UNIVERSITY OF CALIFORNIA
Santa Barbara

Sequential Monte Carlo methods: applications to disease surveillance and fMRI data

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Statistics and Applied Probability

by

Daniel M. Sheinson

Committee in Charge:

Professor Jarad Niemi, Co-chair

Professor Wendy Meiring, Co-chair

Professor John Hsu

Professor Greg Ashby

September 2014

The Dissertation of
Daniel M. Sheinson is approved:

Professor John Hsu

Professor Greg Ashby

Professor Jarad Niemi, Committee Co-chairperson

Professor Wendy Meiring, Committee Co-chairperson

September 2014

Sequential Monte Carlo methods: applications to disease surveillance and fMRI data

by

Daniel M. Sheinson

Acknowledgements

First and foremost, I would like to express my immense gratitude toward my co-advisors, Professor Jarad Niemi and Professor Wendy Meiring. Jarad saw potential in me as a second-year graduate student and has provided me with invaluable professional and academic advice, even after he moved to Iowa to begin his new faculty position. Wendy, while playing an instrumental role in the department as graduate advisor, has always supported me with unwavering dedication. I am deeply grateful for the amount of time that they both have devoted to me, and I owe tremendous thanks to them for their continuous support over the past four years.

I would also like to thank the remaining members serving on my thesis committee, Professor John Hsu and Professor Greg Ashby. John's warmth and kindness played a large part in my decision to come the UC-Santa Barbara, and Greg has been a tremendous help, sharing with me his expertise in the field of functional magnetic resonance imaging. In addition, I would like to thank Professor S. Rao Jammalamadaka, Professor Andrew Carter, Professor Yuedong Wang, and the entire faculty of the Department of Statistics and Applied Probability for their instruction and guidance throughout the development of my academic career.

Finally, I would like to thank my loving family, friends, and fellow graduate students for all of their support and encouragement.

Curriculum Vitæ
Daniel M. Sheinson

Education

2014	University of California, Santa Barbara, CA Doctor of Philosophy in Statistics and Applied Probability Ph.D. emphasis in Quantitative Methods in the Social Sciences
2010	University of California, Santa Barbara, CA Master of Arts in Mathematical Statistics
2009	University of Illinois at Urbana-Champaign, Champaign, IL Bachelor of Science in Statistics Secondary Major in History Minor in Computer Science

Experience

2013-2014	Summer Instructor, Department of Statistics and Applied Probability, University of California, Santa Barbara
2009-2014	Teaching Assistant, Department of Statistics and Applied Probability, University of California, Santa Barbara
2012	Statistics Consultant, Intellectual Ventures Laboratories, Epidemiological Modeling Group, Seattle

Selected Publications

2014	"Comparison of the performance of particle filter algorithms applied to tracking of a disease epidemic", with Jarad Niemi and Wendy Meiring. Mathematical Biosciences 255: 21-32.
2014	"Large Loss Claims: The Market Shift Factor: Justification for a Statistical Solution", with William Novotny in Journal of Advanced Appraisal Studies: 283-302.

Conference Presentations

2014	"Comparison of the performance of particle filter algorithms applied to tracking of a disease epidemic", Joint Statistical Meetings
2013	"Tracking and prediction of a disease epidemic using particle filtering", WNAR Annual Meeting

Awards and Honors

2010	Abraham Wald Memorial Award, UCSB Department of Statistics and Applied Probability

Abstract

Sequential Monte Carlo methods: applications to disease surveillance and fMRI data

Daniel M. Sheinson

We present contributions to epidemic tracking and analysis of fMRI data using sequential Monte Carlo methods within a state-space modeling framework. Using a model for tracking and prediction of a disease outbreak via a syndromic surveillance system, we compare the performance of several particle filtering algorithms in terms of their abilities to efficiently estimate disease states and unknown fixed parameters governing disease transmission. In this context, we demonstrate that basic particle filters may fail due to degeneracy when estimating fixed parameters, and we suggest the use of an algorithm developed by Liu and West (2001), which incorporates a kernel density approximation to the filtered distribution of the fixed parameters to allow for their regeneration. In addition, we show that seemingly uninformative uniform priors on fixed parameters can affect posterior inferences, and we suggest the use of priors bounded only by the support of the parameter. We demonstrate the negative impact of using multinomial resampling and suggest the use of either stratified or residual resampling within the particle filter. We also run a particle MCMC algorithm and show that the performance of the Liu and West (2001) particle filter is competitive with particle MCMC in this particular syndromic surveillance model setting. Finally, the improved performance

of the Liu and West (2001) particle filter enables us to relax prior assumptions on model parameters, yet still provide reasonable estimates for model parameters and disease states.

We also analyze real and simulated fMRI data using a state-space formulation of a regression model with autocorrelated error structure. We demonstrate via simulation that analyzing autocorrelated fMRI data using a model with independent error structure can inflate the false positive rate of concluding significant neural activity, and we compare methods of accounting for autocorrelation in fMRI data by examining ROC curves. In addition, we show that comparing models with different autocorrelated error structures on the basis of the independence of fitted model residuals can produce misleading results. Using data collected from an fMRI experiment featuring an episodic word recognition task, we estimate parameters in dynamic regression models using maximum likelihood and identify clusters of low and high activation in specific brain regions. We compare alternative models for fMRI time series from these brain regions by approximating the marginal likelihood of the data using particle learning. Our results suggest that a regression model with a dynamic intercept is the preferred model for most fMRI time series in the episodic word recognition experiment within the brain regions we considered, while a model with a dynamic slope is preferred for a small percentage of voxels in these brain regions.

Contents

List of Figures

List of Tables

List of Notation and Terminology

capital symbols, e.g. F, G, Φ, Σ	matrices
lowercase symbols, e.g. a, b, β, γ	vectors or scalars
$\mathrm{N}(\mu, \Sigma)$	the normal distribution with mean μ and covariance matrix Σ
$\mathrm{N}_\Omega(\mu, \Sigma)$	the normal distribution truncated onto the set Ω with untruncated mean μ and covariance Σ
$\mathrm{T}(\mu, \Sigma, v)$	the multivariate, nonstandard Student-t distribution with mean μ, scale matrix Σ, and v degrees of freedom
$\mathrm{LN}(\mu, \Sigma)$	the log-normal distribution with mean μ and covariance matrix Σ on the log scale, i.e. $X \sim \mathrm{LN}(\mu, \Sigma) \Leftrightarrow \log X \sim \mathrm{N}(\mu, \Sigma)$
$\mathrm{IG}(a, b)$	the inverse gamma distribution with shape a and rate b, i.e. with pdf given by $p(x\|a, b) = \frac{b^a}{\Gamma(a)} x^{-a-1} \exp(-b/x)$, $x > 0$
$\mathrm{G}(a, b)$	the gamma distribution with shape a and rate b, i.e. with pdf given by $p(x\|a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp(-bx)$, $x > 0$
$\mathrm{Unif}(a, b)$	the continuous uniform distribution on $[a, b]$
$\delta_{x_0}(x)$	the Dirac delta function that places point mass for random vector x at x_0
$. \sim f$	'.' distributed according to f
$. \overset{iid}{\sim} f$	'.' independent and identically distributed according to f
$x \perp y$	x independent of y
$\otimes$	Kronecker product
A'	the transpose of the matrix A

Continued on next page

Continued from previous page

a'	the transpose of the vector a
$A_{(i,j)}$	the i,jth element of the matrix A
$\text{vec}(A)$	column-wise vectorization of the matrix A
$\|A\|$	determinant of the matrix A
I_n	the $n \times n$ identity matrix
y_t	vector of observed data at time t
x_t	unobserved state vector at time t
θ	vector of unknown fixed parameters
$y_{s:t}$	collection of variables $(y_s, y_{s+1}, \ldots, y_{t-1}, y_t)$
$p(y_t\|x_t, \theta)$	observation equation, i.e. conditional likelihood at time t
$p(x_t\|x_{t-1}, \theta)$	state equation, i.e. state transition density from time $t-1$ to t
$p(x_t, \theta\|y_{1:t})$	filtered distribution of current state and fixed parameters at time t
$p(x_{0:t}, \theta\|y_{1:t})$	filtered distribution at time t of state history and fixed parameters
$p(x_s, \theta\|y_{1:t})$, $s < t$	smoothed distribution of state at time s and fixed parameters, conditional on data observed through time t
$p(x_t\|y_{1:t}, \theta)$	filtered distribution of current state at time t conditional on fixed parameters
$p(x_t\|y_{1:t})$	marginal filtered distribution of current state at time t
$p(x_{0:t}\|y_{1:t}, \theta)$	filtered distribution at time t of state history conditional on fixed parameters
$p(x_{0:t}\|y_{1:t})$	marginal filtered distribution of state history at time t
$p(\theta\|y_{1:t}, x_{0:t})$	filtered distribution of fixed parameters conditional on state history throught time t
$p(\theta\|y_{1:t})$	marginal filtered distribution of fixed of fixed parameters at time t
$p(y_{1:t})$	marginal likelihood of the data observed through time t
$p(y_t\|y_{1:t-1})$	one-step ahead predictive density of data at time t given data through time $t-1$
$p(x_0, \theta)$	prior distribution of the initial state and fixed parameters

Continued on next page

Continued from previous page

$p(y_{t+1}\|x_t, \theta)$	conditional predictive distribution of data at time $t+1$ given current state at time t and fixed parameters
$p(x_{t+1}\|y_{t+1}, x_t, \theta)$	filtered distribution of state at time $t+1$ conditional on current state at time t and fixed parameters
$p(x\|\ldots)$	full conditional distribution for any vector x

Chapter 1

Introduction

Time series data, or data consisting of measurements collected sequentially over time, are common in many fields including the social, physical, technological, and biological sciences. Finding innovative ways to analyze and interpret this kind of data has been crucial for advancing these fields, including weather tracking (Dixon and Wiener; 1993), communication signal processing (Gardner; 1994), and social media networks (Smith et al.; 2009). Analysis of sequential data pose several challenges to the researcher. Firstly, the data often exhibit nonlinear behavior. Secondly, the data are typically autocorrelated, meaning that there tends to be a relationship among data points based on their proximity in time. These issues are problematic for analysis using traditional statistical methods that require linear structure and independent observations.

State-space models provide a general framework that is convenient for modeling nonlinear and autocorrelated data by describing a process in terms of a latent, dynamic state. In these models, observations are regarded as conditionally independent given the underlying state of the system, and the state evolves over time according to a linear or nonlinear process. Typically, the general form of these models may be known, but each model will contain unknown fixed parameters that are specific to the application area. In this thesis, we employ state-space models with unknown fixed parameters to study time series data from two specific areas: disease surveillance and functional magnetic resonance imaging (fMRI).

State-space models are frequently used for disease outbreaks to simultaneously model the underlying disease dynamics and the observation process (Martínez-Beneito et al.; 2008; Merl et al.; 2009b; Ludkovski and Niemi; 2010; Skvortsov and Ristic; 2012; Unkel et al.; 2012; Sheinson et al.; 2014). Together with syndromic surveillance systems (Henning; 2004; Wagner et al.; 2006; Wilson et al.; 2006; Hakenewerth et al.; 2009; Ginsberg et al.; 2009), these models are used to identify emerging disease outbreaks (Neill et al.; 2006), estimate their severity (Merl et al.; 2009b), and predict their duration Ludkovski and Niemi (2010). Data from fMRI experiments are used for the purpose of mapping neural activation in the brain (Ashby; 2011; Kiebel and Holmes; 2007; Poldrack et al.; 2011). These data are typically analyzed using a linear regression model that correlates the observed data with the expected brain response to the experimental stimulus (Friston et

al.; 1991, 1995b). We reformulate this regression model within the state-space framework to model autocorrelation in the data in terms of a dynamic state.

In statistical applications where prior knowledge or beliefs about unknown quantities are available, the Bayesian framework is often convenient for performing statistical analysis. Bayesian inference is conducted through the posterior distribution of any unknown quantities, obtained by updating prior information using observed data. However, the calculation of the posterior distribution in state-space models frequently involves complicated integrals without explicit analytical forms. The most common approach to approximate these posterior distributions is Markov chain Monte Carlo (MCMC) (Gelfand and Smith; 1990). In a sequential context, e.g. syndromic surveillance, MCMC is inefficient due to the increase in computational cost incurred by the need to rerun the entire MCMC as each new observation arrives. Sequential Monte Carlo (SMC) - or particle filtering - methods enable on-line inference by updating the estimate of the posterior distribution as new data become available sequentially in time. Furthermore, SMC methods can be flexible, general, easy to implement, and amenable to parallel computing. For a general introduction, please see Doucet et al. (2001) and Cappé et al. (2007).

In this thesis, after reviewing the SMC, MCMC, and PMCMC algorithms and application areas, we analyze data from disease surveillance and fMRI using both MCMC and SMC algorithms. We compare several particle filtering algorithms in

terms of how efficiently they estimate latent, unobserved states and fixed parameters in a state-space model for tracking a disease epidemic similar to one used by Skvortsov and Ristic (2012). We find that an algorithm developed by Liu and West (2001), which regenerates fixed parameter values through the use of a kernel density approximation, outperforms algorithms that incorporate fixed parameters into the state process with degenerate evolutions. We also find, under this particular model, that the Liu and West (2001) algorithm is competitive with particle MCMC (Andrieu et al.; 2010).

Then, we discuss how SMC methods can be used to compare alternative models through approximation of the marginal likelihood of the data. We compare the performance of more recently developed particle filters in terms of how efficiently they identify a true model out of a set of candidate models using data simulated from the true model. We find that a particle learning algorithm (Carvalho et al.; 2010) outperforms both the resample-move particle filter (Gilks and Berzuini; 2001) and the Liu and West (2001) algorithm when applied to data simulated from a local level dynamic linear model (Section 4.3.1 Petris et al.; 2009).

Next, we provide an overview of data generated from fMRI experiments and describe the most common strategies for data analysis used in this field. We underscore the negative impact of analyzing fMRI data without properly accounting for autocorrelation present in the data, and we explore possible models for this autocorrelation. Initially, we estimate unknown parameters in these models using

maximum likelihood, and we compare several candidate models against one another through examination of statistical criteria, namely AIC, AIC corrected for bias, and BIC. We also use simulated fMRI time series to compare candidate models in terms of their false positive and true positive rates of concluding significant brain activation.

Lastly, we implement a particle learning algorithm for estimating latent states and fixed parameters in state-space models for fMRI data. We also estimate the marginal likelihood of the data and compare relative posterior probabilities among several models. Specifically, we consider the likelihood of a regression model with a dynamic slope being suitable for fMRI data from an episodic word recognition experiment, with the notion that a changing slope component could model changes in focus or learning on the part of the subject in the fMRI scanner. Using simulated data, we explore parameter settings under which we can correctly identify the true data-generating model amongst several candidate models for fMRI time series. We compare these models using real fMRI data from a word recognition experiment. Our results suggest that a dynamic slope model may be suitable only for a small percentage of fMRI time series from this specific experiment, but that larger models that incorporate a dynamic slope as well as other components to account for autocorrelation in fMRI data may be appropriate.

This thesis is organized as follows. Chapter 2 contains descriptions of state-space models in general as well as the specific models we use to analyze syndromic

surveillance and fMRI data. Chapter 3 describes MCMC and SMC methods for making inference on latent unobserved states and unknown fixed parameters in these models. Chapter 4 describes an analysis comparing several particle filtering strategies using simulated syndromic data from an influenza-like epidemic outbreak. In Chapter 5, we describe a model comparison strategy using SMC methods and compare several particle filters in terms of their ability to accurately compare first-order dynamic linear models with varying signal-to-noise ratios. In Chapter 6, we investigate current methods for effectively modeling autocorrelated fMRI time series and use an SMC model comparison strategy for assessing the suitability of dynamic regression models for fMRI data. Chapter 7 discusses future directions. The material in Chapter 4 and some of the methodology discussed in 3 are taken from Sheinson et al. (2014).

Chapter 2

Models

In this chapter, we describe the specific models that we use for analyzing fMRI data and tracking a disease epidemic. We also describe models that we simulate data from in order to compare the performance of different particle filtering algorithms. All of these models fall into a general class of models called *state-space models*. In Section 2.1, we describe state-space models in general. In Section 2.2, we describe the model we consider for tracking a disease epidemic. In Section 2.3, we describe a subclass of state-space models called *dynamic linear models* that we use to model fMRI data. In Section 2.4, we describe sequential estimation of states and unknown fixed parameters in state-space models in general and give analytic solutions for special cases.

2.1 State-space models

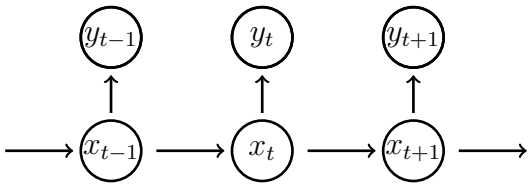
State-space models are a general class of statistical models used for analysis of dynamic data. They are constructed using an observation equation, $y_t \sim p_{y,t}(y_t|x_t, \theta)$, and a state evolution equation, $x_t \sim p_{x,t}(x_t|x_{t-1}, \theta)$, where y_t is the observed response, x_t is a latent, dynamic state, the subscript t is a time index, and θ is an unknown fixed parameter, all of which could be vectors. The y_t's are assumed independent given x_t and θ, and x_t is assumed independent of all states prior to time $t-1$ and all data prior to time t given x_{t-1} and θ. That is, $(y_t \perp y_{1:t-1}, y_{t+1:T})|x_t, \theta$ and $(x_t \perp x_{0:t-2}, y_{1:t-1})|x_{t-1}, \theta$ (see Figure 2.1). The distributions $p_{y,t}$ and $p_{x,t}$ are assumed known conditional on the values of θ and x_t in the observation equation and conditional on θ and x_{t-1} in the evolution equation, respectively. Depending on whether the observations and the states are continuous or discrete, the distributions themselves may be continuous or discrete. The distributions are typically assumed to only vary with x_t and θ, and therefore the t subscript is dropped. For simplicity, we also drop the x and y subscripts and instead let the arguments make clear which distribution we are referring to. Thus, the general state-space model is

$$y_t \sim p(y_t|x_t, \theta) \tag{2.1}$$

$$x_t \sim p(x_t|x_{t-1}, \theta). \tag{2.2}$$

A fully specified Bayesian model is obtained by also specifying the prior $p(x_0, \theta)$.

Figure 2.1: Dependence structure of state-space models



Equations (2.1) and (2.2) describe a very general class of models, including non-Markovian structures and models where the dimension of x_t does not remain constant with respect to t. For instance, we could describe a process where x_t depends on the entire history of states up to t by letting $x_{t-1} = (x_1^*, x_2^*, \ldots, x_{t-1}^*)'$ and defining $x_t = (x_{t-1}, x_t^*)'$, where x_t^* is the new state generated at time t. In addition, the form of equations (2.1) and (2.2) could be linear or nonlinear with respect to x_t or θ. For example, in Section 2.2, we describe a state-space model of a disease outbreak that is nonlinear in the observation equation with respect to x_t and nonlinear in the state equation with respect to θ. In Chapter 4, we compare the performance of several particle filtering algorithms using data simulated from this model.

Special cases of state-space models include finite state-space hidden Markov models (Cappé et al.; 2005), where x_t has discrete support, and dynamic linear models (DLMs) (West and Harrison; 1997; Petris et al.; 2009), where each distribution in (2.1) and (2.2) is Gaussian with mean a linear function of the states and variance that does not depend on the mean. A simple form of a DLM, known as a first-order DLM or local-level model, is described in Section 2.3.1 and used in

Chapter 5 to compare several particle filtering algorithms in terms of their ability to estimate $p(y_{1:t})$, the *marginal likelihood* of the data. In Chapter 6, DLM representations of regression models with autocorrelated errors are used to analyze fMRI data. We describe DLMs in more detail in Section 2.3.

2.2 Model for tracking an epidemic

In this section, we describe a state-space model of an epidemic in which we track the proportion of the population that is susceptible (s_t), infectious (i_t), and recovered (r_t), i.e. no longer able to be infected, at time t. Mathematically, s_t, i_t, and r_t are all nonnegative and $s_t + i_t + r_t = 1$ for all t. When monitoring an epidemic, the true s_t, i_t and r_t are unknown and regarded as hidden states of the model, and the observed data are gathered via syndromic surveillance. In our state-space model of an epidemic, the observation equation specifies how the observed data depend on the state of the epidemic and the state equation describes how the epidemic evolves over time.

2.2.1 SIR model

First, we describe the state equation. Let $x_t = (s_t, i_t)'$ denote the state of the epidemic at time t (by definition $r_t = 1 - s_t - i_t$ and hence r_t is not needed in the state vector). Initially, we consider a compartmental model - or SIR model - of disease transmission that is governed by three parameters:

- β, the contact rate for the spread of illness,
- γ, the recovery time from infection (i.e. the reciprocal of the average infectious period), and
- ν, the mixing intensity of the population.

β, γ, and ν are each restricted to be nonnegative. Define $\theta = (\beta, \gamma, \nu)'$ to be the vector of unknown parameters in our model and let P be the size of the population. Then, we describe the evolution of the epidemic from time t to $t+1$ by

$$x_{t+1} \,|x_t, \theta \sim \mathrm{N}_\Omega \left(f(x_t, \theta), Q(\theta) \right), \tag{2.3}$$

where

$$f(x_t, \theta) = \begin{pmatrix} s_t - \beta i_t s_t^\nu \\ \\ i_t + \beta i_t s_t^\nu - \gamma i_t \end{pmatrix} \qquad Q(\theta) = \frac{\beta}{P^2} \begin{pmatrix} 1 & -1 \\ \\ -1 & 1 + \gamma/\beta \end{pmatrix}$$

and $\Omega = \{(s_t, i_t) : s_t \geq 0, i_t \geq 0, s_t + i_t \leq 1\}$.

In equation (2.3), $Q(\theta)$ is determined by calculating the variances and covariance of s_{t+1} and i_{t+1} in the discrete time approximation of a modified SIR model with stochastic fluctuations (van Herwaarden and Grasman; 1995; Dangerfield et al.; 2009; Anderson et al.; 2004), given by

$$s_{t+1} = s_t - \beta i_t s_t^\nu + \epsilon_\beta \tag{2.4}$$

$$i_{t+1} = i_t + \beta i_t s_t^\nu - \gamma i_t - \epsilon_\beta + \epsilon_\gamma, \tag{2.5}$$

where ϵ_β and ϵ_γ are random components with $\epsilon_\beta \sim \mathrm{N}(0, \sqrt{\beta}/P)$ and $\epsilon_\gamma \sim \mathrm{N}(0, \sqrt{\gamma}/P)$. The variances of these terms come from a scaling law for stochastic fluctuations in a dynamical system generated by random contacts among the population (Ovaskainen and Meerson; 2010; van Herwaarden and Grasman; 1995; Dangerfield et al.; 2009; Skvortsov and Ristic; 2012).

The *basic reproductive number*, $R_0 = \beta/\gamma$, is the average number of people infected by one sick person in a population where everyone is susceptible (Heffernan et al.; 2005). If $R_0 > 1$, then an epidemic can occur. In many cases, prior information about R_0 for a specific type of infection is more readily available than prior knowledge about β or γ individually.

The mixing parameter ν describes the heterogeneity of social interactions within the population, where $\nu = 1$ corresponds to a population with homogenous mixing, i.e. an infectious person is equally likely to infect any susceptible, and $\nu = 0$ corresponds to a population with no social interaction. Values of $\nu > 1$ represent populations with heterogenous mixing, i.e. an individual is more likely to interact with some people more than others, leading to less severe epidemics than those that would occur in homogenous populations for a fixed R_0 (Stroud et al.; 2006; Novozhilov; 2008).

2.2.2 Syndromic surveillance data

The observed data from syndromic surveillance are positive real numbers related to counts of emergency room visits, prescription sales, or calls to a hotline, for example, and we can observe data from these different streams/sources asynchronously in time. That is, at any time t, we can observe data from any subset of the streams (or possibly none of them). Let $y_{l,t} > 0$ represent data coming from stream l at time t, where $l = 1, 2, \ldots, L$ and $t = 1, 2, \ldots, T$. We model the log of the observations by

$$\log y_{l,t} \sim \mathrm{N}\left(b_l i_t^{\varsigma_l} + \eta_l, \sigma_l^2\right), \tag{2.6}$$

where b_l, ς_l, and σ_l are nonnegative constants (Skvortsov and Ristic; 2012) and η_l is a real number that determines the baseline level of incoming syndromic data from stream l.

The form of the mean of $\log y_{l,t}$ in equation (2.6) is derived from a simplification of the power-law relationship, described in Skvortsov and Ristic (2012) and Ginsberg et al. (2009), between syndromic observations and the proportion of the population that is infectious, where b_l is a multiplicative constant that depends on the syndromic data source, ς_l is the power-law exponent, and σ_l is the standard deviation term that determines the magnitude of random fluctuations in the syndromic observations from stream l. In Chapter 4, we first consider the case where b_l, ς_l, σ_l, and η_l are assumed known, as in (Skvortsov and Ristic; 2012), but then relax that assumption in an extended analysis.

Having formulated the data-generating model, we define $y_t = (y_{1,t}, \ldots, y_{L,t})'$ and specify $p(y_t|x_t, \theta)$, i.e. the likelihood of an observation y_t given x_t and θ, according to $\text{LN}(\mu_t, \Sigma_t)$, where μ_t is an L-length vector with element l equal to $b_l i_t^{\varsigma_l} + \eta_l$ and Σ_t is an $L \times L$ diagonal matrix with the l^{th} diagonal equal to σ_l^2. Elements of y_t may be missing, in which case the dimensions of y_t, μ_t, and Σ_t shrink by the number of missing elements. If all elements of y_t are empty (i.e. if no syndromic data are observed at time t), then $p(x_t, \theta|y_{1:t}) = p(x_{t-1}, \theta|y_{1:t-1})$.

Lastly, we specify the full Bayesian model through $p(x_0, \theta)$, the joint prior distribution of the initial state of the epidemic and the fixed parameters. We use a prior of the form $p(x_0, \theta) = p(\theta)p(s_0, i_0)$, where $p(s_0, i_0)$ is specified according to

$$i_0 \sim \text{N}_{[0,1]}(0.002, 0.0005^2) \quad s_0 = 1 - i_0. \tag{2.7}$$

In Chapter 4 we explore the sensitivity of estimation using particle filtering to different choices for $p(\theta)$.

2.3 Dynamic linear models (DLMs)

In this section, we describe DLMs in general and detail specific DLMs analyzed in Chapters 5 and 6. The general form of a DLM is represented as a state space model with observation and state equations given by

$$y_t = F_t x_t + v_t \tag{2.8}$$

$$x_t = G_t x_{t-1} + w_t. \tag{2.9}$$

Here, y_t is a $q \times 1$ observation vector, x_t is a $p \times 1$ state vector, and v_t and w_t are independent and identically distributed (iid) Gaussian random vectors with mean equal to the zero vector (of length q for v_t and length p for w_t) and covariance matrices V_t $(q \times q)$ and W_t $(p \times p)$, respectively. We also assume v_t and $w_{t'}$ independent for all t and t'. F_t is a $q \times p$ matrix that defines the linear dependence between y_t and x_t in the observation equation. Similarly, G_t is a $p \times p$ matrix that defines the linear dependence of x_t on x_{t-1} in the state equation. Lastly, we specify the full Bayesian DLM by defining the distribution of the prior state according to $x_0 \sim \mathrm{N}(m_0, C_0)$, where m_0 is a $p \times 1$ vector and C_0 is a $p \times p$ matrix. The matrices V_t, W_t, F_t, and G_t are allowed to vary with time, and any or all of V_t, W_t, F_t, G_t, and C_0 could possibly contain unknown parameters.

All DLMs discussed in chapters 5 and 6 assume univariate observations, i.e. vector y_t has length $q = 1$. In addition, we assume G_t, V_t, and W_t are time invariant, and so the subscript t is omitted. Some DLMs we consider have time-invariant F_t, e.g. the local level DLM featured in Chapter 5 and the dynamic intercept model discussed in Chapter 6, while others such as the dynamic slope model featured in Chapter 6 incorporate time-varying F_t. Lastly, all DLMs we consider assume F_t is known, while G, V, and W may contain unknown parameters. We provide an overview of these special cases of DLMs in sections 2.3.1 through 2.3.3.

2.3.1 First-order DLM with common variance factor

The first-order DLM – or local level model – for univariate y_t and x_t is specified by setting $F_t = G = 1$ for all t. Note that, in this case, $q = p = 1$ and both V and W are 1×1 matrices. In addition, here we assume that the observation and state variance share an unknown common variance factor, θ, and that the *signal-to-noise ratio*, defined as $\lambda = W/V$, is known. Specifically, we have the following model:

$$y_t \sim \mathrm{N}(x_t, \theta) \tag{2.10}$$

$$x_t \sim \mathrm{N}(x_{t-1}, \theta\lambda) \tag{2.11}$$

with prior distribution $p(x_0, \theta)$ specified by

$$x_0|\theta \sim \mathrm{N}(0, \theta) \quad \theta \sim \mathrm{IG}(a_0, b_0), \tag{2.12}$$

where the hyperparameters a_0 and b_0 are known.

2.3.2 Regression with ARMA errors

A DLM is convenient for representing a linear regression model with autocorrelated errors. To do so, we introduce known covariates and unknown regression coefficients into the model, i.e.

$$y_t = U_t\beta + F_t x_t + v_t \tag{2.13}$$

$$x_t = Gx_{t-1} + w_t, \tag{2.14}$$

where U_t is a known $q \times d$ matrix and β is an unknown $d \times 1$ vector. Alternatively, a regression model could be specified without introducing U_t and β by instead incorporating U_t inside of F_t and including β as part of x_t. However, we write the model as in equations (2.13) and (2.14) to make the separation of fixed regression coefficients and the dynamic state explicit.

In Chapter 6, we consider regression models for univariate y_t with autoregressive-moving average (ARMA) error structure, i.e. models of the form given in equations (2.13) and (2.14) with $q = 1$ and x_t following a zero-mean ARMA(P, Q) stochastic process (Shumway and Stoffer; 2006), where P and Q are the orders of the autoregressive (AR) and moving average (MA) components, respectively. In these models, $x_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,m})'$ is an m-dimensional vector with $m = \max(P, Q+1)$, F_t is a time-invariant $1 \times m$ vector with first element equal to 1 and the rest 0, $v_t \sim \delta_0(v_t)$ where $\delta_a(x)$ is the Dirac delta function that places point mass for random vector x at a (i.e $v_t = 0$ for all t), G is an $m \times m$ matrix that takes the form

$$G = \begin{pmatrix} \phi_1 & \vdots & & & \\ \phi_2 & \vdots & & & \\ \phi_3 & \vdots & & I_{m-1} & \\ \vdots & \vdots & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_m & \vdots & 0 & \cdots & 0 \end{pmatrix},$$

and $W = \sigma^2 ee'$ with $e = (1, \gamma_1, \ldots, \gamma_{m-1})'$. We let $\theta = (\beta', \phi', \gamma', \sigma^2)'$ represent the unknown parameters of the model, where $\phi = (\phi_1, \phi_2, \ldots, \phi_P)'$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_Q)'$ are the coefficients of the AR and MA components, respectively, and σ^2 is the unknown variance of the white noise shocks in the ARMA process. We adopt the convention that $\phi_s = 0$ for $s > P$ and $\gamma_r = 0$ for $r > Q$. Multiplying out the state equation and successively back-substituting the components of x_t (Section 3.2.5, Petris et al.; 2009) yields the more familiar form of a regression model with ARMA errors, given by

$$y_t = U_t\beta + \phi_1 x_{t-1,1} + \phi_2 x_{t-2,1} + \cdots + \phi_P x_{t-P,1} + \epsilon_t + \gamma_1\epsilon_{t-1} + \gamma_2\epsilon_{t-2} + \cdots + \gamma_Q\epsilon_{t-Q} \tag{2.15}$$

for $t \geq m$. Here, $\epsilon_j \overset{iid}{\sim} \mathrm{N}(0, \sigma^2)$ for $j \geq 0$. Note that only the first element of the state vector at each of times $\{t-1, \ldots, t-P\}$ plays a role in equation (2.15) for $t \geq m$.

It is often desired that constraints be imposed on ϕ and γ such that the ARMA process is stationary and invertible. Stationarity ensures that the long term behavior of x_t is predictable and invertibility ensures that current and future states do not depend on the distant past (Shumway and Stoffer; 2006). We require that roots of the AR polynomial

$$\phi(z) = 1 - \phi_1 z - \phi_2 z^2 - \cdots - \phi_P z^P \tag{2.16}$$

lie outside of the unit circle in the complex plane to ensure stationarity. To ensure invertibility, we impose the same constraint on the MA polynomial, given by

$$\gamma(z) = 1 + \gamma_1 z + \gamma_2 z^2 + \cdots + \gamma_Q z^Q. \tag{2.17}$$

In Chapter 6, we impose these constraints when fitting regression models with ARMA errors to fMRI data using maximum likelihood estimation.

In a Bayesian context, these models are completed by specifying the prior distribution $p(x_0, \theta)$, where the initial state, x_0, consists of the presample errors. It is often desired that x_0 come from the stationary distribution of the ARMA process, given by

$$x_0|\theta \sim \mathrm{N}(0, \sigma^2\Omega), \tag{2.18}$$

where $\mathrm{vec}(\Omega) = (I_{m^2} - G \otimes G)^{-1}\mathrm{vec}(ee')$ and ϕ is restricted to the region of stationarity. For a Bayesian treatment of unknown parameters in regression models with stationary and invertible ARMA errors, including results (2.18), we refer the reader to Chib and Greenberg (1994).

2.3.3 Dynamic regression

The regression model with ARMA errors described in Section 2.3.2 is represented as a DLM by setting the observation error v_t in equation (2.13) equal to 0 for all t and completely specifying the error structure through the state equation. By instead letting v_t be random, we can add an additional layer of variance to the

model that represents observation or measurement noise. Furthermore, we can introduce additional structure into the errors through F_t.

Example 1: Dynamic intercept model

For example, consider a simple linear regression model with an intercept and a slope given by β_0 and β_1, respectively, and errors that follow an AR(1) plus white noise (AR(1)+WN) process, i.e.

$$y_t = \beta_0 + \beta_1 u_t + x_t + v_t \tag{2.19}$$

$$x_t = \phi x_{t-1} + w_t, \tag{2.20}$$

where u_t is a known explanatory variable, $v_t \overset{iid}{\sim} \mathrm{N}(0, \sigma_m^2) \perp w_t \overset{iid}{\sim} \mathrm{N}(0, \sigma_s^2)$, and $\theta = (\beta_0, \beta_1, \phi, \sigma_s^2, \sigma_m^2)'$ represents the unknown parameters in the model. Here, ϕ is the lag-1 coefficient of the AR(1) process (note the subscript '1' is removed since there is only one AR coefficient), σ_m^2 is the observation variance, and σ_s^2 is the DLM state variance, or more precisely the variance of the innovations of the AR(1) process for the univariate state. We will refer to this parameter as the *white noise variance* for the state, since the actual variance of the state is $\sigma_s^2/(1-\phi^2)$ (provided $-1 < \phi < 1$ to ensure stationarity of the state process), where σ_s^2 represents the variance of w_t, the "white noise" component of the AR(1) process.

Reexpressing equation (2.19) as

$$y_t = (\beta_0 + x_t) + \beta_1 u_t + v_t \tag{2.21}$$

shows that we can interpret this model as a *dynamic intercept model*, i.e. a simple linear regression model with an intercept that changes over time according to an AR(1) process. We can represent a dynamic intercept model as a DLM defined by equations (2.13) and (2.14) where $U_t = (1, u_t)$, $\beta = (\beta_0, \beta_1)'$, $F_t = 1$ for all t, $V = \sigma_m^2$, $G = \phi$, and $W = \sigma_s^2$.

Example 2: Dynamic slope model

Letting instead $F_t = u_t$ in (2.13) with $U_t = (1, u_t)$, $\beta = (\beta_0, \beta_1)'$, $V = \sigma_m^2$, $G = \phi$, and $W = \sigma_s^2$ yields a *dynamic slope model*, or a simple linear regression model with a slope that changes over time. This can be seen by multiplying out equation (2.13):

$$y_t = \beta_0 + \beta_1 u_t + x_t u_t + v_t = \beta_0 + (\beta_1 + x_t)u_t + v_t. \tag{2.22}$$

Example 3: Dynamic intercept and slope model

Finally, we consider a model with both a dynamic slope and a dynamic intercept by letting $x_t = (x_{t,1}, x_{t,2})'$ be two-dimensional and adjusting F_t, G, and W such that

$$F_t = (1, u_t), \qquad G = \begin{pmatrix} \phi & 0 \\ 0 & \rho \end{pmatrix}, \text{ and } \qquad W = \begin{pmatrix} \sigma_s^2 & 0 \\ 0 & \sigma_b^2 \end{pmatrix}.$$

Multiplying out both equations (2.13) and (2.14), we have

$$y_t = (\beta_0 + x_{t,1}) + (\beta_1 + x_{t,2})u_t + v_t. \tag{2.23}$$

$$x_{t,1} = \phi x_{t-1,1} + w_{t,1} \tag{2.24}$$

$$x_{t,2} = \rho x_{t-1,2} + w_{t,2}, \tag{2.25}$$

where $w_{t,1} \overset{iid}{\sim} \mathrm{N}(0, \sigma_s^2)$, $w_{t,2} \overset{iid}{\sim} \mathrm{N}(0, \sigma_b^2)$, $v_t \overset{iid}{\sim} \mathrm{N}(0, \sigma_m^2)$, $\beta = (\beta_0, \beta_1)'$ and $\theta = (\beta, \phi, \rho, \sigma_s^2, \sigma_b^2, \sigma_m^2)'$ are the unknown parameters. In this model, ϕ now represents the lag-1 autocorrelation for the change in the intercept, ρ is the lag-1 autocorrelation for the change in the slope, σ_s^2 is the white noise variance for the dynamic intercept, and σ_b^2 is the white noise variance for the dynamic slope.

In Chapter 6, we compare the dynamic slope, dynamic intercept, and standard simple linear regression models for fitting fMRI data. For ease of reference, we adopt notation to refer to models within a general class of dynamic regression models. Let M_{ijk} represent a (possibly) dynamic regression model where i is the order of the AR process for the dynamic intercept, j is the order of the AR process for the dynamic slope, and k is either 1 or 0 indicating whether or not the model contains random error in the observation equation (i.e., $k = 0$ implies v_t is restricted to be 0 and $k = 1$ implies $v_t \overset{iid}{\sim} \mathrm{N}(0, \sigma_m^2)$ with $\sigma_m^2 > 0$). Letting either i or j be 0 removes the stochasticity in the corresponding component. Thus, M_{101} corresponds to the dynamic intercept model described by equations 2.21 and 2.20, M_{011} is the dynamic slope model described by equations 2.22 and 2.20, M_{111} is the

model with both a dynamic intercept and dynamic slope described by equations 2.23, 2.24, and 2.25, and M_{001} describes a simple linear regression model with fixed coefficients and independent errors, i.e. equations (2.19) and (2.20) with $\phi = \sigma_s^2 = 0$.

