}^2 \delta(\gamma_{pn}) + \frac{1}{2} \log \tau_p + \frac{1}{2} \log \delta(\gamma_{pn}) + c$$

where $\delta(\gamma_{pn}) = \epsilon$ if $\gamma_{pn} = 0$ and 1 otherwise.

C.3 Posterior distribution

We derive the posterior distribution for $\mathbf{w}_n$, $\mathbf{a}_n$, γ_{pn}, α_k, τ_p, λ_n.

C.3.1 For $\mathbf{w}_n$

Let $\tilde{\mathbf{y}}_{tn} \equiv (y_{t,n}, y_{t-1,n}, ..., y_{t-P,n})$, $\tilde{\mathbf{x}}_{tk} \equiv (x_{t,k}, x_{t-1,k}, ..., x_{t-P,k})$. Then putting x_k together, define $\tilde{\mathbf{X}}_t \equiv (\tilde{\mathbf{x}}_1^T, ..., \tilde{\mathbf{x}}_K^T)^T$. We have

$$\begin{aligned}
\log(\mathbf{w}_n|\cdot) &= \frac{\lambda_n}{2} \sum_t \left[(\tilde{\mathbf{y}}_{tn} - \tilde{\mathbf{X}}_t \mathbf{w}_n)^T \mathbf{a}_n^* \right]^2 - \sum_k \frac{\alpha_k}{2} \mathbf{w}_k^T (\mathbf{S}^T \mathbf{S}) \mathbf{w}_k + c \\
&= -\frac{\lambda_n}{2} \sum_t \left[\tilde{\mathbf{y}}_{tn} \mathbf{a}_n^* - (\tilde{\mathbf{X}}_t \mathbf{a}_n^*)^T \mathbf{w}_n \right]^2 - \sum_k \frac{\alpha_k}{2} \mathbf{w}_k^T (\mathbf{S}^T \mathbf{S}) \mathbf{w}_k + c \\
&= -\frac{1}{2} \mathbf{w}_n^T \Big[\lambda_n \sum_t (\tilde{\mathbf{X}}_t \mathbf{a}_n^*)(\tilde{\mathbf{X}}_t \mathbf{a}_n^*)^T + (\mathbf{S}^T \mathbf{S})_{nn} \text{Diag}(\boldsymbol{\alpha}) \Big] \mathbf{w}_n \\
&+ \Big[\lambda_n \sum_t (\tilde{\mathbf{y}}_{tn} \mathbf{a}_n^*)(\tilde{\mathbf{X}}_t \mathbf{a}_n^*) - \boldsymbol{\alpha} \circ \sum_{n' \neq n} (\mathbf{S}^T \mathbf{S})_{nn'} \mathbf{w}_{n'} \Big]^T \mathbf{w}_n + c
\end{aligned}$$

where $\text{Diag}(\mathbf{v})$ denotes the diagnoal matrix with diagonal elements formed by vector $\mathbf{v}$, and $\circ$ denotes Hadamard product.

Thus

$$\mathbf{w}_n \sim \text{N}(\boldsymbol{\mu}_{\mathbf{W}}^n, \boldsymbol{\Sigma}_{\mathbf{W}}^n)$$

with

$$\begin{aligned}
\boldsymbol{\Sigma}_{\mathbf{W}}^n &= \Big[\lambda_n \sum_t (\tilde{\mathbf{X}}_t \mathbf{a}_n^*)(\tilde{\mathbf{X}}_t \mathbf{a}_n^*)^T + (\mathbf{S}^T \mathbf{S})_{nn'} \text{Diag}(\boldsymbol{\alpha}) \Big]^{-1} \\
\boldsymbol{\mu}_{\mathbf{W}}^n &= \Big[\lambda_n \sum_t (\tilde{\mathbf{y}}_{tn} \mathbf{a}_n^*)(\tilde{\mathbf{X}}_t^T \mathbf{a}_n^*) - \boldsymbol{\alpha} \circ \sum_{n' \neq n} (\mathbf{S}^T \mathbf{S})_{nn'} \mathbf{w}_{n'} \Big] \boldsymbol{\Sigma}_{\mathbf{W}}
\end{aligned}$$

C.3.2 For $\mathbf{a}_n$

Denote $\tilde{\mathbf{y}}^*_{tn} \equiv (y_{t-1,n}, ..., y_{t-P,n})$, $\tilde{\mathbf{x}}^*_{tk} \equiv (x_{t-1,k}, ..., x_{t-P,k})$, $\tilde{\mathbf{X}}^*_t \equiv (\tilde{\mathbf{x}}^{*T}_1, ..., \tilde{\mathbf{x}}^{*T}_K)^T$.

$$\begin{aligned}
\log p(\mathbf{a}_n|\cdot) &= -\frac{\lambda_n}{2}\sum_t \Big[-(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_t\mathbf{w}_n)^T\mathbf{a}_n + (\mathbf{y}_{tn} - \mathbf{X}_t\mathbf{w}_n)\Big]^2 - \sum_p \frac{\tau_p}{2}\delta(\gamma_{pn})a^2_{pn} \\
&= -\frac{1}{2}\mathbf{a}_n^T\Big[\lambda_n\sum_t(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_{tn}\mathbf{w}_n)(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_t\mathbf{w}_n)^T + \text{Diag}(\boldsymbol{\tau}\circ\delta(\boldsymbol{\gamma}_n))\Big]\mathbf{a}_n \\
&+ \lambda_n(y_{tn} - \mathbf{X}_t\mathbf{w}_n)(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_t\mathbf{w}_n)^T\mathbf{a}_n
\end{aligned}$$

Thus

$$\mathbf{a}_n \sim \text{N}(\boldsymbol{\mu}^n_{\mathbf{A}}, \boldsymbol{\Sigma}^n_{\mathbf{A}})$$

with

$$\begin{aligned}
\boldsymbol{\Sigma}^n_{\mathbf{A}} &= \Big[\sum_t(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_t\mathbf{w}_n)(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_t\mathbf{w}_n)^T + \text{Diag}(\boldsymbol{\tau}\circ\delta(\boldsymbol{\gamma}_n))\Big]^{-1} \\
\boldsymbol{\mu}^n_{\mathbf{A}} &= \Big[\lambda_n\sum_t(y_{tn} - \mathbf{X}_t\mathbf{w}_n)(\tilde{\mathbf{y}}^*_{tn} - \tilde{\mathbf{X}}^*_t\mathbf{w}_n)^T\Big]\boldsymbol{\Sigma}^n_{\mathbf{A}}
\end{aligned}$$

C.3.3 For γ_n

$$\begin{aligned}
p(\gamma_{pn} = 1 \mid \cdot) &= \frac{L(\gamma_{pn}=1)p(\gamma_{pn}=1 \mid \gamma_{-pn})}{L(\gamma_{pn}=1)p(\gamma_{pn}=1 \mid \gamma_{-pn}) + L(\gamma_{pn}=0)p(\gamma_{pn}=0 \mid \gamma_{-pn})} \\
&= \frac{L(\gamma_{pn}=1)/L(\gamma_{pn}=0)\exp\Big\{\beta_{0p} + \beta_{1p}\sum_{n'\sim n}\gamma_{pn'}\Big\}}{L(\gamma_{pn}=1)/L(\gamma_{pn}=0)\exp\Big\{\beta_{0p} + \beta_{1p}\sum_{n'\sim n}\gamma_{pn'}\Big\} + 1}
\end{aligned}$$

where $L(\gamma_{pn})$ is the likelihood associated with γ_{pn}. So we have:

$$\begin{aligned}
\frac{L(\gamma_{pn}=1)}{L(\gamma_{pn}=0)} &= \exp\Big\{-\frac{\tau_p}{2}a^2_{pn} + \frac{\epsilon\tau_p}{2}a^2_{pn} - \frac{1}{2}\log\epsilon\Big\} \\
&= \exp\Big\{\frac{(\epsilon-1)\tau_p}{2}a^2_{pn} - \frac{1}{2}\log\epsilon\Big\}
\end{aligned}$$

Thus

$$\gamma_{pn} \sim \text{Ber}(p(\gamma_{pn} = 1 \mid \cdot))$$

C.3.4 Swendsen-Wang update of γ_p

1. For any pair of neighbors (n_1, n_2) that $\gamma_{pn_1} = \gamma_{pn_2}$, form bonds with probability $1 - \exp(-\beta_{1p})$.

2. Let $\{n\}$ denote the set of voxels that belong to one common cluster. For each of the cluster $\{n\}$, calculate:

$$\begin{aligned} p(\gamma_{p\{n\}} &= 1 \mid \cdot) = \frac{L(\gamma_{p\{n\}}=1)\exp\left\{\beta_0 \sum_{n\in\{n\}} \gamma_{pn}\right\}}{L(\gamma_{p\{n\}}=1)\exp\left\{\beta_0 \sum_{n\in\{n\}} \gamma_{pn}\right\} + L(\gamma_{p\{n\}}=0)} \\ &= \frac{L(\gamma_{p\{n\}}=1)/L(\gamma_{p\{n\}}=0)\exp\left\{\beta_0 \sum_{n\in\{n\}} \gamma_{pn}\right\}}{L(\gamma_{p\{n\}}=1)/L(\gamma_{p\{n\}}=0)\exp\left\{\beta_0 \sum_{n\in\{n\}} \gamma_{pn}\right\} + 1} \\ &= \frac{\exp\left\{\beta_0 \sum_{n\in\{n\}} \gamma_{pn} + \frac{1}{2}(\epsilon-1)\tau_p \sum_{\{n\}} a^2_{p\{n\}} - \sum_{\{n\}} \frac{1}{2}\log\epsilon\right\}}{\exp\left\{\beta_0 \sum_{n\in\{n\}} \gamma_{pn} + \frac{1}{2}(\epsilon-1)\tau_p \sum_{\{n\}} a^2_{p\{n\}} - \sum_{\{n\}} \frac{1}{2}\log\epsilon\right\} + 1} \end{aligned}$$

where $L(\gamma_{p\{n\}}=1)$ and $L(\gamma_{p\{n\}}=0)$ is the likelihood associated with $\gamma_{p\{n\}}$.

C.3.5 For α_k

$$\alpha_k \sim G\Big(\frac{N}{2} + q_1 - 1, \Big[\frac{1}{2}\mathbf{w}_k^T(\mathbf{S}^T\mathbf{S})\mathbf{w}_k + \frac{1}{q_2}\Big]^{-1}\Big)$$

C.3.6 For τ_p

$$\begin{aligned} \log p(\tau_p|\cdot) &= \sum_n \left[-\frac{\tau_p}{2}a^2_{pn}\delta(\gamma_{pn}) + \frac{1}{2}\log\tau_p\right] + (u_1-1)\log\tau_p - \tau_p/u_2 \\ &= \left(\frac{N}{2} + u_1 - 1\right)\log\tau_p - \left(\frac{1}{2}\sum_n a^2_{pn}\delta(\gamma_{pn}) + \frac{1}{u_2}\right)\tau_p \end{aligned}$$

$$\tau_p \sim \mathbf{G}\left(\frac{N}{2} + u_1 - 1, \frac{1}{2}\sum_n a^2_{pn}\delta(\gamma_{pn}) + \frac{1}{u_2}\right)$$

C.3.7 For λ_n

$$\lambda_n \sim \mathrm{G}\Big(\frac{T-P}{2} + r_1 - 1, \Big(\frac{1}{2}\sum_t \Big[\sum_p \Big(y_{t-p,n} - \sum_k x_{t-p,k} w_{kn}\Big) a^*_{pn}\Big]^2 + \frac{1}{r_2}\Big)^{-1}\Big)$$

C.4 Updating Scheme

The parameter are updated according to the following sequence:

1. Update $\mathbf{w}_n$ for $n = 1, ..., N$
2. Update $\mathbf{a}_n$ for $n = 1, ..., N$.
3. Update $\boldsymbol{\gamma}_p$ for $p = 1, ..., P$
4. Update α_k for $k = 1, ..., K$.
5. Update τ_p for $p = 1, , , .P$.
6. Update λ_n for $n = 1, ..., N$
7. Repeat step 1-6 for sufficiently long time.

C.5 Proof of neighboring pairs

Since the length of the cubic is V_p, and the neighbors are all 1^{st} order neighbors. It is easy to show that:

The number of voxels having 3 neighbors is 8

The number of voxels having 4 neighbors is $12(V_p - 2)$

The number of voxels having 5 neighbors is $6(V_p - 2)^2$.

The number of voxels having 6 neighbors is $(V_p - 2)^3$

Thus, the total number of neighbors are

$$\frac{1}{2}[6(V_p - 2)^3 + 30(V_p - 2)^2 + 48(V_p - 2) + 24]$$

which is exactly $3V_p^2(V_p - 1)$.