In all of these dynamic regression models, we allow for nonstationarity of the state process. This is intended to enable modeling of a wider range of behavior in fMRI data, as well as estimation using the particle learning algorithm, which we describe in Chapter 3. In this setting, x_t may not have a stationary mean, in which case the model may not be properly identified by unique values of the fixed parameters (Pagan; 1979). To alleviate this concern, we constrain $x_0 = 0$ (or $x_0 = (0, 0)'$ for M_{111}) so that x_t is interpreted as the change in the dynamic slope or intercept (or both) from time $t = 0$. This is equivalent to setting the marginal prior distribution of the initial state as $p(x_0) = \delta_0(x_0)$. In Chapter 6, Bayesian models for M_{101} and M_{011} are analyzed using priors of the form $p(x_0, \theta) = p(\beta|\sigma_m^2)p(\sigma_m^2)p(\phi|\sigma_s^2)p(\sigma_s^2)\delta_0(x_0)$, where

$$\beta|\sigma_m^2 \sim \mathrm{N}(\vartheta_0, \sigma_m^2 B_0) \qquad \sigma_m^2 \sim \mathrm{IG}(a_{m_0}, b_{m_0}) \tag{2.26}$$

$$\phi|\sigma_s^2 \sim \mathrm{N}(\varphi_0, \sigma_s^2 \Phi_0) \qquad \sigma_s^2 \sim \mathrm{IG}(a_{s_0}, b_{s_0}) \tag{2.27}$$

and the hyperparameters ϑ_0, B_0, φ_0, Φ_0, a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} are assumed known.

2.4 Sequential estimation

When data are collected sequentially, it is often of interest to determine the *filtered distribution* $p(x_t, \theta|y_{1:t})$, i.e. the distribution of the current state and parameters conditional on the data observed up to that time. This distribution describes all of the available information up to time t about the current state of the system and any fixed parameters. We may also be interested in the filtered distribution of the entire state history and any fixed parameters, $p(x_{0:t}, \theta|y_{1:t})$. Both can be updated recursively using Bayes' rule:

$$p(x_t, \theta|y_{1:t}) \propto \int p(y_t|x_t, \theta)p(x_t|x_{t-1}, \theta)p(x_{t-1}, \theta|y_{1:t-1})\mathrm{d}x_{t-1} \tag{2.28}$$

$$p(x_{0:t}, \theta|y_{1:t}) \propto p(y_t|x_t, \theta)p(x_t|x_{t-1}, \theta)p(x_{0:t-1}, \theta|y_{1:t-1}), \tag{2.29}$$

where $y_{1:t} = (y_1, \ldots, y_t)$. The *smoothed distribution*, $p(x_s, \theta|y_{1:t})$ for any $s < t$, is then calculated by integrating $p(x_{0:t}, \theta|y_{1:t})$ over all states $\{x_j : j \in \{0, \ldots, t\}/\{s\}\}$.

Only in special cases can these filtered or smoothed distributions be evaluated analytically. The DLM described by equations (2.13) and (2.13) is one such case, provided all fixed parameters (i.e. θ) are known. In this case, an explicit form for $p(x_t|y_{1:t})$ can be found according to $x_t|y_{1:t} \sim \mathrm{N}(m_t, C_t)$, where m_t and C_t are calculated recursively using the Kalman filter (Kalman; 1960). Starting with known m_0 and C_0, the filtering recursions are given by the following equations

(Section 2.7.2 Petris et al.; 2009):

$$\begin{aligned} z_t &= Gm_{t-1} & R_t &= GC_{t-1}G' + W \\ f_t &= U_t\beta + F_t z_t & Q_t &= F_t R_t F_t' + V \\ m_t &= z_t + R_t F_t' Q_t^{-1}(y_t - f_t) & C_t &= R_t - R_t F_t' Q_t^{-1} F_t R_t. \end{aligned} \tag{2.30}$$

When unknown fixed parameters are present in DLMs, analytical tractibility exists in only a few cases, such as the local level DLM described in Section 2.3.1 with common observation and state variance factor (Section 4.3 Petris et al.; 2009). In this case, $p(x_t, \theta|y_{1:t})$ is given by

$$x_t|\theta, y_{1:t} \sim \mathrm{N}(m_t, \theta c_t) \quad \theta|y_{1:t} \sim \mathrm{IG}(a_t, b_t), \tag{2.31}$$

where m_t, c_t, a_t, and b_t are calculated recursively according to

$$\begin{aligned} f_t &= m_{t-1} & q_t &= c_{t-1} + \lambda + 1 \\ m_t &= (1 - c_t)f_t + c_t y_t & c_t &= 1 - \frac{1}{q_t} \\ a_t &= a_{t-1} + \frac{1}{2} & b_t &= b_{t-1} + \frac{(y_t - f_t)^2}{2q_t}, \end{aligned} \tag{2.32}$$

starting with $m_0 = 0$, $c_0 = 1$, and known a_0 and b_0. These equations can also be used to calculate the marginal filtered distribution of the state, $p(x_t|y_{1:t})$, and one-step ahead predictive density, $p(y_t|y_{1:t-1})$, given by

$$x_t|y_{1:t} \sim \mathrm{T}\left(m_t, c_t\frac{b_t}{a_t}, 2a_t\right), \quad t = 1, 2, \ldots \tag{2.33}$$

$$y_t|y_{1:t-1} \sim \mathrm{T}\left(f_t, q_t\frac{b_{t-1}}{a_{t-1}}, 2a_{t-1}\right), \quad t = 2, 3, \ldots \tag{2.34}$$

with initial $y_1 \sim \mathrm{T}\left(f_1, q_1 \frac{b_0}{a_0}, 2a_0\right)$. In Chapter 5, we evaluate the abilities of several particle filters to approximate the marginal likelihood of data generated from this model through comparison with the true marginal likelihood that can be calculated analytically according to

$$p(y_{1:t}) = \left(\prod_{k=2}^{t} p(y_k|y_{1:k-1})\right) p(y_1). \tag{2.35}$$

The remaining DLMs described in sections 2.3.2 and 2.3.3 do not emit analytically tractable forms of the posterior. When analytical tractability is not present, we turn to numerical methods including deterministic versions, e.g. the extended Kalman filter (Section 1.6 Haykin; 2001) and the Gaussian sum filter (Alspach and Sorenson; 1972), or Monte Carlo versions such as particle filters. In Chapter 3, we describe a variety of Markov chain Monte Carlo and sequential Monte Carlo algorithms that we use to estimate states and unknown fixed parameters in state-space models for which filtered distributions are intractable.

Chapter 3

Methods

In Chapters 4 and 6, we consider estimation of states and unknown fixed parameters in Bayesian state-space models for which filtered distributions cannot be calculated analytically. In this case, the most common approach to approximating these distributions is through Markov-chain Monte Carlo (MCMC) methods (Gelfand and Smith; 1990). MCMC is an effective tool for analyzing data in complex modeling situations (Robert and Casella; 2004). However, in sequential analysis using state-space models, where new observations are arriving as time progresses, MCMC is inefficient due to the increase in computational cost incurred by the need for the entire MCMC to be rerun as each new observation arrives.

Sequential Monte Carlo (SMC) - or particle filtering - techniques, on the other hand, enable on-line inference through updating the approximation to the pos-

terior distribution as new data become available (Doucet et al.; 2001; Cappé et al.; 2007). In addition, SMC methods can be flexible, general, easy to implement, amenable to parallel computing, and provide direct estimates of the marginal likelihood. As with MCMC methods, however, the performance of SMC methods suffers as the dimension of the parameter space increases. Furthermore, while MCMC methods directly provide smoothed estimates of states in state-space models, SMC algorithms are inefficient for smoothing and have been mostly used only for filtering (Section 5 Doucet and Johansen; 2009). Each of MCMC and SMC approaches have strengths and limitations in different scenarios.

In Chapters 4 and 5, we compare the performance of several particle filtering algorithms in different model settings. In Chapter 6, we use SMC methods as a tool for model comparison, where we compare the relative posterior probabilities of several models that represent different types/sources of autocorrelation that might be present in fMRI time series data. We discuss possible reasons for the presence of autocorrelation in fMRI time series data and possible modeling approaches in Section 6.2. In each of Chapters 4, 5, and 6, we compare SMC results with MCMC, which has been the standard over the past two decades for Bayesian analysis of analytically intractable models. Thus, in this chapter, we review several strategies for both MCMC and particle filtering.

Specifically, in Section 3.1, we describe the MCMC algorithms we use for the epidemic model described in Section 2.2 (considered further in Chapter 4) and

the dynamic regression models described in Sections 2.3.3 and 2.3.3 (considered further in Chapter 6). In Section 3.2, we describe several particle filtering strategies and how to apply them to the models outlined in Chapter 2. In Section 3.3, we discuss the resampling methods within several particle filtering algorithms. In Section 3.4, we show how SMC techniques can be used for model comparison. Lastly, in Section 3.5, we describe a particle MCMC algorithm that we also use to analyze simulated data from the epidemic model, and which we found to perform more efficiently than standard MCMC.

3.1 Markov chain Monte Carlo (MCMC) algorithms

MCMC methods provide sample-based approximations to the posterior distribution through the generation of dependent samples from distributions whose densities can be evaluated. In this section, we outline the MCMC algorithms that we use to analyze simulated data from the state-space model of a disease epidemic described in Section 2.2, and from the dynamic regression models described in Sections 2.3.3 and 2.3.3. We also describe a particle MCMC (PMCMC) approach that we found to be more efficient when analyzing simulations from the epidemic model. For more comprehensive descriptions of MCMC and PMCMC

methods, we refer the reader to Robert and Casella (2004) and Andrieu et al. (2010) respectively.

3.1.1 MCMC applied to epidemic model

Consider the specific state-space model of an epidemic described in Section 2.2, where $x_t = (s_t, i_t)'$ is the latent disease state, $\theta = (\beta, \gamma, \nu)$ are the unknown fixed parameters, the state equation (2.3) describes the evolution of x_t given x_{t-1} and θ, and the observation equation (2.6) describes the likelihood of new data, y_t, given x_t and θ. We assume the prior distribution, $p(x_0, \theta)$, of the form

$$p(x_0, \theta) = p(x_0)p(\theta) = p(s_0, i_0)p(\beta, \gamma)p(\nu), \tag{3.1}$$

where $p(s_0, i_0)$ is given by equation (2.7).

Suppose we observe syndromic surveillance data, y_t, for $t = 1, 2, \ldots, T$. Using the fact that, for state-space models, we can express the joint density of the data, states, and fixed parameters by

$$p(y_{1:T}, x_{0:T}, \theta) = \prod_{t=1}^{T} \{p(y_t|x_t, \theta)p(x_t|x_{t-1}, \theta)\}\, p(x_0, \theta), \tag{3.2}$$

we derive the *full conditional distribution*, i.e. distribution of a random vector conditional on all of the remaining variables in the model, for each of $x_0, x_1, \ldots, x_T$,

β, γ, and ν as

$$
\begin{aligned}
p(x_0|\ldots) &\propto p(x_1|x_0,\theta)p(x_0) \qquad (3.3)\\
p(x_t|\ldots) &\propto p(y_t|x_t)p(x_{t+1}|x_t,\theta)p(x_t|x_{t-1},\theta), \text{ for } t = 1,\ldots,T-1\\
p(x_T|\ldots) &\propto p(y_T|x_T)p(x_T|x_{T-1},\theta)\\
p(\beta|\ldots) &\propto \prod_{t=1}^{T} \{p(x_t|x_{t-1},\theta)\}\, p(\beta,\gamma)\\
p(\gamma|\ldots) &\propto \prod_{t=1}^{T} \{p(x_t|x_{t-1},\theta)\}\, p(\beta,\gamma)\\
p(\nu|\ldots) &\propto \prod_{t=1}^{T} \{p(x_t|x_{t-1},\theta)\}\, p(\nu),
\end{aligned}
$$

where $p(w|\ldots)$ represents the full conditional distribution of w, for any w. Note that since each of the unknown fixed parameters β, γ, and ν is present only in the state equation (2.3) of the model, their full conditional distributions do not depend on y_t. By the same argument, $p(y_t|x_t,\theta)$ reduces to $p(y_t|x_t)$.

We use these full conditional distributions to generate samples from $p(x_{0:T},\theta|y_{1:T})$ by implementing a Gibbs sampler with adaptive rejection Metropolis-Hastings (MH) steps (Metropolis et al.; 1953; Hastings; 1970; Geman and Geman; 1984; Gilks et al.; 1995). In general, the algorithm works by iteratively sampling each state and fixed parameter, conditional on the current sample, from some proposal distribution, g, and accepting the proposed sample with probability R. R, termed the *Metropolis ratio*, is given by

$$R = \frac{f(w^*)g(w|w^*)}{f(w)g(w^*|w)}, \qquad (3.4)$$

where w is the current sample, w^* is the proposed sample from g, and $f(\cdot)$ is the full conditional distribution of the state or fixed parameter evaluated at '$\cdot$' (Chapter 7 Givens and Hoeting; 2005). In our algorithm, we use Gaussian random-walk proposals for each state and fixed parameter, i.e. each proposed sample is drawn from a normal distribution centered at the current sampled value with standard deviation given by a tuning parameter that is adjusted according to the MH acceptance rate. Because these proposal distributions are symmetric, $g(w|w^*)$ and $g(w^*|w)$ cancel out, reducing the Metropolis ratio to

$$R = \frac{f(w^*)}{f(w)}. \tag{3.5}$$

Let $x_{0:T}^{(j)} = \left(x_0^{(j)}, x_1^{(j)}, \ldots, x_T^{(j)}\right)'$ and $\theta^{(j)} = \left(\beta^{(j)}, \gamma^{(j)}, \nu^{(j)}\right)'$ represent the sampled values of the states and fixed parameters, respectively, at iteration j of the Gibbs sampler. The full Gibbs sampler applied to the epidemic model proceeds as follows:

1. Start with initial draws $x_{0:T}^{(0)} = \left(x_0^{(0)}, x_1^{(0)}, \ldots, x_T^{(0)}\right)'$ and $\theta^{(0)} = \left(\beta^{(0)}, \gamma^{(0)}, \nu^{(0)}\right)'$. Set $j = 1$.

2. Sample the states, $x_t^{(j)}$ for $t = 0, 1, \ldots, T$, from their full conditional distributions. For each $t = 1, 2, \ldots, T$,

   (a) Draw $x_t^* \sim \mathrm{N}\left(x_t^{(j-1)}, \tau_{x_t}^2 I_2\right)$.

(b) Calculate the Metropolis ratio, R, by

$$R = \begin{cases} \dfrac{p\left(x_1^{(j-1)}\middle|x_0^*,\theta^{(j-1)}\right)\ p\left(x_0^*\right)}{p\left(x_1^{(j-1)}\middle|x_0^{(j-1)},\theta^{(j-1)}\right)\ p\left(x_0^{(j-1)}\right)} & \text{, if } t=0 \\ \dfrac{p(y_t|x_t^*)\ p\left(x_{t+1}^{(j-1)}\middle|x_t^*,\theta^{(j-1)}\right)\ p\left(x_t^*\middle|x_{t-1}^{(j)},\theta^{(j-1)}\right)}{p\left(y_t\middle|x_t^{(j-1)}\right)p\left(x_{t+1}^{(j-1)}\middle|x_t^{(j-1)},\theta^{(j-1)}\right)p\left(x_t^{(j-1)}\middle|x_{t-1}^{(j)},\theta^{(j-1)}\right)} & \text{, if } 1\le t\le T-1 \\ \dfrac{p\left(y_T\middle|x_T^*\right)\ p\left(x_T^*\middle|x_{T-1}^{(j)},\theta^{(j-1)}\right)}{p\left(y_T\middle|x_T^{(j-1)}\right)p\left(x_T^{(j-1)}\middle|x_{T-1}^{(j)},\theta^{(j-1)}\right)} & \text{, if } t=T \end{cases}$$

(c) Draw $u \sim \text{Unif}[0,1]$. If $u < \min\{1, R\}$, set $x_t^{(j)} = x_t^*$. Otherwise, set $x_t^{(j)} = x_t^{(j-1)}$.

3. Sample $\beta^{(j)}$ from its full conditional conditional distribution.

(a) Draw $\beta^* \sim \text{N}\left(\beta^{(j-1)}, \tau_\beta^2\right)$.

(b) Calculate Metropolis ratio, R, by

$$\begin{aligned} R &= \frac{p\left(\beta^*|x_{0:T}^{(j)}, \gamma^{(j-1)}, \nu^{(j-1)}\right)}{p\left(\beta^{(j-1)}|x_{0:T}^{(j)}, \gamma^{(j-1)}, \nu^{(j-1)}\right)} \\ &= \frac{\left\{\prod_{t=1}^T p\left(x_t^{(j)}\middle|x_{t-1}^{(j)}, \beta^*, \gamma^{(j-1)}, \nu^{(j-1)}\right)\right\} p(\beta^*, \gamma^{(j-1)})}{\left\{\prod_{t=1}^T p\left(x_t^{(j)}\middle|x_{t-1}^{(j)}, \theta^{(j-1)}\right)\right\} p\left(\beta^{(j-1)}, \gamma^{(j-1)}\right)}. \end{aligned}$$

(c) Draw $u \sim \text{Unif}[0,1]$. If $u < \min\{1, R\}$, set $\beta^{(j)} = \beta^*$. Otherwise, set $\beta^{(j)} = \beta^{(j-1)}$.

4. Sample $\gamma^{(j)}$ from its full conditional conditional distribution.

(a) Draw $\gamma^* \sim \text{N}\left(\gamma^{(j-1)}, \tau_\gamma^2\right)$.

(b) Calculate Metropolis ratio, R, by

$$R = \frac{p\left(\gamma^* | x_{0:T}^{(j)}, \beta^{(j)}, \nu^{(j-1)}\right)}{p\left(\gamma^{(j-1)} | x_{0:T}^{(j)}, \beta^{(j)}, \nu^{(j-1)}\right)} = \frac{\left\{\prod_{t=1}^{T} p\left(x_t^{(j)} \middle| x_{t-1}^{(j)}, \beta^{(j)}, \gamma^*, \nu^{(j-1)}\right)\right\} p(\beta^{(j)}, \gamma^*)}{\left\{\prod_{t=1}^{T} p\left(x_t^{(j)} \middle| x_{t-1}^{(j)}, \beta^{(j)}, \gamma^{(j-1)}, \nu^{(j-1)}\right)\right\} p\left(\beta^{(j)}, \gamma^{(j-1)}\right)}.$$

(c) Draw $u \sim \text{Unif}[0,1]$. If $u < \min\{1, R\}$, set $\gamma^{(j)} = \gamma^*$. Otherwise, set $\gamma^{(j)} = \gamma^{(j-1)}$.

5. Sample $\nu^{(j)}$ from its full conditional conditional distribution.

(a) Draw $\nu^* \sim \text{N}\left(\nu^{(j-1)}, \tau_\nu^2\right)$.

(b) Calculate Metropolis ratio, R, by

$$R = \frac{p\left(\nu^* | x_{0:T}^{(j)}, \beta^{(j)}, \gamma^{(j)}\right)}{p\left(\nu^{(j-1)} | x_{0:T}^{(j)}, \beta^{(j)}, \gamma^{(j)}\right)} = \frac{\left\{\prod_{t=1}^{T} p\left(x_t^{(j)} \middle| x_{t-1}^{(j)}, \beta^{(j)}, \gamma^{(j)}, \nu^*\right)\right\} p(\nu^*)}{\left\{\prod_{t=1}^{T} p\left(x_t^{(j)} \middle| x_{t-1}^{(j)}, \beta^{(j)}, \gamma^{(j)}, \nu^{(j-1)}\right)\right\} p\left(\nu^{(j-1)}\right)}.$$

(c) Draw $u \sim \text{Unif}[0,1]$. If $u < \min\{1, R\}$, set $\nu^{(j)} = \nu^*$. Otherwise, set $\nu^{(j)} = \nu^{(j-1)}$.

6. Set $j = j + 1$ and go back to step 2

The output of this algorithm is a dependent chain of samples which, provided j is large enough, can be assumed to represent draws from the stationary distribution $p(x_{0:T}, \theta | y_{1:T})$ (Chapter 7 Robert and Casella; 2004). Initial values $\theta^{(0)}$ and $x_{0:T}^{(0)}$ could be chosen arbitrarily or by sampling from the prior

$\prod_{t=1}^{T} \{p(x_t|x_{t-1}, \theta)\}\, p(x_0, \theta)$. In either case, the effective sample size of the chain could be sensitive to the initial values. Generating multiple chains from different starting points could help determine reasonable starting values or the burn-in period required before the samples can be assumed to come from the stationary distribution (Givens and Hoeting; 2005). In Section 4.7, we state the initial values used in our implementation of this algorithm applied to data simulated from the epidemic model described in Section 2.2.

The standard deviations of the random-walk proposal distributions, i.e. τ_{x_t} for $t = 0, 1, \ldots, T$, τ_β, τ_γ, and τ_ν, are tuning parameters that are adjusted during the burn-in period of the MCMC. During burn-in, if the proposed value of a state or parameter at any given iteration of the Gibbs sampler is accepted, we adjust the corresponding tuning parameter by multiplying by 1.1. If the proposed value is rejected, we adjust the tuning parameter by dividing by 1.1. The idea here is that a high acceptance rate indicates that proposed samples are in areas of high posterior probability while a low acceptance rate indicates that they are in areas of low posterior probability. We seek proposal distributions that strike a balance in the acceptance rate such that the entire sample space of the posterior is explored. Acceptance rates for optimal mixing of MCMC chains will vary by model and have been explored by Roberts et al. (1997) and Bedard (2008).

3.1.2 MCMC applied to dynamic regression

We now derive an MCMC algorithm to sample from the joint posterior distribution of states and unknown parameters from the dynamic intercept (M_{101}) and dynamic slope (M_{011}) models discussed in Section 2.3.3. Recall that these models are DLMs of the form given by equations (2.13) and (2.14) with

$$U_t = (1, u_t) \qquad \beta = (\beta_0, \beta_1)'$$

$$V = \sigma_m^2 \qquad F_t = \begin{cases} 1, & \text{for } M_{101} \\ u_t, & \text{for } M_{011} \end{cases}$$

$$G = \phi \qquad W = \sigma_s^2,$$

where x_t is the univariate state representing the change in the intercept or slope at time t, and $\theta = (\beta', \phi, \sigma_s^2, \sigma_m^2)'$ are the unknown fixed parameters. We place a prior of the form

$$p(x_0, \theta) = p(x_0)p(\beta|\sigma_m^2)p(\sigma_m^2)p(\phi|\sigma_s^2)p(\sigma_s^2) \tag{3.6}$$

on the initial state and fixed parameters, where $p(x_0) = \delta_0(x_0)$ and, as stated in equations (2.26) and (2.27),

$$\beta|\sigma_m^2 \sim \mathrm{N}(\vartheta_0, \sigma_m^2 B_0) \qquad \sigma_m^2 \sim \mathrm{IG}(a_{m_0}, b_{m_0})$$

$$\phi|\sigma_s^2 \sim \mathrm{N}(\varphi_0, \sigma_s^2 \Phi_0) \qquad \sigma_s^2 \sim \mathrm{IG}(a_{s_0}, b_{s_0})$$

with known ϑ_0, B_0, φ_0, Φ_0, a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0}. The conjugate form of these priors, conditional on x_t, allows for direct sampling from the full conditional distributions of the fixed parameters. Combining this with the forward-filtering

backward sampling (FFBS) algorithm for jointly sampling the states (Carter and Kohn; 1994) allows for a relatively straightforward Gibbs sampler.

Suppose we observe y_t for $t = 1, 2, \ldots, T$ and let $x_{0:T}^{(j)} = \left(x_0^{(j)}, x_1^{(j)}, \ldots, x_T^{(j)}\right)'$ and $\theta^{(j)} = \left(\beta^{(j)\prime}, \phi^{(j)}, \sigma_s^{2(j)}, \sigma_m^{2\ (j)}\right)'$ represent the sampled values of the states and fixed parameters, respectively, at iteration j of the Gibbs sampler. We generate samples from $p(x_{0:T}, \theta | y_{1:T})$ using the following Gibbs sampling algorithm:

1. Start with initial draws $\theta^{(0)} = \left(\beta^{(0)\prime}, \phi^{(0)}, \sigma_s^{2(0)}, \sigma_m^{2\ (0)}\right)'$ and $x_{0:T}^{(0)} = \left(x_0^{(0)}, x_1^{(0)}, \ldots, x_T^{(0)}\right)$. Set $j = 1$.

2. Jointly sample $\sigma_m^{2\ (j)} \sim \text{IG}(a_{m_T}, b_{m_T})$ and $\beta^{(j)} | \sigma_m^{2\ (j)} \sim \text{N}(\vartheta_T, \sigma_m^{2\ (j)} B_T)$, where

$$a_{m_T} = T/2 + a_{m_0} \tag{3.7}$$

$$b_{m_T} = \frac{1}{2}(\text{SS}_y + \vartheta_0' B_0^{-1} \vartheta_0 - \vartheta_T' B_T^{-1} \vartheta_T) + b_{m_0} \tag{3.8}$$

$$\text{SS}_y = \sum_{t=1}^{T} (y_t - F_t x_t)'(y_t - F_t x_t) \tag{3.9}$$

$$\vartheta_T = B_T \left(\sum_{t=1}^{T} U_t'(y_t - F_t x_t) + B_0^{-1} \vartheta_0 \right) \tag{3.10}$$

$$B_T = \left(\sum_{t=1}^{T} U_t' U_t + B_0^{-1} \right)^{-1}. \tag{3.11}$$

3. Jointly sample $\sigma_s^{2(j)} \sim \text{IG}(a_{s_T}, b_{s_T})$ and $\phi^{(j)}|\sigma_s^{2(j)} \sim \text{N}(\varphi_T, \sigma_s^{2(j)}\Phi_T)$, where

$$a_{s_T} = T/2 + a_{s_0} \tag{3.12}$$

$$b_{s_T} = \frac{1}{2}(\text{SS}_x + \varphi_0'\Phi_0^{-1}\varphi_0 - \varphi_T'\Phi_T^{-1}\varphi_T) + b_{s_0} \tag{3.13}$$

$$\text{SS}_x = \sum_{t=1}^{T} x_t^2 \tag{3.14}$$

$$\varphi_T = \Phi_T \left(\sum_{t=1}^{T} x_t x_{t-1} + \Phi_0^{-1}\varphi_0 \right) \tag{3.15}$$

$$\Phi_T = \left(\sum_{t=1}^{T} x_{t-1}^2 + \Phi_0^{-1} \right)^{-1}. \tag{3.16}$$

4. Sample $x_{0:T}^{(j)}$ using the following FFBS (forward filtering, backward sampling) algorithm (Section 4.4 Petris et al.; 2009), setting $\theta = \theta^{(j)}$ from Steps 2 and 3:

   (a) Start with initial values $m_0 = C_0 = 0$.

   (b) Calculate z_t, R_t, m_t and C_t for $t = 1, 2, \ldots, T$ using the Kalman filter given by equation (2.30).

   (c) Draw $x_T^{(j)} \sim \text{N}(m_T, C_T)$. Then, for $t = T-1, \ldots, 0$, draw $x_t^{(j)} \sim \text{N}(h_t, H_t)$, where

$$h_t = m_t + C_t G' R_{t+1}^{-1}(x_{t+1}^{(j)} - z_{t+1})$$

$$H_t = C_t - C_t G' R_{t+1}^{-1} G C_t.$$

5. Set $j = j + 1$ and go back to Step 2.

We discuss initial values that we use in Step 1 when we apply this algorithm to simulated fMRI data in Section 6.4.1. Note that both this algorithm and the MCMC for the epidemic model described in Section 3.1.1 provide joint samples from $p(x_{0:T}, \theta|y_{1:T})$. Samples from the smoothed distributions, $p(x_s, \theta|y_{1:T})$ for $s < T$, can be directly obtained from these joint samples through Monte Carlo integration (Chapter 3 Robert and Casella; 2004).

3.2 Particle filtering

Particle filtering is an SMC inferential technique based on repeated use of importance sampling. It aims to approximate the filtered distribution at time t through a weighted Monte Carlo realization from this distribution in terms of J particles, i.e.

$$p(x_t, \theta|y_{1:t}) \approx \sum_{j=1}^{J} w_t^{(j)} \delta_{\left(x_t^{(j)}, \theta^{(j)}\right)}(x_t, \theta), \tag{3.17}$$

where $\left(x_t^{(j)}, \theta^{(j)}\right)$ is the location of the j^{th} particle at time t and $w_t^{(j)}$ is the weight of that particle with $\sum_{j=1}^{J} w_t^{(j)} = 1$. A variety of SMC techniques have been developed to provide more efficient approximations to $p(x_t, \theta|y_{1:t})$ in the sense that with the same computation time a better approximation is achieved. In this section, we describe five particle filtering techniques: the bootstrap filter (BF), the auxiliary particle filter (APF), the kernel density particle filter (KDPF), the resample-move particle filter (RM), and particle learning (PL).

Each of these five strategies has its own advantages and disadvantages. The BF and APF are the simplest and most straightforward to implement, but are unequipped to efficiently deal with state-space models that contain unknown fixed parameters. PL performs the most efficiently, but can only be applied to special cases of state-space models such as DLMs. The RM, while capable of handling state-space models of any form, requires an MCMC step in addition to the SMC, and thus is not a truly sequential algorithm. The KDPF, while being the only truly sequential particle filtering algorithm that can be applied to any state-space model, is outperformed by the RM and PL in many model settings.

In Chapter 4, we compare the efficiency of the BF, APF, and KDPF in the syndromic surveillance context. In Chapter 5, we compare the KDPF, RM, and PL in terms of their efficiency for estimating the marginal likelihood of data generated from the local level DLM described in Section 2.3.1. Finally, in Chapter 6, we employ PL for estimating states and unknown fixed parameters in DLMs using real and simulated fMRI data.

3.2.1 Bootstrap filter (BF)

The BF is first successful version of the particle filter (Gordon et al.; 1993; Kitagawa; 1996). Since this method and the APF were developed for when θ is known, we will (for the moment) drop θ from the notation. Given an approximation to

$p(x_t|y_{1:t})$ as in equation (3.17) (with θ omitted), we obtain an approximation to $p(x_{t+1}|y_{1:t+1})$ by performing the following steps for each particle $j = 1, \ldots, J$:

1. Resample: sample an index $k \in \{1, \ldots, j, \ldots, J\}$ with associated probabilities $\left\{w_t^{(1)}, \ldots, w_t^{(j)}, \ldots, w_t^{(J)}\right\}$,

2. Propagate: sample $x_{t+1}^{(j)} \sim p\left(x_{t+1} \middle| x_t^{(k)}\right)$, and

3. Calculate weights and renormalize:

$$\tilde{w}_{t+1}^{(j)} = p\left(y_{t+1} \middle| x_{t+1}^{(j)}\right) \qquad w_{t+1}^{(j)} = \tilde{w}_{t+1}^{(j)} \Bigg/ \sum_{l=1}^{J} \tilde{w}_{t+1}^{(l)} .$$

This procedure can be applied recursively beginning with an initial set of weights $w_0^{(j)}$ and locations $x_0^{(j)}$ for all j. For all particle filters that we implement, we initialize the algorithm by sampling from the prior with uniform weights.

3.2.2 Auxiliary particle filter (APF)

One problem that arises in implementing the BF is that $w_t^{(j)}$ will be small for particles for which $p\left(y_t \middle| x_t^{(j)}\right)$ is small, and these particles will contribute little to the approximation to $p(x_t|y_{1:t})$. The APF aims to mitigate this by anticipating which particles will have small weight using a look ahead strategy (Pitt and Shephard; 1999). Given an approximation to $p(x_t|y_{1:t})$, the APF approximates $p(x_{t+1}|y_{1:t+1})$ by the following:

1. For each particle j, calculate a point estimate of $x_{t+1}^{(j)}$ called $\mu_{t+1}^{(j)}$, e.g.

$$\mu_{t+1}^{(j)} = E\left(x_{t+1} \middle| x_t^{(j)}\right).$$

2. Calculate auxiliary weights and renormalize:

$$\tilde{g}_{t+1}^{(j)} = w_t^{(j)} p\left(y_{t+1} \middle| \mu_{t+1}^{(j)}\right) \qquad g_{t+1}^{(j)} = \tilde{g}_{t+1}^{(j)} \Big/ \sum_{l=1}^{J} \tilde{g}_{t+1}^{(l)}.$$

3. For each particle $j = 1, \ldots, J$,

   (a) Resample: sample an index $k \in \{1, \ldots, j, \ldots, J\}$ with associated probabilities $\left\{g_{t+1}^{(1)}, \ldots, g_{t+1}^{(j)}, \ldots, g_{t+1}^{(J)}\right\}$,

   (b) Propagate: sample $x_{t+1}^{(j)} \sim p\left(x_{t+1} \middle| x_t^{(k)}\right)$, and

   (c) Calculate weights and renormalize:

$$\tilde{w}_{t+1}^{(j)} = \frac{p\left(y_{t+1} \middle| x_{t+1}^{(j)}\right)}{p\left(y_{t+1} \middle| \mu_{t+1}^{(k)}\right)} \qquad w_{t+1}^{(j)} = \tilde{w}_{t+1}^{(j)} \Big/ \sum_{l=1}^{J} \tilde{w}_{t+1}^{(l)}.$$

The point estimate used in Step 1 can be any point estimate, although the expectation is commonly used. Step 3 is exactly the same as the BF with appropriate modifications to the weight calculation to adjust for the 'look ahead' in steps 1 and 2. APF weights tend to be closer to uniform than BF weights, in which case a better approximation to $p(x_t|y_{1:t})$ is achieved.

The BF and the APF were constructed with the idea that all fixed parameters are known. In order to simultaneously estimate the time-evolving states and fixed parameters using either the BF or APF, it is necessary to incorporate the fixed parameters into the state with degenerate evolutions. That is, one regards the fixed parameters as elements of x_t and specifies the state evolution equation such that these elements do not change over time. Due to the possible duplication

of some particles and elimination of others through resampling, the number of unique values of the fixed parameters in the particle set will decrease over time, resulting in *degeneracy* in the fixed parameters (Liu and West; 2001).

3.2.3 Kernel density particle filter (KDPF)

The particle filter introduced by Liu and West (2001), which we refer to as the KDPF, builds on the APF and provides a general way of fighting degeneracy in fixed parameters. This is done by approximating the set of fixed parameter values by a kernel density estimate and then regenerating values from this approximation. This filter approximates $p(x_t, \theta|y_{1:t})$ via equation (3.17). To make the notation transparent, we introduce subscripts for our fixed parameters, e.g. $\theta_t^{(j)}$ represents the value for θ at time t for particle j. This does not imply that the true θ is dynamic, but rather that particle j can have different values for θ throughout time.

Let $\bar{\theta}_t$ and V_t be the weighted sample mean and weighted sample covariance matrix of $\theta_t^{(1)}, \ldots, \theta_t^{(J)}$. The KDPF uses a tuning parameter Δ, the discount factor that takes values in $(0, 1)$, and two derived quantities $h^2 = 1 - ((3\Delta - 1)/2\Delta)^2$ and $a^2 = 1 - h^2$ that determine how smooth the kernel density approximation is. Lower values of Δ result in a smoother approximation. However, the goal here is simply to jitter particles around to refresh values of the fixed parameters and

reduce the chance of degeneracy, and so Δ is typically taken to be between 0.95 and 0.99 (Liu and West; 2001).

Given an approximation to the filtered distribution at time t as in equation (3.17), the KDPF provides an approximation to $p(x_{t+1}, \theta | y_{1:t+1})$ by the following steps:

1. For each particle j, set $m_t^{(j)} = a\theta_t^{(j)} + (1-a)\bar{\theta}_t$ and calculate a point estimate of $x_{t+1}^{(j)}$ called $\mu_{t+1}^{(j)}$, e.g. $\mu_{t+1}^{(j)} = E\left(x_{t+1} \middle| x_t^{(j)}, \theta_t^{(j)}\right)$.

2. Calculate auxiliary weights and renormalize:

$$\tilde{g}_{t+1}^{(j)} = w_t^{(j)} p\left(y_{t+1} \middle| \mu_{t+1}^{(j)}, m_t^{(j)}\right) \qquad g_{t+1}^{(j)} = \tilde{g}_{t+1}^{(j)} \Big/ \sum_{l=1}^{J} \tilde{g}_{t+1}^{(l)}.$$

3. For each particle $j = 1, \ldots, J$,

   (a) Resample: sample an index $k \in \{1, \ldots, j, \ldots, J\}$ with associated probabilities $\left\{g_{t+1}^{(1)}, \ldots, g_{t+1}^{(j)}, \ldots, g_{t+1}^{(J)}\right\}$,

   (b) Regenerate the fixed parameters: sample $\theta_{t+1}^{(j)} \sim \mathrm{N}\left(m_t^{(k)}, h^2 V_t\right)$,

   (c) Propagate: sample $x_{t+1}^{(j)} \sim p\left(x_{t+1} \middle| x_t^{(k)}, \theta_{t+1}^{(j)}\right)$, and

   (d) Calculate weights and renormalize:

$$\tilde{w}_{t+1}^{(j)} = \frac{p\left(y_{t+1} \middle| x_{t+1}^{(j)}, \theta_{t+1}^{(j)}\right)}{p\left(y_{t+1} \middle| \mu_{t+1}^{(k)}, m_t^{(k)}\right)} \qquad w_{t+1}^{(j)} = \tilde{w}_{t+1}^{(j)} \Big/ \sum_{l=1}^{J} \tilde{w}_{t+1}^{(l)}.$$

The KDPF adds the kernel density regeneration to the auxiliary particle filter. Here, we use a mixture distribution that places normal kernels around each particle, where the mean of each kernel is a weighted average between the particle

value and the overall mean of all particles. This ensures that the variance of regenerated fixed parameter values within a specific iteration of the particle filter is the same as the variance of the fixed parameter value prior to regeneration (Liu and West; 2001).

To use the KDPF with normal kernels, it is necessary to parameterize the fixed parameters so that their support is on the real line. This is not a constraint, but rather a practical implementation detail. We typically use logarithms for parameters that have positive support and the logit function for parameters in the interval (0,1). A parameter ψ bounded on the interval (a,b) can first be rebounded to (0,1) through $(\psi - a)/(b - a)$, and then the logit transformation can be applied. We investigate the sensitivity of the performance of the particle filters to the choice of transformation in Chapter 4.

3.2.4 Resample-move algorithm (RM)

In Chapter 4, we show that the KDPF can be an effective tool for estimating unknown fixed parameters in state-space models. However, the choice of a mixture normal distribution for regenerating fixed parameter values is somewhat arbitrary, and efficiency of the algorithm can be increased by using a kernel that matches $p(\theta|y_{1:t})$ more closely. The RM, introduced by Gilks and Berzuini (2001), aims to do this by regenerating fixed parameter values from an MCMC transition kernel with stationary distribution equal to $p(\theta|y_{1:t})$. The algorithm works by running

one or a few iterations of an MCMC algorithm within each step of the particle filter for the purpose of jittering fixed parameter values. Since the weighted sample of fixed parameter values already represents an approximation to $p(\theta|y_{1:t})$, the resulting sample after running an MCMC for each particle yields a sample that can only improve the approximation (Section 4.4 Doucet and Johansen; 2009).