APPENDIX D

Supplementary figures for Chapter III

D.0.1 Simulation One

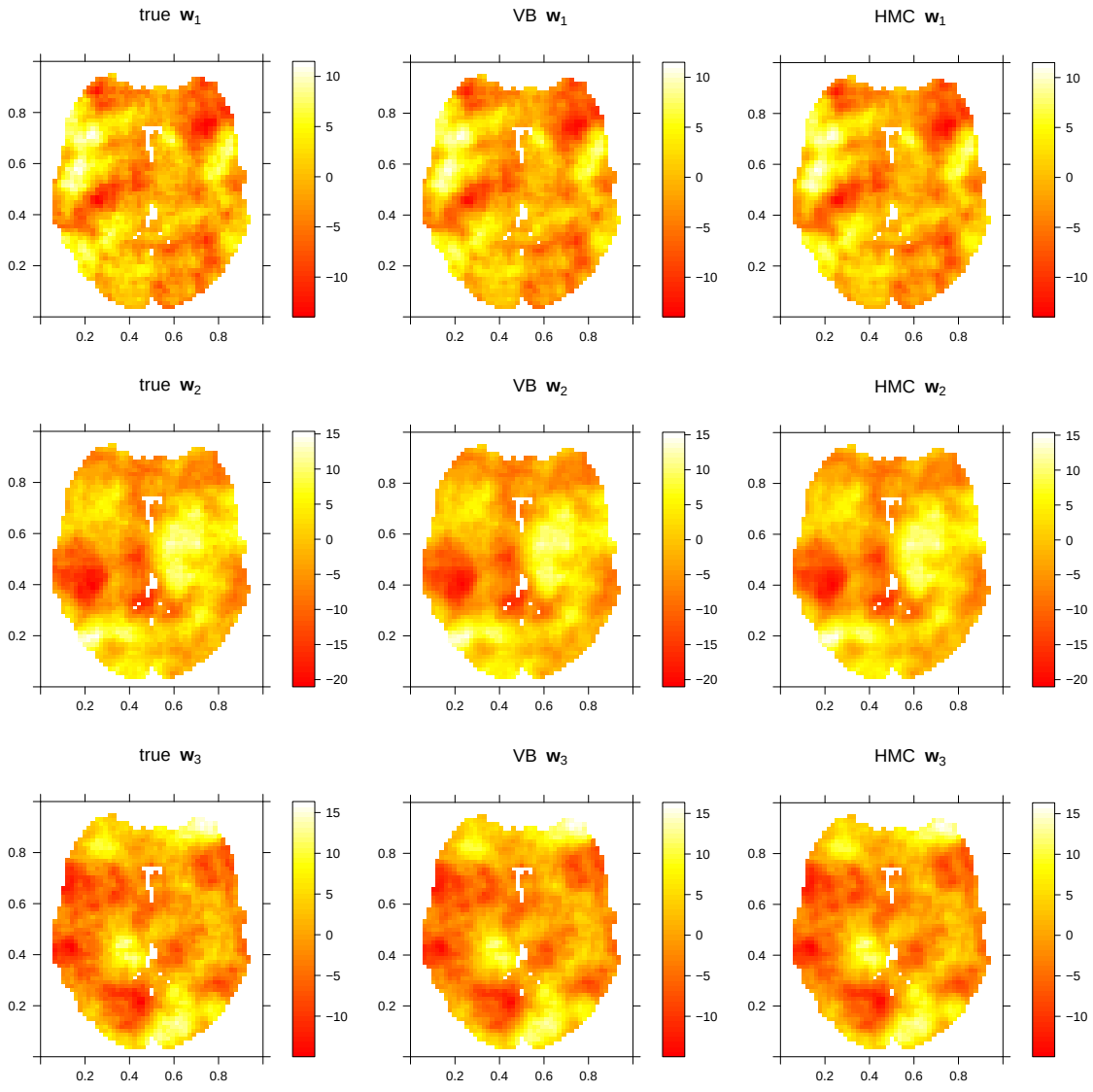


Figure D.1: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ from HMC and VB for Simulation One. The estimates are compared with true image in each row.

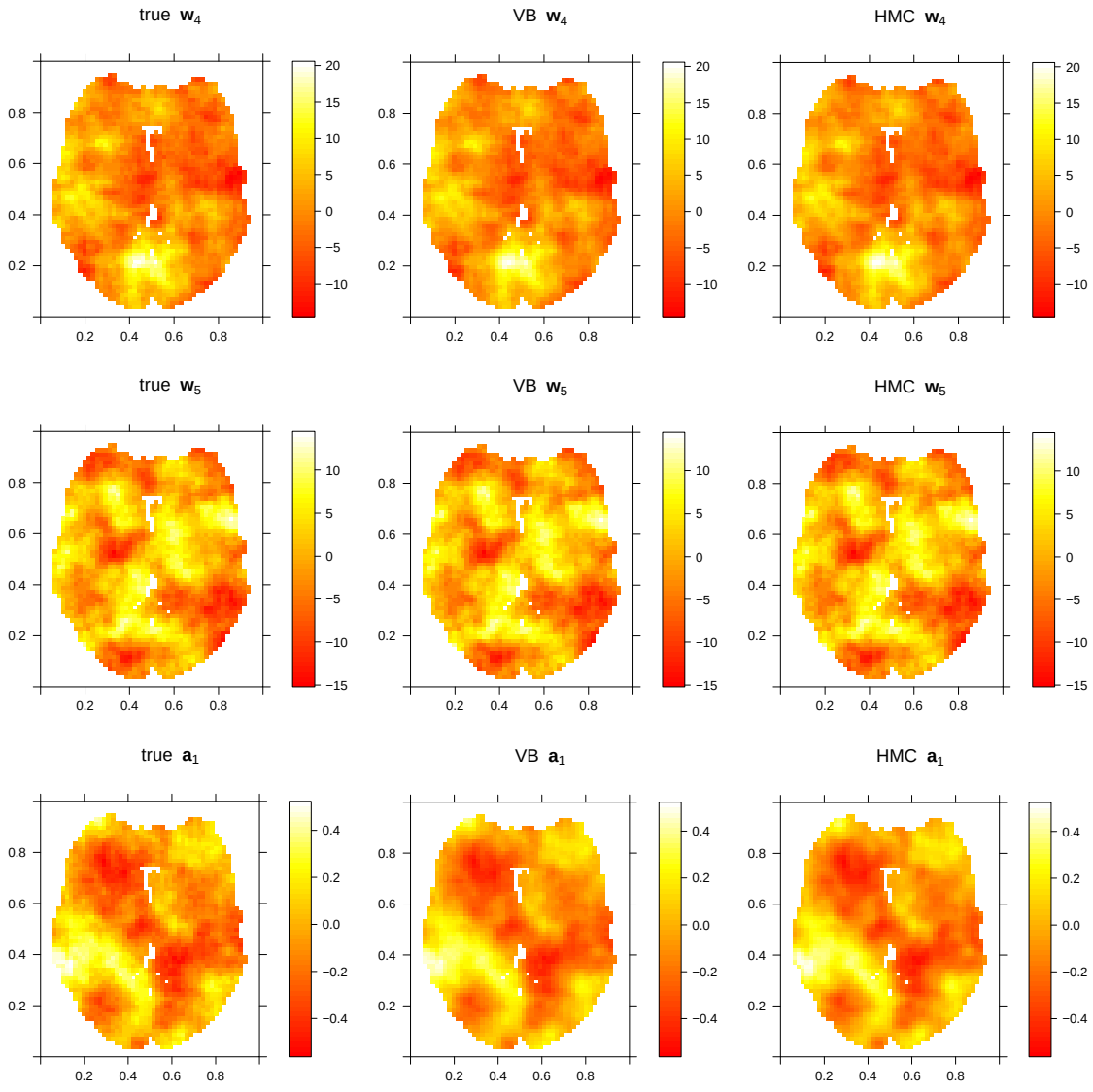


Figure D.2: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_4$, $\mathbf{w}_5$, $\mathbf{a}_1$ from HMC and VB for Simulation One. The estimates are compared with true image in each row.

D.0.2 Simulation Two

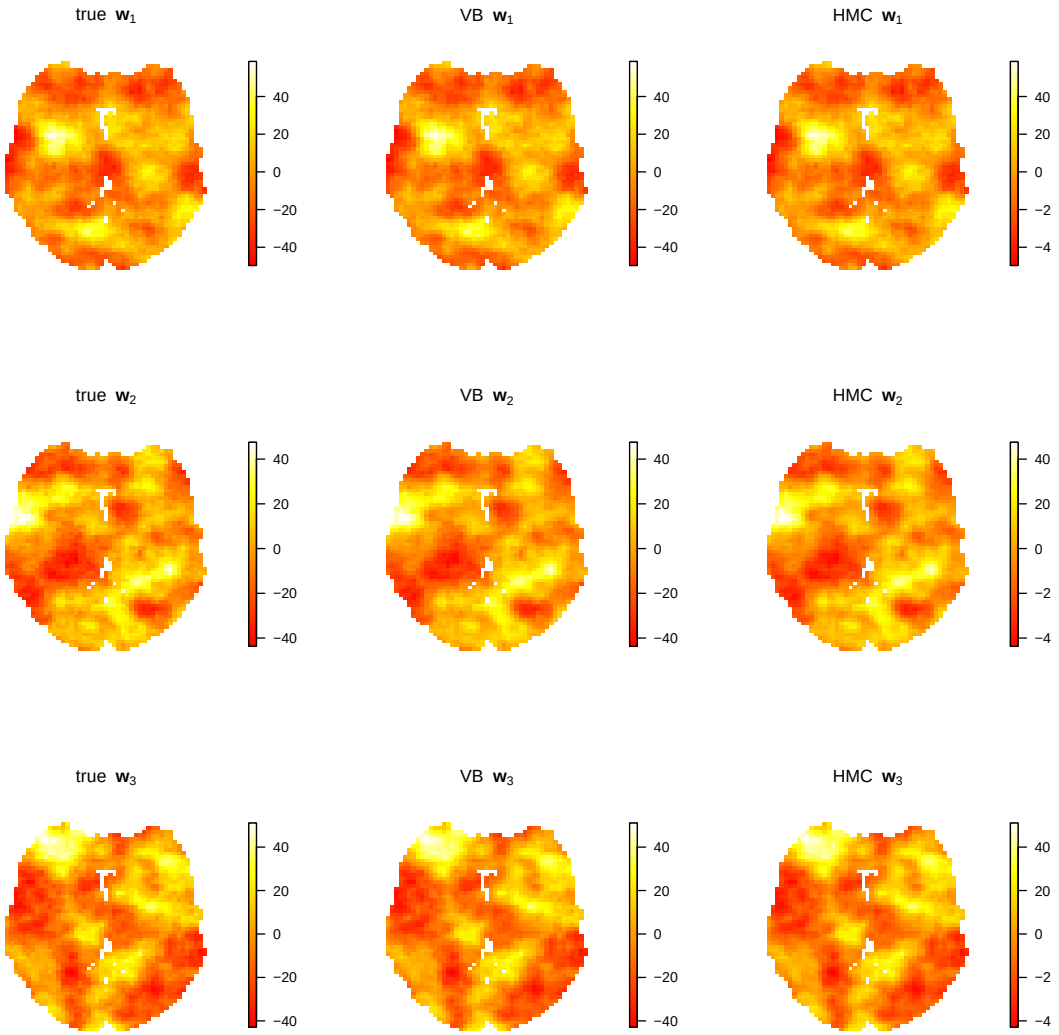


Figure D.3: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_1$, $\mathbf{w}_2$, $\mathbf{w}_3$ from HMC and VB for Simulation Two. The estimates are compared with true image in each row.