Since distributions that need to evaluated in MCMC algorithms often depend on all of the observed data and unobserved states, we must track the entire history of states within each particle. Thus, we now represent particle j by $\left(x_{0:t}^{(j)}, \theta_t^{(j)}\right)$ with weight $w_t^{(j)}$, where $x_{0:t}^{(j)} = \left(x_0^{(j)}, x_1^{(j)}, \ldots, x_t^{(j)}\right)$ represents the sample path of the state from time 0 to time t for particle j. The entire collection of J particles now represents an approximation to $p(x_{0:t}, \theta|y_{1:t})$.

The general RM algorithm proceeds in the following way. Given a particle approximation to $p(x_{0:t}, \theta|y_{1:t})$, we move to a particle approximation to $p(x_{0:t+1}, \theta|y_{1:t+1})$ by the following steps for each particle $j = 1, \ldots, J$:

1. Propagate: draw $\tilde{x}_{t+1}^{(j)}$ from $p\left(x_{t+1}|x_t^{(j)}, \theta_t^{(j)}\right)$. Incorporate $\tilde{x}_{t+1}^{(j)}$ into particle j and denote the new augmented particle by $\left(\tilde{x}_{0:t+1}^{(j)}, \theta_t^{(j)}\right)$, where $\tilde{x}_{0:t+1}^{(j)} = \left(\tilde{x}_0^{(j)}, \tilde{x}_1^{(j)}, \ldots, \tilde{x}_t^{(j)}, \tilde{x}_{t+1}^{(j)}\right)$,

2. Calculate weights and renormalize:

$$\tilde{w}_t^{(j)} = p\left(y_{t+1}|\tilde{x}_{t+1}^{(j)}, \theta_t^{(j)}\right) \qquad w_{t+1}^{(j)} = \tilde{w}_{t+1}^{(j)} \Big/ \sum_{l=1}^{J} \tilde{w}_{t+1}^{(l)},$$

3. Resample: sample an index k from $\{1, \ldots, j, \ldots, J\}$ with associated probabilities $\{w_{t+1}^{(1)}, \ldots, w_{t+1}^{(j)}, \ldots, w_{t+1}^{(J)}\}$, and

4. Move particles: draw a new particle $\left(x_{0:t+1}^{(j)}, \theta_{t+1}^{(j)}\right)$ from some transition kernel $q\left(x_{0:t+1}, \theta | \tilde{x}_{0:t+1}^{(k)}, \theta_t^{(k)}\right)$ with invariant distribution $p(x_{0:t+1}, \theta | y_{1:t+1})$.

RM for the local level DLM with common observation and state variance factor

In Chapter 5, we run this algorithm on data simulated from the local level DLM with unknown common variance factor, θ, described in Section 2.3.1. To do this, we need to define an MCMC kernel, q, for the "Move particles" step in the above algorithm. In this case, we sample from q as follows: For a given particle j and sampled index k,

1. Sample $\theta_{t+1}^{(j)} \sim \text{IG}(a_{t+1}, b_{t+1})$, where

$$a_t = a_0 + 1/2 + t + 1$$
$$b_t = b_0 + \frac{1}{2}\left(\sum_{i=1}^{t+1}\left(y_i - \tilde{x}_i^{(k)}\right)^2 + \frac{1}{\lambda}\sum_{i=1}^{t}\left(\tilde{x}_i^{(k)} - \tilde{x}_{i-1}^{(k)}\right)^2 + \left(\tilde{x}_0^{(k)}\right)^2\right), \text{ and}$$

2. Sample $x_{0:t+1}^{(j)}$ using the FFBS algorithm detailed in Steps 4b and 4c of the Gibbs sampler from Section 3.1.2 with $T = t + 1$ and

$$m_0 = 0 \quad C_0 = V = \theta_{t+1}^{(j)} \quad W = \theta_{t+1}^{(j)}\lambda \quad F_t = G = 1.$$

Note that the RM is not a truly sequential particle filter because of the increase in computation required with increasing t, due to the increasing dimension of the state component.

3.2.5 Particle learning (PL)

We consider a particle filtering algorithm called particle learning (Carvalho et al.; 2010) that can be applied to a particular class of state-space models which includes DLMs. For models within this class, particle learning prescribes a truly sequential algorithm that samples new values for θ from $p(\theta|y_{1:t})$ using conditional sufficient statistics. Let s_t denote the sufficient statistics for θ conditional on the states $x_{0:t}$ (unrelated to s_t in the epidemic model from Section 2.2). Then, we incorporate the sampled values of the sufficient statistics, $s_t^{(j)}$, into the particles, i.e. particle j at time t is now represented by $\left(x_t^{(j)}, s_t^{(j)}, \theta_t^{(j)}\right)$. We move from an approximation to $p(x_t, \theta|y_{1:t})$ to that of $p(x_{t+1}, \theta|y_{1:t+1})$ by the following procedure for each particle $j = 1, 2, \ldots, J$:

1. Calculate weights and renormalize:

$$\tilde{w}_{t+1}^{(j)} = p\left(y_{t+1} \middle| x_t^{(j)}, \theta_t^{(j)}\right) \qquad w_{t+1}^{(j)} = \tilde{w}_{t+1}^{(j)} \Bigg/ \sum_{l=1}^{J} \tilde{w}_{t+1}^{(l)} \,,$$

2. Resample: sample an index $k \in \{1, \ldots, j, \ldots, J\}$ with associated probabilities $\left\{w_{t+1}^{(1)}, \ldots, w_{t+1}^{(j)}, \ldots, w_{t+1}^{(J)}\right\}$,

3. Propagate: sample $x_{t+1}^{(j)} \sim p\left(x_{t+1} \middle| y_{t+1}, x_t^{(k)}, \theta_t^{(k)}\right)$,

4. Update sufficient statistics: calculate $s_{t+1}^{(j)} = S\left(y_{t+1}, x_{t+1}^{(j)}, s_t^{(k)}\right)$, and

5. Regenerate: sample $\theta_{t+1}^{(j)} \sim p\left(\theta \left| s_{t+1}^{(j)} \right.\right)$.

Note that this algorithm requires the ability to evaluate the conditional predictive distribution $p(y_{t+1}|x_t, \theta)$ and sample from the conditional filtered distributions $p(x_{t+1}|y_{t+1}, x_t, \theta)$ and $p(\theta|s_t)$. Thus, particle learning is only applicable to models for which the form of these distributions is analytically tractable. In addition, we must define the recursive map S to update the sufficient statistics based on the new observation y_{t+1} and the newly sampled state $x_{t+1}^{(j)}$. In Chapter 5, we run PL on simulated data from the local level DLM described in Section 2.3.1, and in Chapter 6, we apply PL to the dynamic regression models described in Section 2.3.3 using real and simulated fMRI data. We now show how to implement PL for these specific models.

PL for the local level DLM with common observation and state variance factor

To implement a particle learning algorithm for the local level DLM given by equations (2.10) and (2.11), we derive the conditional predictive distribution of y_{t+1} given x_t and θ, and the filtered distribution of x_{t+1} given x_t and θ, for each

t. These distributions are given by

$$y_{t+1}|x_t, \theta \sim \mathrm{N}\left(x_t, \theta(1+\lambda)\right) \tag{3.18}$$

$$x_{t+1}|y_{t+1}, x_t, \theta \sim \mathrm{N}(\mu_t, \tau^2), \tag{3.19}$$

with

$$\mu_t = \frac{\lambda}{1+\lambda}(y_{t+1} + x_t/\lambda) \qquad \tau^2 = \theta\frac{\lambda}{1+\lambda}. \tag{3.20}$$

We also derive the filtered distribution of θ conditional on the states, $p(\theta|y_{1:t}, x_{0:t})$, expressed by

$$\theta|y_{1:t}, x_{0:t} \sim \mathrm{IG}(a_t, b_t), \tag{3.21}$$

where

$$\begin{aligned} a_t &= t + 1/2 + a_0 \\ b_t &= b_0 + \frac{1}{2}\left(\sum_{k=1}^{t}(y_k - x_k)^2 + \frac{1}{\lambda}\sum_{k=1}^{t}(x_k - x_{k-1})^2 + x_0^2\right). \end{aligned}$$

Thus, $s_t = (a_t, b_t)$ are the conditional sufficient statistics for θ at time t, which can be updated according to the recursive map S defined by

$$a_{t+1} = a_t + 1, \quad t \geq 1 \tag{3.22}$$

$$b_{t+1} = \frac{1}{2}\left((y_{t+1} - x_{t+1})^2 + \frac{1}{\lambda}(x_{t+1} - x_t)^2\right) + b_t, \quad t \geq 1 \tag{3.23}$$

with initial conditions

$$\begin{aligned} a_1 &= 3/2 + a_0 \\ b_1 &= \frac{1}{2}\left((y_1 - x_1)^2 + \frac{1}{\lambda}(x_1 - x_0)^2 + x_0^2\right) + b_0. \end{aligned}$$

PL for dynamic regression models

Consider the dynamic regression models M_{101} and M_{011} described in Section 2.3.3 and given by equations (2.13) and (2.14) with

$$U_t = (1, u_t) \qquad \beta = (\beta_0, \beta_1)'$$

$$V = \sigma_m^2 \qquad F_t = \begin{cases} 1, & \text{for } M_{101} \\ u_t, & \text{for } M_{011} \end{cases}$$

$$G = \phi \qquad W = \sigma_s^2.$$

We specify the prior distributions $p(\beta, \sigma_m^2)$ and $p(\phi, \sigma_s^2)$ according to equations (2.26) and (2.27), restated below as

$$\beta|\sigma_m^2 \sim \mathrm{N}(\vartheta_0, \sigma_m^2 B_0) \qquad \sigma_m^2 \sim \mathrm{IG}(a_{m_0}, b_{m_0}) \tag{3.24}$$

$$\phi|\sigma_s^2 \sim \mathrm{N}(\varphi_0, \sigma_s^2 \Phi_0) \qquad \sigma_s^2 \sim \mathrm{IG}(a_{s_0}, b_{s_0}) \tag{3.25}$$

with $x_0 = 0$ (i.e. $p(x_0) = \delta_0(x_0)$) and the hyperparameters ϑ_0, B_0, φ_0, Φ_0, a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} assumed known. s_0 that shows up in the subscripts of a_{s_0} and b_{s_0} is unrelated to the sufficient statistic s_0 as well as s_0 from the epidemic model from Section (2.2).

To implement a particle learning algorithm for this model, we derive the conditional predictive and conditional filtered distributions

$$y_{t+1}|x_t, \theta \sim \mathrm{N}(U_{t+1}\beta + F_t\phi x_t, F_t^2\sigma_s^2 + \sigma_m^2) \tag{3.26}$$

$$x_{t+1}|y_{t+1}, x_t, \theta \sim \mathrm{N}(\mu_t, \tau_t^2), \tag{3.27}$$

where

$$\mu_t = \tau_t^2 \left(\frac{(y_{t+1} - U_{t+1}\beta)F_{t+1}}{\sigma_m^2} + \frac{\phi x_t}{\sigma_s^2} \right) \qquad \tau_t^2 = \left(\frac{F_{t+1}^2}{\sigma_m^2} + \frac{1}{\sigma_s^2} \right)^{-1}.$$

In addition, we derive $p(\theta|y_{1:t}, x_{0:t})$ using the fact that

$$p(\theta|y_{1:t}, x_{0:t}) \propto \left(\prod_{k=1}^{t} p(y_k|x_k, \beta, \sigma_m^2) p(x_k|x_{k-1}, \phi, \sigma_s^2) \right) p(\beta, \sigma_m^2) p(\phi, \sigma_s^2). \quad (3.28)$$

The filtered distribution for θ conditional on the states is then given by

$$\beta|\sigma_m^2, y_{1:t}, x_{0:t} \sim \mathrm{N}(\vartheta_t, \sigma_m^2 B_t) \qquad \sigma_m^2|y_{1:t}, x_{0:t} \sim \mathrm{IG}(a_{m_t}, b_{m_t}) \quad (3.29)$$

$$\phi|\sigma_s^2, y_{1:t}, x_{0:t} \sim \mathrm{N}(\varphi_t, \sigma_s^2 \Phi_t) \qquad \sigma_s^2|y_{1:t}, x_{0:t} \sim \mathrm{IG}(a_{s_t}, b_{s_t}), \quad (3.30)$$

where ϑ_t, B_t, φ_t, Φ_t, a_{m_t}, b_{m_t}, a_{s_t}, and b_{s_t} are calculated according to the equations in Steps 2 and 3 of the Gibbs sampler outlined in Section 3.1.2 (with $T = t$). We let $s_t = (\vartheta_t, B_t, a_{m_t}, \xi_{m_t}, \varphi_t, \Phi_t, a_{s_t}, \xi_{s_t})$ denote the sufficient statistics for θ and update them through the recursive map given by

$$B_t^{-1}\vartheta_t = B_{t-1}^{-1}\vartheta_{t-1} + U_t'(y_t - F_t x_t) \qquad B_t^{-1} = B_{t-1}^{-1} + U_t'U_t \quad (3.31)$$

$$a_{m_t} = a_{m_{t-1}} + 1/2 \qquad \xi_{m_t} = \xi_{m_{t-1}} + (y_t - F_t x_t)^2$$

$$\Phi_t^{-1}\varphi_t = \Phi_{t-1}^{-1}\varphi_{t-1} + x_t x_{t-1} \qquad \Phi_t^{-1} = \Phi_{t-1}^{-1} + x_{t-1}^2$$

$$a_{s_t} = a_{s_{t-1}} + 1/2 \qquad \xi_{s_t} = \xi_{s_{t-1}} + x_t^2,$$

where $\xi_{m_0} = \xi_{s_0} = 0$. We update ξ_{m_t} and ξ_{s_t} in the recursive map and calculate the inverse-gamma rate parameters b_{m_t} and b_{s_t} according to

$$b_{m_t} = \frac{1}{2}\left(\xi_{m_t} + \vartheta_0' B_0^{-1}\vartheta_0 - \vartheta_t' B_t^{-1}\vartheta_t\right) + b_{m_0} \quad (3.32)$$

$$b_{s_t} = \frac{1}{2}\left(\xi_{s_t} + \varphi_0' \Phi_0^{-1}\varphi_0 - \varphi_t' \Phi_t^{-1}\varphi_t\right) + b_{s_0}.$$

3.3 Resampling

Successful implementation of any particle filtering algorithm depends on which resampling scheme to use and when to resample. Resampling is sampling (with replacement) random indices between 1 and J, where index j has probability $w^{(j)}$ of being selected. Throughout our discussion, we have explicitly used multinomial resampling, but alternative resampling schemes exist including residual, stratified, and systematic resampling (Randal et al.; 2005). Residual resampling deterministically samples $\lfloor w^{(j)}J \rfloor$ copies of particle j, for each j, and distributes the remaining $J - \sum_{j=1}^{J} \lfloor w^{(j)}J \rfloor$ particles according to a multinomial distribution with associated probabilities $(w^{(j)}J - \lfloor w^{(j)}J \rfloor)/(J - \sum_{j=1}^{J} \lfloor w^{(j)}J \rfloor)$, where $\lfloor . \rfloor$ is the largest integer less than or equal ".". Stratified resampling samples uniformly over the interval $[(j-1)/J, j/J]$, for $j = 1, 2, \ldots, J$, and calculates the number of copies of particle j according to the empirical cumulative distribution function of the particle indices (i.e. the "inversion method"). Finally, systematic resampling is similar to stratified resampling, except that only one uniform draw is initially sampled from $[0, 1/J]$ and the remaining $J-1$ are calculated by adding $(j-1)/J$ to the sampled value prior to applying the inversion method.

Resampling is meant to rebalance the weights of the particles in order to avoid degeneracy, but this introduces additional Monte Carlo variability to the particle sample. Despite systematic resampling only requiring a single uniform draw, Randal et al. (2005) show via example that it can introduce more Monte

Carlo variability than the other three resampling schemes. In Chapter 4, we discuss some advantages and disadvantages of the different resampling methods when applied to our specific model of a disease outbreak and suggest the use of stratified or residual resampling.

The frequency of resampling should be reduced to balance the loss of information due to degeneracy with the loss of information due to the additional Monte Carlo variability introduced during resampling. Typically, a measure of the nonuniformity of particle weights is used to determine if resampling should be performed at a given iteration of a particle filter. The common measures are effective sample size, coefficient of variation, and entropy. We use effective sample size (Liu et al.; 1998), a value ranging between 1 and J that can be interpreted as the number of independent particle samples. An effective sample size of J corresponds to all particle weights being equal, and a value of 1 corresponds to one particle weight being 1 with the rest 0. Using this measure of nonuniformity, we set a threshold of $0.8J$, meaning that if the number of independent samples is less than 80% of the total number of particles at time t, resampling is performed at that time.

The algorithms described in Sections 3.2.1 through 3.2.5 were constructed under the assumption that resampling is performed at every iteration of the filter. However, in practice, we omit the resampling step in each algorithm at each time point where the effective sample size exceeds $0.8J$. If resampling is not performed,

we modify the algorithm at that timepoint by 1) omitting the 'Resample' step, 2) replacing all instances of the sampled index k with the particle index j, and 3) adjusting the calculation of $\tilde{w}_{t+1}^{(j)}$ by multiplying by $w_t^{(j)}$ (in the BF, RM, and, PL) or $\tilde{g}_{t+1}^{(j)}$ (in the APF and KDPF), i.e. the particle weights get carried over. For the KDPF, RM, and PL, regeneration is not performed when resampling is not performed since, in this case, there is no reduction in the number of unique fixed parameter values. In this case, we let $\theta_{t+1}^{(j)} = \theta_t^{(j)}$ for all j.

3.4 Model comparison

Each of the particle filters described in previous sections, in addition to generating a weighted sample approximation to $p(x_t, \theta|y_{1:t})$, provide an approximation to the marginal likelihood, $p(y_{1:t})$. We first note that, given $p(y_{1:t-1})$, $p(y_{1:t})$ can be updated recursively by

$$p(y_{1:t}) = p(y_t|y_{1:t-1})p(y_{1:t-1}), \text{ for } t \geq 2. \tag{3.33}$$

Thus, $p(y_{1:t})$ can be calculated through $p(y_1)$ and the one-step ahead predictive densities $p(y_k|y_{1:k-1})$ for $k = 2, \ldots, t$, according to

$$p(y_{1:t}) = \left(\prod_{k=2}^{t} p(y_k|y_{1:k-1}) \right) p(y_1). \tag{3.34}$$

At any step of the particle filter (i.e. for any time $t \geq 1$), an approximation to $p(y_t|y_{1:t-1})$ (or $p(y_1)$ if $t = 1$) can be obtained by the following equations:

If $t = 1$,

$$p(y_1) \approx \begin{cases} \sum_{j=1}^{J} w_0^{(j)} \tilde{w}_1^{(j)}, & \text{for the BF, RM, and PL} \\ \left(\frac{1}{J}\sum_{j=1}^{J} \tilde{w}_0^{(j)}\right)\left(\sum_{j=1}^{J} \tilde{g}_1^{(j)}\right) & \text{for the APF and KDPF} \end{cases}, \text{ and} \tag{3.35}$$

if $t \geq 2$,

$$p(y_t|y_{1:t-1}) \approx \begin{cases} \sum_{j=1}^{J} w_{t-1}^{(j)} \tilde{w}_t^{(j)}, & \text{for the BF, RM, and PL} \\ \left(\frac{1}{J}\sum_{j=1}^{J} \tilde{w}_{t-1}^{(j)}\right)\left(\sum_{j=1}^{J} \tilde{g}_t^{(j)}\right) & \text{for the APF and KDPF} \end{cases}. \tag{3.36}$$

(Section 4.2 Doucet and Johansen; 2009). Given approximations to $p(y_1)$ and $p(y_k|y_{1:k-1})$ for $k = 2, \ldots, t$, the marginal likelihood can be approximated via equation (3.34).

Having prescribed a method for approximating $p(y_{1:t})$ sequentially using particle filtering, we can compare a set of N possible models $M_1, M_2, \ldots, M_N$ according to their posterior model probabiities, given by

$$p(M_i|y_{1:t}) = \frac{p(y_{1:t}|M_i)p(M_i)}{\sum_{i=1}^{N} p(y_{1:t}|M_i)p(M_i)}. \tag{3.37}$$

In Chapter 5, we compare estimated marginal likelihoods using the KDPF, RM and PL with the true marginal likelihood that can be calculated analytically under the local level DLM described in Section 2.3.1. In Chapter 6 we compare relative posterior probabilities among the models M_{101}, M_{011}, and M_{001} using PL.

3.5 Particle MCMC

At the beginning of this chapter, some of the advantages and disadvantages of both MCMC and SMC were mentioned. A significant amount of research has focused on combining aspects of both types of methods to create more efficient algorithms for sampling from high dimensional posterior distributions. One such algorithm is the RM described in Section 3.2.4 is one such example, which incorporates an MCMC algorithm within the particle filter as a way to avoid degeneracy in fixed parameter values. Particle MCMC (PMCMC) (Andrieu et al.; 2010) is another example. This method incorporates a particle filter within an iteration of an MCMC algorithm in order to increase efficiency when ideal proposal distributions are intractable.

The MCMC algorithm proposed in Section 3.1.1 for analyzing data from the the epidemic model described in 2.2 could be made more efficient by using PMCMC instead. Instead of using Gaussian random walk proposals for sampling each of $x_1, \ldots, x_T$ from their full conditional distributions, as is done in Step 2 of the Gibbs sampler in Section 3.1.1, we could instead propose a sample path $x^*_{0:T}$

from $p(x_{0:T}|\theta, y_{1:T})$ using a particle filter. The `pmcmc` function within R package `pomp` (King et al.; 2014) implements this kind of algorithm to generate samples for the fixed parameters that are asymptotically (as $J \to \infty$) distributed according to $p(\theta|y_{1:T})$ (Andrieu and Roberts; 2009).

Let $\theta^{(j)} = \left(\beta^{(i)}, \gamma^{(i)}, \nu^{(i)}\right)'$ represent the sampled values of the fixed parameters in the epidemic model at iteration i of the MCMC chain. The general algorithm implemented by `pmcmc`, called the particle marginal Metropolis-Hastings sampler (PMMH) (Section 2.4.2 Andrieu et al.; 2010), proceeds as follows:

1. Initialization:

   (a) Set initial values of the fixed parameters $\theta^{(0)}$,

   (b) Treating the fixed parameters as known $\theta^{(0)}$, run an SMC algorithm to generate an approximation to $p(x_{0:T}|y_{1:T}, \theta^{(0)})$ via

   $$\hat{p}\left(x_{0:T}|y_{1:T}, \theta^{(0)}\right) = \sum_{j=1}^{J} w_T^{(j)} \delta_{\left(x_{0:T}^{(j)}\right)}(x_{0:T}),$$

   (c) Sample $x_{0:T}^* \sim \hat{p}\left(x_{0:T}|y_{1:T}, \theta^{(0)}\right)$ and calculate an estimate of the marginal likelihood (conditional on $\theta^{(0)}$), denoted $\hat{p}\left(y_{1:T}|\theta^{(0)}\right)$, via equations (3.36) and (3.34), and

   (d) Set $i = 1$.

2. For $i \geq 1$,

   (a) Sample θ^* from some proposal distribution $q(\theta|\theta^{(i-1)})$,

(b) Treating the fixed parameters as known θ^*, run an SMC algorithm to generate an approximation to $p(x_{0:T}|y_{1:T},\theta^*)$ via

$$\hat{p}(x_{0:T}|y_{1:T},\theta^*) = \sum_{j=1}^{J} w_T^{(j)} \delta_{\left(x_{0:T}^{(j)}\right)}(x_{0:T}),$$

(c) Sample $x_{0:T}^* \sim \hat{p}(x_{0:T}|y_{1:T},\theta^*)$ and calculate an estimate of the marginal likelihood (conditional on θ^*), denoted $\hat{p}(y_{1:T}|\theta^*)$, via equations (3.36) and (3.34),

(d) With probability

$$R = \min\left(1, \frac{\hat{p}(y_{1:T}|\theta^*)p(\theta^*)}{\hat{p}\left(y_{1:T}|\theta^{(i-1)}\right)p\left(\theta^{(i-1)}\right)} \frac{q\left(\theta^{(i-1)}|\theta^*\right)}{q\left(\theta^*|\theta^{(i-1)}\right)}\right),$$

set

$$\theta^{(i)} = \theta^* \qquad \hat{p}\left(y_{1:T}|\theta^{(i)}\right) = \hat{p}(y_{1:T}|\theta^*).$$

Otherwise, set

$$\theta^{(i)} = \theta^{(i-1)} \qquad \hat{p}\left(y_{1:T}|\theta^{(i)}\right) = \hat{p}\left(y_{1:T}|\theta^{(i-1)}\right).$$

Extensions to this algorithm to provide samples for the unobserved states approximately distributed according to $p(x_{0:T}|y_{1:T})$ or joint samples for states and fixed parameters approximately distributed according to $p(x_{0:T},\theta|y_{1:T})$ have yet to be implemented in `pomp` (Section 2.4.3 Andrieu et al.; 2010). We implement this algorithm on data simulated from the epidemic model described in Section 3.1.1 using `pmcmc`, which samples fixed parameter values using Gaussian random walk proposals with prespecified standard deviations, and the algorithm uses a

plain bootstrap filter to sample states and estimate the marginal likelihood. In Chapter 4, we compare the performance of the PMCMC algorithm with that of the KDPF and standard MCMC described in Section 3.1.1.

Chapter 4

Simulation study: tracking a disease epidemic

In this chapter, we compare the performance of the BF, APF and KDPF using simulated data from the epidemic model described in Section 2.2. This data is analogous to that analyzed by Skvortsov and Ristic (2012) using the BF. In addition, using the KDPF, we compare the performance of bounded versus unbounded priors on the fixed parameters as well as different resampling schemes. Lastly, we discuss the role of the discount factor Δ when implementing the KDPF, and we compare results from running the KDPF with results from running the MCMC and PMCMC algorithms described in Sections 3.1.1 and 3.5.

In Section 4.1, we describe how data from the epidemic model described in Section 2.2 were simulated. In Section 4.2, we describe how the BF, APF, and KDPF

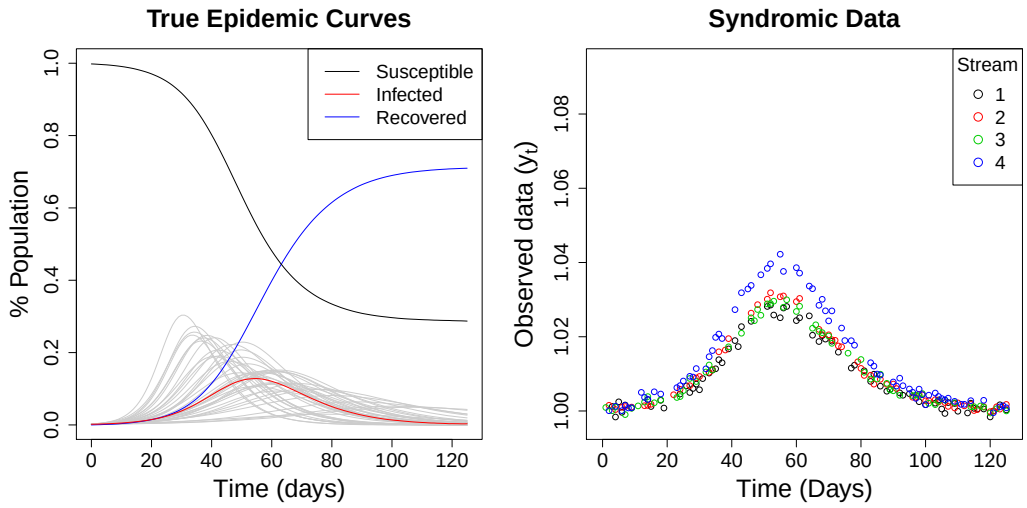
were implemented using both uniform and log-normal priors on the fixed parameters. In Section 4.3, we compare the performance of the BF, APF, and KDPF using uniform priors and systematic resampling. In Section 4.4, we compare the sensitivity to analysis using the KDPF to placing uniform versus log-normal priors on the fixed parameters. In Section 4.5, we compare the multinomial, residual, stratified, and systematic resampling schemes within the KDPF with log-normal priors on the fixed parameters.

4.1 Simulated epidemic data

Forty epidemics were simulated according to equation (2.3) for a population of size $P = 5000$ and $T = 125$ days. True values of β, γ, and ν were different for each simulated outbreak, determined by sampling from the log-normal prior distribution, $p(\theta)$, that we define in equation (4.2) in Section 4.2. For all simulations, infection was introduced in 10 people in the population at day 0 (i.e. true $i_0 = 10/5000$ and $s_0 = 4990/5000$). Among the 40 simulations, the average time at which the epidemics peak is 57 days and the average proportion of the population that has been infected by $t = 125$ is 74%. The left panel of Figure 4.1 shows the evolution of s_t, i_t, and r_t for a single simulation. The evolution of i_t for the remaining 39 simulations are superimposed (light gray).

Data from randomly selected streams at each day were generated from equation (2.6). The right panel of Figure 4.1 displays the observed data from the simulated

Figure 4.1: Simulated epidemic data



Simulated epidemic curves (left) and syndromic observations (right) for a single simulated epidemic (colored lines) with $\beta = 0.254$, $\gamma = 0.111$, and $\nu = 1.246$ along with infectious curves, i_t, for the remaining simulations (light gray).

Table 4.1: Values of known constants in epidemic model

l	b_l	ς_l	σ_l
1	0.25	1.07	0.0012
2	0.27	1.05	0.0008
3	0.23	1.01	0.0010
4	0.29	0.98	0.0011

epidemic shown on the left. Values of known constants for $L = 4$ streams were kept the same for each simulation and are given in Table 4.1 (η_l was set to 0 for all l). Values for b_l, ς_l, and σ_l were chosen to be the same as those used in the numerical study carried out by Skvortsov and Ristic (2012). The values chosen for ς_l were motivated by evidence based on real syndromic data that suggests values close to 1 (Chew and Eysenbach; 2010).

4.2 Particle filter runs

For each simulated data set, the BF, APF, and KDPF were run using $J = 100, 1000, 10000,$ and 20000 particles to obtain weighted sample approximations to $p(x_t, \theta | y_{1:t})$ for $t = 1, \ldots, T$. For each J, separate runs using multinomial, residual, stratified, and systematic resampling were implemented (Niemi; 2012), and an effective sample size threshold was set at 80% of the total number of particles to determine when to resample (Liu et al.; 1998). For the KDPF, sensitivity to changes in the discount factor, Δ, is explored by running with $\Delta = 0.9, 0.95, 0.96, 0.97, 0.98, 0.99$.

To start each particle filter run, values for the initial state and fixed parameters for J particles were sampled from the prior density $p(x_0, \theta) = p(\theta)p(i_0, s_0)$, where $p(i_0, s_0)$ is the joint pdf of the random variables s_0 and i_0 and $p(\theta)$ is the joint prior density of β, γ, and ν. We let $i_0 \sim \mathrm{N}_{[0,1]}(0.002, 0.0005^2)$ and set $s_0 = 1 - i_0$, as in equation (3.1). This is motivated by the fact that a very small percentage of the population is infected during the initial stage of an epidemic, and no infected individuals have yet recovered from illness.

To investigate the impact of different prior distributions for θ on the performance of the particle filters, the runs described above were performed once using uniform priors on θ and then again using log-normal priors. We first ran the particle filters using uniform priors on θ that were chosen to be the same as those

used in Skvortsov and Ristic (2012), i.e. $p(\theta) = p(\beta)p(\gamma)p(\nu)$ with

$$\begin{aligned} \beta &\sim \text{Unif}(0.14, 0.50) \\ \gamma &\sim \text{Unif}(0.09, 0.143) \\ \nu &\sim \text{Unif}(0.95, 1.3). \end{aligned} \tag{4.1}$$

These priors allow for values of R_0 in a range of approximately 1 to 5.5 and an average infectious period in a range of roughly 7 to 11 days. R_0 values for strains of influenza have been estimated to be around 2-3 (Mills et al.; 2004; Heffernan et al.; 2005; Zhang; 2011). Thus, while these priors impose restrictive bounds on the parameters, they are not particularly informative for tracking a flu epidemic.

We then ran the particle filters using our own log-normal priors on θ that we define by $p(\theta) = p(\beta, \gamma)p(\nu)$ (i.e. β and γ are not independent). When implementing the particle filtering algorithms, the prior draws for β were determined by multiplying sampled values of γ by the basic reproductive rate $R_0 = \beta/\gamma$. All parameters were sampled independently with priors

$$\begin{aligned} R_0 &\sim \text{LN}(0.7520, 0.1768^2) \\ \gamma &\sim \text{LN}(-2.1764, 0.1183^2) \\ \nu &\sim \text{LN}(0.1055, 0.0800^2). \end{aligned} \tag{4.2}$$

Here, we incorporate prior information on R_0 instead of β directly, since prior knowledge of the basic reproductive number may be easier to obtain than for the contact rate itself. These log-normal priors constrain β, γ and ν to be positive.

The mean and variance of the prior distributions on $\log \gamma$ and $\log \nu$ were chosen such that random draws of γ and ν would fall within the bounds of their respective uniform priors (in equation (4.1)) with 95% probability. The mean and variance of $\log R_0$ were chosen such that R_0 would fall between 1.5 and 3 with 95% probability.

It is important to note that particle filters do not perform well when diffuse priors are placed on unknown states or fixed parameters. This is because priors that are too vague yield a small number of prior draws sampled in areas of high likelihood, resulting in degeneracy of the particle filter after only a few time points. We discuss this challenge in more detail in Section 4.9.

Logit and log transformations were applied to the components of θ in the manner described at the end of Section 3.2.3 so that the normal mixture kernel could be used in the KDPF while constraining β, γ, and ν to be within their respective prior domains (i.e. logit was used with uniform priors and log with log-normal priors).

4.3 Comparison of particle filter algorithms under uniform priors

First, we compare the performance of the particle filters using uniform priors on the elements of θ and systematic resampling, since these priors and resampling scheme were used in Skvortsov and Ristic (2012). For ease of comparison, the

same prior draws were used in each particle filter for fixed J. Figure 4.2 shows 95% credible bounds of $p(\beta|y_{1:t})$, $p(\gamma|y_{1:t})$, and $p(\nu|y_{1:t})$ for $t = 1, 2, \ldots, T$ and $J = 100, 1000, 10000, 20000$ using the simulated data displayed in Figure 4.1. Initially, the credible bounds for the BF and APF match those of the KDPF, but quickly degenerate toward a single value due to elimination of unique particles during resampling. Although the time of degeneracy increases as J gets larger, the BF and APF bounds become misleading during the second half of the epidemic, even for $J = 20000$. The bounds for the KDPF, on the other hand, have dramatically reduced degeneracy since new values of θ are regenerated from the kernel density approximation.

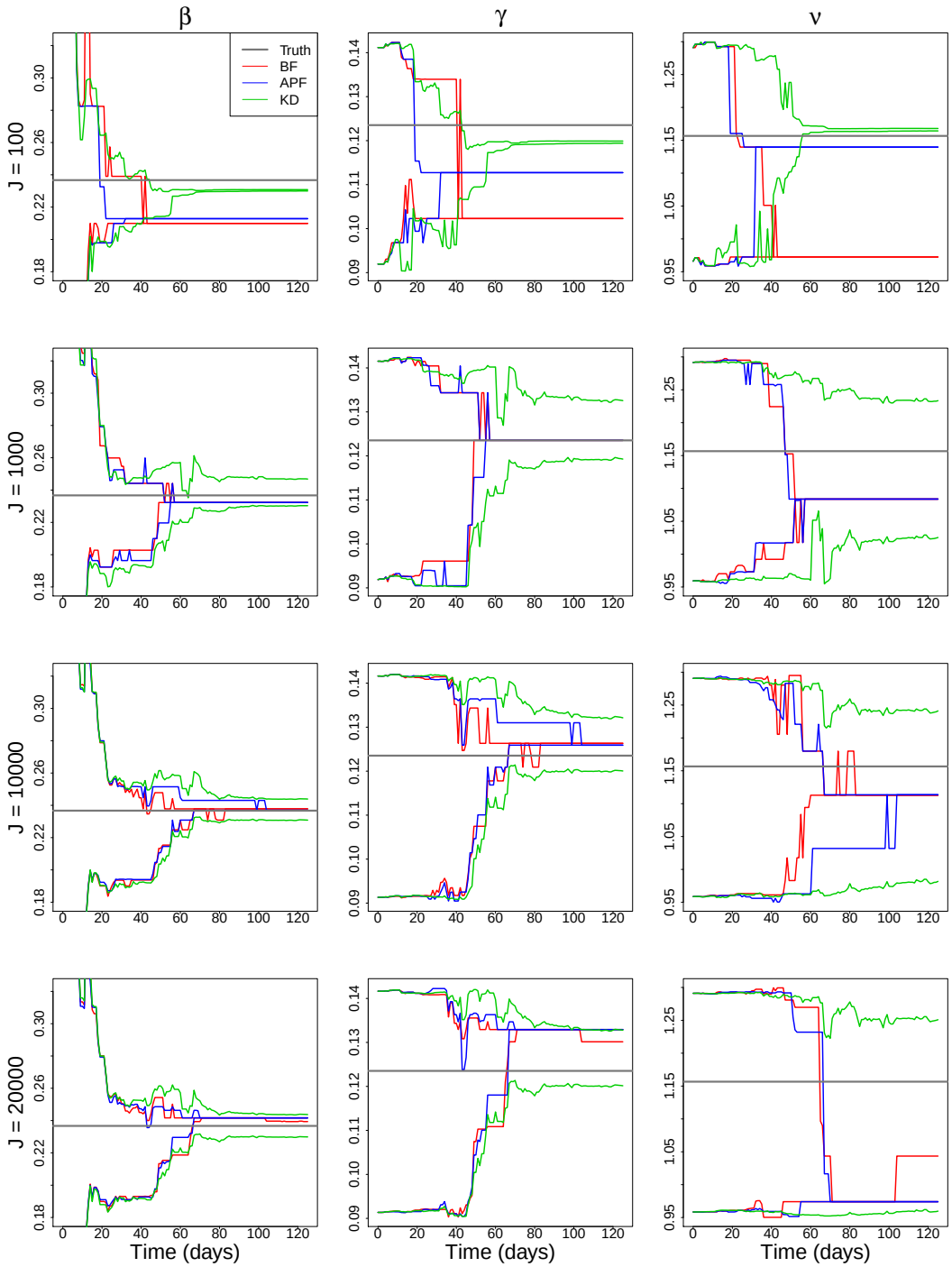
The KDPF also has an advantage over the BF and APF in terms of computational efficiency. Notice that the bounds for the KDPF become wider as J increases, but they do not change much between $J = 10000$ and $J = 20000$. This suggests that by 10000 particles, the weighted sample approximation of $p(x_t, \theta|y_{1:t})$ has converged to the true posterior over the entire epidemic period, unlike with the BF and APF. Even though the bounds for β for the BF and APF seem to roughly match those of the KDPF for $J = 20000$ over the first half of the epidemic, the KDPF provides the same measure of uncertainty for $J = 10000$ and does not degenerate during the second half of the epidemic.

Estimation of ν is more challenging than for β or γ because of the nonlinear nature of the state equation with respect to this parameter. We notice from the

plots for ν with $J \geq 10000$ that very little information is gained over the course of the epidemic about this parameter relative to its uniform prior. Furthermore, the 95% credible intervals for ν expand between $t = 70$ and $t = 80$, while we typically expect to see the width of credible intervals for an unknown fixed parameter decrease monotonically as data is accumulated. A plausible explanation here is that $p(\nu|y_{1:t})$ has been squeezed against the lower bound of the prior for ν. Rerunning the analysis (results not shown) using our log-normal priors relaxes the prior bounds on ν and shows a shift in the distribution toward higher values around $t = 70$ as opposed to the widening of the interval that we see in Figure 4.2.

Table 4.2 shows that the behavior of the BF, APF, and KDPF illustrated in Figure 4.2 is consistent across the 40 simulations. For instance, for $J = 20000$, the 95% credible intervals at $t = 125$ for each of β, γ and ν cover the truth for 39 out of 40 (97.5%) simulations using the KDPF. The BF and APF runs using the same number of particles, on the other hand, yield 95% credible intervals at $t = 125$ that cover the truth for no more than 13 out of 40 (32.5%) simulations when considering β, γ, and ν marginally.

Figure 4.2: Comparing credible intervals for the BF, APF, and KDPF



Sequential 95% credible intervals for β (left column), γ (middle column), and ν (right column) for increasing number of particles (rows) for the BF (red), APF (blue), and KDPF (green), compared with the truth (black lines), when using systematic resampling and uniform priors. Data were generated from the simulated epidemic shown in Figure 4.1. For the KDPF, Δ was set to 0.99.

Table 4.2: Comparing credible intervals for the BF, APF, and KDPF

	β			γ			ν		
J	BF	APF	KDPF	BF	APF	KDPF	BF	APF	KDPF
100	0.000	0.000	0.175	0.000	0.000	0.175	0.000	0.000	0.100
1000	0.000	0.000	0.800	0.000	0.000	0.900	0.000	0.000	0.800
10000	0.150	0.150	0.975	0.150	0.200	0.950	0.175	0.250	0.925
20000	0.325	0.275	0.975	0.325	0.175	0.975	0.300	0.175	0.975

Proportion of simulated data sets (out of 40 total) for which 95% credible intervals obtained from the marginal filtered distributions of the fixed parameters (columns) at the end of the epidemic (i.e. $t = 125$) cover the true value used for simulation for increasing number of particles (rows) using the BF, APF, and KDPF.

4.4 Illustration of the negative impact of priors with truncated support

As mentioned in Section 3.2.3, implementing the KDPF using a normal kernel density approximation to $p(\theta|y_{1:t})$ to regenerate the fixed parameters requires applying some transformation to the components of θ so that their support is on the real line. The logit function is a convenient choice for mapping fixed parameters with bounded support to the real line; the log function is convenient for fixed parameters with positive support. Thus, we investigate the sensitivity of the results with $J = 10000$ to these two types of transformations using the KDPF.

Figure 4.3 compares scatterplots of β versus γ sampled jointly from the filtered distribution $p(\beta, \gamma|y_{1:t})$ at $t = 0, 20, 40, 60$. In the top row, the same truncated support prior as in Skvortsov and Ristic (2012) is used, i.e. $\beta \sim \text{Unif}(0.14, 0.50)$ and $\gamma \sim \text{Unif}(0.09, 0.143)$ independently. In order to ensure regeneration in the

KDPF does not extend past these bounds, a logit transformation was applied in the manner described at the end of Section 3.2.3 with $a = 0.14$ and $b = 0.50$ for β and $a = 0.09$ and $b = 0.143$ for γ. The kernel density is then created on this transformed space. In the top row of Figure 4.3, the samples concentrate on the boundaries of the uniform prior on γ, particularly for $t = 20$ and $t = 40$. This suggests that the truncated support prior bounds on γ are too restrictive to account for the uncertainty in this recovery time.

To test this hypothesis, we reran the KDPF using exactly the same prior draws as those used in the first row of Figure 4.3, but we apply a log transformation to θ (to ensure β, γ and ν are positive) instead of the logit transformation. The results are shown in the second row of Figure 4.3. Despite the particles starting within the uniform bounds at $t = 0$, the samples stray outside the uniform bounds for γ, suggesting that the data are informing us that reasonable parameter values can be found outside the bounds that would have been imposed by the truncated support prior.

The bottom row of Figure 4.3 displays results from running the KDPF using prior samples taken from the log-normal priors on the elements of θ described be equation (4.2). As in the second row, a log transformations were applied to the fixed parameters so that the normal kernel density could be used for jittering particles. The log-normal priors on the fixed parameters are more informative than the uniform priors in the sense that a greater number of particles are concentrated

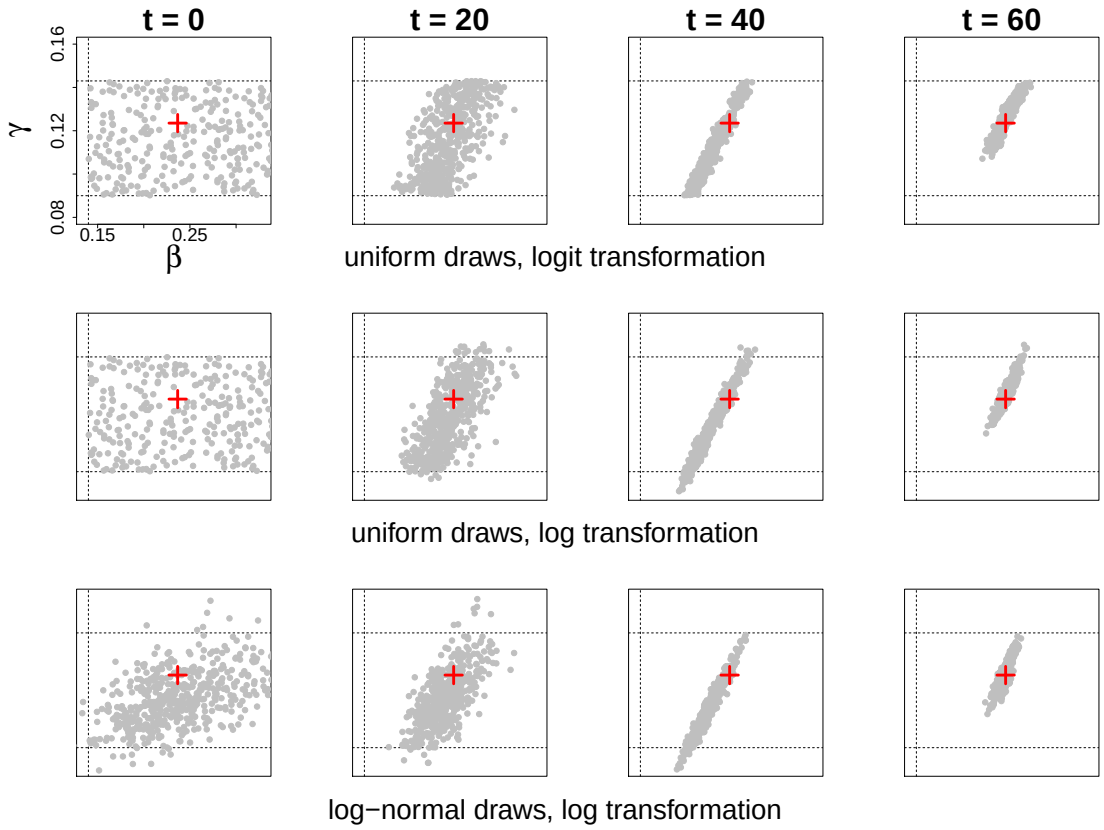
near the true values of β and γ at $t = 0$. Yet, the distribution of particles at $t = 20$ and $t = 40$ appear more spread out in the bottom row than in the top row because the log-normal priors are less restrictive on the sample space of β and γ than are the truncated support uniform priors.

In the bottom row of Figure 4.3, sampled particle values at $t = 60$ have moved inside the bounds that would have been imposed by the uniform prior and form an ellipse-shaped distribution similar to what is shown in the second row of Figure 4.3 at $t = 60$. This suggests that the tail of points concentrated along the upper uniform bound at $t = 60$ in the top row of Figure 4.3 is an artifact of the over-restrictive uniform prior and not influenced by the data. We suggest using log-normal priors on positive elements of θ as opposed to uniform priors which bound the range of possible parameter values. This allows us to use prior knowledge of the epidemic to encourage points to lie in a reasonable range while retaining flexibility in the event of model mis-specification either in the likelihood or the prior.

4.5 Comparison of resampling schemes

As mentioned in Section 3.3, resampling is meant to rebalance the weights of the particles in order to avoid degeneracy, but this comes at the cost of increasing the Monte Carlo variability of the particle sample. Up to this point, we have used only systematic resampling, as in Skvortsov and Ristic (2012). Alternatively, we

Figure 4.3: Comparing priors in the KDPF



Scatterplots of β (horizontal) versus γ (vertical) with true values (red crosses) at $t = 0, 20, 40, 60$ days using the KDPF with $J = 10000$ particles, systematic resampling, and $\Delta = 0.99$. The logit transformation (top row) on θ shifted and scaled to $(0, 1)$ and log transformation (second row and bottom row) were used before regenerating the fixed parameters. To aid comparison, the same uniform draws of θ were sampled at $t = 0$ in each of the first two rows. Log-normal prior draws were sampled in the bottom row. For demonstration, each panel shows 500 particles sampled from the weighted sample approximation to $p(x_t, \theta|y_{1:t})$. Axes are the same in each panel. Dashed horizontal and vertical lines indicate the bounds of the uniform priors on γ and β, respectively. The upper bound on the uniform prior on β is not shown because it lies outside the range of the horizontal axis.

could have chosen multinomial, residual, or stratified resampling. Randal et al. (2005) explains each of these methods in detail and shows that 1) multinomial resampling introduces more Monte Carlo variability than do residual or stratified resampling, 2) residual and stratified resampling introduce the same amount of Monte Carlo variability, on average, and 3) systematic resampling can introduce more Monte Carlo variability than does multinomial resampling.

With this in mind, we turn to a comparison of different techniques for the resampling step using the KDPF with log-normal priors on the fixed parameters and $\Delta = 0.99$. To aid in comparison of the different resampling techniques, the same prior draws were used in all particle filter runs for fixed J. We would like to choose the resampling scheme for which the filtered distribution, $p(x_t, \theta | y_{1:t})$, approaches the true posterior the fastest as a function of the number of particles. If the filtered distributions have converged to the true posterior, then 95% credible intervals should cover the true parameter value about 95% of the time.

Using $J = 100$ particles (not shown), sequential 95% credible intervals over the second half of the epidemic for each of β, γ, and ν cover the true parameter value for less than half of the 40 simulated data sets, indicating that more particles are needed to approximate the true posterior. Figure 4.4 shows that coverage probabilities approach the nominal level for all four resampling techniques as J increases. Multinomial resampling, however, appears to be outperformed by the other three resampling techniques, as coverage for all three model parameters

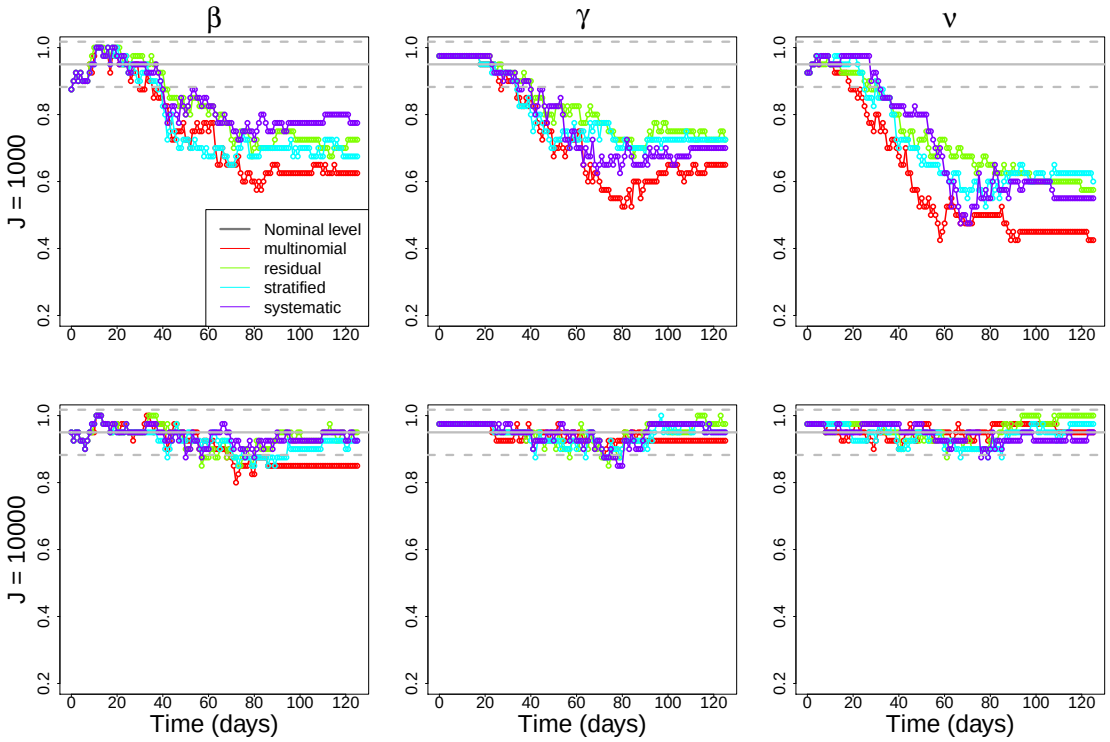
using $J = 1000$ particles dips lower during the second half of the epidemic for multinomial resampling than it does for any of the other three methods. This is also true for β with $J = 10000$ particles. By increasing the number of particles to $J = 20000$ (not shown), all four resampling methods yield coverage probabilities for each parameter that remain within the 95% confidence bounds around the nominal coverage level throughout the epidemic.

Although residual, stratified, and systematic resampling perform about the same with this specific model, we prefer to use either residual or stratified resampling because of an example shown in Randal et al. (2005) where systematic resampling adds more Monte Carlo variability than any of the other three resampling schemes.

4.6 Discount factor

We recommend, based on the results from Sections 4.3, 4.4, and 4.5, that the KDPF with residual or stratified resampling and prior distributions bounded only by the support of the parameters be used in preference to other choices mentioned. With this implementation of the particle filter, the practitioner is still left to choose a value for the discount factor, Δ. As mentioned in Section 3.2.3, Δ is a tuning parameter that determines the smoothness of the kernel density approximation to $p(\theta|y_{1:t})$ when implementing the KDPF. Choosing Δ close to 0 results in a smoother approximation and more substantial jittering of particles

Figure 4.4: Comparing resampling schemes in the KDPF



Proportion of the 40 simulated data sets for which 95% credible intervals for β (left), γ (middle), and ν (right) cover the true value used for simulation for different t (x-axis) and J (rows) using the KDPF with log-normal priors on θ and $\Delta = 0.99$. Solid gray horizontal line denotes nominal coverage (95%) and dashed lines give 95% confidence bounds around the true coverage.

while Δ close to 1 leads to a choppier approximation and more subtle jittering of particles.

To test the sensitivity of the KDPF to different values of Δ, we ran the KDPF with log-normal priors and stratified resampling on each of the 40 simulated data sets using different values of Δ (0.9, 0.95, 0.96, 0.97, 0.98, and 0.99) and J (100, 1000, 10000, and 20000). We then calculated 95% credible intervals for each of the unknown parameters. The results (not shown) indicate that lower values of Δ lead to a higher proportion of 95% credible intervals covering the truth, but that coverage probabilities for all values of Δ are close to the nominal level for all t if $J \geq 10000$. We use $\Delta = 0.99$ because we seek an implementation of the KDPF that works well when enough particles are used to provide a good approximation to the true posterior. Liu and West (2001) recommends choosing a value between 0.95 and 0.99.

4.7 Comparison with MCMC

For a comparison with our KDPF results, MCMC analyses were run using both the Gibbs sampling algorithm described in Section 3.1.1 and the PMCMC approach of Andrieu et al. (2010) described in Section 3.5. The Gibbs sampler for the standard MCMC was implemented by running three chains for 10,100,000 iterations each, and each with a burn-in period of 100000 iterations. The final sample was thinned by taking every 1000th iteration. Chains were initialized by

drawing values for the fixed parameters from the prior $p(\theta)$. Initial values of the states, $x_{0:T}^{(0)}$, were sampled by 1) drawing $x_0^{(0)}$ from $p(x_0)$, 2) drawing θ_t^* from $p(\theta)$ for $t = 1, \ldots, T$, and 3) drawing $x_t^{(0)}$ from $p(x_t|x_{t-1}^{(0)}, \theta_t^*)$ for $t = 1, \ldots, T$.

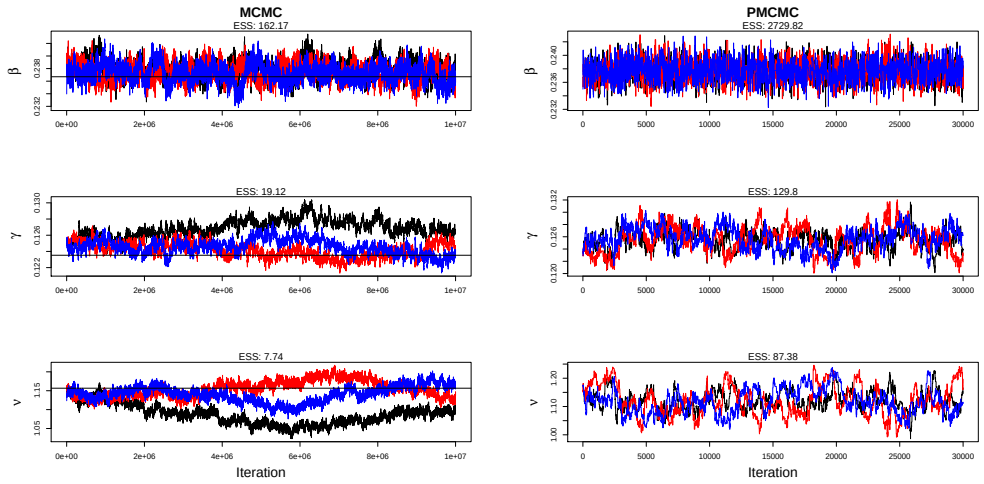
Initial values of the tuning parameters τ_β, τ_γ, τ_ν (i.e. the standard deviations of the Gaussian random-walk proposal distributions for each fixed parameter) were set to 0.01, 0.001, and 0.01, respectively. Initial values of τ_{x_t} for $t = 1, \ldots, T$ (the tuning parameters for the joint draws of the epidemic states s_t and i_t), were set to 0.001 for all t. These tuning parameters were adjusted during the burn-in period, as described in Section 3.1.1, by multiplying by 1.1 if a proposed sample was accepted and dividing by 1.1 if a proposed sample was rejected. As detailed in the Gibbs sampler described in Section 3.1.1, proposed values for $x_t = (s_t, i_t)'$ were jointly accepted or rejected depending on the value of the Metropolis ratio, while proposed values for each of β, γ, and ν were accepted or rejected marginally depending on the values of their respective Metropolis ratios.

A PMCMC algorithm was implemented, as described in Section 3.5, using the `pmcmc` function within the R package `pomp` (King et al.; 2014). Samples from the posterior distribution of the fixed parameters were generated conditional on the first T observations from the simulated data set pictured in Figure 4.1 for $T = 5, 10, 15, \ldots, 125$. For each T, three chains consisting of 30000 PMCMC iterations were generated and 95% credible intervals were calculated based on each chain. PMCMC chains were initialized at the true values of the fixed parameters used

for simulating the data. Tuning parameters representing the standard deviations of the random-walk proposal distributions of β, γ, and ν were set to 0.005, 0.001, and 0.01, respectively. These values were chosen because they allowed the chains to mix well within reasonable computing time, but it is possible that different values could provide better mixing and hence improved estimates of the fixed parameters within the same computing time. For more information on PMCMC and using functions within the `pomp` package, we refer the reader to Andrieu et al. (2010) and King et al. (2014).

Figure 4.5 compares the efficiency of the standard MCMC and PMCMC algorithms run on the entire data set ($T = 125$) using traceplots and effective sample size calculations for the fixed parameters. Traceplots of multiple chains for each parameter show how well the chains mixed and can indicate whether the entire sample space of the posterior was adequately explored. Effective sample size (ESS), which we calculate using R package `coda`, gives an estimate of the number of independent MCMC samples by adjusting for the autocorrelation present in the total sample (Plummer; 2005). Despite being run for over 10 million iterations (which took about two weeks), the extremely low values of ESS and poor mixing of the chains, particularly for γ and ν, for the standard MCMC on the left of Figure 4.5, relative to PMCMC on the right, suggest that PMCMC is much better suited for analyzing data from this particular epidemic model.

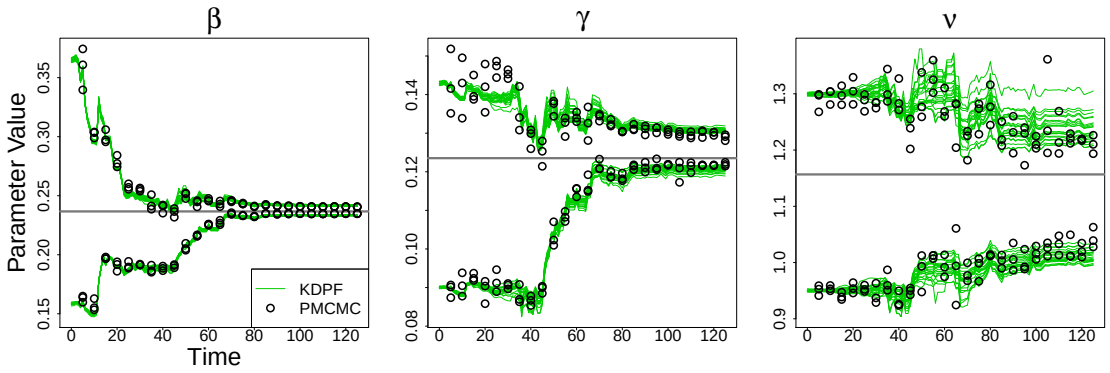
Figure 4.5: Traceplots comparing the MCMC versus PMCMC



Traceplots and effective sample sizes of three different MCMC chains of 10,000,000 iterations each for the standard MCMC (left) and 30000 iterations each for the PMCMC (right). Both MCMC algorithms analyze data for the entire epidemic period ($T = 125$). Only every 1000th iteration is plotted for the MCMC on the left.

Figure 4.6 compares the performance of the KDPF with PMCMC in terms of marginal 95% credible intervals for the fixed parameters. The intervals obtained from PMCMC samples for each chain at each T are compared with those produced by 20 separate runs of the KDPF on the same data set. The KDPF was run using 20000 particles, log-normal priors on the components of θ, stratified resampling, and $\Delta = 0.99$. Multiple runs of the KDPF and PMCMC on the same data allow us to assess the uncertainty in the 95% credible intervals for the filtered distributions at each T. For instance, the high variance of the interval estimates for ν in the rightmost panel of the figure demonstrates the challenge in estimating this parameter. The performance of the KDPF compares well with PMCMC in this study, as the bounds of the 95% credible intervals obtained from

Figure 4.6: Comparing the KDPF versus PMCMC



Sequential 95% credible intervals for β (left), γ (middle), and ν (right) obtained from 20 different runs of the KDPF (green lines) using $J = 20000$ particles, log-normal priors, stratified resampling, and $\Delta = 0.99$ compared with 95% credible intervals obtained from 3 different PMCMC chains (black circles) run for 30000 iterations on data collected up until day T for $T = 5, 10, \ldots, 125$. All KDPF and PMCMC runs used observations taken from the same simulated data set pictured in Figure 4.1.

the PMCMC chains over the course of the epidemic are more variable than the bounds obtained from the separate KDPF runs, particularly for γ prior to day $T = 30$. Furthermore, a single PMCMC chain run on the full data set (i.e. for $T = 125$) took 8-9 hours to complete while the KDPF with $J = 20000$ particles provided results for all time points in about 15 minutes in our study. Section 4.9 provides further discussion of different scenarios where either PMCMC or the KDPF might be preferred.

4.8 Additional Unknown Parameters

Within SMC approaches, an advantage of using a more computationally efficient algorithm is to allow reduced model assumptions. We therefore turn our fo-

cus to extending the analysis using the KDPF to include $\{b_l, \varsigma_l, \sigma_l, \eta_l : l \in 1, \ldots, L\}$ as unknown parameters, thereby increasing the number of unknown parameters beyond those considered by Skvortsov and Ristic (2012) using the BF. For this section, we consider data coming from only one stream ($L = 1$) and let $\theta = (\beta, \gamma, \nu, b, \varsigma, \sigma, \eta)'$, dropping the subscript l. Keeping the same simulated evolution of the true epidemic as shown in the left panel of Figure 4.1, a single stream of syndromic observations, y_t for $t = 1, \ldots, T$, was simulated from equation (2.6) with true values of b, ς, σ, and η set to 0.258, 1.028, 0.000737 and 2.346, respectively. Days at which data were observed from the single stream were randomly selected.

The KDPF with tuning parameter Δ set to 0.99 was run with $J = 60000$ particles and stratified resampling was used with an effective sample size threshold of $0.8J$. As before, fixed parameter values were regenerated only when resampling was performed. Initial particles for states and parameters were sampled from their prior with $p(x_0)p(\theta) = p(i_0, s_0)p(\beta, \gamma)p(\nu)p(b)p(\varsigma)p(\sigma)p(\eta)$. The prior for the state and log-normal priors for R_0, γ, and ν are the same as those defined in Section

4.2. The priors for b, ς, σ, and η are

$$
\begin{aligned}
b &\sim \mathrm{LN}(-1.6090, 0.3536^2) \\
\varsigma &\sim \mathrm{LN}(-0.0114, 0.0771^2) \\
\sigma &\sim \mathrm{LN}(-7.0516, 0.2803^2) \\
\eta &\sim N(2.5, 1)
\end{aligned}
\tag{4.3}
$$

independently. The choice of prior mean and standard deviation on the log scale were made such that random draws of b, ς, and σ on the original scale would be within $(0.1, 0.4)$, $(0.85, 1.15)$, and $(0.0005, 0.0015)$, respectively, with 95% probability. To assess the loss in precision of our estimates due to incorporating additional unknown parameters into our analysis, we compared with results from running the KDPF using 60000 particles with b, ς, σ, and η assumed known at their true values used for simulating the data (we refer to the run with b, ς, σ, and η assumed known as the initial analysis).

Figure 4.7 shows sequential 95% credible intervals for both the extended (blue lines) and the initial (red lines) analyses. Most noticeable from Figure 4.7 is that the intervals for β, γ, ν, s_t, and i_t are wider for the extended analysis than they are for the initial. This is due to the added uncertainty in b, ς, σ, and η in the extended analysis. Nonetheless, we are still able to obtain credible intervals for the unknown parameters that cover the true values for this simulated data set, as well as intervals for the states that cover the true epidemic curves, using a higher

number of particles ($J = 60000$) than was used in the initial KDPF analysis in prior sections.

In Figure 4.7, the lines appear choppy or block-like. This results from data coming from only one stream, leading to more time points where no data are available and making the analysis more sensitive to abnormal data. Gaps in the data lead to a lack of resampling of particles and cause more drastic shifts in the filtered distribution once data arrive. For instance, we notice a spike in the s_t curve right after $t = 40$ because of a gap in the data and a shift in the trajectory of data points near the epidemic peak.

Lastly, we comment on a widening of the credible intervals for ν. This phenomenon suggests that the log-normal priors used on ν are too restrictive, and that our model provides even less insight about this parameter than our prior belief. Scarce knowledge about ν is gained over the course of the epidemic in the initial analysis due to the nonlinear nature of the evolution equation with respect to ν, and we in fact lose information about ν in the extended analysis relative to our specified prior. While the extended analysis could be rerun with a different prior, we present this specific analysis to illustrate the sensitivity of the filtered distribution of ν to assumptions about other parameters. The improved efficiency of the KDPF provides insight into this sensitivity (within reasonable computing time) in the absence of assumptions that were made about fixed parameters in

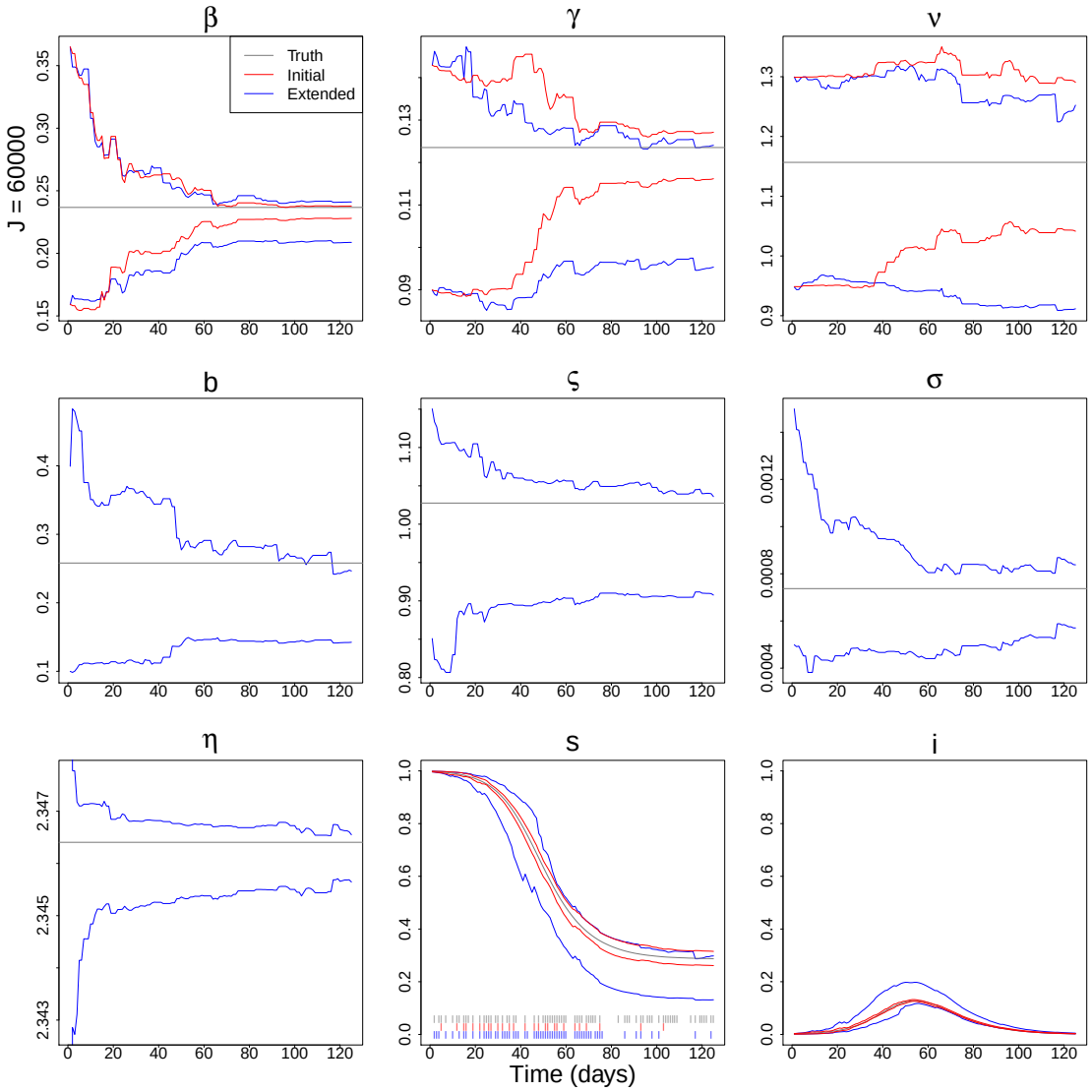
the observation equation in both our initial analysis and in the prior BF analysis by Skvortsov and Ristic (2012).

4.9 Discussion

Presented in this chapter is a strategy for simultaneous estimation of the current outbreak state and fixed parameters related to disease transmission using syndromic data. We describe a stochastic epidemiological compartment model of a disease outbreak for data from syndromic surveillance that could possibly be multivariate and have any pattern of missingness. We suggest the use of the kernel density particle filter (Liu and West; 2001) using priors on fixed parameters that are bounded only by their support. We suggest the use of stratified or residual resampling when effective sample size has dropped markedly and that regeneration of fixed parameter values should only occur when resampling is performed. We showed how this approach is capable of estimating a model with additional unknown fixed parameters.

Advanced techniques exist that are better than the KDPF at fighting particle degeneracy, but require more practitioner input. For example, particle degeneracy could be combated within an SMC algorithm by incorporating a MCMC step to refresh fixed parameter values (Gilks and Berzuini; 2001; Storvik; 2002), e.g. the resample-move algorithm described in Section 3.2.4. However, this would require the practitioner to define an MCMC algorithm in addition to the SMC

Figure 4.7: Analyzing epidemic model with additional unknown parameters



Sequential 95% credible intervals for the states and fixed parameters from the original (red) and extended (blue) analyses where the KDPF with $J = 60000$ particles was run with stratified resampling and $\Delta = 0.99$. Tick marks are shown along the bottom of the plot for s_t at time points when data were observed (dark gray) and when particles were resampled (blue and red for the extended and original analyses, respectively).

algorithm. In addition to this requirement, the algorithm would no longer be truly sequential as the computational effort would increase with time. Alternatively, if the practitioner is willing to modify their model, they can take advantage of a sufficient statistic structure (Fearnhead; 2002), Rao-Blackwellization (Doucet et al.; 2000), or both (Carvalho et al.; 2010), as in in the particle learning algorithm described in Section 3.2.5. Possible modifications to the model in Section 2.2.1 to allow alternative strategies include setting $\nu = 1$, removing fixed parameters from Q, and eliminating the truncation in equation (2.2).

The KDPF provides a sequential inferential strategy that is easy to implement, applies to a very broad class of models, and reduces particle degeneracy when applied to models with unknown fixed parameters. However, along with its methodological strengths, the algorithm has weaknesses that are reflective of all SMC methods in general. For instance, SMC methods do not perform well in high-dimensional parameter space. In addition, while particle filters perform well when run over fixed-length time intervals, they eventually degenerate if run over long periods of time due to the accumulation of approximation errors. Lastly, as mentioned in Section 4.2, all SMC methods suffer from degeneracy if vague priors are used. A common solution to this problem is to first run an MCMC based on the first few data points to find a reasonable particle cloud from which to draw prior samples (Chap 5, Petris et al.; 2009).

In high-dimensional settings, PMCMC methods provide better estimates of unknown states and fixed parameters by using SMC methods to construct efficient proposal distributions for a joint sample of all dynamic states (Andrieu et al.; 2010). Since it is likely that more complicated models than what we presented in this thesis may be required for monitoring disease outbreaks in real-life situations (Shaman and Karspeck; 2012; Bhadra et al.; 2011), PMCMC may offer a better solution in certain situations such as when x_t and θ are high-dimensional. However, PMCMC is a non-sequential method and only valuable for on-line analysis provided the computation time required is not too burdensome. A sequential analysis could be more valuable for processing data collected at shorter time intervals when an immediate decision regarding an intervention policy is needed (Merl et al.; 2009a; Ludkovski and Niemi; 2010; Dukic et al.; 2012). In addition, efficient comparison of competing models for an epidemic outbreak (Bhadra et al.; 2011) could be made more feasible by running an SMC algorithm that could assess the fit of the data to multiple models more quickly. SMC methods can also provide an approximation to the marginal likelihood of the data if formal model comparison or model averaging is desired (Doucet and Johansen; 2009; Zhou et al.; 2013). We believe both the KDPF and PMCMC are valuable tools available to the practitioner.

In this chapter, we outline a strategy for real time tracking of a disease epidemic using data from syndromic surveillance, but this strategy can be applied

to many other fields requiring on-line data analysis. We present improved particle filtering methods in general within the framework of sequential estimation of states and unknown fixed parameters in state-space models to inspire future work in epidemiological modeling and other scientific areas as well.

Chapter 5

Simulation study: SMC model comparison of local level DLMs

In Chapter 4, we illustrated the improved performance of the KDPF over the BF and APF within the context of tracking an epidemic using a model that contains unknown fixed parameters. In this chapter, we compare the KDPF with more advanced strategies, specifically the RM and PL. In particular, we focus on how efficiently these algorithms perform in terms of estimating the marginal likelihood and comparing possible data-generating models. To evaluate the relative performance of the particle filters, we simulate data from the local level DLM with common observation and state variance factor described in Section 2.3.1, since this model leads to an analytically tractable form of the marginal likelihood of the data.

This short chapter consists of three sections. Section 5.1 describes the simulated data set, as well as estimation of states and the unknown common precision factor (i.e. the inverse of the unknown common variance factor) using analytical forms of their respective marginal filtered distributions. Section 5.2 describes estimation of states and the unknown common precision factor using the KDPF, RM, and PL. Finally, in Section 5.3, we compare the efficiency of the KPDF, RM, and PL when estimating the marginal likelihood of the data as well as relative posterior probabilities among these local level DLMs with varying signal-to-noise ratios in equations (2.10) and (2.11).

5.1 Simulated data and analytical forms for estimation

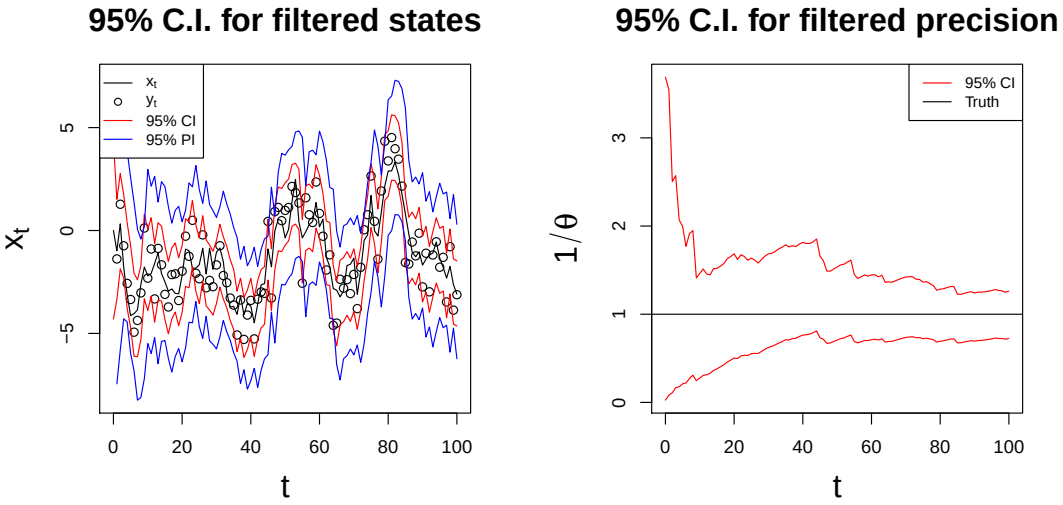
Consider the local level DLM with unknown common state and observation variance factor, θ, and known signal-to-noise ratio, λ, given by equations (2.10) and (2.11), namely

$$y_t \sim \mathrm{N}(x_t, \theta)$$

$$x_t \sim \mathrm{N}(x_{t-1}, \theta\lambda).$$

A time series of length $T = 100$ was simulated from this model with true $\theta = \lambda = 1$. The data, y_t, and true unobserved states x_t, for $t = 1, \ldots, T$, are shown by black dots and lines, respectively, in the left panel of Figure 5.1.