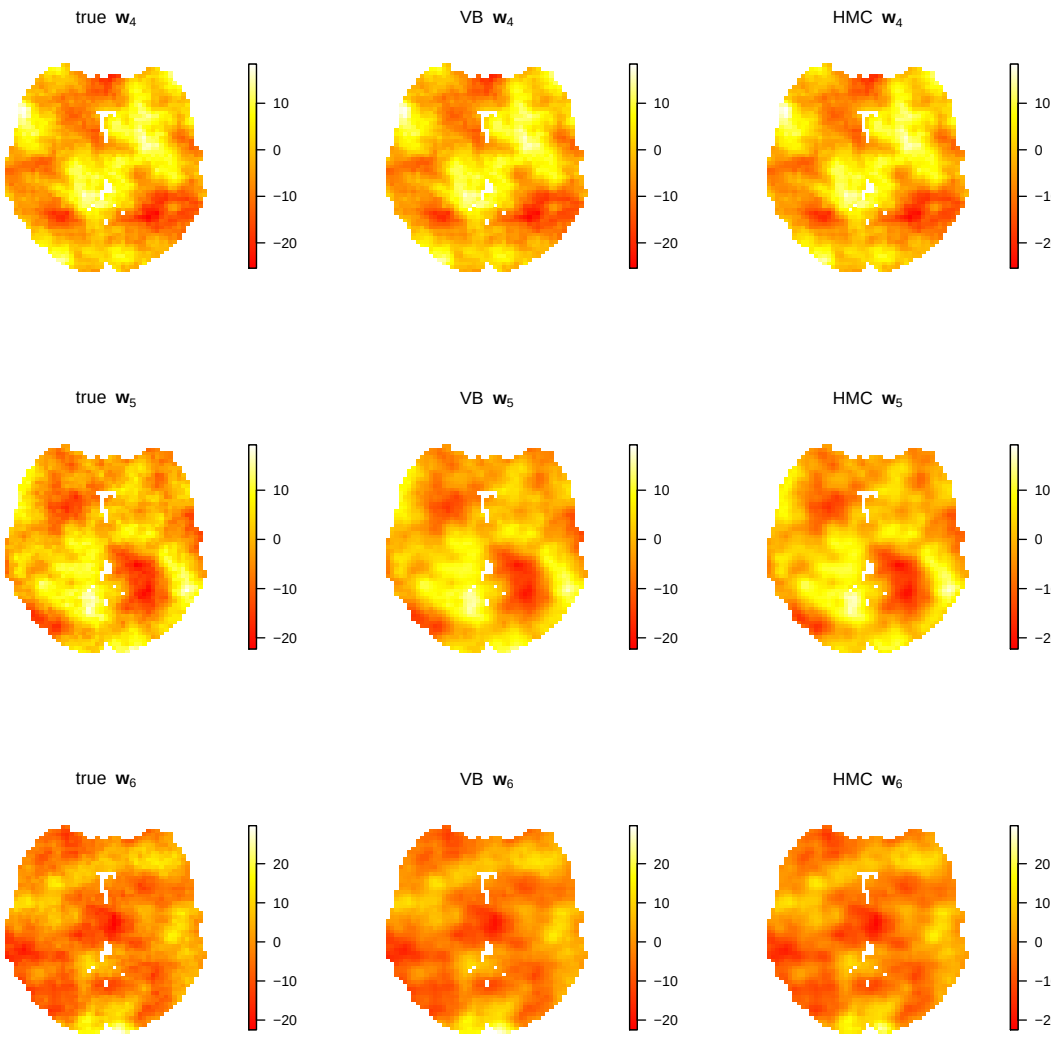


Figure D.4: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_4$, $\mathbf{w}_5$, $\mathbf{w}_6$ from HMC and VB for Simulation Two. The estimates are compared with true image in each row.

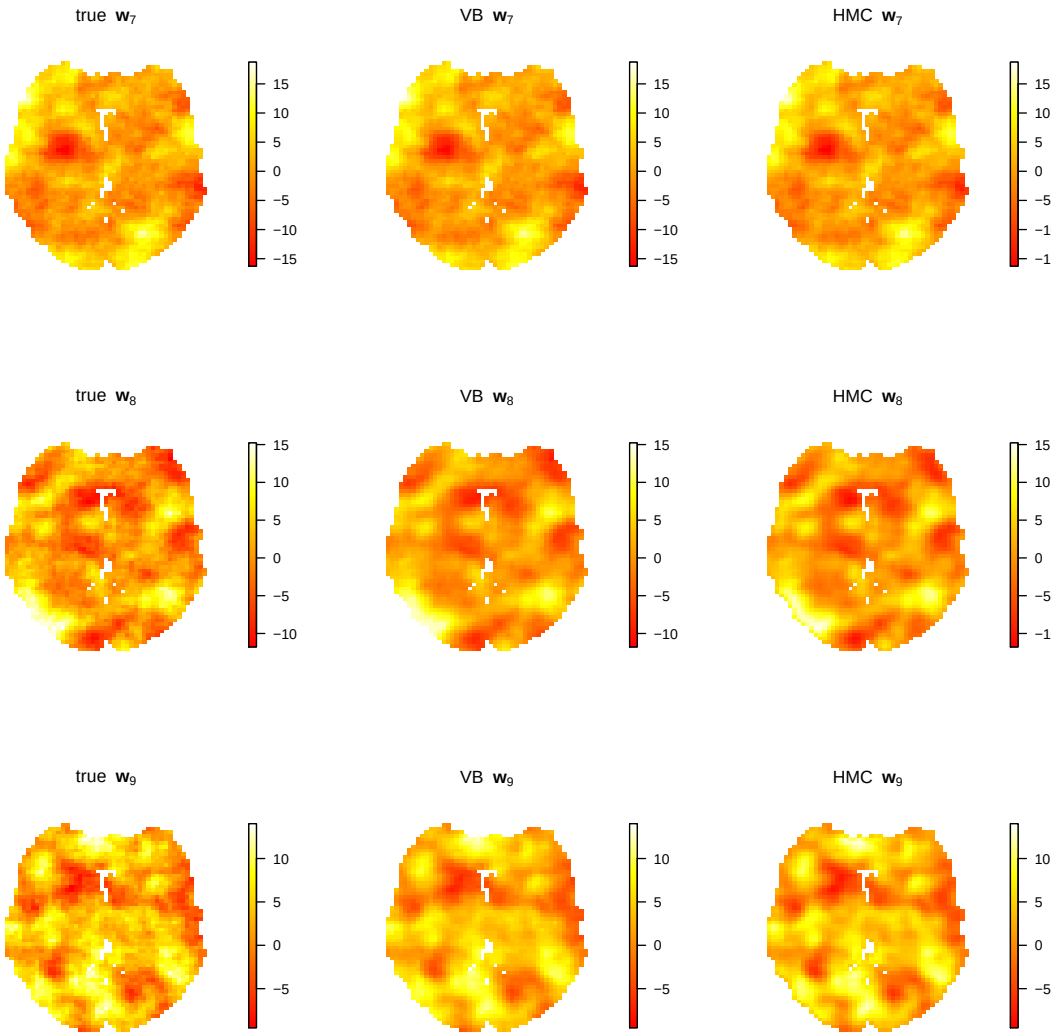


Figure D.5: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_7$, $\mathbf{w}_8$, $\mathbf{w}_9$ from HMC and VB for Simulation Two. The estimates are compared with true image in each row.

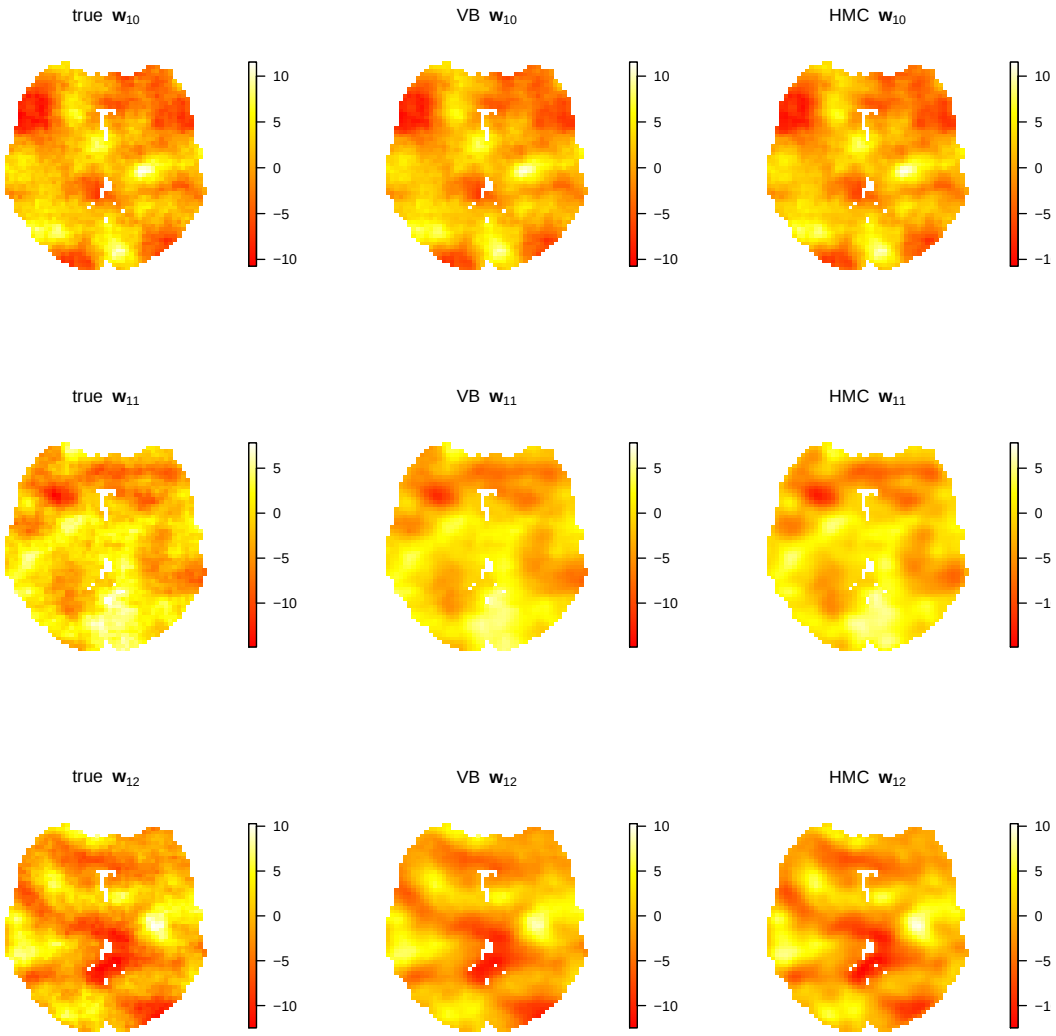


Figure D.6: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_{10}$, $\mathbf{w}_{11}$, $\mathbf{w}_{12}$ from HMC and VB for Simulation Two. The estimates are compared with true image in each row.

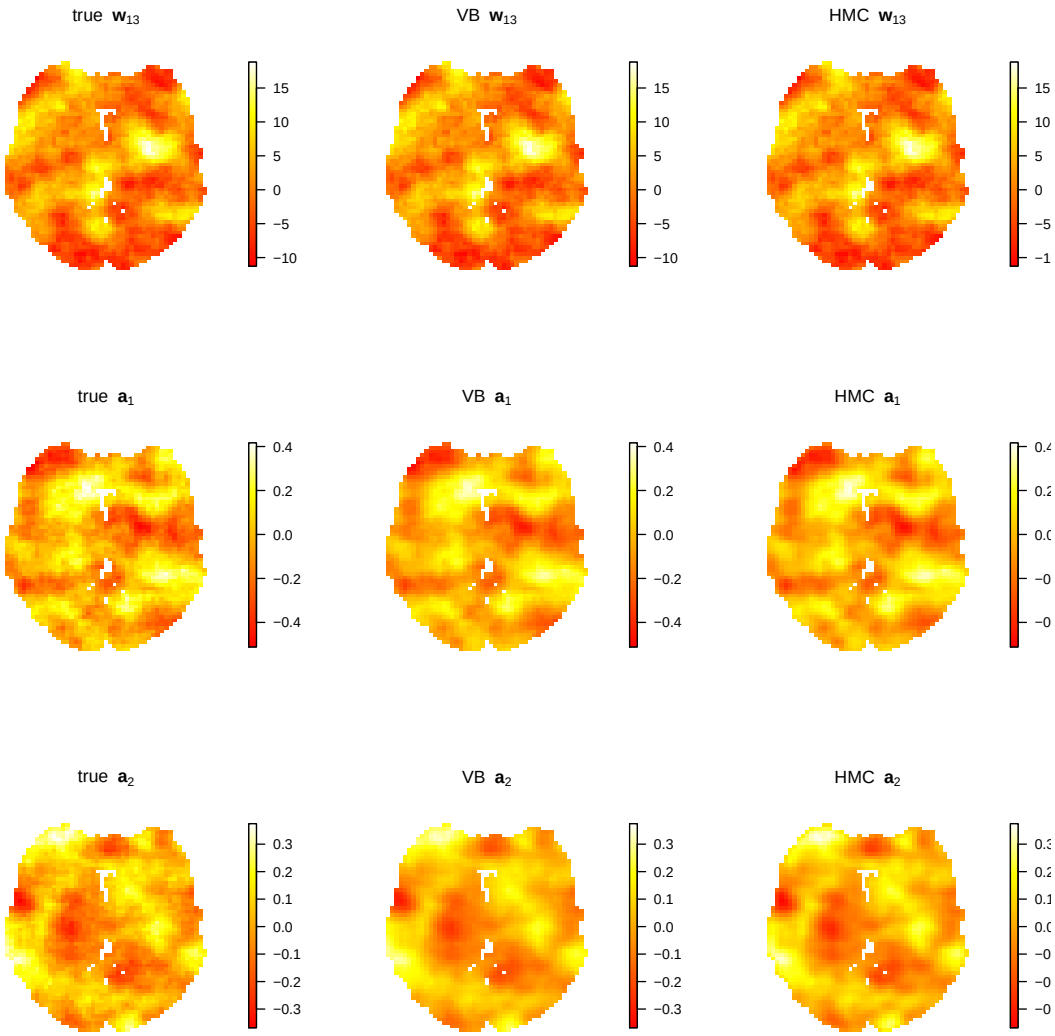


Figure D.7: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{w}_{13}$, $\mathbf{a}_1$, $\mathbf{a}_2$ from HMC and VB for Simulation Two. The estimates are compared with true image in each row.

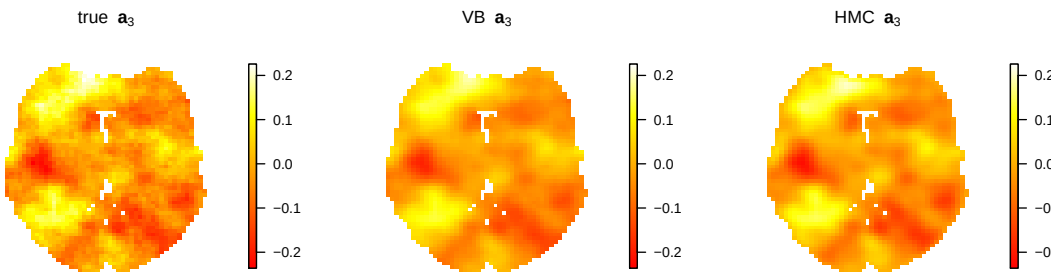


Figure D.8: Image of average (over simulation replicates) posterior mean estimate of $\mathbf{a}_3$ from HMC and VB for Simulation Two. The estimates are compared with true image.

D.0.3 Real applicatioin

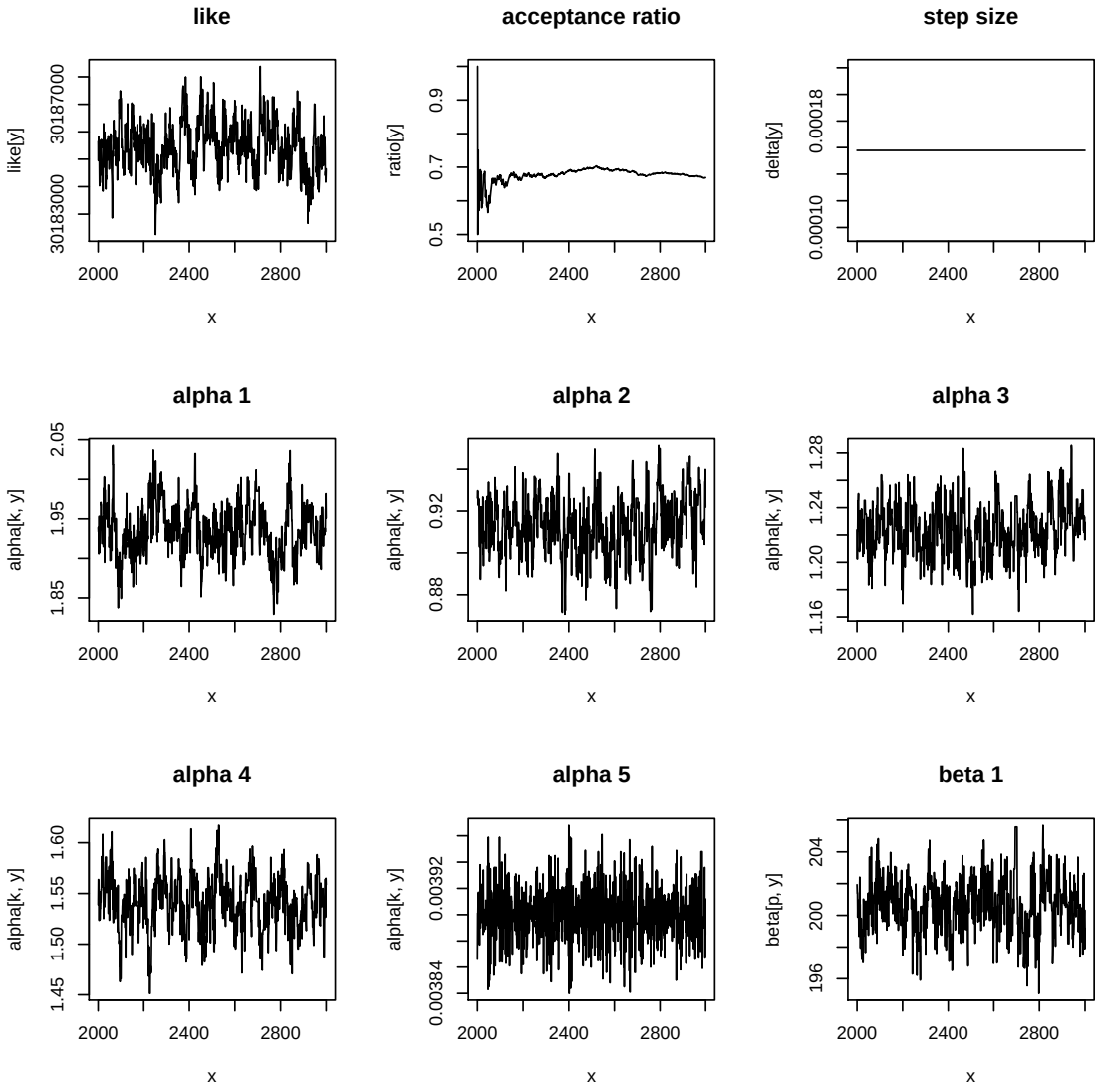


Figure D.9: Traceplot for the parameters from HMC. The chain runs for 3000 iterations, with first 2000 as burn-in and thrown away. The three figures on top row (from left to right) are likelihood, acceptance ratio of Metropolis-Hastings step, and leapfrog step size δ respectively. The rest shows the trace plots from $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$.

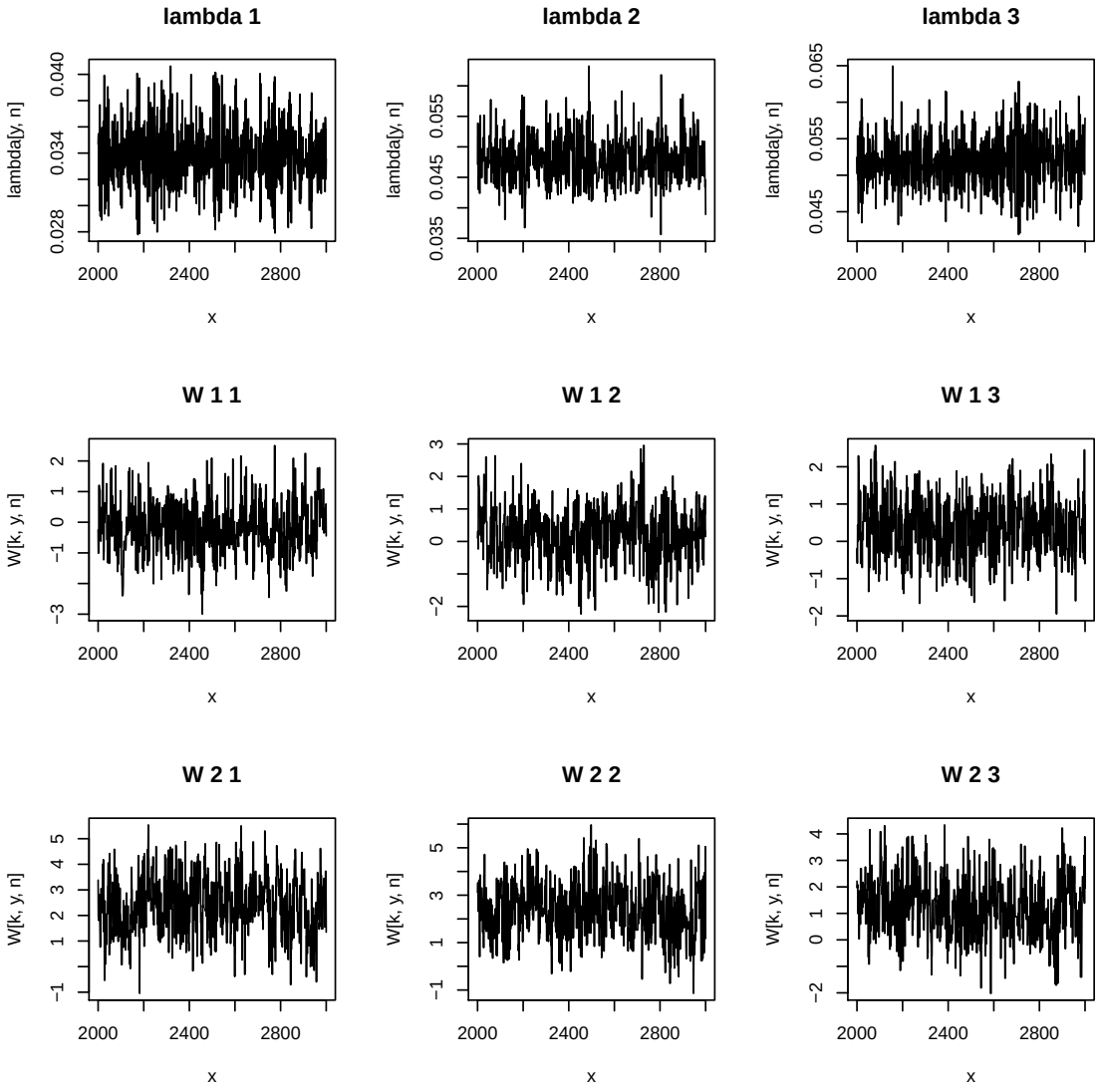


Figure D.10: Traceplot for the parameters from HMC. The chain runs for 3000 iterations, with first 2000 as burn-in and thrown away. The top row represents the trace plots for $\lambda_1, \lambda_2, \lambda_3$. The second and third row shows trace plots from w_{11}, w_{12}, w_{13} and w_{21}, w_{22}, w_{23}. We just show the trace plots from first three voxels out of 56527 voxels due to a limited space.

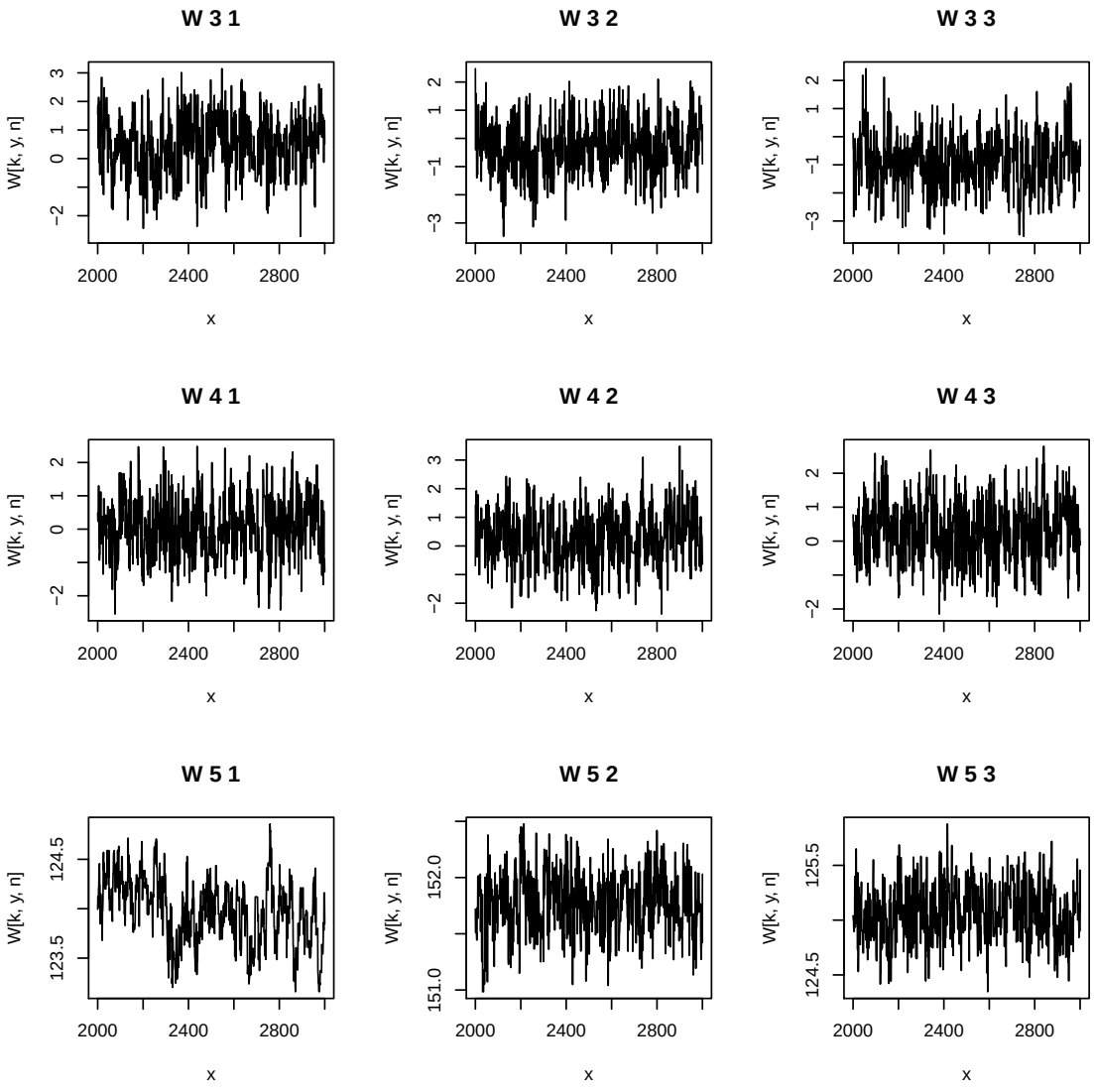


Figure D.11: Traceplot for the parameters ($\mathbf{w}_3$ to $\mathbf{w}_5$) from HMC. The chain runs for 3000 iterations, with first 2000 as burn-in and thrown away.