Figure 5.1: Simulated data and analytical estimates for local level DLM



True observed data (black dots) and unobserved states (black lines) for data simulated from the local level DLM from equations (2.10) and (2.11) with $\theta = \lambda = 1$, along with marginal 95% credible intervals (red lines) for the states (left) and precision (right), as well as 95% one-step ahead prediction intervals (blue lines) for the data (left).

We assume the prior distribution $p(x_0, \theta)$ from equation (2.12) with $a_0 = b_0 = 1$. The red and blue lines in the left panel of Figure 5.1 show 95% credible intervals, at each time t, for the marginal filtered distribution of x_t, $p(x_t|y_{1:t})$, and the one-step ahead predictive distribution for y_t, $p(y_t|y_{1:t-1})$, respectively. We can obtain these intervals without need for running a particle filter because these distributions can be calculated analytically according to equations (2.33) and (2.34). Similarly, we can calculate sequential 95% credible intervals for the unknown common state and observation precision factor, $1/\theta$, using the analytical form of the marginal filtered distribution $p(1/\theta|y_{1:t})$, which is known to be a gamma with shape a_t and rate b_t (CI's displayed in right panel of Figure 5.1).

These shapes and rates are calculated recursively according to equation (2.32). We estimate the common precision factor instead of the common variance factor because quantiles of the gamma distribution are easier to obtain than quantiles of the inverse-gamma distribution.

In practice, particle filters are not needed to analyze data from this model since analytical forms for the filtered distributions of states and unknown parameter are available. However, this model is carefully chosen since it is this availability of analytical distributions (which we refer to as the "true posterior" or "true filtered distribution") that provides an exact benchmark for assessing the performance of the KDPF, RM, and PL in the remainder of this chapter. Specifically, we search for the algorithm which yields the best approximation to the true posterior for a fixed number of particles.

5.2 Estimation using particle filters

The KDPF, RM, and PL were each run twenty times on the simulated data set using $J = 100, 500, 1000,$ and 5000 particles. For each particle filter run, the data were assumed to be generated from the local level DLM described by equations (2.10) and (2.11) with $\lambda = 1$ (i.e. the true model from which the data were simulated). Resampling was performed in each algorithm at time points where the effective sample size dropped below $0.8J$ (this is the same resampling threshold used in Chapter 4). For the KDPF, the discount factor Δ was set to

0.99, and the RM and PL were implemented as described in Sections 3.2.4 and 3.2.5. Each algorithm was run assuming a prior of the form given in (2.12) with $a_0 = b_0 = 1$.

Sequential 95% credible intervals for the marginal filtered distributions of x_t and $1/\theta$ for each J are displayed in Figure 5.2. Compared with the 95% credible intervals of the true filtered distribution of the states, all algorithms seem to perform well, yielding credible intervals in line with the truth for $J > 100$. The KDPF, however, is outperformed by the other two algorithms, with credible intervals for the filtered precision that are inaccurate for $J < 1000$ and exhibit wide variability around the true upper bounds of the intervals for $J >= 1000$.

The RM and PL both perform well, yielding sequential 95% credible intervals for both x_t and $1/\theta$ near those for their respective true filtered distributions for $J = 100$. By $J = 5000$ the true and estimated bounds for both x_t and $1/\theta$ become almost indistinguishable from one another using either algorithm. Looking at the filtered precision for $J = 100$, however, credible intervals generated by the RM within a single particle filter run appear to exhibit more variability than those generated within a single run of the PL algorithm.

5.3 Comparing models with varying signal-to-noise ratios

We now discuss estimation of the marginal likelihood of the simulated data and calculating posterior model probabilities. Recall that, given the density of the one-step ahead predictions, the marginal likelihood can be calculated by equation (2.35). Since the distribution of the one-step ahead predictions, $p(y_t|y_{1:t-1})$ for all t, is known for a local level DLM with common observation and state variance factor, the marginal likelihood $p(y_{1:T})$ can be calculated analytically. For computational stability, we calculate the log marginal likelihood, $\log p(y_{1:T})$, when comparing different models.

The true signal-to-noise ratio λ for our simulated data is 1. However, we can consider the marginal likelihood of the data under models with a different λ. Figure 5.3 displays the true log marginal likelihood of the data when different values of λ are assumed. Notice that, starting at $\lambda = 0$, there is a sharp increase in $\log p(y_{1:T})$ to maximum value as we approach the true λ of 1, with a gradual decrease in $\log p(y_{1:T})$ for $\lambda > 1$.

We ran each particle filter twenty more times for each of $J = 100, 500, 1000,$ and 5000 under the local level DLM described by equations (2.10) and (2.11) with λ assumed to be 0.5. We then repeated these runs for $\lambda = 2$. We consider these two values of λ because, as seen from Figure 5.3, these two values of λ

yield log marginal likelihoods that are lower than the true value but fairly close to one another. Figure 5.4 shows kernel density approximations to the empirical distributions of the twenty log marginal likelihood estimates, $\log p(y_{1:T})$, generated using each particle filter for each J and λ. For $\lambda = 1$ and 2, the PL appears to provide a better estimate of $\log p(y_{1:T})$ the fastest as a function J, as indicated by more concentrated densities around the truth relative to the RM and KDPF. For $J > 100$ and $\lambda = 0.5$, the RM is competitive with PL, while the KDPF appears to be outperformed in all scenarios.

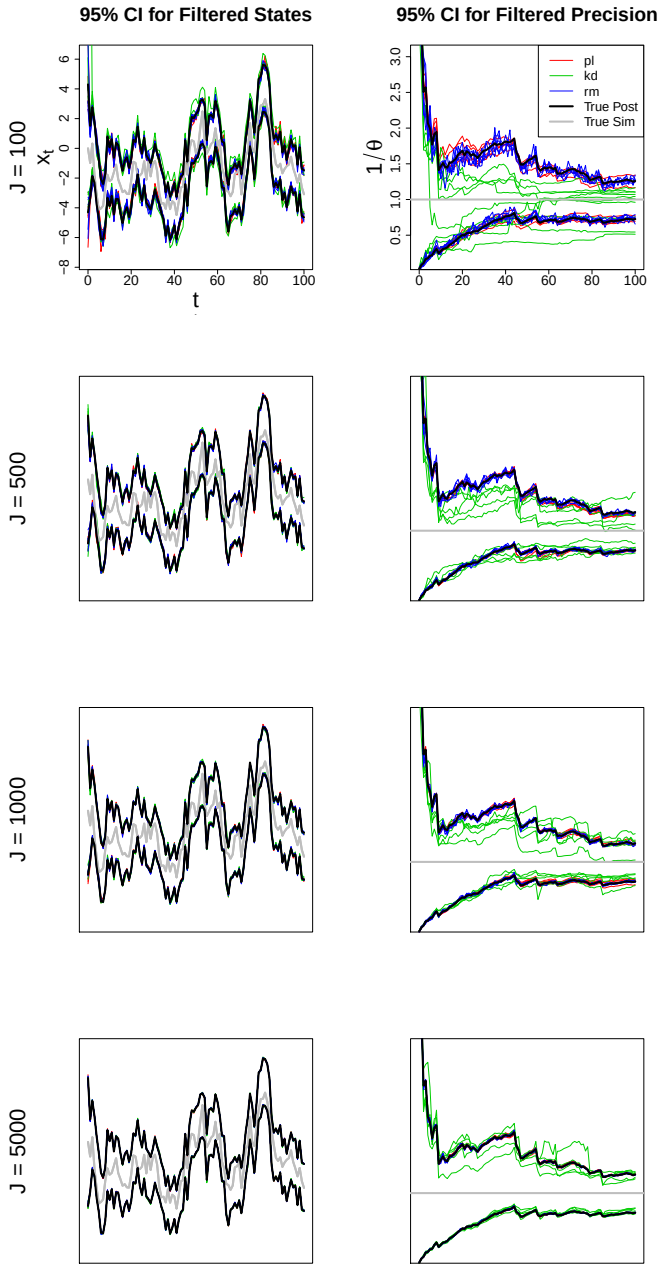
Lastly, we can consider posterior model probabilities among the set of models that assume $\lambda = 0.5, 1,$ and 2. We assume the prior probability of each model is 1/3, and calculate posterior model probabilities according to equation (3.37) with $N = 3$ and M_1, M_2, and M_3 representing models with $\lambda = 0.5, 1,$ and 2, respectively. Given a single estimate of the log marginal likelihood of the data under each of M_1, M_2, and M_3, a set of posterior probabilities among the three models can be calculated according to equation (3.37). Thus, for each of the KDPF, RM, and PL, we generate twenty such sets. We also calculate the set of true posterior probabilities among the three models by plugging in the true marginal likelihood of the data under each model into equation (3.37).

Figure 5.5 summarizes these calculations using compositional plots, where each corner of the ternary diagrams represents one of the three possible models. Each point in a diagram represents a set of three posterior probabilities, one for each

model, estimated by a specific particle filtering algorithm using J particles. The PL appears to perform the best, as the clustering of red points hones in around the point representing the true posterior probabilities the fastest with increasing J. The KDPF, again, is outperformed, needing at least 5000 particles to even start to cluster around the true posterior.

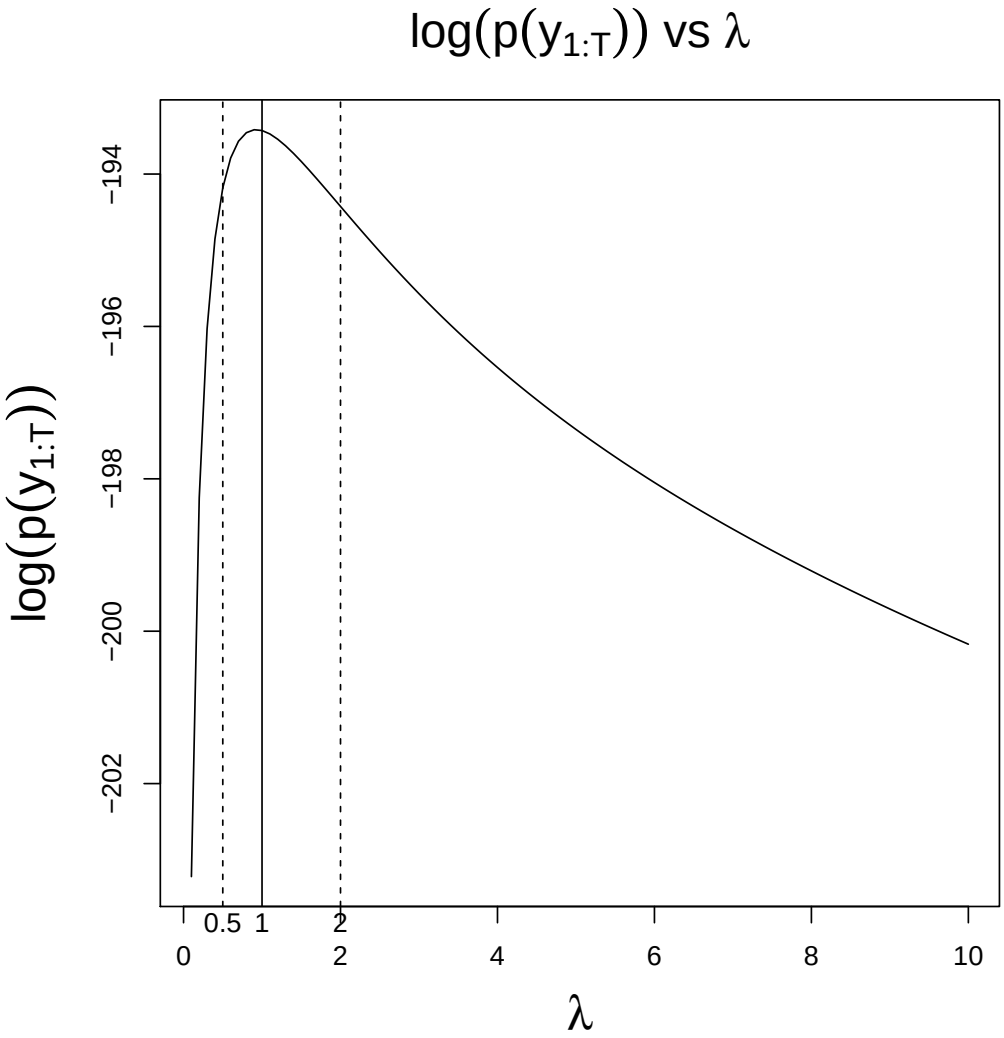
In this chapter, we've compared the relative performance of the KDPF, RM, and PL for comparing local level DLMs with common state and observation variance factor and different signal-to-noise ratios. The superior performance of PL relative to the RM is convenient from the practitioner's perspective, since this algorithm is easier to implement than the RM, which requires the specification of an MCMC algorithm in addition to the SMC. However, PL can only be used for specific models where the the distributions $p(y_{t+1}|x_t, \theta)$, $p(x_{t+1}|y_t, x_t, \theta)$, and $p(\theta|y_{1:t}, x_{0:t})$ are analytically tractable. The dynamic regression models described in Sections 2.3.3 and 2.3.3 emit analytical tractable forms of these distributions, and so we use PL for comparing these models in Chapter 6.

Figure 5.2: Comparing sequential credible intervals for KDPF, RM, and PL



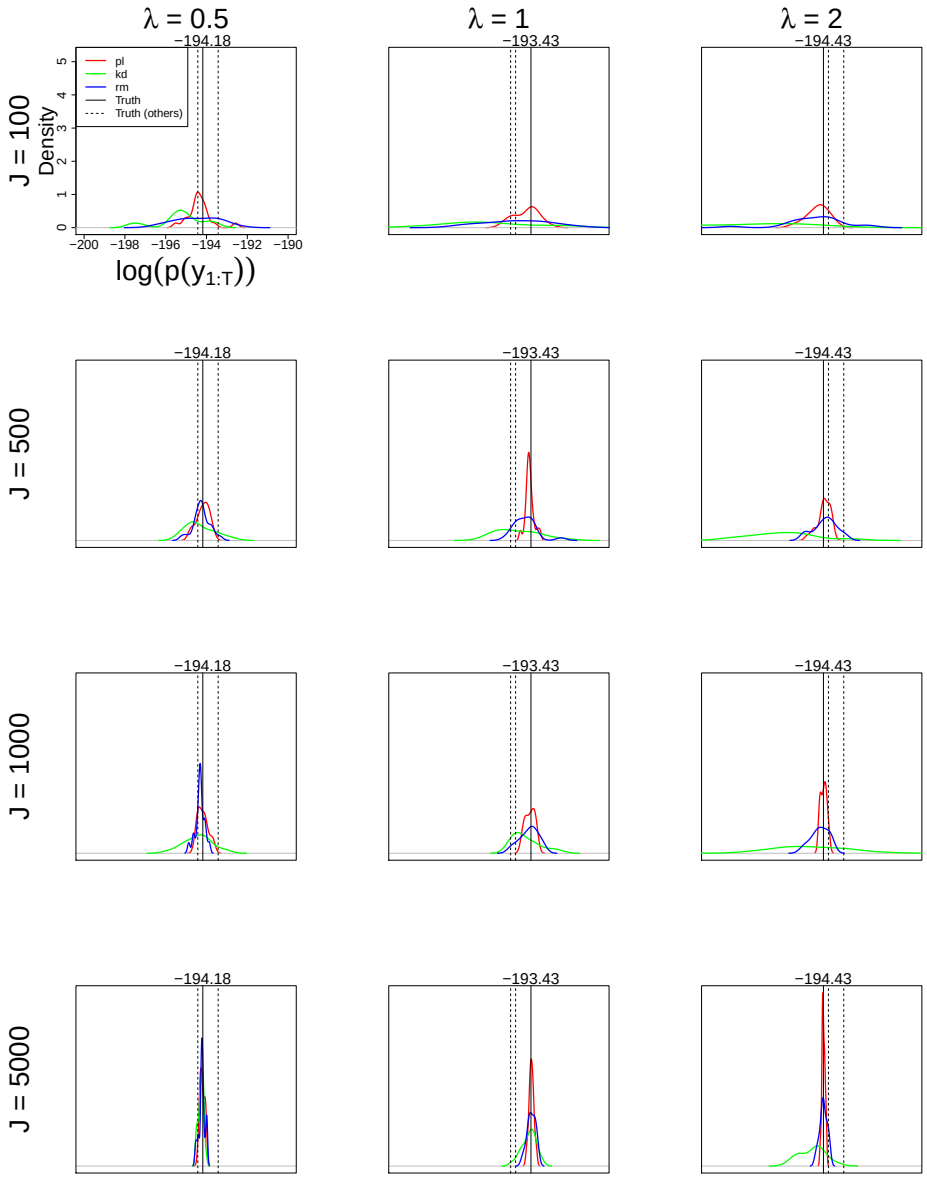
Sequential 95% credible intervals for the marginal filtered distribution of the states (left) and precision (right) for increasing number of particles J (rows) for the KDPF (green lines), RM (blue lines), and PL (red lines) compared with true simulated values (gray lines) and sequential 95% credible intervals obtained from the true filtered distributions (black lines). All axes within columns are on the same scale.

Figure 5.3: Log marginal likelihood versus λ



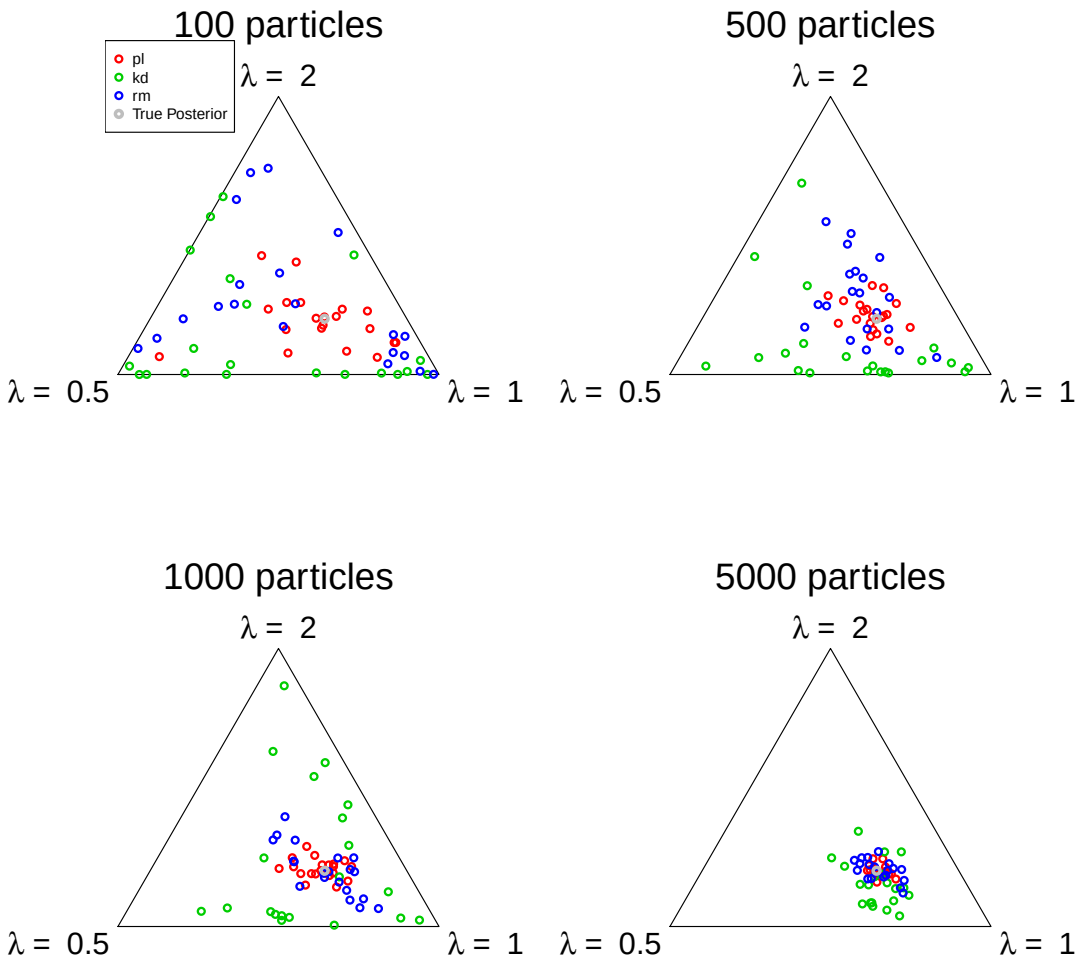
Solid black curve shows true log marginal likelihood (y-axis) of data simulated from the local level DLM described by equations (2.10) and (2.11) with $\theta = \lambda = 1$, calculated for increasing assumed values of the signal-to-noise ratio λ (x-axis). Height of the solid curve when intersecting with the solid vertical line denotes the true log marginal likelihood under the model with $\lambda = 1$, and the height of the solid curve at intersection with each of the dashed vertical lines represent the true log marginal likelihood under each of $\lambda = 0.5$ and $\lambda = 2$.

Figure 5.4: Comparing estimated log marginal likelihoods for KDPF, RM, and PL



Kernel density estimates of the distribution of twenty estimates of the log marginal likelihood of data simulated from the local level DLM described by equations (2.10) and (2.11) with $\theta = \lambda = 1$, obtained by running each of the KDPF (green lines), RM (blue lines), and PL (red lines) under different values of the signal-to-noise ratio λ (columns) for increasing number of particles J (rows), compared with the true log marginal likelihood (solid black lines with values at top). Dashed black vertical lines correspond to the true log marginal likelihood under models with λ equal to values from the other columns. Axes for all plot panels are on the same scale.

Figure 5.5: Comparing posterior model probabilities for KDPF, RM, and PL



Estimated posterior model probabilities among local level DLMs with common state and observation variance factor and three different signal-to-noise ratios λ (corners of triangles) for twenty runs of each of the KDPF (green dots), RM (blue dots), and PL (red dots) using increasing number of particles J (plot panels) on data simulated from the local level DLM described by equations (2.10) and (2.11) with $\theta = \lambda = 1$. Each point represents a set of posterior model probabilities (one for each λ), and the proximity of each point to a particular corner of the triangle represents the posterior probability of the model in that corner relative to the other models. The set of true posterior model probabilities is represented by the gray dot (the same for all panels).

Chapter 6

Statistical analysis of fMRI data

In this chapter, we use models and tools described in Chapters 2 and 3 to analyze time series data collected from an fMRI experiment. We describe the most common method of statistical analysis used in the field, i.e. the correlation-based general linear model (GLM) approach (Friston et al.; 1991, 1995b), and discuss challenges associated with analyzing fMRI data using this method. Autocorrelated time series invalidate results obtained using the standard GLM, which assumes independence of the error terms in the model. We explore variations of the GLM to account for this autocorrelation and show via simulation the negative consequences of using the standard GLM to analyze autocorrelated data.

We then use the dynamic regression models described in Section 2.3.3, with maximum likelihood estimation, to describe variation in fMRI data collected from a word recognition experiment. We propose a strategy to compare different dy-

namic regression models using PL. Using simulated data, we evaluate our ability to identify true model parameters via maximum likelihood estimation. Then, we use PL to examine conditions under which we can correctly identify a true data-generating model amongst several candidate models. Finally, we analyze real fMRI data using PL and discuss the appropriateness of the dynamic slope model for this data and as a tool for future fMRI studies.

In Section 6.1, we provide an overview of fMRI, standard estimation of fMRI time series, and the experimental data set. This material is mainly taken from Ashby (2011). In Section 6.2, we compare several techniques for estimating parameters in fMRI time series models and explore their impact on fitted model residuals as well as false positive/true positive rates of concluding significant brain activation. In Section 6.3, we investigate our ability to identify true values of model parameters in the dynamic regression models described in Section 2.3.3, and we fit real fMRI data using these models by maximum likelihood. Finally, in Section 6.4, we use simulated data to examine the ability to compare these dynamic regression models against each other using PL, and we discuss results from applying PL under these models to real fMRI data.

6.1 Overview of fMRI

Functional MRI provides an indirect measure of neural activation in the brain in near real time. Most fMRI experiments measure the *blood oxygen level-dependent*

(BOLD) signal, or ratio of oxygenated to deoxygenated hemoglobin in the blood. Evidence suggests that a type of neural activity called the local field potential is closely related to the BOLD signal recorded in an fMRI experiment (Logothetis; 2003; Logothetis et al.; 2001). By providing a noninvasive way to study functional changes in the brain over time, fMRI has allowed researchers to study topics that had previously seemed impossible to give a detailed scientific investigation, such as the nature of consciousness (Lloyd; 2002), meditation (Cahn and Polich; 2006), and moral judgement (Greene et al.; 2001).

6.1.1 The haemodynamic response

The BOLD response to a neural impulse is characterized by an increase in the BOLD signal from a baseline level to its peak at around 6 seconds post-stimulus, followed by a gradual decay back to baseline over the next 20-25 seconds. This typical BOLD response to an impulse as a function of time is referred to as the *haemodynamic response function (hrf)*, and knowledge about this function is crucial for effectively analyzing data from fMRI experiments. Although studies have shown that the hrf varies from person to person based on factors such as age (Richter and Richter; 2003), most analyses assume a known form of the hrf for all subjects. A commonly used hrf that is thought to represent an average BOLD response for a typical subject is defined by the SPM software package for analysis of fMRI data (http://www.fil.ion.ucl.ac.uk/spm/doc/). This hrf is known as the

canonical hrf (see bottom panel of Figure 6.1). Another commonly used hrf is the gamma function proposed by Boynton et al. (1996), given by

$$h(s) = \frac{(s/\tau)^{n-1} e^{-s/\tau}}{\tau(n-1)!}, \tag{6.1}$$

where s is time in seconds and τ and n are free parameters that determine the shape of the hrf. We use the canonical hrf for analyzing data from a word recognition experiment in Sections 6.2.1, 6.3.2, and 6.4.5, and we use the gamma hrf for simulating fMRI data in Sections 6.2.2, 6.2.3, 6.3.1, 6.4.1, 6.4.2, and 6.4.3.

6.1.2 The scanning session

An fMRI scanning session consists of one or more runs in which a human subject is presented with a simple task designed to stimulate the brain while scans are taken every few seconds. Runs can typically last anywhere between 10 and 30 minutes, and the *repetition time (TR)*, or length between individual scans, can be anywhere between 1 and 3 seconds. Each scan within a TR involves creating cross-sectional images, or *slices*, across the whole brain. Although TRs less than one second are possible on some machines, decreasing the TR length often comes at the cost of sacrificing spatial resolution of the images resulting from each scan.

Each whole brain image consists of a three-dimensional array of volumetric pixels, or *voxels*, and each voxel contains a value of the BOLD response for a small area of the brain. Voxel size and TR must be determined prior to run-

ning an experiment based on desired spatial and temporal resolution of the data. An average experiment might involve three 10-minute scanning runs with a TR of 2 seconds. An average scan might consist of 36 slices, where each slice consists of a 64 by 64 array of 3 mm^3 voxels. In this average scenario, each image would be made up of 147,456 voxels, and data from the entire scanning session would contain 132,710,400 BOLD values. Combining this with the fact that many studies involve multi-subject experiments, the sheer sizes of fMRI data sets pose significant challenges for data analysis.

In addition to choosing scanning parameters such as voxel size and TR, designing an fMRI experiment also involves deciding how the experimental stimulus is presented to the subject in the scanner. Three experimental designs frequently used in fMRI are block designs, slow event-related designs, and rapid event-related designs. Block designs divide functional runs into blocks of continuous activity and continuous rest, usually lasting anywhere from 30 seconds to a couple of minutes. During the activation blocks, subjects are instructed to perform the same task continuously over the entire block. In event-related designs, the stimulus *onsets* (i.e., TRs at which an experimental stimulus is presented to the subject) are chosen randomly, with the time between onsets, or *delay*, usually somewhere between 2 and 16 seconds. Slow event-related designs include rest periods that last around 30 seconds, while rapid event-related designs use shorter rest periods.

The long rest periods included in block and slow event-related designs are meant to allow the BOLD response to decay back to baseline before the next stimulus presentation. This helps increase the power of statistical tests designed to identify neural activation or distinguish between event types. However, these designs result in longer experiments which are more expensive and incur a greater risk of having the subject's mind wander during rest periods and generate non-task related BOLD signal. Rapid event-related designs have become more popular with the development of statistical methods such as the GLM approach that make analysis of data collected from these experiments possible.

This section is intended to provide a quick overview of fMRI for the purpose of giving context to the analyses discussed in the rest of this chapter. For more information on fMRI and designing fMRI experiments, we refer the reader to Ashby (2011); Poldrack et al. (2011).

6.1.3 The correlation-based GLM approach

The standard correlation-based GLM analysis of fMRI data models the observed fMRI data in a single voxel of the brain as a linear function of the expected BOLD response from a voxel responding to the experimental stimulus, i.e.

$$y_t = \beta_0 + \beta_1 \text{conv}_t + \epsilon_t, \tag{6.2}$$

where y_t is the observed fMRI signal at TR t, conv_t is the expected BOLD response at TR t in an active voxel, $\beta = (\beta_0, \beta_1)'$ are unknown fixed regression coefficients,

and $\epsilon_t \overset{iid}{\sim} \mathrm{N}(0, \sigma^2)$ are independent random errors. The expected BOLD response, conv_t, is calculated by convolving the hrf with an "on-off" boxcar function that is equal to 1 when the experimental stimulus is on, and 0 when it is off. Specifically,

$$\mathrm{conv}_t = \int_0^{t'} N(s)h(t-s)\mathrm{d}s, \tag{6.3}$$

where $N(s)$ represents the value of the neural activation boxcar at time s in seconds. Although $N(s)$ and $h(s)$ are defined with respect to time in seconds, we observe fMRI data at discrete time points determined by the TR. Thus, we define the time index t in units of TRs and let $t' = s/TR$. Expected responses to different event types can be included in this model as additional covariates by convolving the hrf with the boxcar function associated with each event. In this chapter, we restrict ourselves to experiments with a single event type.

Under the model given by equation (6.2), the hypothesis test

$$H_0 : \beta_1 = 0 \quad H_A : \beta_1 > 0 \tag{6.4}$$

is usually of interest, where rejection of H_0 in favor of H_A is interpreted as evidence of neural activation in the voxel from which the fMRI time series came from. To test this hypothesis, ordinary least squares estimates of the unknown fixed parameters β and σ^2 are computed, i.e.

$$\hat{\beta} = (\hat{\beta}_0, \hat{\beta}_1)' = (X'X)^{-1}X'y \qquad \hat{\sigma}^2 = \frac{1}{T-2}\|y - X\hat{\beta}\|^2, \tag{6.5}$$

where T is the total number of TRs in the functional run, X is the T by 2 design matrix with first column all 1's and second column equal to conv_t, $y =$

$(y_1, \ldots, y_T)'$, and $\|\cdot\|$ is the Euclidean norm. The test statistic, T^*, and p-value, p^*, are then calculated by

$$T^* = \frac{\hat{\beta}_1}{\sqrt{\hat{\sigma}^2 (X'X)^{-1}_{(2,2)}}} \qquad p^* = \mathrm{P}(T^* > t^*_{obs} | H_0), \tag{6.6}$$

where $(X'X)^{-1}_{(2,2)}$ is the element in the second row and second column of $(X'X)^{-1}$, t^*_{obs} is the realization of the random variable T^*, and $P(A|H_0)$ is the probability of event A assuming H_0 is true. Thus, p^* is calculated under the assumption that $t^* \sim \mathrm{T}(0, 1, T-2)$, and H_0 is rejected if p^* is less than some significance threshold α.

The majority of fMRI studies perform this hypothesis test independently for every voxel, resulting in a statistical parametric map of brain activation. With this approach, an adjustment to the significance threshold α must be made to account for multiple hypothesis tests being performed simultaneously. For example, if a false positive rate of $\alpha = 0.05$ is desired, a corrected threshold α^* must be used for each independent test so that the probability of at least one false positive among all tests is 0.05. Because of the spatial relationship among voxels, these hypothesis tests are not actually independent of each other, and this complicates the problem of finding the necessary correction. Typically, an approach relying on the theory of Gaussian random fields is used (Worsley; 1995; Worsley et al.; 1996, 1992; Friston et al.; 1991). We refer the reader to Ashby (Chapter 6, 2011) for more information on the multiple comparisons problem.

The standard GLM approach to analyzing fMRI data, which models univariate time series separately for each voxel, is convenient because regression theory allows for simple forms of estimators and fast computation. However, aside from using a multiple comparisons correction, the spatial nature of fMRI data and the connection between voxel-wise time series is ignored using this approach. Hence, a second-stage connectivity analysis is required to gain any insight into neural networks (Chapters 8 and 9 Ashby; 2011). The development of multivariate methods that analyze activation and connectivity simultaneously has gained popularity over the last decade. These include independent components analysis (Beckmann and Smith; 2004), multi-voxel pattern recognition Norman et al. (2006), representation similarity analysis (Nili et al.; 2014), and Bayesian spatio-temporal modeling approaches such as those developed by Woolrich et al. (2004), Bowman et al. (2008), Alicia et al. (2010), and Zhang et al. (2014), to name a few.

While the development of efficient numerical approximation algorithms have decreased the computational burden of analyzing fMRI data using Bayesian spatio-temporal models, they are still slower and more difficult to implement than the standard GLM approach. Thus, voxel-wise hypothesis tests are still the norm in fMRI data analysis, and we operate within the framework of univariate voxel-specific time series models for the remainder of this chapter. For more information on fMRI and standard statistical techniques used in the field, see Ashby (2011); Penny et al. (2011).

6.1.4 Word recognition task

The data set that we analyze in Sections 6.3 and 6.4 comes from an episodic word recognition experiment for one human subject. The task the subject worked on, described in Bennett and Miller (2013), consisted of an encoding session that took place outside the scanner and a recognition session that took place inside the scanner. During encoding, the subject was presented with a list of words one at a time and told to memorize the words so that if they saw one of them again, they would recognize it. During the recognition session, the subject was presented with another list of words, some of which they saw during encoding and some of which were new. The subject was asked to respond as to whether they thought each word was old or new based on their memory.

While in the scanner, the words were presented according to a rapid-event related design with random delays between onsets lasting somewhere between 2 and 10 seconds. The expected BOLD response (conv_t) for this design was then constructed by convolving the canonical hrf with a boxcar function that is equal to 1 during TRs when a word was presented to the subject and 0 otherwise. The middle panel of Figure 6.1 shows conv_t for this experiment. Scans of the subject's brain were taken every 1.5 seconds for about 6 minutes ($T = 245$ total TRs) with a voxel size of 3 mm^3. Although whole brain data were recorded, we look specifically at time series from 5 by 5 by 5 voxel cubes (125 voxels per cube) extracted from six different brain regions, namely the left frontal pole (FP), left intraparietal sulcus

(IPS-left), right intraparietal sulcus (IPS-right), primary visual cortex (PV), left secondary visual cortex (SV-left), and right secondary visual cortex (SV-right).

An important step that is performed prior to analyzing fMRI data is preprocessing of the raw data that comes directly out of the scanner. For example, images need to be spatially realigned to reduce the effect of the subject's head movements while inside the scanner. In addition, a high-resolution structural image taken prior to the functional run can be used to discern the exact location of voxels that are difficult to locate in lower resolution functional images. This process is called *coregistration* of the functional and structural data. The coregistered data then needs to be normalized to a standard brain atlas so that active voxels can be assigned to a neuroanatomic brain structure.

For this data set, preprocessing proceeded as outlined in Bennett and Miller (2013):

> "The functional time series were spatially realigned to the first image using a least squares approach with a 6-parameter rigid body affine transformation (Friston et al.; 1995c). Realigned images were then "unwarped" to reduce the influence of residual movement-related variance on BOLD signal intensity (Andersson et al.; 2001). The functional data were coregistered to a high-resolution T1 anatomical image using mutual information maximization with a 6-parameter rigid body affine transform (Ashburner et al.; 1997). Then, the images

> were normalized to the standard 3D brain atlas defined by the International Consortium for Brain Mapping using a combination of a 12-parameter linear affine transformation and 3 by 2 by 3 nonlinear three-dimensional discrete cosine transform. A 7th degree B-spline was used as the interpolation method for creating normalized images (Ashburner and J.; 1999; Mazziotta et al.; 1995)."

Other common preprocessing steps include spatial smoothing and slice-timing correction. Spatial smoothing is intended to get rid of some of the noise in the data and allow for the use of Gaussian random field theory when applying a correction for multiple hypothesis tests. Slice-timing correction adjusts for the fact that brain slices taken during a particular TR don't occur at the same time. For the word recognition experiment, the data were spatially smoothed with an 8 mm full-width at half maximum isotropic Gaussian kernel. Slice-timing correction was not applied to the data and is not typically used when the TR is less than 2 seconds (Penny et al.; 2011).

Preprocessing of fMRI data is intended to improve statistical analysis by removing *artifacts*, or abnormalities in the data due to non-task related events. However, it is possible that preprocessing can also add artifacts to the data. For instance, the compound effect of applying both slice-timing correction and spatial realignment can affect the signal in the data. While researchers hope to find signals in the data that provide insight into task-related neural activity, it is possible

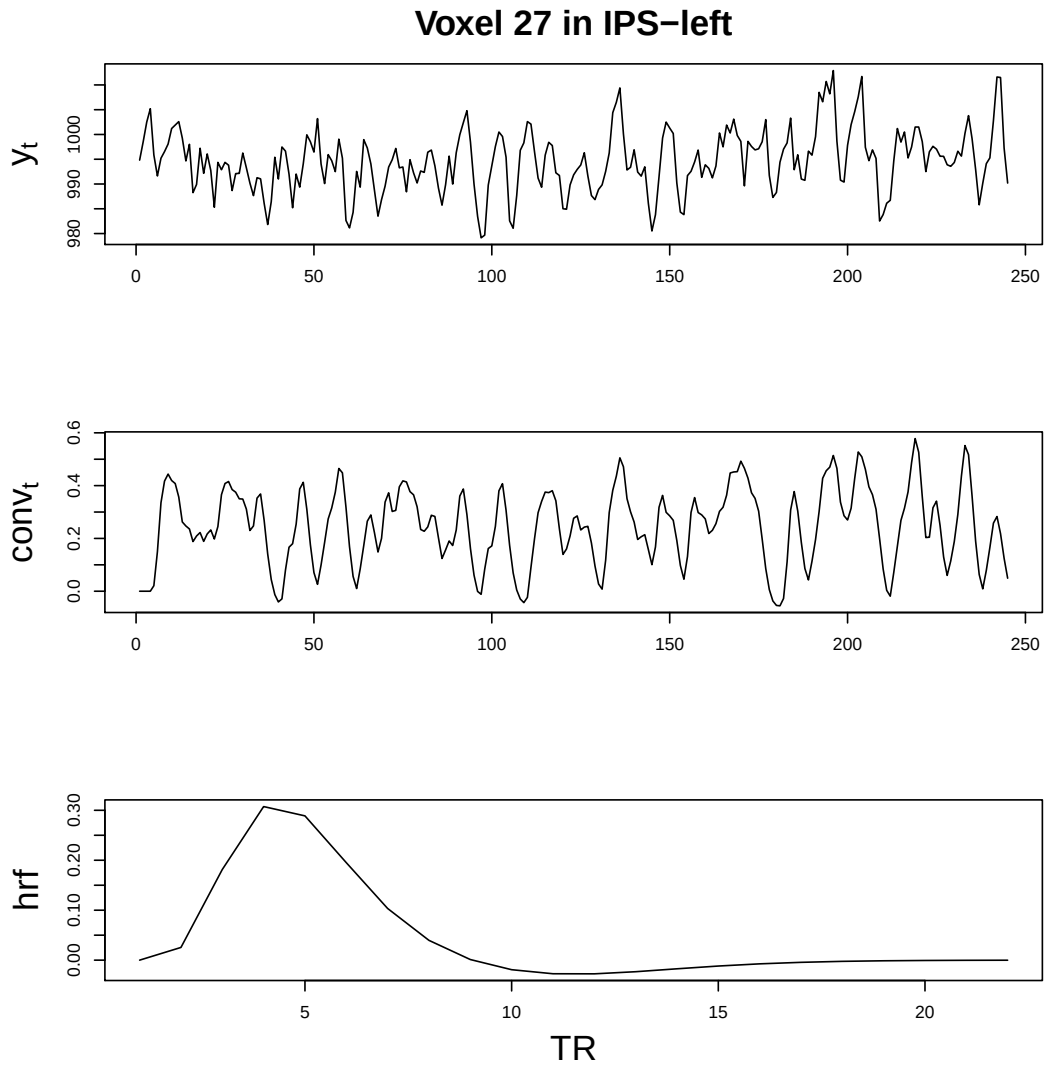
that the way data are preprocessed can affect results. For more information on preprocessing of fMRI data, see Ashby (Chapter 4, 2011).

The top panel of Figure 6.1 shows the preprocessed fMRI time series for a voxel in IPS-left. Notice that the pattern of the observed time series somewhat mirrors the active response displayed in the middle panel for TRs greater than 75, but not for TRs less than 75. This type of behavior motivates our thinking that a regression model with a changing slope, such as M_{011}, might be appropriate for modeling fMRI data.

6.2 Temporal autocorrelation

The standard GLM approach for identifying task-related activity in a single voxel of the brain relies on an assumption of independence of the error terms, ϵ_t, in equation (6.2). This assumption is not reasonable for fMRI data since random departures between the observed and predicted BOLD responses are likely to be similar among voxels near to each other in time and space. One reason for this is that the BOLD response to neural activation is not uniform across space (Aguirre et al.; 1998), so any assumed hrf is guaranteed to be at least slightly inaccurate. Thus, if the BOLD response in a voxel at one particular TR is greater than average, it is likely to also be greater than average in nearby voxels and at subsequent TRs (Chapter 1 Ashby; 2011). Other factors that contribute to spatially and temporally autocorrelated errors include unaccounted for signals in

Figure 6.1: Single voxel time series from fMRI experiment



Time series data (top), expected BOLD response (middle), and haemodynamic response function (bottom) versus TR for voxel 27 in the left intraparietal sulcus.

the data, such as non-task related cognitive activity on the part of the subject, and small movements caused by heartbeat and respiration (Locasio et al.; 1997).

An approach to handling temporally autocorrelated fMRI time series that was developed early on is to "color" the data using a low-pass temporal filter to reduce high-frequency noise and amplify the signal in the data (Friston et al.; 1995a; Worsley and Friston; 1995). An alternate approach used by Bullmore et al. (1996) involves a two-stage procedure where the data are *prewhitened* by first estimating the autocorrelation in the errors using the residuals from a GLM fit and then transforming the data to remove the autocorrelation. The standard GLM analysis is then applied to the prewhitened data. The prewhitening approach is an improvement over coloring the data because it yields minimum variance unbiased estimates of the regression coefficients, provided the autocorrelation is accurately estimated from the residuals (Friston et al.; 2002). Woolrich et al. (2001) use resting state data to demonstrate that prewhitening performs more efficiently than coloring in their data, and that bias in estimating the autocorrelation during prewhitening can be lowered by carrying out spatial and temporal smoothing during preprocessing.

The prewhitening approach to handling autocorrelated errors is the standard technique used in the FSL software package (http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/). Other approaches attempt to account for temporal autocorrelations explicitly through modeling. For example, Lund et al. (2006) measured effects thought

to contribute to autocorrelated noise such as heartbeat, respiration, and magnetic field strength, and included them as additional covariates in the GLM. SPM uses an approach developed by Kiebel and Holmes (2007) that models the correlation in the errors by

$$\epsilon \sim \mathrm{N}(0, \sigma^2 I_T + \lambda Q) \quad Q_{i,j} = \begin{cases} 0, & \text{if } i = j \\ e^{-|i-j|}, & \text{if } i \neq j, \end{cases} \tag{6.7}$$

where T is the total number of TRs in the experiment, $\epsilon = (\epsilon_1, \ldots, \epsilon_T)'$, and λ is another unknown fixed parameter. Restricted maximum likelihood estimation (REML) is carried out to estimate the unknown fixed parameters, and the hypothesis test in equation (6.4) is performed using the test statistic

$$T^* = \frac{\hat{\beta}_1}{\sqrt{\left((X'X)^{-1}X'(\hat{\sigma}^2 I_T + \hat{\lambda} Q)X(X'X)^{-1}\right)_{(2,2)}}}, \tag{6.8}$$

where $\hat{\beta}_1$, $\hat{\sigma}^2$, and $\hat{\lambda}$ are the REML estimates. We discuss REML further in Section 6.2.2. Under the null hypothesis, $t^* \sim \mathrm{T}(0, 1, df)$, where the degrees of freedom df is computed by the Satterthwaite approximation (Worsley and Friston; 1995). In Section 6.2.2, we compare ordinary least squares (OLS) estimation, prewhitening, and REML in terms of false positive and true positive rates of significant brain activation using simulated data.

6.2.1 Exploration of ARMA models

Numerous studies have attempted to account for temporal autocorrelation in fMRI data by replacing the error term in equation (6.2) by first and second

order autoregressive processes (Bullmore et al.; 1996; Locasio et al.; 1997). We now explore the class of regression models with ARMA(P, Q) errors discussed in Section 2.3.2 using maximum likelihood estimation to investigate whether other ARMA error structures might be appropriate for the word recognition data set.

Recall from Section 2.3.2 that the DLM formulation of a regression model with ARMA(P, Q) errors is given by equations (2.13) and (2.14) with $x_t = (x_{t,1}, x_{t,2}, \ldots, x_{t,m})'$ an m-dimensional vector, $m = \max(P, Q+1)$, F_t a time-invariant $1 \times m$ vector with first element equal to 1 and the rest 0, $v_t = 0$ for all t, G a $m \times m$ matrix that takes the form

$$G = \begin{pmatrix} \phi_1 & \vdots & & & \\ \phi_2 & \vdots & & & \\ \phi_3 & \vdots & & I_{m-1} & \\ \vdots & \vdots & & & \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \phi_m & \vdots & 0 & \cdots & 0 \end{pmatrix},$$

and $W = \sigma^2 ee'$ with $e = (1, \gamma_1, \ldots, \gamma_{m-1})'$. The unknown fixed parameters are given by $\theta = (\beta', \phi', \gamma', \sigma^2)'$, where $\beta = (\beta_0, \beta_1)'$, $\phi = (\phi_1, \phi_2, \ldots, \phi_P)'$ and $\gamma = (\gamma_1, \gamma_2, \ldots, \gamma_Q)'$. We let $U_t = (1, \text{conv}_t)$, where conv_t is the convolution, evaluated at TR t via equation (6.3), of the canonical hrf with the on-off boxcar function representing the stimulus pattern for the word recognition experiment. We adopt the convention that $\phi_s = 0$ for $s > P$ and $\gamma_r = 0$ for $r > Q$.

Maximization of the likelihood is performed using the R function `arima` (R Core Team; 2013), which calls on `optim` to minimize the negative log likelihood, given by

$$-\log p(y_{1:T}|\theta) = \frac{1}{2}\sum_{t=1}^{T}\log|Q_t| + \frac{1}{2}\sum_{t=1}^{T}\zeta_t' Q_t^{-1}\zeta_t \tag{6.9}$$

(Shumway and Stoffer; 2006, Chapter 6). Here, $\zeta_t = y_t - f_t$ are the *innovations* of the ARMA process, with f_t and Q_t being the mean and variance of the one-step ahead forecasts for y_t. For fixed θ, f_t and Q_t are computed via the Kalman filter in equation (2.30), provided the initial values m_0 and C_0. These initial values are chosen automatically by `arima` such that the stationarity constraint given in equation (2.18) is satisfied (Gardner et al.; 1980). Given an initial set of fixed parameter values, optimization is performed using an iterative algorithm that alternates between running the Kalman filter conditional on θ and minimizing equation (6.9) conditional on $f_{1:T}$ and $Q_{1:T}$ (Durbin and Koopman; 2012). Fixed parameter values are constrained to their regions of stationarity, given by equations (2.16) and (2.17), using the transformation method of Jones (1980).

Time series regression models with ARMA errors for all combinations of P and Q up to order 10 were fit to data from five randomly chosen voxels from each of the six brain regions using the `arima` function. We then evaluated each model fit using three criteria: Akaike's information criterion (AIC) (Sakamoto et al.; 1986), AIC corrected for bias (AICC) (Sugiura; 1978; Hurvich and Tsai; 1989), and Bayes' information criterion (BIC) (Schwarz; 1980). For our model with a

single regression covariate, the formulas for these criteria are given by

$$AIC = -2\log p(y_{1:T}|\hat{\theta}) + 2(P+Q+3) \tag{6.10}$$

$$AICC = -2\log p(y_{1:T}|\hat{\theta}) + 2T\frac{P+Q+2}{T-P-Q-3} \tag{6.11}$$

$$BIC = -2\log p(y_{1:T}|\hat{\theta}) + (\log T)(P+Q+3), \tag{6.12}$$

where $\hat{\theta}$ is the MLE of the unknown fixed parameters and $\log p(y_{1:T}|\hat{\theta})$ is the value of the log-likelihood at convergence of the maximum likelihood optimization procedure.

When performing model selection using one of the above criteria, the goal is to select a model that minimizes the specific chosen criterion. The first term is the same for all criteria and should be smaller for better model fits. The second term is a penalty for the number of parameters in the model (a measure of model complexity). BIC imposes the strongest penalty for having more parameters and is the most likely out of the three to prefer simpler models. In our study, we recorded the values of P and Q that minimized each of these criteria for each randomly selected voxel, and the average P and Q for each brain region are shown in Table 6.1. From these averages, it appears that AIC and AICC prefer P and Q near 3 while BIC prefers $P = 1$ and an $Q = 0$ or 1. We prefer to use the simpler models chosen by BIC and will primarily focus on models that incorporate first-order autoregressive errors in the remainder of this chapter.

Table 6.1: Mean AR and MA orders for experimental fMRI data

Region	Criterion					
	AIC		AICC		BIC	
	P	Q	P	Q	P	Q
Left frontal pole	2.80	3.20	2.80	2.90	1.70	0.90
Left intraparietal sulcus	3.75	3.50	3.50	3.25	1.81	0.06
Right intraparietal sulcus	3.20	2.80	3.20	2.80	0.80	0.80
Primary visual	3.10	3.00	3.10	2.70	0.90	1.70
Secondary visual left	3.20	2.90	2.10	2.40	1.40	0.00
Secondary visual right	3.20	3.00	3.10	2.50	0.70	0.70
Mean across regions	3.21	3.07	2.97	2.76	1.22	0.69

Mean AR and MA orders (P and Q, respectively) chosen according to AIC, AICC, and BIC for maximum likelihood fits of regression models with ARMA errors to voxel-wise time series from 5 by 5 by 5 voxel cubes taken from 6 different brain regions.

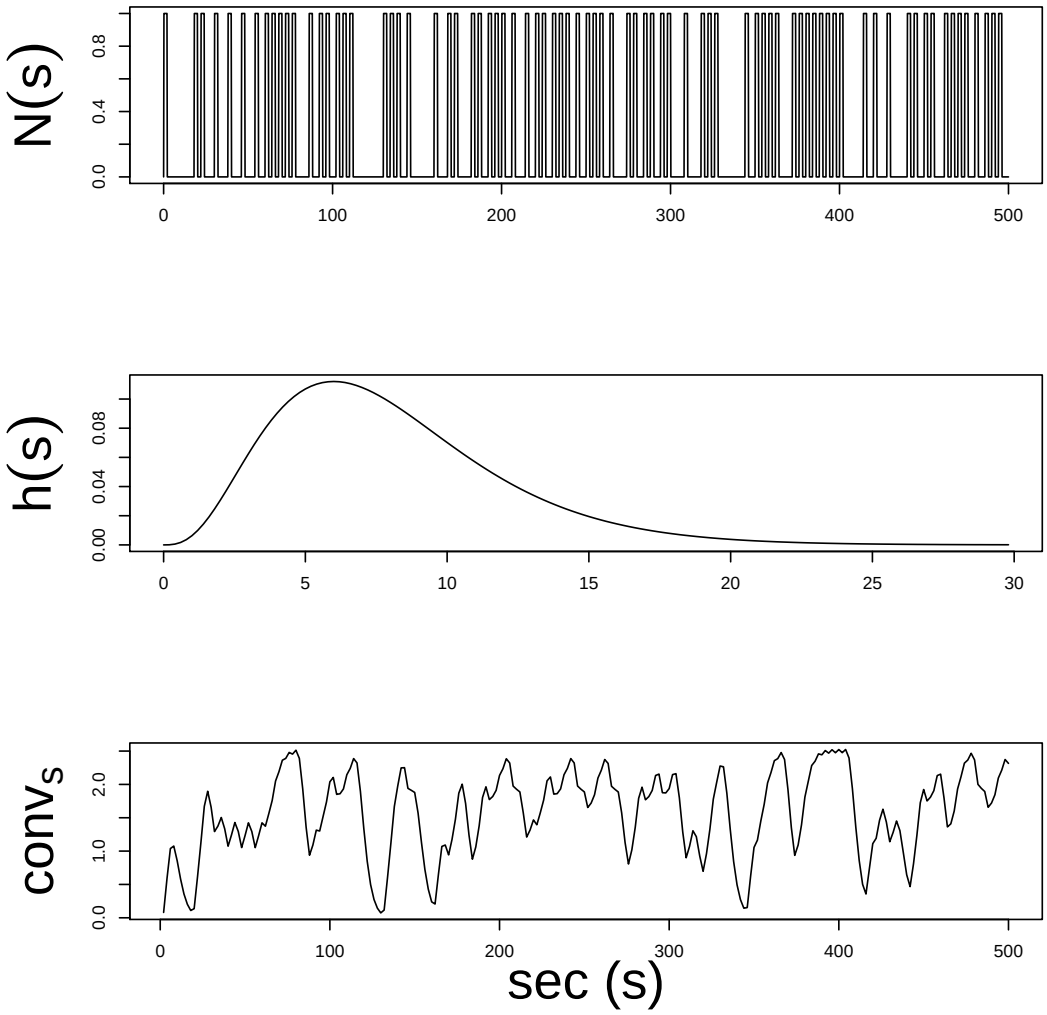
6.2.2 False positive and true positive rates

In this section, we consider the impact of different approaches to testing for brain activation in autocorrelated voxel-wise time series. Specifically, we examine false positive and true positive rates. The false positive rate is the rate of concluding significant neural activation in a voxel when there is no activity present. The true positive rate (or power) is the rate of concluding significant neural activation when there is in fact activity present. We prefer methods that yield a high true positive rate while keeping the false positive rate low. In this section, we examine the effect on the false positive rate of using standard OLS estimation of the regression slope in equation (6.2), and we compare different methods for accounting for temporal autocorrelation in terms of their effect on false postive and true positive rates.

To analyze false positive and true positive rates, we simulated fMRI data from the regression model described in Section 6.2.1 with AR(1) error structure (i.e. $P = 1$ and $Q = 0$). An experiment with a rapid event-related design and a single event type was created by simulating random times between onsets according to a truncated geometric distribution with a maximum time-to-event of 10 TRs. We used a TR of 2 seconds and let the experiment run for 250 total TRs. Onset of the stimuli were assumed to last the length of the TR, and an on-off boxcar function of 1's and 0's was constructed to match the stimulus pattern. The explanatory variable in the regression, conv_t, was constructed by convolving the boxcar function with the gamma hrf from equation (6.1) with $\tau = 2$ and $n = 4$. Figure 6.2 displays the simulated experimental design and expected BOLD response (conv_t) for active voxels.

Time series of length $T = 250$ were then simulated according to M_{100}, i.e. a regression model with AR(1) errors as in equations (2.13) and (2.14) with $m = P = 1$, $Q = 0$, $F_t = 1$ for all t, $v_t \sim \delta_0(v_t)$ for all t (i.e., $v_t = 0$ for all t), $G = \phi$, $W = \sigma^2$, $\beta = (\beta_0, \beta_1)'$, and $U_t = (1, \text{conv}_t)$. For these simulations, we let $\beta_0 = 750$ and $\sigma^2 = 15$, and one thousand time series were generated for every $(\beta_1, \phi) \in \{0, 1, 2, 3\} \times \{0.25, 0.50, 0.75, 0.95\}$. Different values of β_1 were used so that we could analyze false positive rates (for $\beta_1 = 0$) and power (for $\beta_1 > 0$). Similarly, different values of ϕ were used for simulation so that we

Figure 6.2: Simulated rapid-event related design of fMRI experiment



Simulated boxcar function (top), hrf (middle), and convolution of the boxcar with the hrf (bottom) for a rapid-event related design of an fMRI experiment.

could analyze false positive rates and true positive rates for increasing amounts of autocorrelation in the data.

For each simulated time series, we tested the hypothesis in equation (6.4) using the OLS method, i.e. where the test statistic and p-value are calculated according to equations (6.5) and (6.6). We also performed hypothesis tests using prewhitening (PW) and two variations of a REML approach. Our PW and REML approaches each assume that the data are generated from M_{100}, as described in the preceding paragraph, where the AR(1) process for x_t is stationary. This model can be reformulated as

$$y = X\beta + \epsilon, \quad \text{where} \tag{6.13}$$

$$\epsilon \sim \mathrm{N}(0, \sigma^2\Lambda),$$

$$\Lambda_{i,j} = \frac{\phi^{|i-j|}}{1-\phi^2},$$

$\beta = (\beta_0, \beta_1)'$ are the fixed regression coefficients, y and X are data vector and design matrix, respectively, as defined immediately after equation (6.5), and $\epsilon = (\epsilon_1, \ldots, \epsilon_T)'$ is the vector of error terms.

The PW approach uses the fact that if Λ is known, the data can be transformed via

$$y^* = S^{-1}y \qquad X^* = S^{-1}X, \tag{6.14}$$

where S is the Cholesky decomposition of Λ (i.e. $SS' = \Lambda$), to yield the independent error GLM given by

$$y^* = X^*\beta + \epsilon \quad \epsilon \sim \mathrm{N}(0, \sigma^2 I_T). \tag{6.15}$$

We carry out the PW method on a time series from a single voxel by first estimating β_0 and β_1 using OLS. Then, a zero mean AR(1) process is fitted to the residuals from the resulting fit, using maximum likelihood via the `arima` function in R. The maximum likelihood estimate of the autocorrelation in the residuals, denoted by $\hat{\phi}$, is used to approximate ϕ in equation (6.13), and an estimate of the covariance matrix Λ is computed according to

$$\hat{\Lambda}_{(i,j)} = \frac{\hat{\phi}^{|i-j|}}{1-\hat{\phi}^2}. \tag{6.16}$$

The data are then transformed according to

$$\hat{y}^* = \hat{S}^{-1}y \qquad \hat{X}^* = \hat{S}^{-1}X, \tag{6.17}$$

where $\hat{S}$ is the Cholesky decomposition of $\hat{\Lambda}$. Lastly, the hypothesis test is performed using the OLS method with $\hat{y}^*$ and $\hat{X}^*$ used in place of y and X in equations (6.5) and (6.6).

Estimation using the restricted likelihood is intended to remove bias in estimating variance components that is due to fixed regression coefficients being included in the model. This is achieved by integrating β out of the full likelihood. For hypothesis tests using REML, we first obtain an estimate of ϕ by maximizing

the profiled log-restricted likelihood, derived by Harville (1977) and given by

$$\log p(y|\phi) \propto -(T-2)\log\left|\left|y^* - X^*\left[(X^*)'X^*\right]^{-1}(X^*)'y^*\right|\right| - \frac{1}{2}\log|(X^*)'X^*| - \frac{1}{2}\log|\Lambda| \tag{6.18}$$

(Pinheiro and Bates; 2000, Page 205). Here, y^*, X^*, and Λ are implicitly functions of ϕ, with y^* and X^* described by equation (6.14) and Λ described by the third line of equation (6.13). Once $\hat{\phi}$ that maximizes $\log p(y|\phi)$ in equation (6.18) is found, $\hat{\Lambda}$ is calculated according to equation (6.16), $\hat{X}^*$ and $\hat{y}^*$ are calculated according to equation (6.17), and REML estimates of β and σ^2 are computed according to

$$\hat{\beta} = \left[(\hat{X}^*)'\hat{X}^*\right]^{-1}(\hat{X}^*)'\hat{y}^* \qquad \hat{\sigma}^2 = \frac{1}{T-2}\left|\left|\hat{y}^* - \hat{X}^*\hat{\beta}\right|\right|^2. \tag{6.19}$$

We used the `gls` function in R package `nlme` to find $\hat{\phi}$ that maximizes the profiled log-restricted likelihood in equation (6.18) (Pinheiro and Bates; 2000), and then used equations (6.19) to calculate $\hat{\beta}$ and $\hat{\sigma}^2$. We then performed the t-test in equation (6.6) using these estimates of $\hat{\beta}_1$ and $\hat{\sigma}^2$, with $\hat{X}^*$ used in place of X. However, because ϕ is estimated in addition to σ^2, the null distribution of the test statistic T^* no longer follows a t-distribution. Nonetheless, we can perform the t-test conditional on the REML estimate of ϕ by using $\mathrm{T}(0, 1, T-2)$ as an approximation to the null distribution of T^*.

A better approximation to the null distribution of T^* can be achieved by using a t-distribution with an adjusted degrees of freedom. For example, Kiebel and Holmes (2007) adjusts the degrees of freedom of the t-test for the model described

by equation (6.7) using the Satterthwaite approximation described in Worsley and Friston (1995). We consider a strategy prescribed by Dawdy and Matalas (1964), where the problem of autocorrelated time series within the context of statistical tests that depend on an assumption of independent random samples is circumvented by calculating the so-called "effective sample size". Specifically, we use an effective sample size adjustment for time series with first-order autocorrelation, given by

$$T' = T\frac{1-\hat{\phi}}{1+\hat{\phi}}, \tag{6.20}$$

and we adjust the degrees of freedom of the t-test by using $T' - 2$. We will refer to the method that approximates the null distribution of T^* using $\mathrm{T}(0, 1, T-2)$ as the REML approach, and we will refer to the method that approximates this distribution using $\mathrm{T}(0, 1, T'-2)$ as the REMLc approach (for corrected REML).

Table 6.2 displays false positive rates of rejecting H_0 using significance thresholds $\alpha = 0.001, 0.01,$ and 0.05 for the 1000 simulations using $\beta_1 = 0$ and for each of four values of ϕ. Methods of estimation that accurately assess the uncertainty in $\hat{\beta}_1$ should yield false positive rates equal to α. The results from Table 6.2 illustrate that using OLS inflates the false positive rate, and furthermore that the ratio of the false positive rate to α increases as α decreases.

Figure 6.3 displays false positive rates with increasing α for each method and each true value of ϕ, as well as 95% confidence intervals for the false positive rates calculated using a normal approximation to the distribution of the proportion of

false positives out of the 1000 simulations. The 95% confidence intervals around the false positive rate for PW and REML contain the nominal threshold α for all values of ϕ. REMLc appears to give slightly lower false positive rates than REML and PW for $\phi \leq 0.75$ and decidedly lower false positive rates for $\phi = 0.95$ (for which the approximate confidence intervals for REMLc do not contain the nominal value of α). While at first glance this may seem to be an advantage of REMLc, a decrease in the false positive rate can come at the cost of a decrease in the true positive rate as well. Figure 6.4 illustrates this point using ROC curves. The ROC curves in this figure display the true positive rate versus the false positive rate for the hypothesis tests performed according to each method. The methods with the largest area under the ROC curve performed the best in terms of distinguishing between the null and alternative hypotheses. The curve corresponding to $\phi = 0.95$ shows that REMLc is outperformed by PW and REML in our simulation, suggesting that although REMLc offers a decrease in the false positive rate for highly autocorrelated data relative to the other methods, it does not correctly identify as many truly active voxels.

6.2.3 Testing independence of residuals

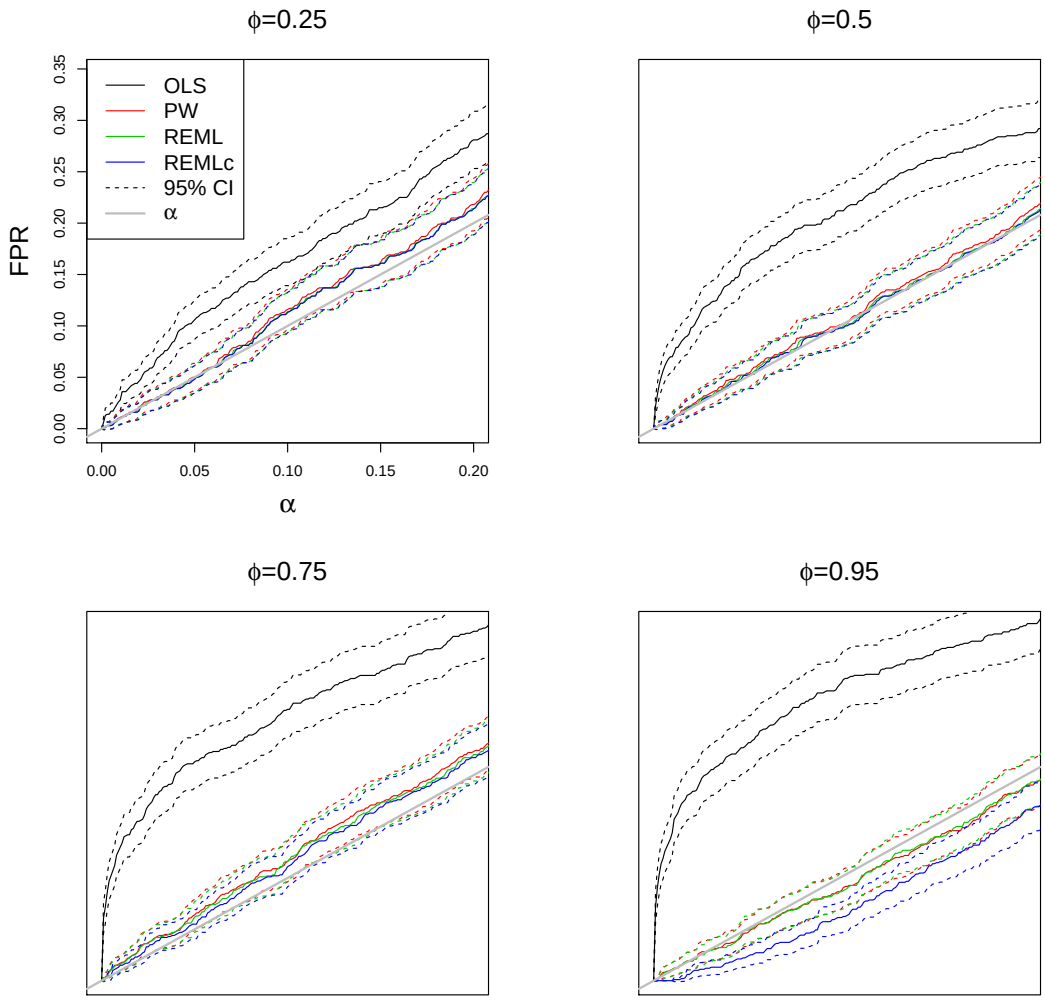
In the previous section, we used false positive rates to compare several methods that attempt to capture the autocorrelation in fMRI time series. To do this, we simulated non-active voxels by letting the true $\beta_1 = 0$ in M_{001}. In practice, how-

Table 6.2: False positive rates for simulated fMRI data

α	OLS	PW	REML	REMLc
	$\phi = 0.25$			
0.001	0.006	0.001	0.001	0.001
0.010	0.028	0.011	0.010	0.010
0.050	0.106	0.050	0.049	0.048
	$\phi = 0.50$			
0.001	0.022	0.002	0.002	0.002
0.010	0.070	0.010	0.009	0.007
0.050	0.161	0.054	0.052	0.052
	$\phi = 0.75$			
0.001	0.061	0.001	0.001	0.000
0.010	0.130	0.017	0.016	0.015
0.050	0.213	0.064	0.062	0.055
	$\phi = 0.95$			
0.001	0.072	0.000	0.000	0.000
0.010	0.144	0.011	0.011	0.000
0.050	0.230	0.046	0.049	0.023

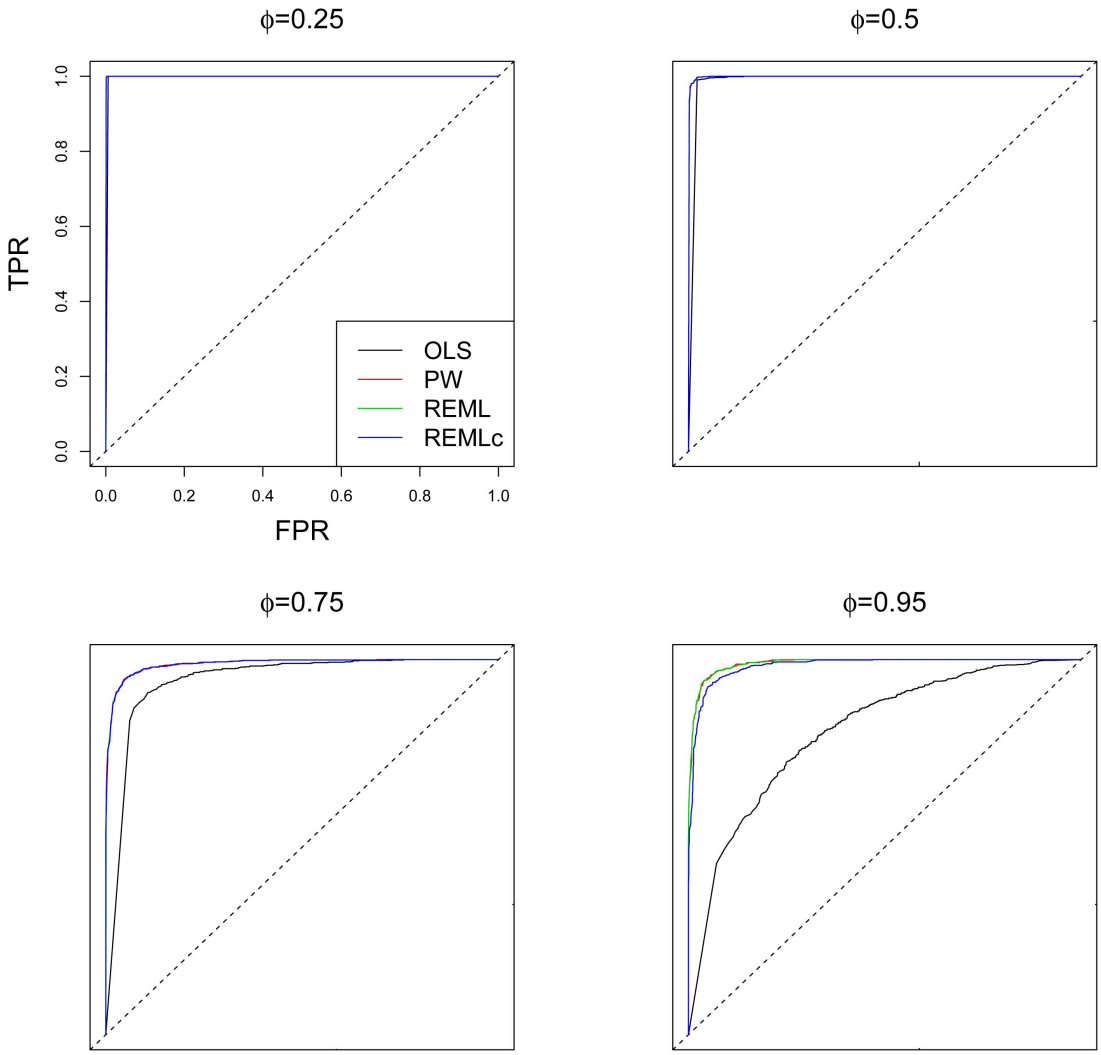
False positive rates at significance levels $\alpha = 0.001, 0.01,$ and 0.05 (rows) for testing $H_0 : \beta_1 = 0$ vs $H_A : \beta_1 > 0$ using OLS, PW, REML, and REMLc (columns) on 1000 simulated data sets of length $T = 250$ from M_{100} with $\beta = (750, 3)$, $\sigma^2 = 15$, and for each $\phi \in \{0.25, 0.50, 0.75, 0.95\}$.

Figure 6.3: False positive rates for simulated fMRI data



False positive rates (FPR, solid lines) and 95% confidence intervals (dashed lines) for testing $H_0 : \beta_1 = 0$ vs $H_A : \beta_1 > 0$ plotted against the nominal threshold level α (gray line) using OLS (black lines), PW (red lines), REML (green lines), and REMLc (blue lines) on simulated data from M_{100} with $T = 250$, $\beta = (750, 0)$, $\sigma^2 = 15$, and increasing ϕ (plot panels). Each panel is based on the same set of 1000 simulations (using the specified ϕ) for all values of α, and plot axes are the same across all panels.

Figure 6.4: ROC curves for simulated fMRI data



ROC curves for testing $H_0 : \beta_1 = 0$ vs $H_A : \beta_1 > 0$ using OLS (black lines), PW (red lines), REML (green lines), and REMLc (blue lines) on simulated data from M_{100} with $T = 250$, $\beta = (750, 3)$, $\sigma^2 = 15$, and increasing ϕ (plot panels). In each panel, the vertical axis is the true positive rate (TPR) and the horizontal axis is the false positive rate (FPR). The axes are the same in all panels. Diagonal dashed line would occur if TPR and FPR equalled each other.

ever, we don't know beforehand which voxels are inactive, making false positive rates difficult to obtain. Many studies have attempted to analyze false positive rates using *resting state* data, or fMRI data generated from a subject at rest (Purdon and Weisskoff; 1998; Burock and Dale; 2000; Woolrich et al.; 2001). These studies have effectively provided insight into the impact of certain autocorrelation estimation algorithms, sampling rates, and experimental designs on false positive rates. However, it is difficult to assess whether voxels from a subject who is resting are really inactive. For example, the subject's mind could wander during the experiment and generate a BOLD signal. Furthermore, research suggests that some brain areas exhibit some intrinsic activation during resting state, such as those associated with the default mode network (Greicius et al.; 2003, 2009).

An alternate strategy that has been used for evaluating different models of autocorrelation in fMRI data is to examine the residuals from a fitted model and test whether or not they are uncorrelated (Luo and Nichols; 2003; Leonski et al.; 2008). In particular Leonski et al. (2008) compares several autocorrelation estimation algorithms used in different software packages such as SPM and FSL, and then suggests that an AR(2) process be used for modeling the errors in fMRI time series. The latter recommendation is based in part on the fact that residuals from AR(2) fits to fMRI data used in their study were determined to be uncorrelated more often than for residuals from other models according statistical tests such as the Durbin-Watson and cumulative periodogram tests.

While an AR(2) error structure may be appropriate for modeling some data sets, we contend that an analysis of residuals should not be the basis for this decision. This is because of the potential for overfitting. To illustrate, we fit regression models with each of independent, AR(1), and AR(2) error structures to the 1000 simulated data sets of length $T = 250$ described in Section 6.2.2 with $\beta = (750, 3)'$, $\sigma^2 = 15$, and for each $\phi \in \{0.25, 0.50, 0.75, 0.95\}$. The regression models with independent errors were fit using OLS, and those with AR(1) and AR(2) errors were fit using the `arima` function in R. For each fitted model, we tested the independence of the residuals using the Ljung-Box test for lag-1 autocorrelations (Ljung and Box; 1978). Specifically, we test the null hypothesis (H_0) that the residuals are independently distributed against the alternative hypothesis (H_A) that they are not independently distributed by assuming that the test statistic

$$Q = \frac{T(T+2)}{\hat{\omega}^2(T-1)} \tag{6.21}$$

can be approximated by a chi-squared distribution with 1 degree of freedom under H_0, where $\hat{\omega}$ is the lag-1 sample autocorrelation in the residuals.

The results in Table 6.3 show that even though the true data-generating model has AR(1) errors, H_0 was not rejected when using residuals from the AR(2) fit just as often if not more often than when using residuals from the AR(1) fit. For this reason, we do not recommend evaluating models with autocorrelated errors based on an assessment of independence of residuals. Instead, in Section 6.4, we explore a model comparison strategy based on PL.

Table 6.3: Proportion of times null hypothesis of independent errors was not rejected for simulated fMRI data

α	OLS	AR(1)	AR(2)
	$\phi = 0.25$		
0.001	0.332	1.000	1.000
0.010	0.134	1.000	1.000
0.050	0.028	1.000	1.000
	$\phi = 0.50$		
0.001	0.000	1.000	1.000
0.010	0.000	1.000	1.000
0.050	0.000	0.999	1.000
	$\phi = 0.75$		
0.001	0.000	1.000	1.000
0.010	0.000	1.000	1.000
0.050	0.000	0.994	1.000
	$\phi = 0.95$		
0.001	0.000	1.000	1.000
0.010	0.000	0.991	1.000
0.050	0.000	0.958	1.000

Proportion of times H_0 of independent residuals is not rejected in favor of H_A of non-independent residuals based on 1000 simulations of length $T = 250$ from M_{100} with $\beta = 3$, $\sigma^2 = 15$, and increasing ϕ (embedded tables), as determined by Ljung-Box test at varying significance levels α (rows) from fitting regression models with independent (OLS), AR(1), and AR(2) error structures.

6.3 Fitting dynamic regression models

We now turn our attention to the dynamic regression models described in Section 2.3.3. Specifically, we examine the dynamic intercept model (M_{101}), dynamic slope model (M_{011}), and a model with both a dynamic intercept and a dynamic slope (M_{111}). M_{101} can be thought of as a regression model with AR(1)+WN errors, which has been used to analyze fMRI time series (Purdon and Weisskoff; 1998; Burock and Dale; 2000) and is similar to the model used in SPM (Kiebel and Holmes; 2007). Of particular interest to us is the possibility of modeling fMRI time series using M_{011} or M_{111} . While models with a constant slope and autocorrelation included only in the error term of the regression model, as in M_{101}, have been the norm for analyzing fMRI data, we explore the possibility that a model with a changing slope, such as M_{011}, can improve on existing methods through the ability to adapt to behaviors, such as learning or changes in focus, on the part of the subject.