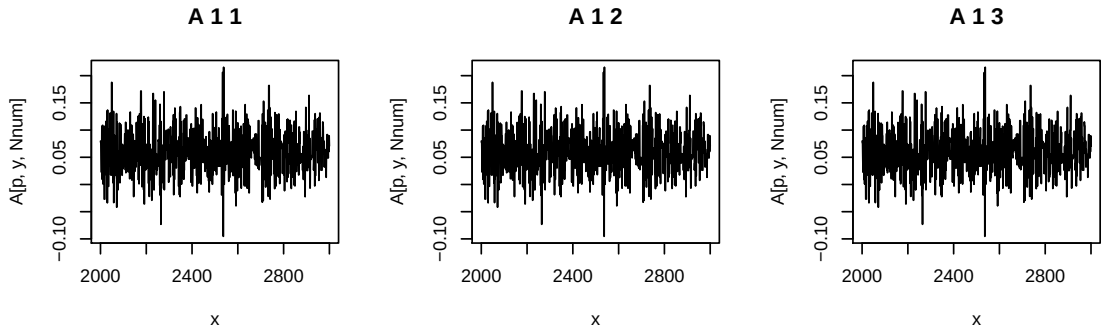


Figure D.12: Traceplot for the auto-regressive coefficient $\mathbf{a}_1$ from HMC. The chain runs for 3000 iterations, with first 2000 as burn-in and thrown away.

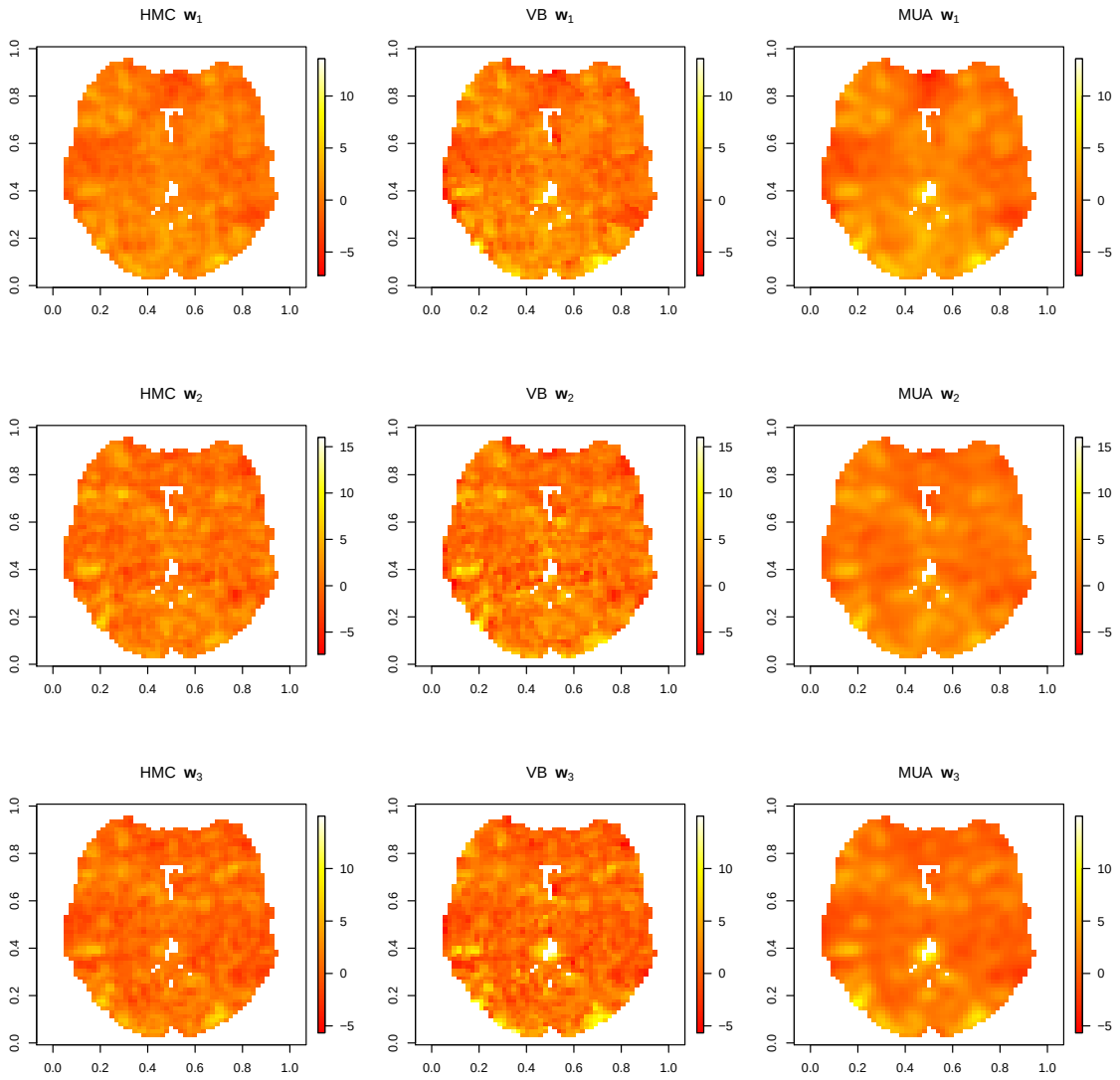


Figure D.13: Image of posterior mean estimate of $\mathbf{w}_1 - \mathbf{w}_3$ from HMC, VB and MUA. These are the estimates from 26^{th} slices on the z-axis. We only provide this slice due to a limited space. The result is similar in other slices.

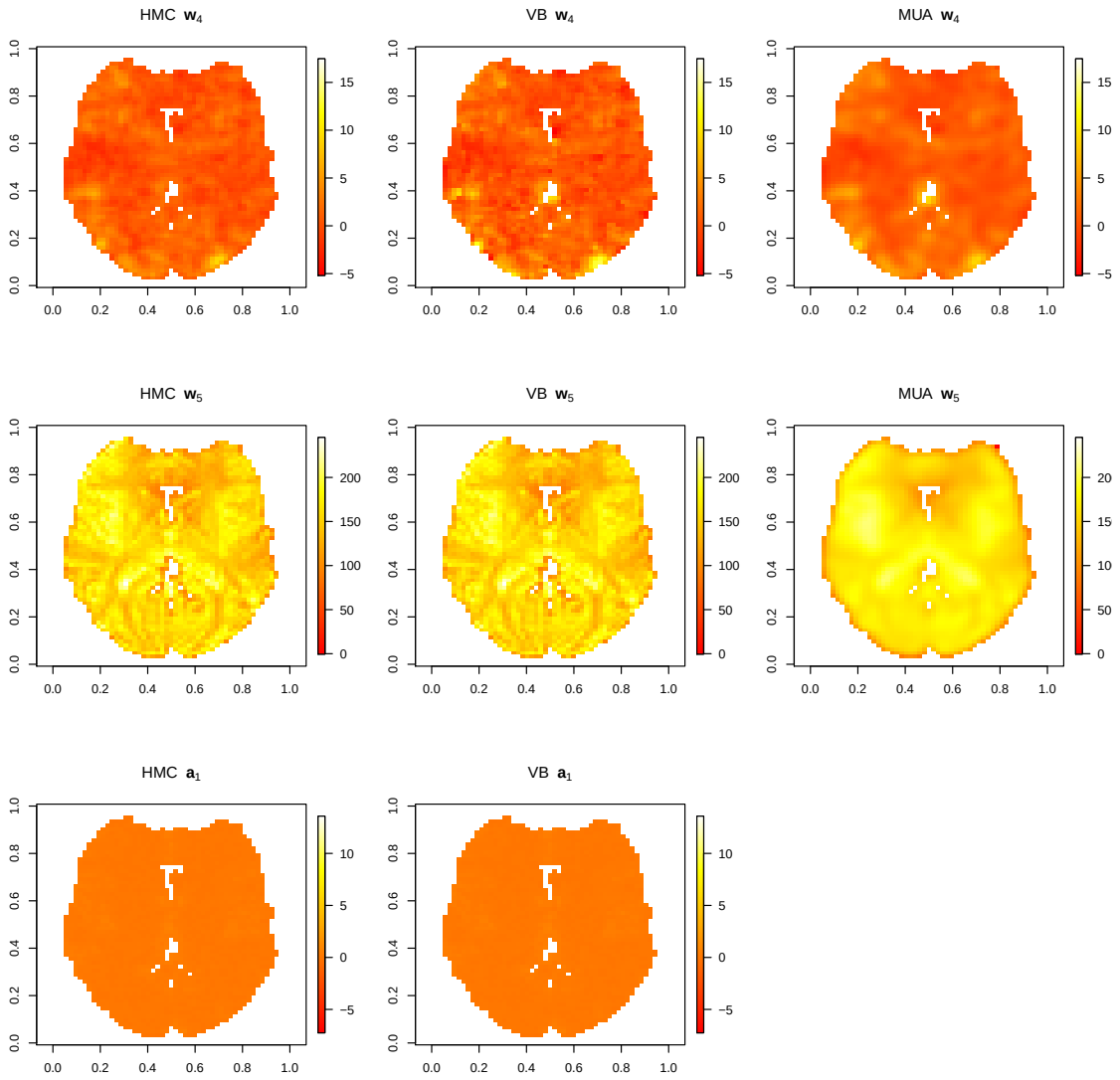


Figure D.14: Image of posterior mean estimate of $\mathbf{w}_4, \mathbf{w}_5, \mathbf{a}_1$ from HMC, VB and MUA. These are the estimates from 26^{th} slices on the z-axis. Because MUA do not provide estimates of auto-regressive coefficients, we omit it here.

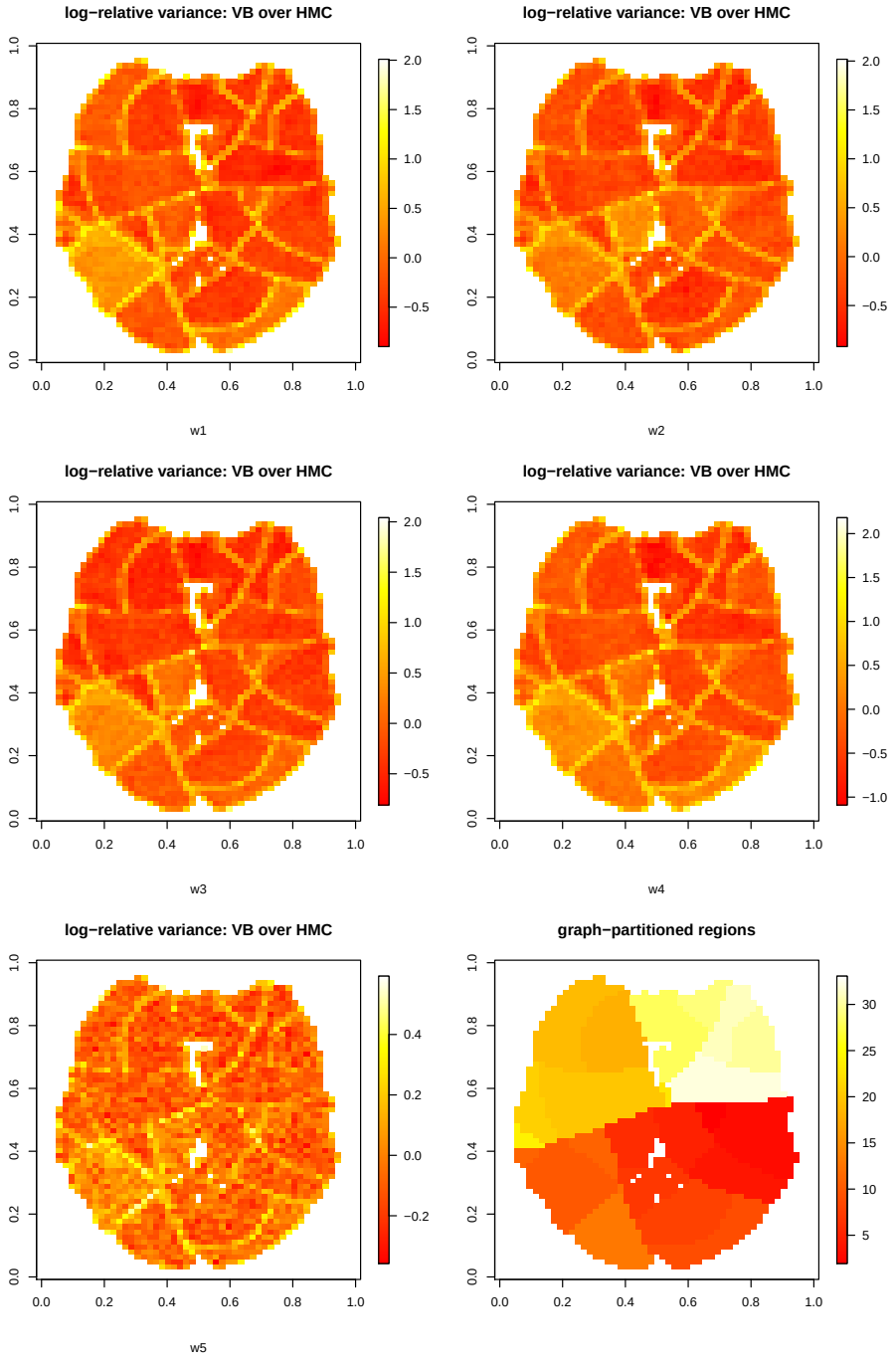


Figure D.15: Log-relative ratio of marginal posterior variance from VB over HMC. The first five image corresponds to $\mathbf{w}_1$ to $\mathbf{w}_5$, the last one is the graph-partitioned regions by SPM VB. This is also the 26^{th} slice.

APPENDIX E

Supplementary figures for Chapter IV

E.1 Simulation One

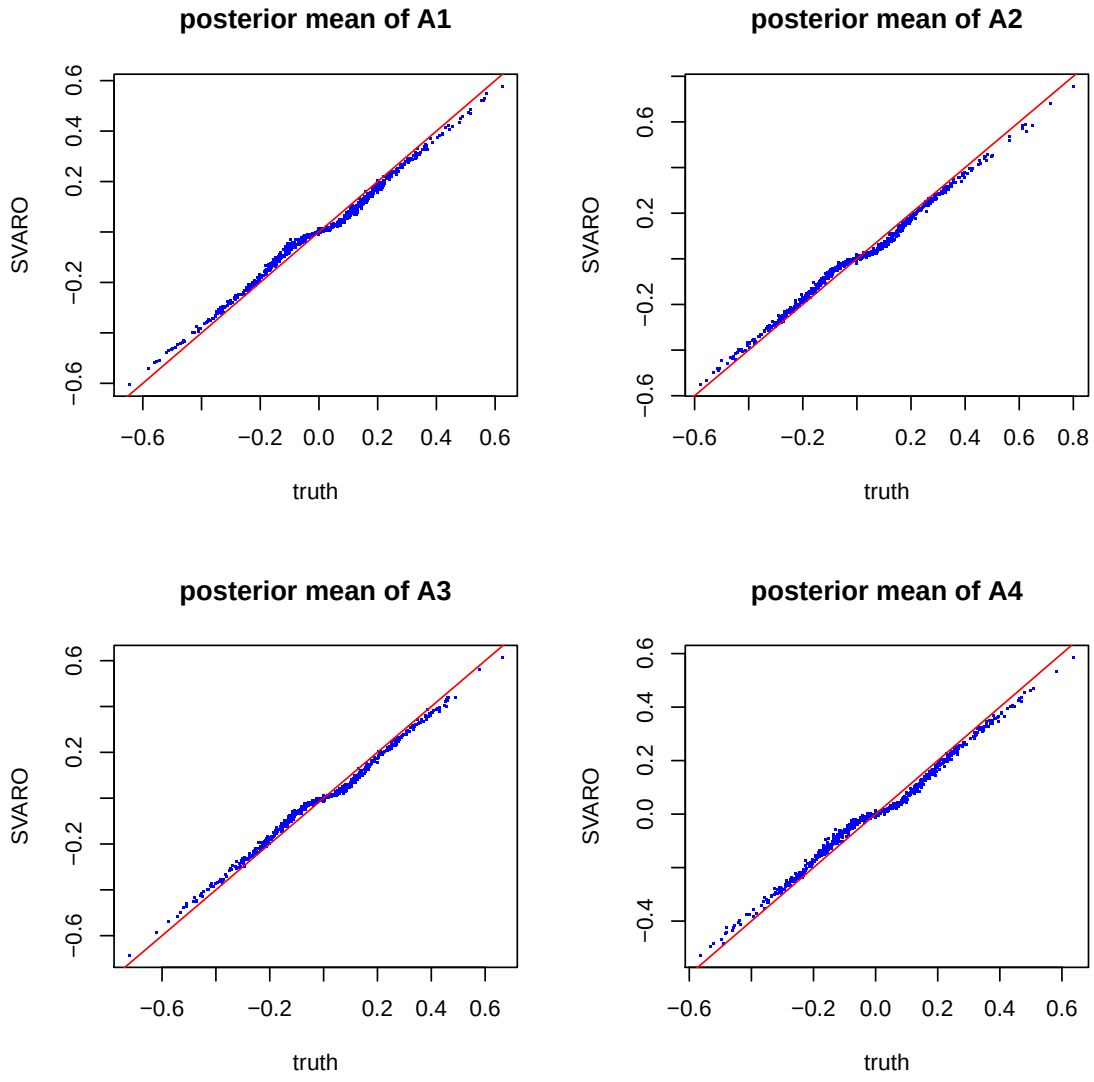


Figure E.1: Scatter plot of posterior mean of the first 4 AR coefficients for SVARO versus the true AR coefficients.

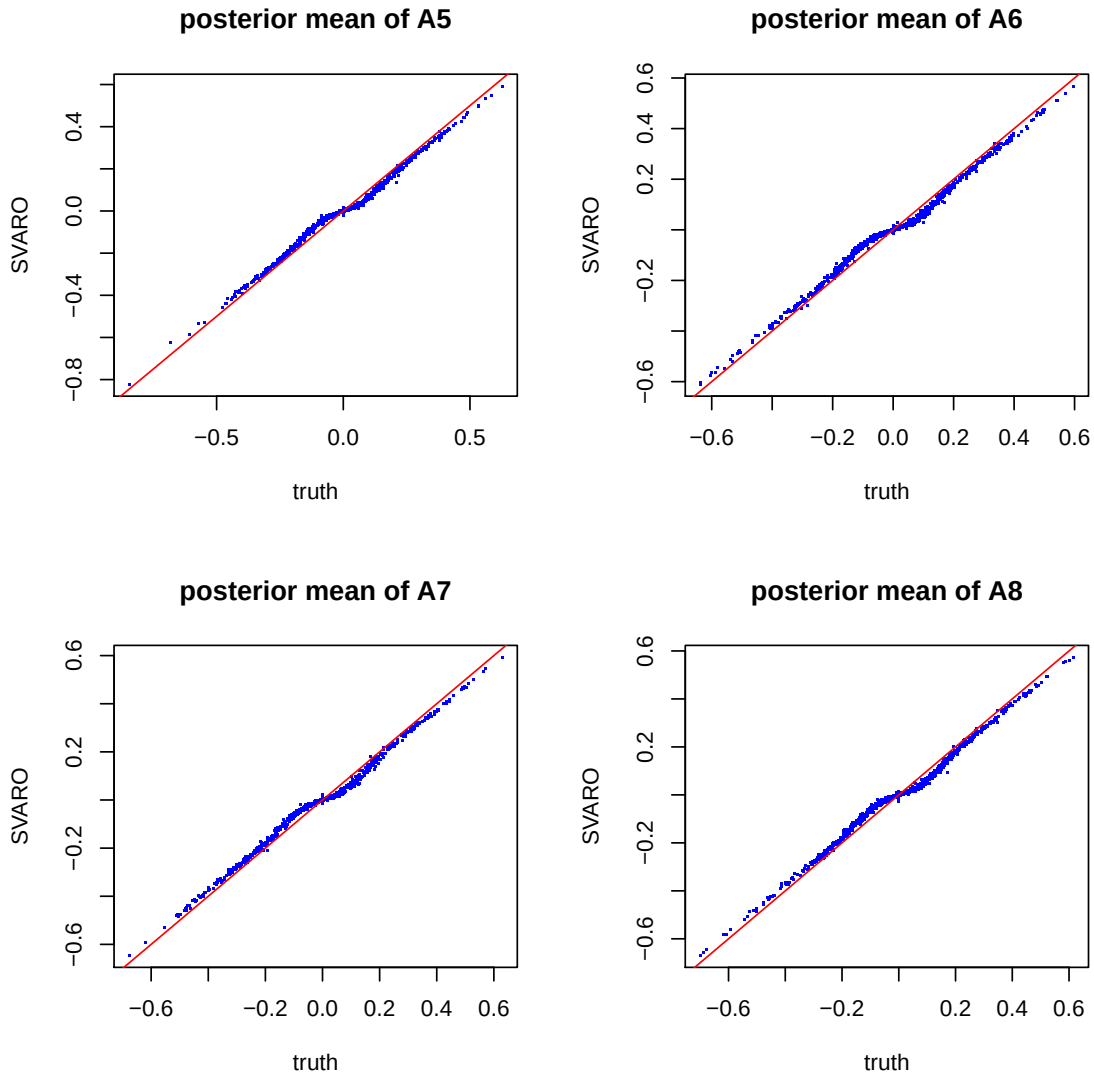


Figure E.2: Scatter plot of posterior mean of the last 4 AR coefficients for SVARO versus the true AR coefficients.

E.2 Simulatin Two

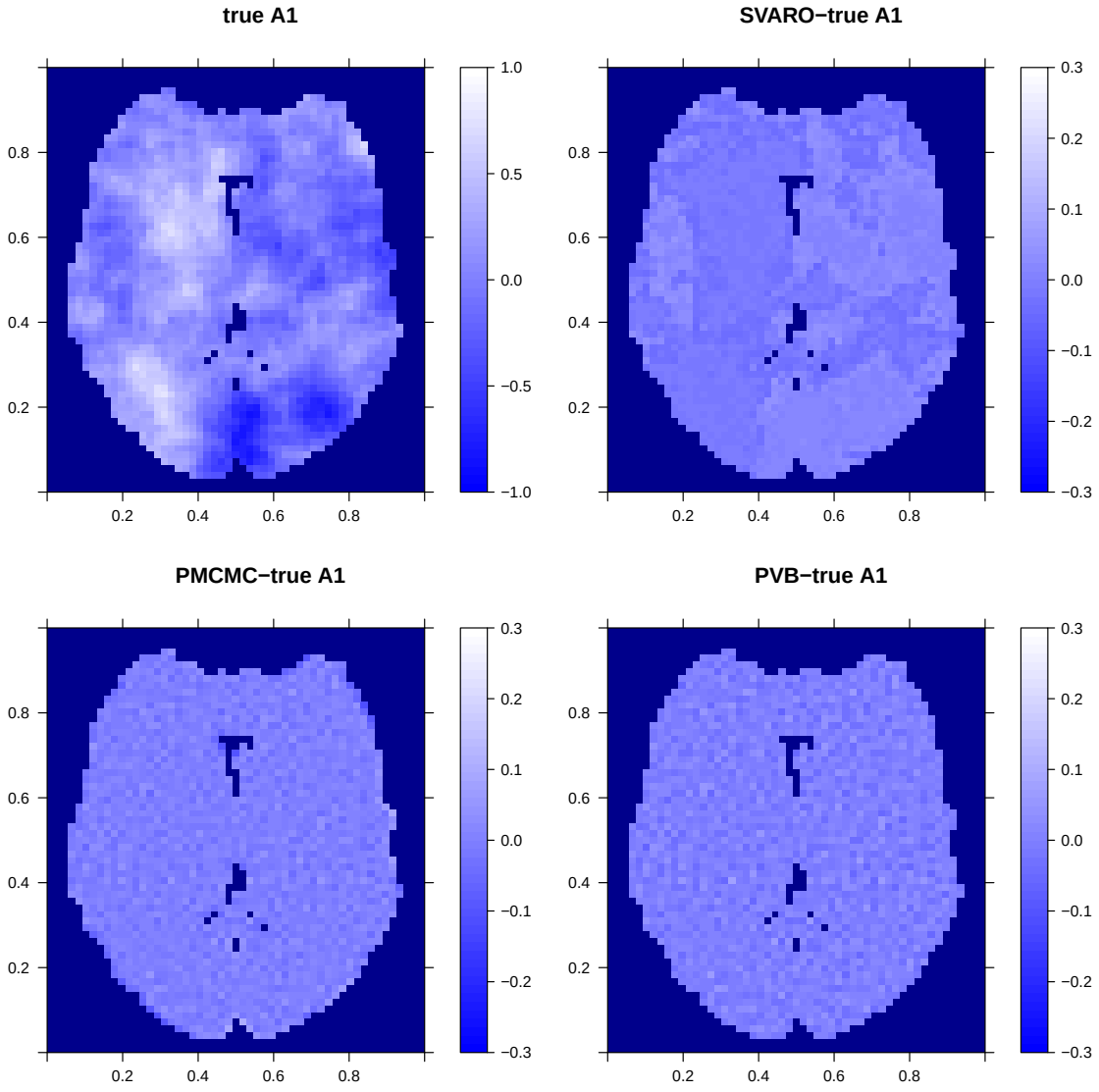


Figure E.3: The top left image (with scale -1 to 1) denotes the true AR coefficients for $1st$ order. The 2nd, 3rd, and 4th image are the corresponding difference between truth and posterior mean of SVARO, PMCMC and PVB respectively. The color scale for the rest of the three images are truncated from -0.3 to 0.3 to remain accordance with the previous simulation. The posterior means are all averaged over 100 replicates. Because the true order is 1, so we omit the figures of other orders of AR coefficients for SVARO.