6.3.1 Identifiability of dynamic regression models

Before applying the dynamic regression models to actual fMRI data, we examine whether we can identify these models using simulated data. That is, if we simulated multiple time series from M_{011}, for example, with the same true values of the model parameters, could we expect to recover these true values from the data? Since maximum likelihood estimators of unknown fixed parameters in

these models are asymptotically normal and consistent (Section 10.1 Casella and Berger; 2002), we'd expect that MLEs obtained from fitting repeated simulations from M_{011} to the same model would crowd around the true parameter values if the generated time series were long enough. We must investigate if the voxel-wise time series generated from the word recognition experiment are long enough to identify model parameters in this way.

To investigate this, we simulated 1000 time series of length $T = 250$ (in the actual word recognition experiment analyzed in Sections 6.2.1, 6.3.2, and 6.4.5, $T = 245$) from each of M_{101}, M_{011}, and M_{111} with $u_t = \text{conv}_t$, the expected BOLD response from the simulated experiment pictured in the middle panel of Figure 6.2. We let the true $\beta = (750, 15)'$ and $\sigma_m^2 = 10$ in each of these simulation models, and repeated the simulations for various values of ϕ, σ_s^2, ρ, and σ_b^2 (these last two parameters are only relevant for M_{111}). Specifically, for M_{101} and M_{011}, we simulate for all combinations of $\phi \in \{0.1, 0.5, 0.9\}$ and $\sigma_s^2 \in \{1, 5, 10, 15, 20\}$. For M_{111}, we simulate for all combinations of $\phi \in \{0.3, 0.6, 0.9\}$, $\sigma_s^2 \in \{1, 5, 10, 15, 20\}$, $\sigma_b^2 \in \{1, 5, 10, 15, 20\}$, and $\rho \in \{0.3, 0.6, 0.9\}$. The choice of these particular values for β and range of values for the variance terms was motivated by the MLEs from fitting real fMRI data in Section 6.3.2.

For each simulated time series, we calculate MLEs for the unknown fixed parameters using the `dlmMLE` function in R package `dlm` (Petris et al.; 2009). We use this function instead of `arima` because it allows us to incorporate the

observation error, v_t, into the model. However, `dlmMLE` operates on models of the form given by equations (2.8) and (2.9), where the additional regression term $U_t\beta$ is not included. Thus, to use this function to find MLEs of fixed parameters in the dynamic regression models, β must be incorporated into x_t and U_t into F_t. To allow for estimation via `dlmMLE`, we therefore reformulate M_{011} and M_{101} as

$$y_t = \tilde{F}_t \tilde{x}_t + v_t \tag{6.22}$$

$$\tilde{x}_t = \tilde{G}\tilde{x}_{t-1} + w_t, \tag{6.23}$$

where $\tilde{x}_t = (\beta_0, \beta_1, x_t)'$, $\tilde{F}_t = (1, \text{conv}_t, F_t)$,

$$\tilde{G} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \phi \end{pmatrix},$$

and

$$v_t \overset{iid}{\sim} \text{N}(0, \sigma_m^2) \perp w_t \overset{iid}{\sim} \text{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \sigma_s^2 \end{pmatrix} \right).$$

For M_{011}, we let $F_t = \text{conv}_t$, and for M_{101}, we let $F_t = 1$ for all t. In these models, x_t represents either the change in regression slope or the change in the regression intercept, respectively, as in Section (2.3.3).

To allow maximum likelihood estimation of model parameters in M_{111} using R function `dlmMLE`, we reformulate the model from equations (2.23) through (2.25) into the form of equations (6.22) and (6.23) with $\tilde{x}_t = (\beta_0, \beta_1, x_{1,t}, x_{2,t})'$, $\tilde{F}_t =$

$(1, \text{conv}_t, 1, \text{conv}_t)$,

$$\tilde{G} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \phi & 0 \\ 0 & 0 & 0 & \rho \end{pmatrix},$$

and

$$v_t \overset{iid}{\sim} \text{N}(0, \sigma_m^2) \perp w_t \overset{iid}{\sim} \text{N}\left(\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \sigma_s^2 & 0 \\ 0 & 0 & 0 & \sigma_b^2 \end{pmatrix} \right).$$

Here, $x_{1,t}$ and $x_{2,t}$ represent the change in the regression intercept and slope, respectively.

Similar to `arima`, `dlmMLE` uses a call to `optim` to minimize the negative log likelihood, expressed as

$$-\log p(y_{1:T}|\theta) \propto \frac{1}{2}\sum_{t=1}^{T} \log|Q_t| + \frac{1}{2}\sum_{t=1}^{T}(y_t - \tilde{F}_t z_t)' Q_t^{-1}(y_t - \tilde{F}_t z_t) \tag{6.24}$$

(Petris et al.; 2009, Chapter 4). Here, z_t and Q_t depend implicitly on θ and are calculated according to the Kalman filter given by equation (2.30) with $\tilde{F}_t$ and $\tilde{G}$ used in place of F_t and G, respectively, and initial values $m_0 = C_0 = 0$ (to constrain $x_0 = 0$).

Maximum likelihood estimates of ϕ, σ_s^2, and σ_m^2 (and ρ and σ_b^2 for M_{111}) – denoted $\hat{\phi}$, $\hat{\sigma}_s^2$, and $\hat{\sigma}_m^2$ (and $\hat{\rho}$ and $\hat{\sigma}_b^2$) – are obtained directly from `dlmMLE`, while MLEs for β are obtained from the first and second elements of m_T after running

the Kalman filter in equation (2.30) conditional on $\hat{\phi}$, $\hat{\sigma}_s^2$, and $\hat{\sigma}_m^2$ (and $\hat{\rho}$ and $\hat{\sigma}_b^2$) with $m_0 = C_0 = 0$, $F_t = \tilde{F}_t$, $V = \hat{\sigma}_m^2$, $G = \tilde{G}$, and $W = \tilde{W}$. Since U_t and β are already included in F_t and x_t, respectively, the Kalman filter is implemented with the middle line in equation (2.30) reading $f_t = \tilde{F}_t z_t$ (instead of $f_t = U_t\beta + \tilde{F}_t z_t$).

Unlike in Section 6.2.1, we do not restrict ϕ (or ρ) to the region of stationarity. This is intended to enable modeling of a wider range of behavior in fMRI data, as well as placing priors on the unknown fixed parameters that are conjugate conditional on the states, so that estimation using the particle learning algorithm (described in Section 3.2.5) can be performed. In some cases, we obtain estimates of ϕ that lay outside the region of stationarity when analyzing fMRI time series from the word recognition data set using maximum likelihood in Section 6.3.2 and particle learning in Section 6.4.5.

Figure 6.5 displays either univariate histograms or two-dimensional kernel density estimates (Wand and Ripley; 2006) of the MLEs for fits of M_{011} to data simulated from the same model with $\phi = 0.1$ and increasing signal-to-noise ratio σ_s^2/σ_m^2. We characterize the model as being "well identified" if the maximum likelihood estimates appear to be approximately normally distributed and concentrated around the true parameter values, since asymptotic distribution theory for MLEs guarantees the MLEs for the fixed parameters are asymptotically normal and consistent (Casella and Berger; 2002, Section 10.1). In this figure, β appears to be well identified for all values of the signal-to-noise ratio, as evidenced by

the two-dimensional kernel density estimates in the first column of Figure 6.5 that show an ellipse with a clear mode near the true value. However, for true $\sigma_s^2/\sigma_m^2 = 0.1$, ϕ, σ_s^2, and σ_m^2, don't appear to be well identified. By increasing $\sigma_s^2/\sigma_m^2 > 0.1$, identification of these parameters seems to improve.

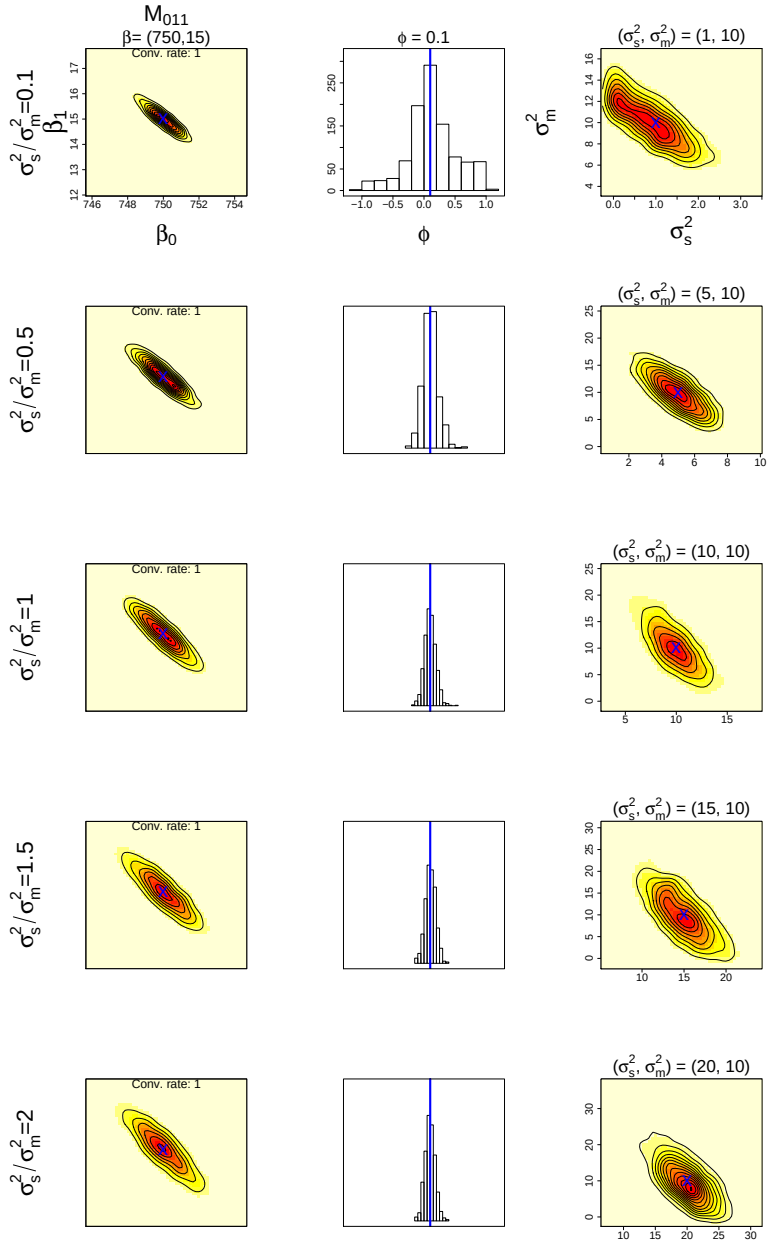
Figure 6.6 shows similar plots with σ_s^2/σ_m^2 fixed at 0.1 and increasing true values of ϕ. While identification of ϕ, σ_s^2, and σ_m^2 appears to be poor for $\phi = 0.1$, a drastic improvement is shown by increasing ϕ to 0.5 and 0.9. β, again, is identified well for all combinations of fixed parameter values shown in the figure. In addition, the rate at which the `optim` function in R successfully converged at the minimum of the negative log likelihood for a given set of 1000 simulations using fixed true values of the unknown parameters (displayed along the top of the plots in the first column of Figures 6.5 and 6.6) is 1 for all true values of σ_s^2/σ_m^2, indicating that a clear maximum value of the likelihood always exists for these simulated time series from M_{011}.

Lastly, we comment on the skewness of the distribution of the MLEs for ϕ when the true value is 0.9, as illustrated by the histogram in the last row of Figure 6.6. Asymptotic distribution theory for MLEs guarantees that this distribution should appear more bell-shaped as T approaches infinity (Casella and Berger; 2002, Section 10.1). However, for finite T, this distribution is skewed toward lower values of ϕ. This is because values of ϕ larger than 1 lead to a model with a nonstationary dynamic slope, behavior that is fundamentally different from

that exhibited by the simulated data with a stationary dynamic slope. Similar results using larger T (not shown) show less skewness and smaller variance in the distribution for ϕ, with a more bell-shaped histogram concentrated around the true value of $\phi = 0.9$.

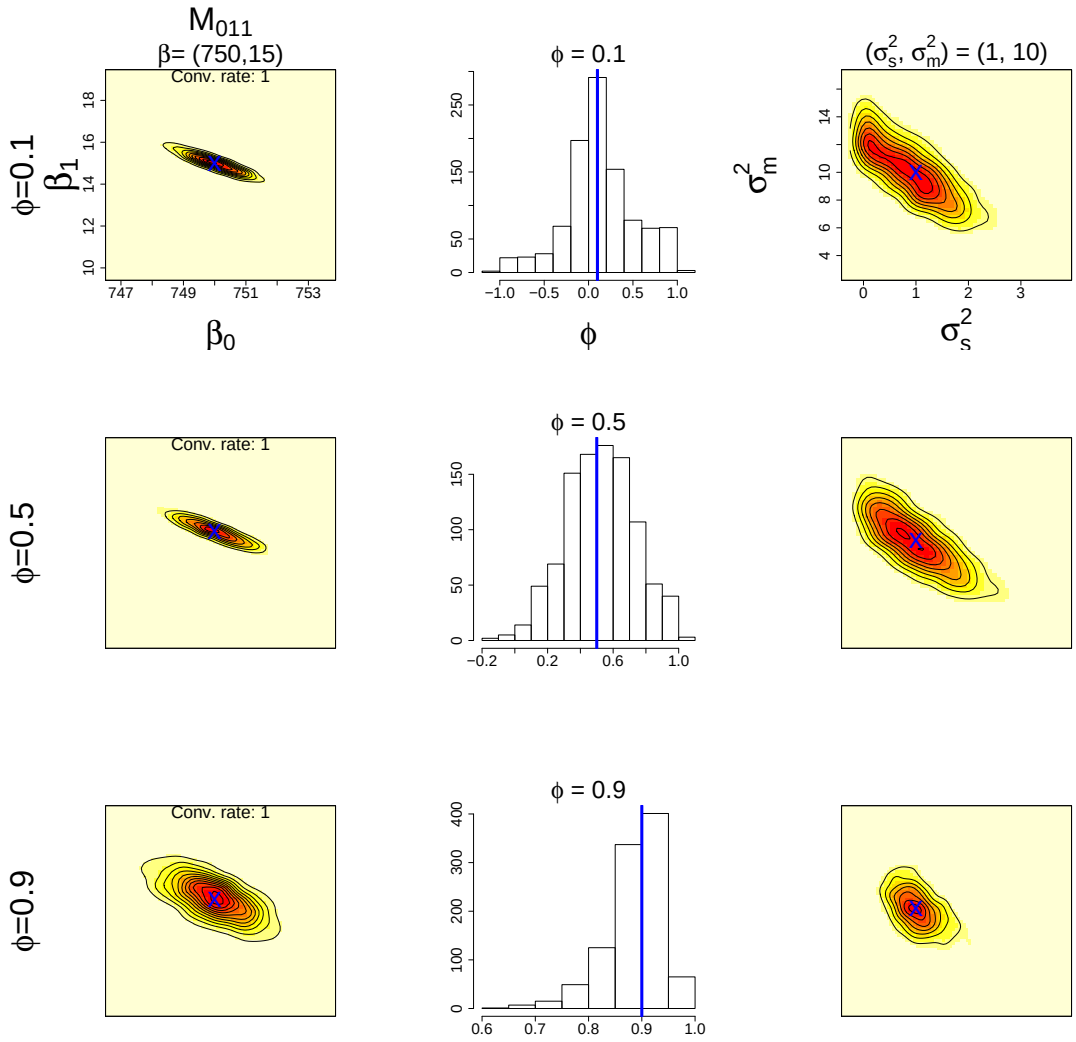
Figures 6.5 and 6.6 provide evidence that M_{011} is well-identified provided the true signal-to-noise ratio and true autocorrelation coefficient are not too low. Similar plots shown in Figures 6.7 and 6.8 reveal that identifiability of true model parameters in M_{101} is more challenging. When the true ϕ is fixed at 0.1, identification of ϕ, σ_s^2, and σ_m^2 in M_{101} is poor for all $\sigma_s^2 \in \{1, 5, 10, 15, 20\}$, indicated by the bimodal two-dimensional kernel density estimates of the MLEs for (σ_s^2, σ_m^2) and non-Gaussian distributions of MLEs for ϕ shown in Figure 6.5. However, when the true σ_s^2/σ_m^2 is fixed at 0.1, increasing the true ϕ to 0.9 results in much better identification of these model parameters, as illustrated by the histograms and two-dimensional kernel density estimates of the MLEs for ϕ and (σ_s^2, σ_m^2), respectively, that concentrate near their corresponding true values in the last row of Figure 6.8. In each of these cases, the distributions of the MLEs for β appear to be normally distributed around the true values. The challenge in identifying true parameter values in M_{101}, relative to M_{011}, is further highlighted by the existence of a small percentage of simulations for which the `optim` function in R does not successfully converge at the minimum of the negative log likelihood (see conver-

Figure 6.5: Identifying dynamic slope model by increasing signal-to-noise ratio



Histograms in 1D (for ϕ, second column) and 2D kernel density estimates (for β in first column and (σ_s^2, σ_m^2) in the third column) of MLEs of fits of M_{011} to data simulated from M_{011} with true $\beta = (750, 15)'$, $\phi = 0.1$, $\sigma_m^2 = 10$, and increasing σ_s^2 (rows). Blue crosses indicate the true values of β and (σ_s^2, σ_s^2) in each of the corresponding image panels, and blue vertical lines indicate the true value of ϕ. Plot axes are the same within the first and second columns, but differ within the third column due to differing true signal to noise ratios σ_s^2/σ_m^2.

Figure 6.6: Identifying dynamic slope model by increasing autocorrelation



Histograms in 1D (for ϕ, second column) and 2D kernel density estimates (for β in first column and (σ_s^2, σ_m^2) in third column) of MLEs of fits of M_{011} to data simulated from M_{011} with true $\beta = (750, 15)'$, $\sigma_s^2 = 1$, $\sigma_m^2 = 10$, and increasing ϕ (rows). Blue crosses indicate the true values of β and (σ_s^2, σ_s^2) in each of the corresponding image panels, and blue vertical lines indicate the true values of ϕ. Plot axes are the same within the first and third columns, but differ within the second column due to differing true lag-1 autocorrelations ϕ.

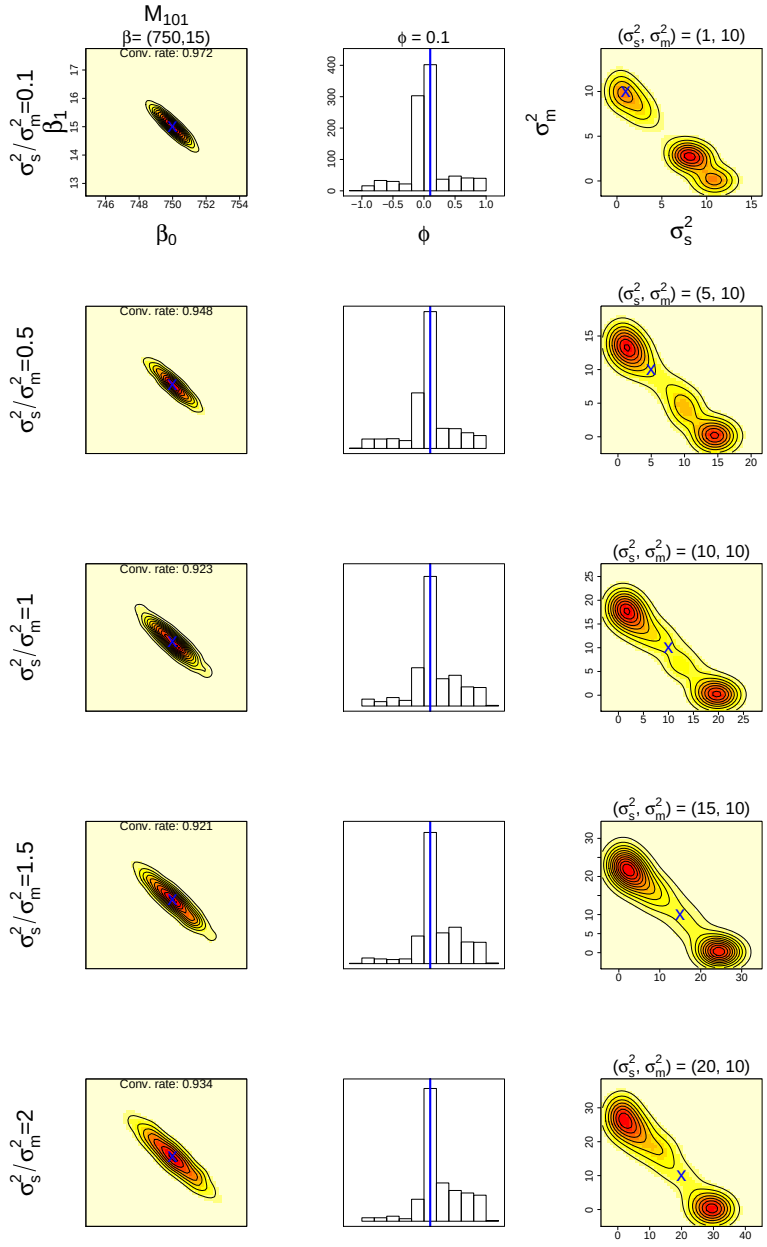
gence rates displayed along the top of the plots in the first column of Figures 6.5 and 6.6).

Lastly, identification of model parameters in M_{111} appears to be the most challenging. Figure 6.9 shows that for fixed $\phi = 0.9$, $\rho = 0.6$, $\sigma_b^2 = 1$, and $\sigma_m^2 = 10$, increasing the true white noise variance of the dynamic intercept, σ_s^2, to 15 improves identification of σ_m^2 while the distributions of MLEs for ϕ and σ_b^2 are skewed and not centered at the true values. On the other hand, if σ_b^2 is increased to 20 as in Figure 6.10, identification of ϕ and σ_b^2 improves with increasing σ_s^2, but the distribution of MLEs for σ_m^2 is now skewed and off-center. As with M_{011} and M_{101}, β appears to be better identified than other model parameters.

Unimodal and elliptical distributions of MLEs for β under all three models provides some confidence that we can expect estimates of the regression coefficients to be unbiased. However, when the true autocorrelation and signal-to-noise ratio are low, bimodal or non-normal distributions of MLEs appear for other model parameters, suggesting that inference on β, and specifically the conclusion of the hypothesis test in equation (6.4), could be incorrect. Figures 6.5 through 6.10 suggest that out of the three models, M_{011} is the most likely to be well identified for a given set of values for true model parameters, while M_{111} is least likely to be well-identified.

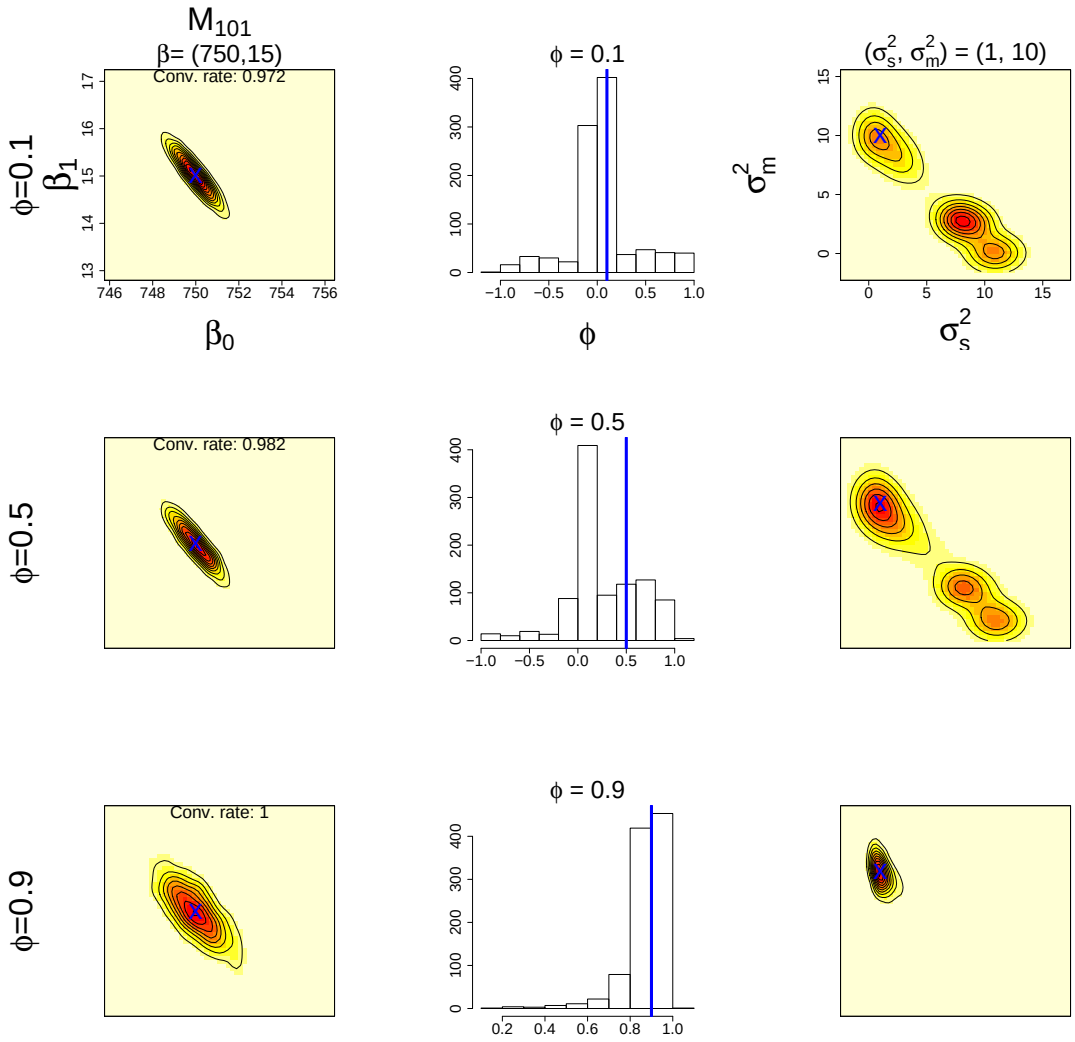
Poor identification of M_{111} is further illustrated by the decrease in the proportion of simulations for which the `optim` function successively converges to the

Figure 6.7: Identifying dynamic intercept model by increasing signal-to-noise ratio



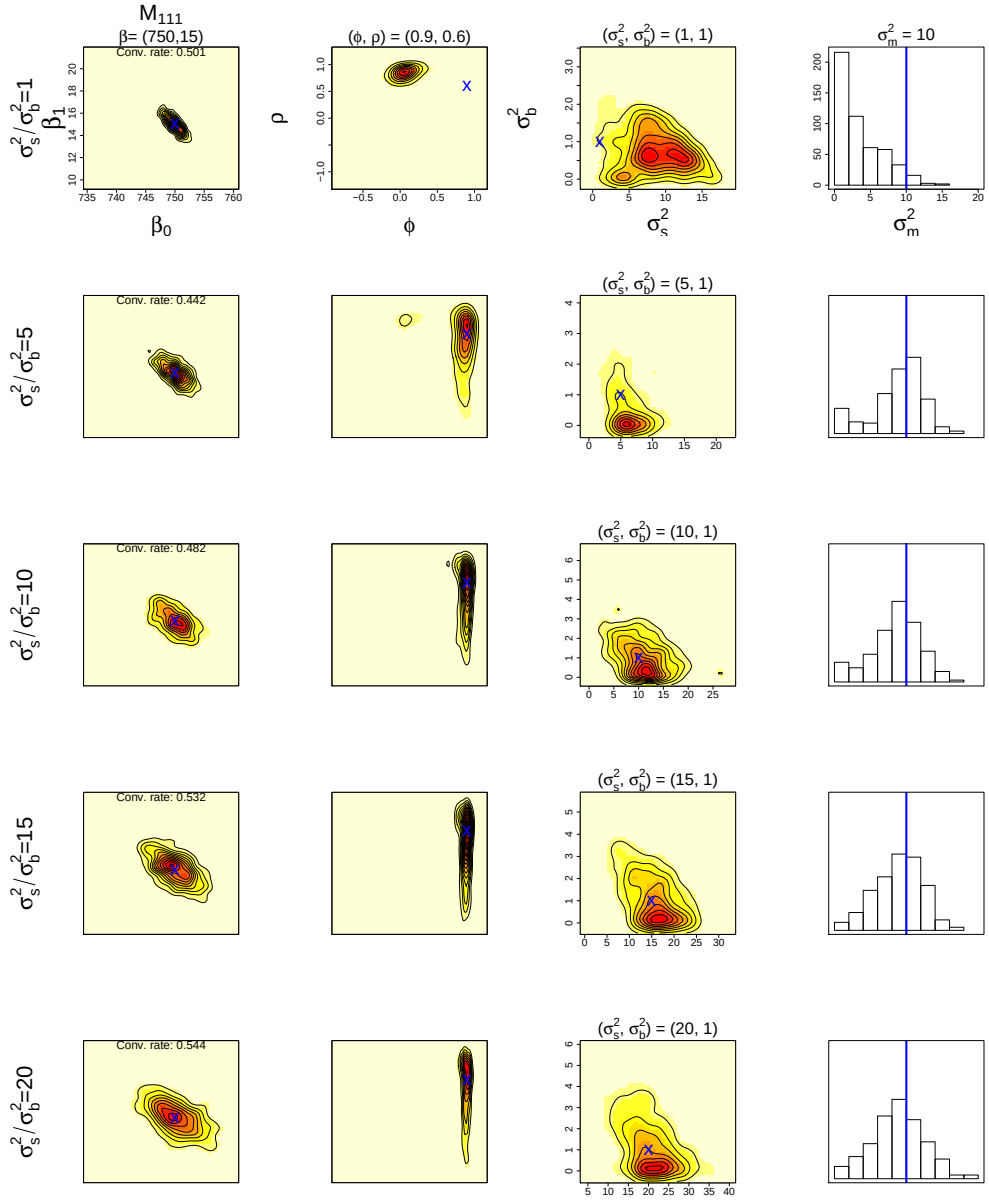
Histograms in 1D (for ϕ, second column) and 2D kernel density estimates (for β in first column and (σ_s^2, σ_m^2) in third column) of MLEs of fits of M_{101} to data simulated from M_{101} with true $\beta = (750, 15)'$, $\phi = 0.1$, $\sigma_m^2 = 10$, and increasing σ_s^2 (rows). Blue crosses indicate true values of β and (σ_s^2, σ_s^2) in each of the corresponding image panels, and blue vertical lines indicate the true value of ϕ. Plot axes are the same within the first and second columns, but differ within the third column due to differing true signal to noise ratios σ_s^2/σ_m^2.

Figure 6.8: Identifying dynamic intercept model by increasing autocorrelation



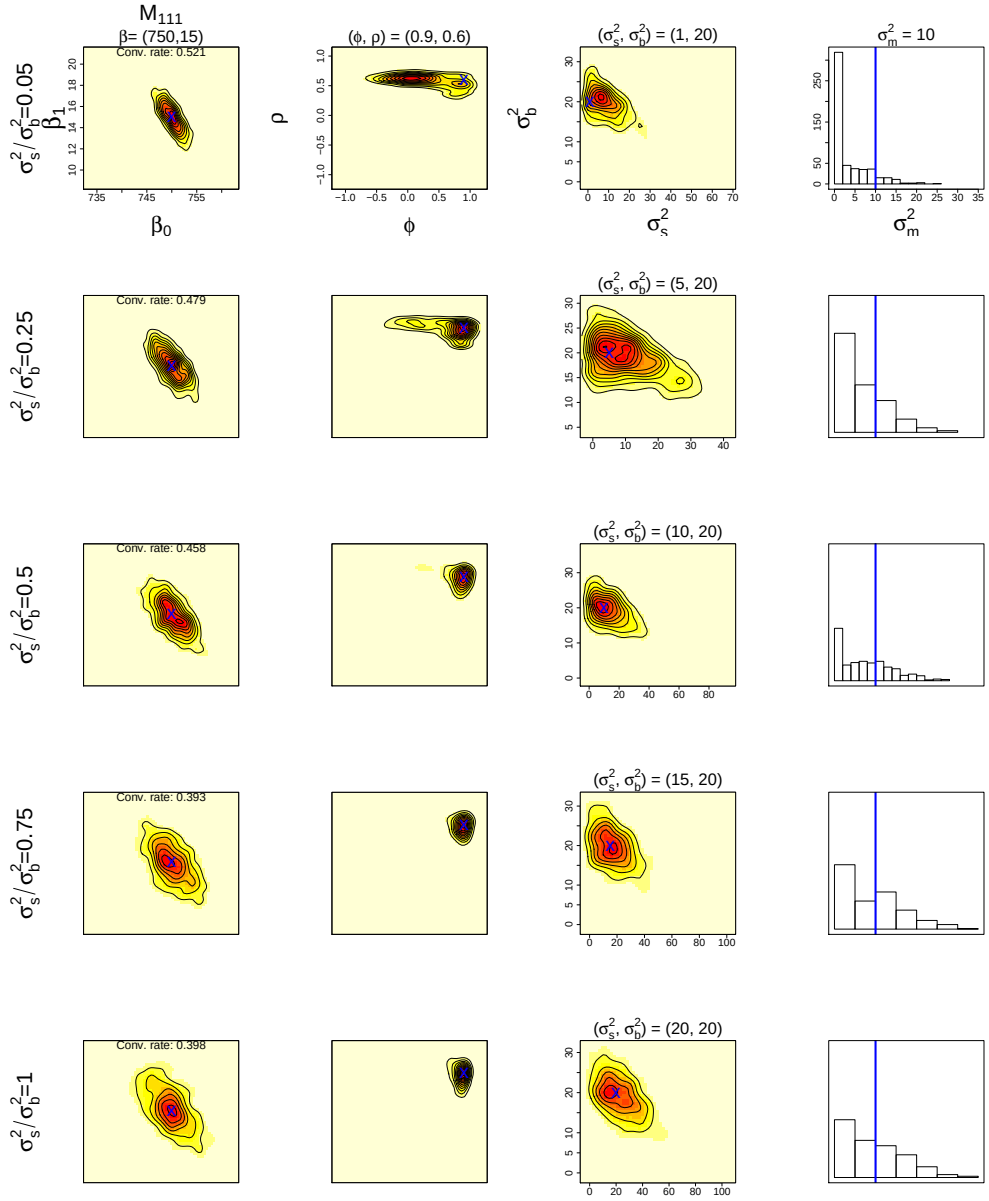
Histograms in 1D (for ϕ, second column) and 2D kernel density estimates (for β in first column and (σ_s^2, σ_m^2) in third column) of MLEs of fits of M_{101} to data simulated from M_{101} with true $\beta = (750, 15)'$, $\sigma_s^2 = 1$, $\sigma_m^2 = 10$, and increasing ϕ (rows). Blue crosses indicate the true values of β and (σ_s^2, σ_s^2) in each of the corresponding image panels, and blue vertical lines indicate the true values of ϕ. Plot axes are the same within the first and third columns, but differ within the second column due to differing true lag-1 autocorrelations ϕ.

Figure 6.9: Identifying model with both dynamic slope and intercept with small slope variance



Histograms in 1D (for σ_m^2, last column) and 2D kernel density estimates (for β in first column, (ϕ, ρ) in second column, and (σ_s^2, σ_b^2) in third column) of MLEs of fits of M_{111} to data simulated from M_{111} with true $\beta = (750, 15)'$, $\phi = 0.9$, $\rho = 0.6$, $\sigma_b^2 = 1$, $\sigma_m^2 = 10$, and increasing σ_s^2 (rows). Blue crosses indicate true values of β, (ϕ, ρ), and (σ_s^2, σ_b^2) in each of the corresponding image panels, and blue vertical lines indicate the true value of σ_m^2. Plot axes are the same within the first, second, and fourth columns, but differ within the third column due to differing true signal to noise ratios σ_s^2/σ_b^2.

Figure 6.10: Identifying model with both dynamic slope and intercept with large slope variance



Histograms in 1D (for σ_m^2, last column) and 2D kernel density estimates (for β in first column, (ϕ, ρ) in second column, and (σ_s^2, σ_b^2) in third column) of MLEs of fits of M_{111} to data simulated from M_{111} with true $\beta = (750, 15)'$, $\phi = 0.9$, $\rho = 0.6$, $\sigma_b^2 = 20$, $\sigma_m^2 = 10$, and increasing σ_s^2 (rows). Blue crosses indicate the true values of β, (ϕ, ρ), and (σ_s^2, σ_b^2) in each of the corresponding image panels, and blue vertical lines indicate the true value of σ_m^2. Plot axes are the same within the first, second, and fourth columns, but differ within the third column due to differing true signal to noise ratios σ_s^2/σ_b^2.

MLEs (see the convergence rates displayed under the plots for β in Figures 6.9 and 6.10). Models with identifiability issues are sometimes characterized by flat likelihoods that can make it difficult to find local maxima using iterative routines.

Identifiability of true model parameters in all three models improves by increasing the length of the simulated time series, T, or by simulating with the explanatory variable $u_t \overset{iid}{\sim} \mathrm{N}(0,1)$, instead of simulating with correlated explanatory variable conv_t (results not shown). Improvement in identifiability when simulating using uncorrelated explanatory variable, compared to simulations with conv_t, may be associated with confounding caused by autocorrelation present in both the regression covariate and the error term when conv_t is used (Hodges and Reich; 2010). However, these are factors that we don't have much control over in fMRI experiments, since increasing the length of fMRI scanning sessions is expensive, and the expected BOLD response from an fMRI experiment typically exhibits autocorrelation due to nature of the hrf. Due to these identifiability concerns, we discard M_{111} at this point and examine only M_{011} and M_{101} in the remainder of this chapter.

6.3.2 Fitting word recognition data

We now provide results from fitting M_{101} and M_{011} to the word recognition data set, using maximum likelihood estimation. As in Section 6.3.1, the `dlmMLE` function was used to obtain MLEs for all the fixed parameters in M_{101} and M_{011}.

In addition, we offer a comparison with standard GLM fits of M_{001} using OLS. That is, we use OLS to fit the model of the form shown in equation (2.13) with $U_t = (1, \text{conv}_t)$, $\beta = (\beta_0, \beta_1)'$, $F_t = 0$ for all t (making the state equation (2.14) irrelevant), and $v_t \overset{iid}{\sim} \text{N}(0, \sigma_m^2)$. OLS estimates are obtained according to equation (6.5), and in this case we let $\hat{\sigma}_m^2 = \hat{\sigma}^2$ from equation (6.5). We let $\hat{\theta} = (\hat{\beta}', \hat{\phi}, \hat{\sigma}_s^2, \hat{\sigma}_m^2)'$ denote the MLEs of the fixed parameters in M_{011} and M_{101}, and $\hat{\theta} = (\hat{\beta}', \hat{\sigma}_m^2)'$ be the MLEs of the fixed parameters in M_{001}. All three model were fit with conv_t as the expected BOLD response for the word recognition experiment shown in the middle panel of Figure 6.1.