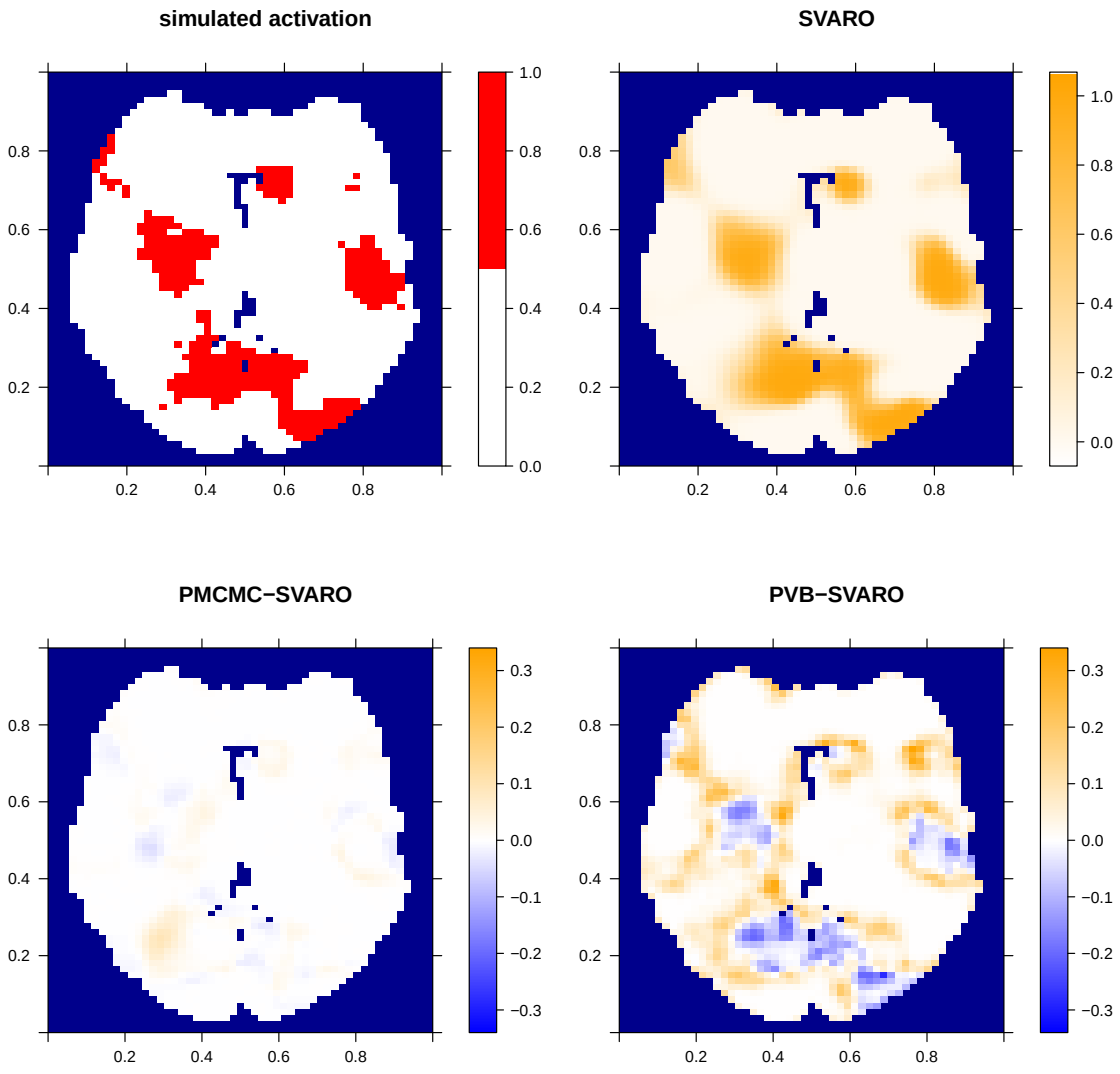


Figure E.4: Topleft depicts the true activation map (red dots denote activation). The remaining panels are posterior probability maps (PPM) of activation obtained using SVARO, (SVARO-PMCMC) and (SVARO-PVB). The latter two reflect the difference of the two alternative approaches relative to SVARO.

BIBLIOGRAPHY

BIBLIOGRAPHY

Adams, R. P., Murray, I., and MacKay, D. J. (2009), "Tractable nonparametric Bayesian inference in Poisson processes with Gaussian process intensities," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ACM, pp. 9–16.

Alder, B. J. and Wainwright, T. (1959), "Studies in molecular dynamics. I. General method," *The Journal of Chemical Physics*, 31, 459–466.

Ashburner, J., Barnes, G., Chen, C., Daunizeau, J., Flandin, G., Friston, K., Kiebel, S., Kilner, J., Litvak, V., Moran, R., et al. (2014), "SPM12 Manual," *Wellcome Trust Centre for Neuroimaging, London (UK)*.

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014), *Hierarchical modeling and analysis for spatial data*, Crc Press.

Basu, S. and Dassios, A. (2002), "A Cox process with log-normal intensity," *Insurance: mathematics and economics*, 31, 297–302.

Bernardo, J., Berger, J., Dawid, A., Smith, A., et al. (1998), "Regression and classification using Gaussian process priors," *Bayesian statistics*, 6, 475.

Besag, J. (1994), "Discussion on the paper by Grenander and Miller," *Journal of the Royal Statistical Society, Series B*, 56, 591–592.

Beskos, A., Pillai, N., Roberts, G., Sanz-Serna, J.-M., Stuart, A., et al. (2013), "Optimal tuning of the hybrid Monte Carlo algorithm," *Bernoulli*, 19, 1501–1534.

Bezener, M., Eberly, L. E., Hughes, J., Jones, G., and Musgrove, D. R. (2016), "Bayesian Spatiotemporal Modeling for Detecting Neuronal Activation via Functional Magnetic Resonance Imaging," *Handbook of Big Data Analytics, Springer Handbooks of Computational Statistics. Springer*.

Bishop, C. M. (2006), "Pattern Recognition," *Machine Learning*.

Blei, D. M., Jordan, M. I., et al. (2006), "Variational inference for Dirichlet process mixtures," *Bayesian analysis*, 1, 121–143.

Bullmore, E., Brammer, M., Williams, S. C., Rabe-Hesketh, S., Janot, N., David, A., Mellers, J., Howard, R., and Sham, P. (1996), "Statistical methods of estimation and inference for functional MR image analysis," *Magnetic Resonance in Medicine*, 35, 261–277.

Bullmore, E., Fadili, J., Maxim, V., Şendur, L., Whitcher, B., Suckling, J., Brammer, M., and Breakspear,

M. (2004), "Wavelets and functional magnetic resonance imaging of the human brain," *Neuroimage*, 23, S234–S249.

Castruccio, S., Ombao, H., and Genton, M. G. (2016), "A multi-resolution spatio-temporal model for brain activation and connectivity in fMRI data," *arXiv preprint arXiv:1602.02435*.

Chang, C. and Glover, G. H. (2010), "Time–frequency dynamics of resting-state brain connectivity measured with fMRI," *Neuroimage*, 50, 81–98.

Chiu, S. N., Stoyan, D., Kendall, W. S., and Mecke, J. (2013), *Stochastic geometry and its applications*, John Wiley & Sons.

Cox, D. and Isham, V. (1980), "Point Processes (Monographs on Applied Probability and Statistics)," .

Daley, D. J. and Vere-Jones, D. (1988), *An introduction to the theory of point processes*, vol. 2, Springer.

Daunizeau, J., Friston, K., and Kiebel, S. (2009), "Variational Bayesian identification and prediction of stochastic nonlinear dynamic causal models," *Physica D: nonlinear phenomena*, 238, 2089–2118.

Della-Maggiore, V., Chau, W., Peres-Neto, P. R., and McIntosh, A. R. (2002), "An empirical comparison of SPM preprocessing parameters to the analysis of fMRI data," *Neuroimage*, 17, 19–28.

Diggle, P. (1985), "A kernel method for smoothing point process data," *Applied Statistics*, 138–147.

Diggle, P. J. and Gratton, R. J. (1984), "Monte Carlo methods of inference for implicit statistical models," *Journal of the Royal Statistical Society. Series B (Methodological)*, 193–227.

Diggle, P. J. et al. (1983), *Statistical analysis of spatial point patterns.*, Academic Press.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987), "Hybrid monte carlo," *Physics letters B*, 195, 216–222.

Fadili, M. and Bullmore, E. (2002), "Wavelet-generalized least squares: a new BLU estimator of linear regression models with 1/f errors," *NeuroImage*, 15, 217–232.

Ferkingstad, E., Rue, H., et al. (2015), "Improving the INLA approach for approximate Bayesian inference for latent Gaussian models," *Electronic Journal of Statistics*, 9, 2706–2731.

Filler, A. G. (2010), "The history, development and impact of computed imaging in neurological diagnosis and neurosurgery: CT, MRI, and DTI," *Internet Journal of Neurosurgery*, 7, 5.

Fishman, G. S. and Yarberry, L. S. (1997), "An implementation of the batch means method," *INFORMS Journal on Computing*, 9, 296–310.

Friston, K., Josephs, O., Zarahn, E., Holmes, A., Rouquette, S., and Poline, J.-B. (2000), "To smooth or not to smooth?: Bias and efficiency in fmri time-series analysis," *NeuroImage*, 12, 196–208.

Friston, K. J., Holmes, A. P., Poline, J., Grasby, P., Williams, S., Frackowiak, R. S., and Turner, R. (1995), "Analysis of fMRI time-series revisited," *Neuroimage*, 2, 45–53.

Friston, K. J., Holmes, A. P., Worsley, K. J., Poline, J.-P., Frith, C. D., and Frackowiak, R. S. (1994), "Statistical parametric maps in functional imaging: a general linear approach," *Human brain mapping*, 2, 189–210.

Geisser, S. and Eddy, W. F. (1979), "A predictive approach to model selection," *Journal of the American Statistical Association*, 74, 153–160.

Gelfand, A. E. and Dey, D. K. (1994), "Bayesian model choice: asymptotics and exact calculations," *Journal of the Royal Statistical Society. Series B (Methodological)*, 501–514.

Gelman, A. and Meng, X.-L. (1998), "Simulating normalizing constants: From importance sampling to bridge sampling to path sampling," *Statistical science*, 163–185.

Gelman, A., Meng, X.-L., and Stern, H. (1996), "Posterior predictive assessment of model fitness via realized discrepancies," *Statistica Sinica*, 6, 733–760.

George, E. I. and McCulloch, R. E. (1993), "Variable selection via Gibbs sampling," *Journal of the American Statistical Association*, 88, 881–889.

Girolami, M. and Calderhead, B. (2011), "Riemann manifold langevin and hamiltonian monte carlo methods," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 123–214.

Gonçalves, F. B. and Gamerman, D. (2015), "Exact Bayesian inference in spatio-temporal Cox processes driven by multivariate Gaussian processes," *arXiv preprint arXiv:1504.06638*.

Gössl, C., Auer, D. P., and Fahrmeir, L. (2001), "Bayesian spatiotemporal inference in functional magnetic resonance imaging," *Biometrics*, 57, 554–562.

Harrison, L., Penny, W. D., and Friston, K. (2003), "Multivariate autoregressive modeling of fMRI time series," *Neuroimage*, 19, 1477–1491.

Harrison, L. M., Penny, W., Daunizeau, J., and Friston, K. J. (2008a), "Diffusion-based spatial priors for functional magnetic resonance images," *Neuroimage*, 41, 408–423.

Harrison, L. M., Penny, W., Flandin, G., Ruff, C. C., Weiskopf, N., and Friston, K. J. (2008b), "Graph-partitioned spatial priors for functional magnetic resonance images," *NeuroImage*, 43, 694–707.

Henson, R., Shallice, T., Gorno-Tempini, M., and Dolan, R. (2002), "Face repetition effects in implicit and explicit memory tests as measured by fMRI," *Cerebral Cortex*, 12, 178–186.

Higdon, D. M. (1998), "Auxiliary variable methods for Markov chain Monte Carlo with applications," *Journal of the American Statistical Association*, 93, 585–595.

Humphreys, K. and Titterington, D. (2000), "Approximate Bayesian inference for simple mixtures," in *COMPSTAT*, Springer, pp. 331–336.

Illian, J., Penttinen, A., Stoyan, H., and Stoyan, D. (2008), *Statistical analysis and modelling of spatial point patterns*, John Wiley & Sons.

Illian, J. B., Møller, J., and Waagepetersen, R. P. (2009), "Hierarchical spatial point process analysis for a plant community with high biodiversity," *Environmental and Ecological Statistics*, 16, 389–405.

Ising, E. (1925), "Beitrag zur theorie des ferromagnetismus," *Zeitschrift für Physik*, 31, 253–258.

Jeong, J., Vannucci, M., and Ko, K. (2013), "A Wavelet-Based Bayesian Approach to Regression Models with Long Memory Errors and Its Application to fMRI Data," *Biometrics*, 69, 184–196.

Johnson, T. D., Liu, Z., Bartsch, A. J., and Nichols, T. E. (2013), "A Bayesian non-parametric Potts model with application to pre-surgical FMRI data," *Statistical methods in medical research*, 22, 364–381.

Johnson, V. E. and Rossell, D. (2012), "Bayesian model selection in high-dimensional settings," *Journal of the American Statistical Association*, 107, 649–660.

Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999), "An introduction to variational methods for graphical models," *Machine learning*, 37, 183–233.

Kang, J., Johnson, T. D., Nichols, T. E., and Wager, T. D. (2011), "Meta analysis of functional neuroimaging data via Bayesian spatial point processes," *Journal of the American Statistical Association*, 106, 124–134.

Kim, S., Smyth, P., and Stern, H. (2010), "A Bayesian mixture approach to modeling spatial activation patterns in multisite fMRI data," *IEEE transactions on medical imaging*, 29, 1260–1274.

Lazar, N. (2008), *The statistical analysis of functional MRI data*, Springer Science & Business Media.

Lee, K.-J., Jones, G. L., Caffo, B. S., and Bassett, S. S. (2014), "Spatial Bayesian variable selection models on functional magnetic resonance imaging time-series data," *Bayesian Analysis (Online)*, 9, 699.

Li, F., Zhang, T., Wang, Q., Gonzalez, M. Z., Maresh, E. L., Coan, J. A., et al. (2015), "Spatial Bayesian variable selection and grouping for high-dimensional scalar-on-image regression," *The Annals of Applied Statistics*, 9, 687–713.

Lindgren, F., Rue, H., and Lindström, J. (2011), "An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73, 423–498.

Lindquist, M. A. et al. (2008), "The statistical analysis of fMRI data," *Statistical Science*, 23, 439–464.

Lloyd, C., Gunter, T., Osborne, M. A., and Roberts, S. J. (2014), "Variational inference for Gaussian process modulated Poisson processes," *arXiv preprint arXiv:1411.0254*.

Locascio, J. J., Jennings, P. J., Moore, C. I., and Corkin, S. (1997), "Time series analysis in the time domain and resampling methods for studies of functional magnetic resonance brain imaging," *Human brain mapping*, 5, 168–193.

MacKay, D. J. (1997), "Ensemble learning for hidden Markov models," Tech. rep., Technical report, Cavendish Laboratory, University of Cambridge.

Makni, S., Ciuciu, P., Idier, J., and Poline, J.-B. (2006), "Joint detection-estimation of brain activity in fMRI using an autoregressive noise model," in *Biomedical Imaging: Nano to Macro, 2006. 3rd IEEE International Symposium on*, IEEE, pp. 1048–1051.

Matérn, B. (1960), "Spatial Variation," *Meddelanden från Statens Skogforskningsinstitut*, 49.

Meyer, F. G. (2003), "Wavelet-based estimation of a semiparametric generalized linear model of fMRI time-series," *IEEE transactions on medical imaging*, 22, 315–322.

Minka, T. P. (2001), "Expectation propagation for approximate Bayesian inference," in *Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence*, Morgan Kaufmann Publishers Inc., pp. 362–369.

Mitchell, T. J. and Beauchamp, J. J. (1988), "Bayesian variable selection in linear regression," *Journal of the American Statistical Association*, 83, 1023–1032.

Møller, J., Syversveen, A. R., and Waagepetersen, R. P. (1998), "Log gaussian cox processes," *Scandinavian journal of statistics*, 25, 451–482.

Møller, J. and Waagepetersen, R. P. (2003), *Statistical inference and simulation for spatial point processes*, CRC Press.

Moran, P. A. (1950), "Notes on continuous stochastic phenomena," *Biometrika*, 37, 17–23.

Murray, I., Adams, R. P., and MacKay, D. J. (2010), "Elliptical slice sampling," *The Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, 9, 541–548.

Murray, I., Ghahramani, Z., and MacKay, D. (2012), "MCMC for doubly-intractable distributions," *arXiv preprint arXiv:1206.6848*.

Musgrove, D. R., Hughes, J., and Eberly, L. E. (2016), "Fast, fully Bayesian spatiotemporal inference for fMRI data," *Biostatistics*, 17, 291–303.

Nathoo, F. (2010), "Joint spatial modeling of recurrent infection and growth with processes under intermittent observation," *Biometrics*, 66, 336–346.

Nathoo, F., Babul, A., Moiseev, A., Virji-Babul, N., and Beg, M. (2014), "A variational Bayes spatiotemporal model for electromagnetic brain mapping," *Biometrics*, 70, 132–143.

Nathoo, F., Lesperance, M., Lawson, A., and Dean, C. (2013), "Comparing variational Bayes with Markov chain Monte Carlo for Bayesian computation in neuroimaging," *Statistical methods in medical research*, 22, 398–423.

Neal, R. (2011), "MCMC using Hamiltonian dynamics," in *Handbook of Markov Chain Monte Carlo*, eds. Brooks, S., Gelman, A., Jones, G. L., and Meng, X.-L., CRC Press, pp. 113–162.

Neal, R. M. (1995), "Bayesian learning for neural networks," Ph.D. thesis, University of Toronto.

— (2003), "Slice sampling," *Annals of statistics*, 705–741.

Neal, R. M. and Hinton, G. E. (1998), "A view of the EM algorithm that justifies incremental, sparse, and other variants," in *Learning in graphical models*, Springer, pp. 355–368.

Nguyen, T. V. and Bonilla, E. V. (2014), "Automated variational inference for Gaussian process models," in *Advances in Neural Information Processing Systems*, pp. 1404–1412.

Ogawa, S., Lee, T.-M., Kay, A. R., and Tank, D. W. (1990), "Brain magnetic resonance imaging with contrast dependent on blood oxygenation," *Proceedings of the National Academy of Sciences*, 87, 9868–9872.

O'Hara, R. B., Sillanpää, M. J., et al. (2009), "A review of Bayesian variable selection methods: what, how and which," *Bayesian analysis*, 4, 85–117.

Opper, M. and Archambeau, C. (2009), "The variational Gaussian approximation revisited," *Neural computation*, 21, 786–792.

Paisley, J., Blei, D., and Jordan, M. (2012), "Variational Bayesian inference with stochastic search," *arXiv preprint arXiv:1206.6430*.

Pascual-Marqui, R. D., Michel, C. M., and Lehmann, D. (1994), "Low resolution electromagnetic tomography: a new method for localizing electrical activity in the brain," *International Journal of psychophysiology*, 18, 49–65.

Penny, W., Flandin, G., and Trujillo-Barreto, N. (2007), "Bayesian comparison of spatially regularised general linear models," *Human brain mapping*, 28, 275–293.

Penny, W., Kiebel, S., and Friston, K. (2003), "Variational Bayesian inference for fMRI time series," *NeuroImage*, 19, 727–741.

Penny, W. D., Trujillo-Barreto, N. J., and Friston, K. J. (2005), "Bayesian fMRI time series analysis with spatial priors," *NeuroImage*, 24, 350–362.

Robert, C. and Casella, G. (2013), *Monte Carlo statistical methods*, Springer Science & Business Media.

Roberts, G. O. and Rosenthal, J. S. (1998), "Optimal scaling of discrete approximations to Langevin diffusions," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60, 255–268.

Roberts, G. O., Rosenthal, J. S., et al. (2001), "Optimal scaling for various Metropolis-Hastings algorithms," *Statistical science*, 16, 351–367.

Rosenblatt, M. et al. (1956), "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, 27, 832–837.

Rue, H. and Held, L. (2005), *Gaussian Markov random fields: theory and applications*, CRC Press.

Rue, H., Martino, S., and Chopin, N. (2009), "Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations," *Journal of the royal statistical society: Series b (statistical methodology)*, 71, 319–392.

Schmidt, D. F. and Makalic, E. (2013), "Estimation of stationary autoregressive models with the Bayesian LASSO," *Journal of Time Series Analysis*, 34, 517–531.

Sengupta, B., Friston, K. J., and Penny, W. D. (2016), "Gradient-based MCMC samplers for dynamic causal modelling," *NeuroImage*, 125, 1107–1118.

Shu, H., Nan, B., and Koeppe, R. (2015), "Multiple testing for neuroimaging via hidden Markov random field," *Biometrics*, 71, 741–750.

Sidén, P., Eklund, A., Bolin, D., and Villani, M. (2017), "Fast Bayesian whole-brain fMRI analysis with spatial 3D priors," *NeuroImage*, 146, 211–225.

Simpson, D., Lindgren, F., and Rue, H. (2012), "Think continuous: Markovian Gaussian models in spatial statistics," *Spatial Statistics*, 1, 16–29.

Skudlarski, P., Constable, R. T., and Gore, J. C. (1999), "ROC analysis of statistical methods used in functional MRI: individual subjects," *Neuroimage*, 9, 311–329.

Smith, M. and Fahrmeir, L. (2007), "Spatial Bayesian variable selection with application to functional magnetic resonance imaging," *Journal of the American Statistical Association*, 102, 417–431.

Stanley, H. E., Stauffer, D., Kertesz, J., and Herrmann, H. J. (1987), "Dynamics of spreading phenomena in two-dimensional Ising models," *Physical review letters*, 59, 2326.

Swendsen, R. H. and Wang, J.-S. (1987), "Nonuniversal critical dynamics in Monte Carlo simulations," *Physical review letters*, 58, 86.

Taylor, B. M. and Diggle, P. J. (2013), "INLA or MCMC? A tutorial and comparative evaluation for spatial prediction in log-Gaussian Cox processes," *Journal of Statistical Computation and Simulation*, ahead-of-print, 1–19.

Wang, C. and Blei, D. M. (2013), "Variational inference in nonconjugate models," *The Journal of Machine Learning Research*, 14, 1005–1031.

— (2015), "A General Method for Robust Bayesian Modeling," *arXiv preprint arXiv:1510.05078*.

Wang, H., Li, G., and Tsai, C.-L. (2007), "Regression coefficient and autoregressive order shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69, 63–78.

Wang, J., Wu, T., Liu, Y., Deng, W., and Oh, H. (2016), "Modeling and performance analysis of dynamic spectrum sharing between DSRC and Wi-Fi systems," *Wireless Communications and Mobile Computing*, 16, 2743–2758.

Wood, A. T. A. and Chan, G. (1994), "Simulation of stationary Gaussian processes in $[0, 1]^d$," *Journal of Computational and Graphical Statistics*, 3, 409–432.

Woolrich, M. W., Behrens, T. E., and Smith, S. M. (2004a), "Constrained linear basis sets for HRF modelling using Variational Bayes," *NeuroImage*, 21, 1748–1761.

Woolrich, M. W., Jenkinson, M., Brady, J. M., and Smith, S. M. (2004b), "Fully Bayesian spatio-temporal modeling of fMRI data," *Medical Imaging, IEEE Transactions on*, 23, 213–231.

Woolrich, M. W., Ripley, B. D., Brady, M., and Smith, S. M. (2001), "Temporal autocorrelation in univariate linear modeling of FMRI data," *Neuroimage*, 14, 1370–1386.

Worsley, K. J. and Friston, K. J. (1995), "Analysis of fMRI time-series revisited—again," *Neuroimage*, 2, 173–181.

Zammit-Mangion, A., Sanguinetti, G., and Kadirkamanathan, V. (2012), "Variational estimation in spatiotemporal systems from continuous and point-process observations," *Signal Processing, IEEE Transactions on*, 60, 3449–3459.

Zarahn, E., Aguirre, G. K., and D'Esposito, M. (1997), "Empirical analyses of BOLD fMRI statistics," *NeuroImage*, 5, 179–197.