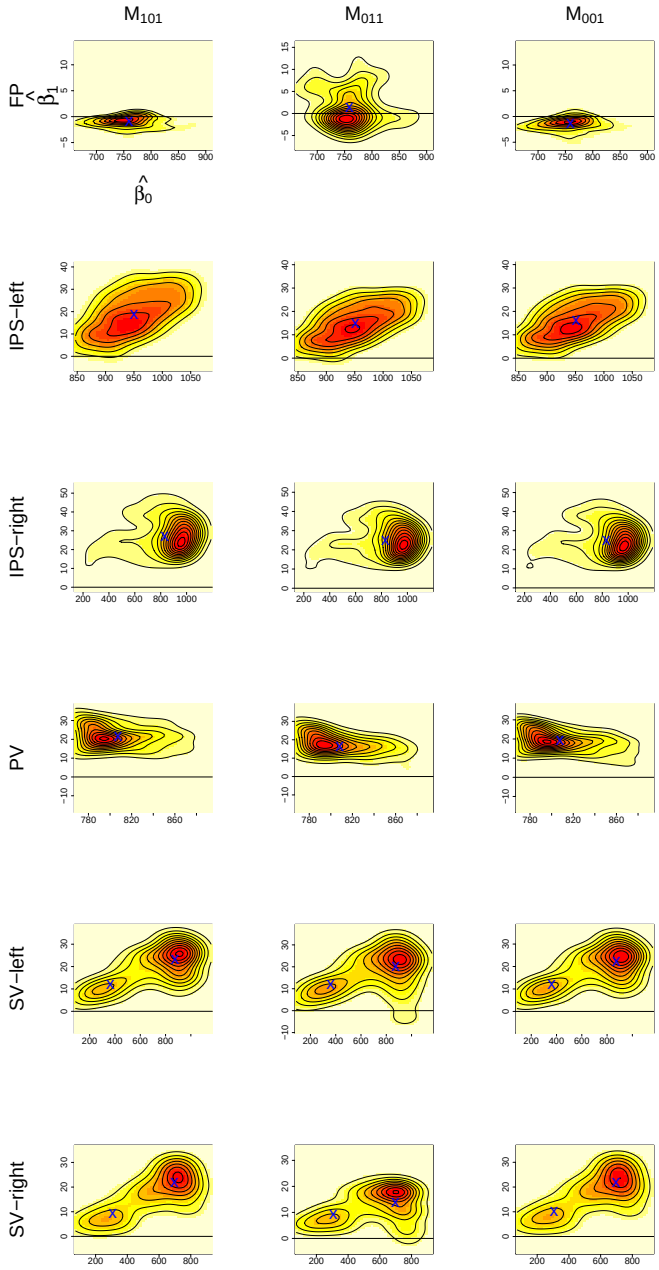
Figure 6.11 displays two-dimensional kernel density estimates of $\hat{\beta}$ over the 125 voxels from each of the 6 brain regions in Table 6.1 according to each model. These density estimates suggest that significant brain activation is present in all regions except for FP. Also, estimates are relatively consistent across models for all brain regions with the exception of FP, where histograms for M_{011} include a cloud of more active voxels while the other two models don't.

Also apparent from this figure is the bimodal nature of brain activation in SV-left and SV-right. For each model, voxels in these two regions were divided into high and low clusters, denoted by "Cluster H" and "Cluster L", respectively, using the k-means clustering method (Hartigan and Wong; 1978) applied to $\hat{\theta}$. Table 6.6 shows that about 65% of voxels from these two regions fall into a cluster of higher activation and higher baseline BOLD response, evidenced by the higher

regional average values of $\hat{\beta}_1$ and $\hat{\beta}_0$ in Table 6.5. The clustering of voxels in these brain regions could provide support for the M_{011} model, since an apparent change in the regression slope over such close space might suggest that a change in the regression slope over time might also be reasonable.

Unlike MLEs for β, MLEs for ϕ, σ_s^2, and σ_m^2 are not consistent between M_{011} and M_{101}. Tables 6.4 and 6.5 show that in IPS-left, IPS-right, and PV, estimates for ϕ are, on average, higher under M_{011} than they are under M_{101}. In addition, the average signal-to-noise ratio, σ_s^2/σ_m^2, is estimated to be much higher in these three regions for M_{101}. In contrast, average estimates of ϕ in SV-left and SV-right are lower under M_{011} than they are under M_{101}, and the opposite relationship is true for the average estimates of σ_s^2/σ_m^2. The opposing nature of autocorrelation and signal-to-noise estimates between M_{101} and M_{011} suggests that these models account for variation and autocorrelation in the data differently. What M_{101} models as increased signal-to-noise, M_{011} interprets as increased autocorrelation in the dynamic slope. Conversely, what M_{011} models as increased signal-to-noise, M_{101} interprets as increased autocorrelation in the errors.

Figure 6.11: Kernel density estimates of MLEs of regression coefficients



Two-dimensional kernel density estimates of MLEs for β using fMRI data from a word recognition experiment extracted from six brain regions (rows) fit to models M_{011}, M_{101}, and M_{001} (columns). Blue crosses denote the marginal averages of the MLEs from each brain region, and for each of two clusters in SV-left and SV-right.

Table 6.4: Average MLEs in single cluster brain regions

Parameter	FP	IPS-left	IPS-right	PV
	M_{011}			
β_0	759.155	951.101	831.359	808.257
β_1	1.395	15.009	24.894	16.492
ϕ	0.736	0.853	0.871	0.832
σ_s^2	8.746	27.268	53.171	70.646
σ_m^2	10.031	22.522	41.340	21.826
	M_{101}			
β_0	759.875	950.038	830.901	807.434
β_1	-1.032	18.879	26.838	21.247
ϕ	0.746	0.637	0.654	0.596
σ_s^2	3.323	16.970	29.565	23.498
σ_m^2	6.105	0.534	1.382	1.727
	M_{001}			
β_0	759.204	950.773	831.191	807.846
β_1	-1.448	16.087	24.688	19.039
σ_m^2	12.480	29.579	54.511	37.629

Average MLEs calculated marginally for each fixed parameter (rows) using fMRI data from a word recognition experiment extracted from four different brain regions (columns) based on fitting models M_{011}, M_{101}, and M_{001} (embedded tables).

Table 6.5: Average MLEs in bi-cluster brain regions

Parameter	SV-left		SV-right	
	Cluster H	Cluster L	Cluster H	Cluster L
	M_{011}			
β_0	874.999	363.601	697.816	308.080
β_1	23.133	12.203	21.767	9.415
ϕ	0.566	0.520	0.489	0.630
σ_s^2	11.475	4.378	13.162	4.889
σ_m^2	0.502	0.393	2.423	3.906
	M_{101}			
β_0	875.319	362.054	698.025	307.655
β_1	20.009	11.708	13.883	9.361
ϕ	0.821	0.761	0.979	0.864
σ_s^2	11.941	7.986	1.727	7.427
σ_m^2	14.116	4.953	16.345	8.719
	M_{001}			
β_0	875.084	361.957	697.769	307.538
β_1	22.414	12.135	21.723	10.168
σ_m^2	17.409	6.332	19.450	11.108

Average MLEs calculated marginally for each fixed parameter (rows) using fMRI data from a word recognition experiment extracted from each cluster of secondary visual left and secondary visual right (columns) fit to dynamic regression models (embedded tables).

Table 6.6: Proportion of voxels with high activation

Model	SV-left	SV-right
M_{011}	0.672	0.648
M_{101}	0.677	0.648
M_{001}	0.672	0.648

Proportion of voxels in each of secondary visual left and right (columns) classified into high activation cluster after applying the k-means clustering algorithm to MLEs of θ under each model (rows).

6.4 Comparing dynamic regression models using particle learning

In the previous section, we explored fits of M_{011}, M_{101}, and M_{001} to the word recognition data using maximum likelihood estimation. In this section, we investigate the relative appropriateness of these models for the data using an SMC model comparison strategy. Results in Chapter 5 showed that the performance of PL is superior to the RM and KDPF in terms of efficiently and accurately estimating the marginal likelihood and posterior model probabilities within the context of the local level DLM with common state and observation variance factor. Since the dynamic regression models we consider for fMRI data admit tractable forms of the distributions needed to implement the PL, we use this algorithm to compare models in this section.

In Section 3.2.5, we described a PL scheme to estimate the filtered distributions of states and unknown fixed parameters in M_{011} (letting $F_t = \text{conv}_t$) and M_{101} (letting $F_t = 1$). Using this algorithm, we also have a way of estimating the marginal likelihood of the data through equation (3.36). For M_{001}, an exact form of the marginal likelihood is available (O'Hagan; 1994), given by

$$p(y_{1:T}) = \frac{1}{(2\pi)^{T/2}} \sqrt{\frac{|B_0^{-1}|}{|B_T^{-1}|}} \left(\frac{(b_{m_0})^{a_{m_0}}}{(b_{m_T})^{a_{m_T}}} \right) \left(\frac{\Gamma(a_{m_T})}{\Gamma(a_{m_0})} \right), \tag{6.25}$$

where

$$B_T = (X'X + B_0^{-1})^{-1} \qquad \vartheta_T = B_T(X'y + B_0^{-1}\vartheta_0) \tag{6.26}$$

$$a_{m_T} = a_{m_0} + T/2 \qquad b_{m_T} = b_{m_0} + \frac{1}{2}(y'y + \vartheta_0' B_0^{-1}\vartheta_0 - \vartheta_T' B_T^{-1}\vartheta_T),$$

and ϑ_0, B_0, a_{m_0}, and b_{m_0} are prior hyperparameters that are assumed known (see equation 2.26). In equation (6.26), y and X are the data vector and design matrix, respectively, as defined in equation (6.5). Given the exact marginal likelihood of the data under M_{001}, and approximations to the marginal likelihood under M_{101} and M_{011}, relative posterior model probabilities among the three models can be computed according to equation (3.37).

In order to implement PL on fMRI time series data from a single voxel, we need to specify the prior distribution, $p(x_0, \theta)$, and the number of particles to use in the particle filter. For all three models under consideration, we use a prior of the form given by equations (2.26) and (2.27), i.e. $p(x_0, \theta) = p(\beta|\sigma_m^2)p(\sigma_m^2)p(\phi|\sigma_s^2)p(\sigma_s^2)\delta_0(x_0)$ with

$$\beta|\sigma_m^2 \sim \mathrm{N}(\vartheta_0, \sigma_m^2 B_0) \qquad \sigma_m^2 \sim \mathrm{IG}(a_{m_0}, b_{m_0}) \tag{6.27}$$

$$\phi|\sigma_s^2 \sim \mathrm{N}(\varphi_0, \sigma_s^2 \Phi_0) \qquad \sigma_s^2 \sim \mathrm{IG}(a_{s_0}, b_{s_0}). \tag{6.28}$$

For M_{001}, only equation (6.27) is needed (since $\phi = \sigma_s^2 = 0$). The hyperparameters ϑ_0, B_0, φ_0, Φ_0, a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} are assumed known, and we let B_0 and Φ_0

take the form

$$B_0 = \kappa^2 \begin{pmatrix} 1000 & 0 \\ 0 & 225 \end{pmatrix} \quad \Phi_0 = \kappa^2 \times 0.25. \tag{6.29}$$

We let $\kappa = 1$ in Sections 6.4.1 and 6.4.5, but let $\kappa > 0$ in Section 6.4.2 to examine the sensitivity of the marginal likelihood of the data to more diffuse priors.

6.4.1 Analyzing simulated fMRI data using particle learning

In this section, we simulate data from each of M_{011} and M_{101}, and we test the PL algorithm by running it under the true model for increasing number of particles. Based on the resulting particle samples, we obtain estimates of the filtered distributions of the dynamic regression coefficients and fixed parameters. Specifically, we compare sequential 95% credible intervals for unknown states and fixed parameters obtained from PL with those from running the MCMC algorithm described in Section 3.1.2.

Time series of length $T = 250$ were simulated from both models with true fixed parameter values set to $\beta = (750, 15)'$, $\phi = 0.95$, $\sigma_s^2 = 10$, and $\sigma_m^2 = 10$. The same conv_t generated from the simulated rapid event-design illustrated in Figure 6.2 was used as the regression covariate. The simulated time series, y_t, and the simulated change in the regression slope, x_t, from M_{011} are pictured in Figure 6.12. Notice from this figure that y_t mirrors the convolution function better at

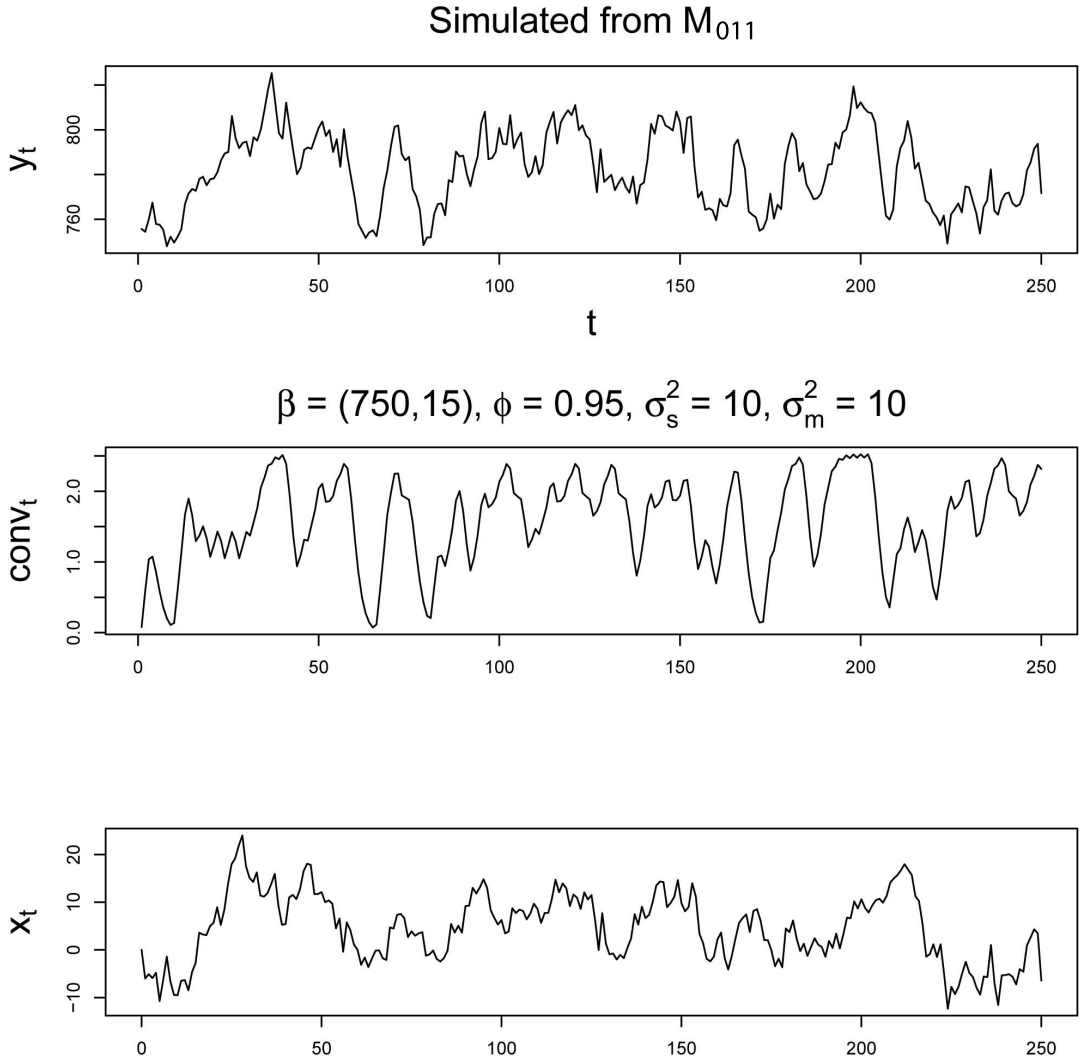
TRs where x_t is high, since higher values of the dynamic slope amplify the signal in the data relative to the noise.

For both PL and MCMC, the prior distribution on the initial state and fixed parameters were specified according to equations (6.27) and (6.28) with $\kappa = 1$. We let the hyperparameters b_0 and φ_0 be equal to the true values of β and ϕ used for simulation, respectively. The inverse-gamma hyperparameters a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} were set such that the prior means for each of σ_s^2 and σ_m^2 were equal to their respective true values used for simulation, and such that each prior variance was equal to $\kappa^2 \times 500$.

We ran the PL algorithm under the true model on time series simulated from both models using 500, 1000, 5000, 10000, and 20000 particles. In addition, we ran three MCMC chains for each simulation under the true model using the same priors. For each chain, initial values of β_0, β_1, ϕ were set to 0 and initial values for σ_s^2 and σ_m^2 were set to 1. Initial values of the states, x_t for $t = 0, 1, \ldots, T$, were drawn from a standard normal distribution. Twenty-five thousand iterations were run for each chain including a burn-in period of 5000 iterations, and every 20th iteration was saved.

The results for M_{011} shown in Figure 6.13 indicate that the filtered distributions of the fixed parameters and dynamic slope estimated by PL seem to have converged if at least 10000 particles are used. In addition, the filtered distributions at time $t = T = 250$ appear to agree with the MCMC estimates for the dynamic

Figure 6.12: Simulated fMRI data from dynamic slope model



Simulated fMRI time series y_t (top) from M_{011} with $\beta = (750, 15)'$, $\phi = 0.95$, $\sigma_s^2 = 10$, and $\sigma_m^2 = 10$. Convolution of the hrf and neural activation pattern conv_t and simulated change in dynamic slope (x_t) are displayed in the middle and bottom panels, respectively.

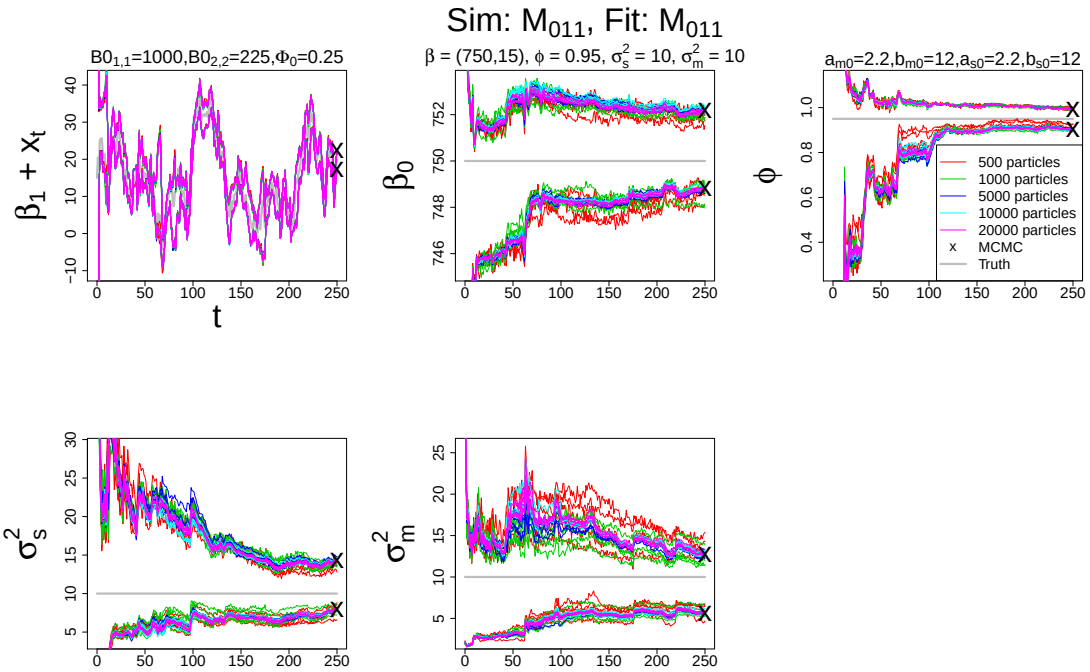
slope and each fixed parameter. MCMC estimates can only be compared with PL at $t = T$ since MCMC provides smoothed estimates of the dynamic slope and fixed parameters for $t < T$, while PL provides only filtered estimates. Analagous plots for the data simulated from M_{101} (Figure 6.14) show similar results for the dynamic intercept and fixed parameters in M_{101}.

6.4.2 Distinguishing dynamic regression models using particle learning

The results from the previous section indicate that stable estimates of dynamic regression coefficients and fixed parameters can be obtained by PL if enough particles are used. We now examine estimation of the marginal likelihood using PL. In particular, we are interested in gaining understanding about the parameter settings under which the true model can be distinguished amongst M_{011}, M_{101}, and M_{001} by looking at the marginal likelihood.

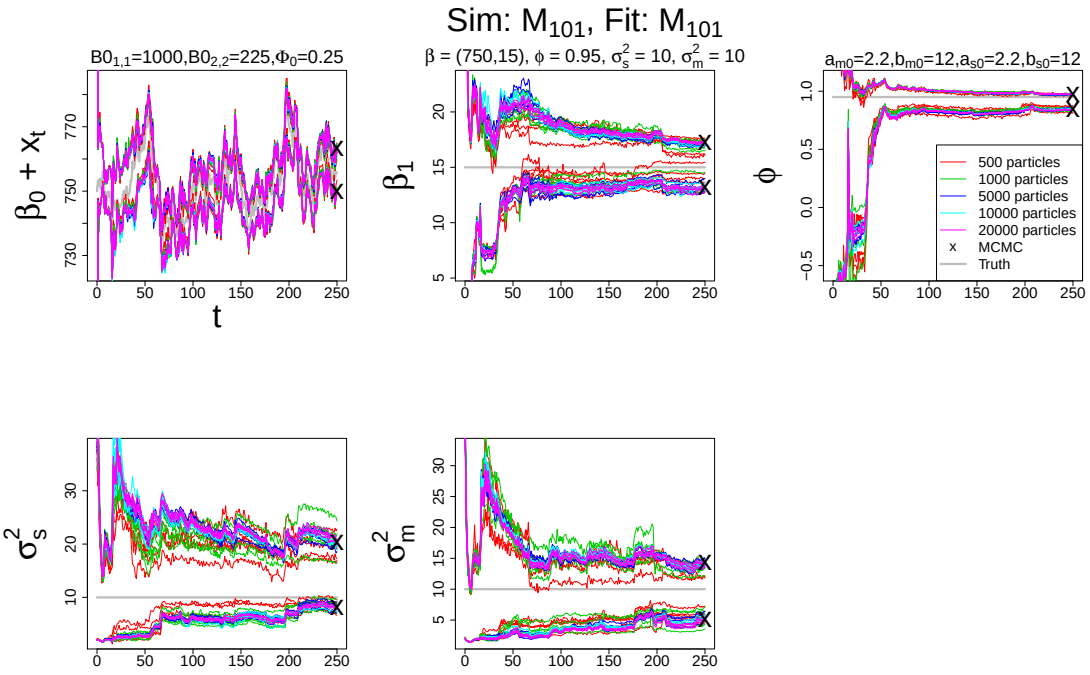
To study this, we simulated time series of length $T = 250$ from both M_{011} and M_{101} with the same convolution function used in the previous section, $\beta = (750, 15)'$, and various values of ϕ, σ_s^2, and σ_m^2. Specifically, we simulated time series for all combinations of $\phi \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.99\}$, $\sigma_s^2 \in \{1, 2, 3, 4, 5, 10, 15, 20\}$, and $\sigma_m^2 = 10$. We then ran the PL algorithm, under both M_{011} and M_{101}, three times on each simulated time series using 500 particles. The prior hyperparameters B_0 and Φ_0 were specified by letting $\kappa = 1$ in equation

Figure 6.13: Credible intervals from PL compared with MCMC for simulated fMRI data



Sequential 95% credible intervals for the dynamic slope (top left) and fixed parameters (other panels) in M_{011} using PL with increasing number of particles (colors) compared with MCMC (black crosses, only displayed for $T = 250$ since MCMC was run using entire data set) run on simulated data of length $T = 250$ from M_{011} with true $\beta = (750, 15)'$, $\phi = 0.95$, $\sigma_s^2 = 10$, and $\sigma_m^2 = 10$ (displayed above top middle panel). The true values of fixed parameters used for simulation and the true simulated dynamic slopes are represented by gray lines/curves, respectively. Credible interval estimates from MCMC are displayed only for $\beta_1 + x_T$ and for each of the fixed parameters conditional on all the data ($T = 250$). The same prior distributions on the initial state and fixed parameters were used for running both PL and MCMC, with $p(x_0) = \delta_0(x_0)$, b_0 and ϕ_0 set to the true β and ϕ, respectively, and the remaining hyperparameters displayed above the top left and right panels.

Figure 6.14: Credible intervals from PL compared with MCMC for simulated fMRI data



Sequential 95% credible intervals for the dynamic intercept (top left) and fixed parameters (other panels) in M_{101} using PL with increasing number of particles (colors) compared with MCMC (black crosses, only displayed for $T = 250$ since MCMC was run using entire data set) run on simulated data from M_{101} with $\beta = (750, 15)'$, $\phi = 0.95$, $\sigma_s^2 = 10$, and $\sigma_m^2 = 10$ (displayed above top middle panel). The true values of fixed parameters used for simulation and the true simulated dynamic slopes are represented by gray lines/curves, respectively. Credible interval estimates from MCMC are displayed only for $\beta_0 + x_T$ and for each of the fixed parameters conditional on all the data ($T = 250$). The same prior distributions on the initial state and fixed parameters were used for running both PL and MCMC, with $p(x_0) = \delta_0(x_0)$, b_0 and ϕ_0 set to the true β and ϕ, respectively, and the remaining hyperparameters displayed above the top left and right panels.

(6.29). For each simulation, we calculated the MLEs of the unknown fixed parameters using `dlmMLE`, as in Section 6.3, prior to running the PL, and we let the hyperparameters b_0 and φ_0 be equal to the MLEs of β and ϕ, respectively. The inverse-gamma hyperparameters a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} were set such that the prior means for each of σ_s^2 and σ_m^2 were equal to their respective MLEs, and such that each prior variance was equal to $\kappa^2 \times 500$. The marginal likelihood is sensitive to specification of the prior distribution, as we discuss further in Section 6.4.3. While setting priors in this way can be construed as data snooping, we do this in an attempt to limit the influence of the prior on comparison of the three models via the marginal likelihood.

Some PL runs suffered from numerical instability caused by values of the conditional likelihood, $p(y_t|x_t, \theta)$, being too low to be evaluated for some particles. For each PL run that did not encounter this issue, we computed estimates of the log marginal likelihood. In addition, we computed the exact log marginal likelihood under M_{001} for each simulated time series using equation (6.25).

Figures 6.15 and 6.16 show the results for the data simulated from M_{011} and M_{101}, respectively. Both figures indicate that the log marginal likelihood under the true data-generating model is larger than the log marginal likelihood under the other models provided the true ϕ and signal-to-noise ratio, σ_s^2/σ_m^2, are large enough. However, it appears more difficult to identify M_{101} as the true model than it does for M_{011}. For example, when the true model is M_{011} and the true

signal-to-noise ratio is 1.5, 2, or 2.5 (bottom row of Figure 6.15), the log marginal likelihood obtained from each PL run under M_{011} is larger than those obtained from all PL runs under the other models for any $\phi \geq 0.1$. In contrast, when the true signal-to-noise ratio under M_{101} is 1.5, 2, or 2.5 (bottom row of Figure 6.16), $\phi \geq 0.6$ is required for all log marginal likelihoods obtained from PL runs under M_{101} to be larger than those obtained for all PL runs with 500 particles under the other models.

6.4.3 Sensitivity of the marginal likelihood to priors

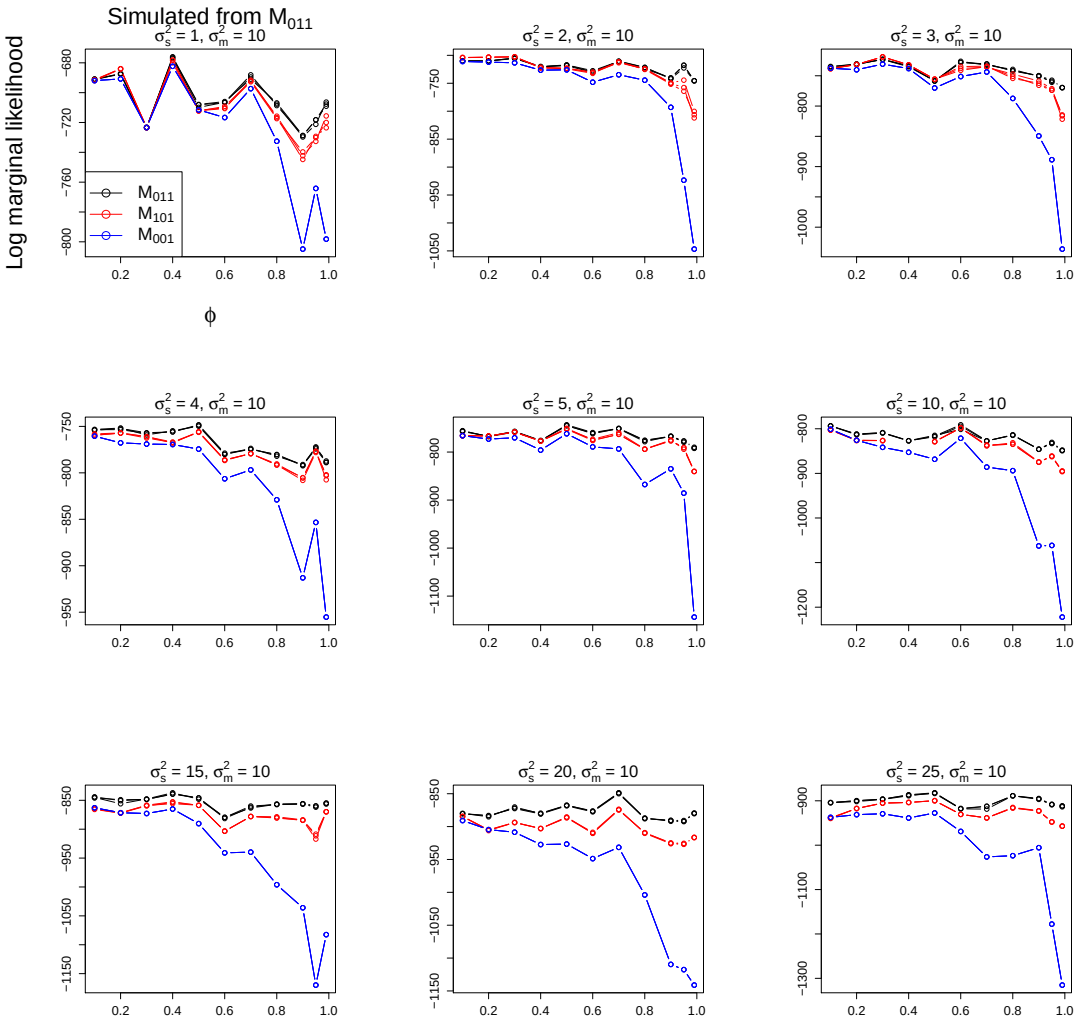
It is important to understand that comparing models in terms of the log marginal likelihood, as in Figures 6.15 and 6.16, is sensitive to the specified prior distribution on the initial state and fixed parameters. This is because the marginal likelihood is computed by integrating out the states and fixed parameters from the joint likelihood, i.e.

$$p(y_{1:T}) = \int_{\theta} \int_{x_0} \int_{x_1} \cdots \int_{x_T} \prod_{t=1}^{T} \left(p(y_t|x_t, \theta) p(x_t|x_{t-1}, \theta) \right) p(x_0, \theta) \mathrm{d}x_{0:T} \mathrm{d}\theta. \quad (6.30)$$

Thus, if $p(x_0, \theta)$ is diffuse relative to the joint posterior, $p(x_{0:T}, \theta|y_{1:T})$, $p(y_{1:T})$ will be much smaller than it would be for $p(x_0, \theta)$ that is more concentrated around $p(x_{0:T}, \theta|y_{1:T})$.

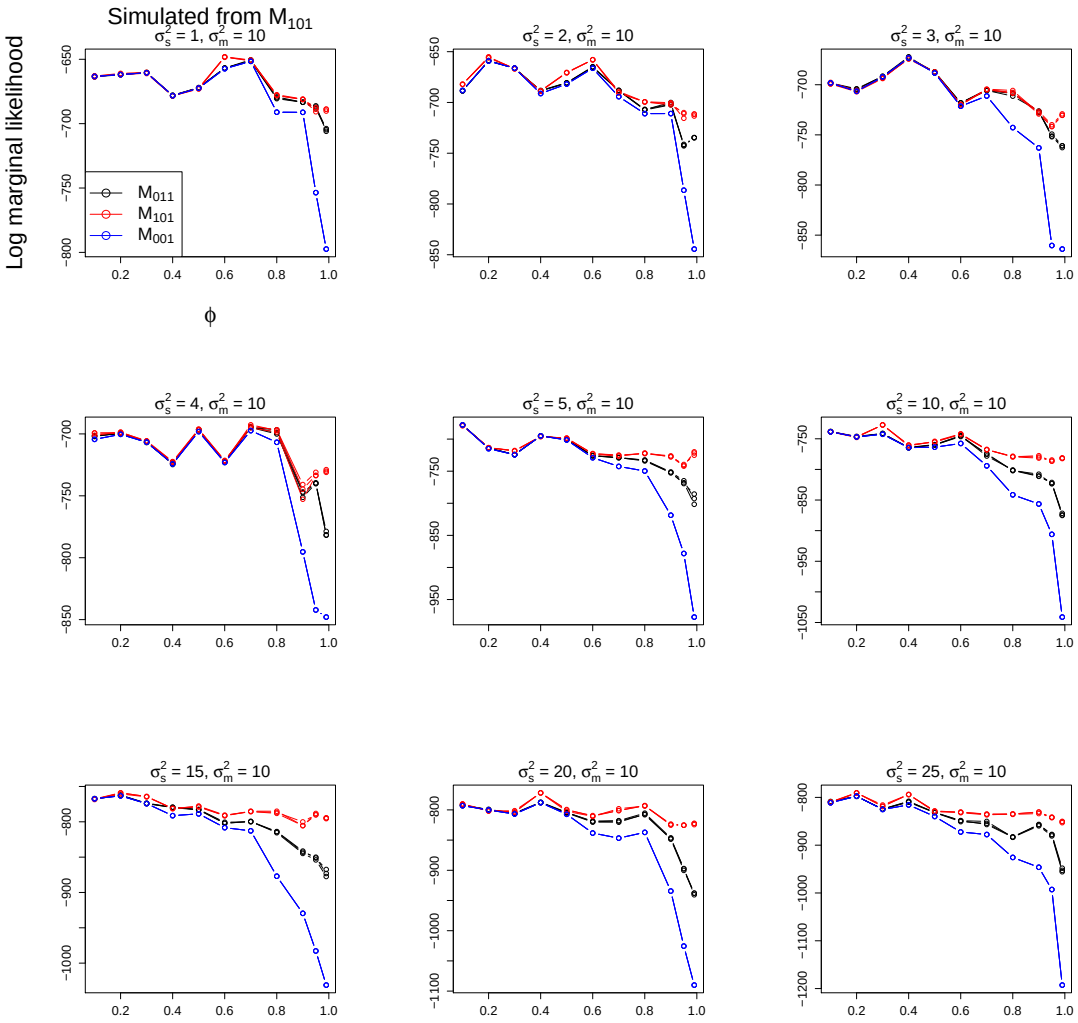
To examine the sensitivity of comparing M_{011}, M_{101}, and M_{001} to specified priors of the form given by equations (6.27) and (6.28), we simulated two more time series under both M_{011} and M_{101} for each set of fixed parameter values. Then,

Figure 6.15: Distinguishing the dynamic slope model from the dynamic intercept and simple linear regression models



Log marginal likelihoods of data simulated from M_{011} with $\beta = (750, 15)$, $\sigma_m^2 = 10$, increasing ϕ (x-axis) and increasing σ_s^2 (plot panels) under M_{011} (black lines), M_{101} (red lines), and M_{001} (blue lines). Log marginal likelihoods from three independent PL runs with 500 particles under each model for each simulation are displayed by colored points. When running the PL, prior hyperparameters B_0 and Φ_0 were specified by equation (6.29) with $\kappa = 1$, and b_0 and φ_0 were set to the MLEs of β and ϕ, respectively. The inverse-gamma hyperparameters a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} were set such that the prior means for each of σ_s^2 and σ_m^2 were equal to their respective MLEs, and such that each prior variance was equal to 500. Points are not displayed for PL runs that did not complete due to numerical instability.

Figure 6.16: Distinguishing the dynamic intercept model from the dynamic slope and simple linear regression models



Log marginal likelihoods of data simulated from M_{101} with $\beta = (750, 15)$, $\sigma_m^2 = 10$, increasing ϕ (x-axis) and increasing σ_s^2 (plot panels) under M_{011} (black lines), M_{101} (red lines), and M_{001} (blue lines). Log marginal likelihoods from three independent PL runs with 500 particles under each model for each simulation are displayed by colored points. When running the PL, prior hyperparameters B_0 and Φ_0 were specified by equation (6.29) with $\kappa = 1$, and b_0 and φ_0 were set to the MLEs of β and ϕ, respectively. The inverse-gamma hyperparameters a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} were set such that the prior means for each of σ_s^2 and σ_m^2 were equal to their respective MLEs, and such that each prior variance was equal to 500. Points are not displayed for PL runs that did not complete due to numerical instability.

we ran the PL algorithm three times under the true data-generating model using $\kappa = 5$ and 500 particles for each of the now three total simulations generated using each set of true fixed parameter values (remaining prior hyperparameters were set based on the MLEs as in Section 6.4.2). This process was then repeated for $\kappa = 10$ and $\kappa = 15$. The goal here is that we want to examine the effect of increasing κ on which true values of ϕ and σ_s^2/σ_m^2 would be required for the log marginal likelihoods obtained from all three PL runs under the true data-generating model to be higher than the log marginal likelihoods obtained from all three PL runs under the non-true dynamic regression model (either M_{011} or M_{101}) and analytical estimate of the log marginal likelihood under M_{011} with $\kappa = 1$. For example, for running the PL under M_{011} with $\kappa = 1$ on data simulated from the same model with true $\sigma_s^2/\sigma_m^2 = 1$, a true value of $\phi \geq 0.7$ is required for the log marginal likelihoods obtained from all three PL runs to be higher than those obtained from all PL runs under the other models (see plot in the second row and third column of Figure 6.15).

Theoretically, we should be able to increase κ to the point that the marginal likelihood of the true model is lower than that of the other models regardless of the true values of ϕ and σ_s^2/σ_m^2. Figures 6.17 and 6.18 illustrate this point. In Figure 6.17, it is clear that as κ is increased, larger true values of ϕ and σ_s^2/σ_m^2 are required to identify M_{011} as the true model. This phenomenon is even more pronounced when considering M_{101} as the true model, as in Figure 6.18. For

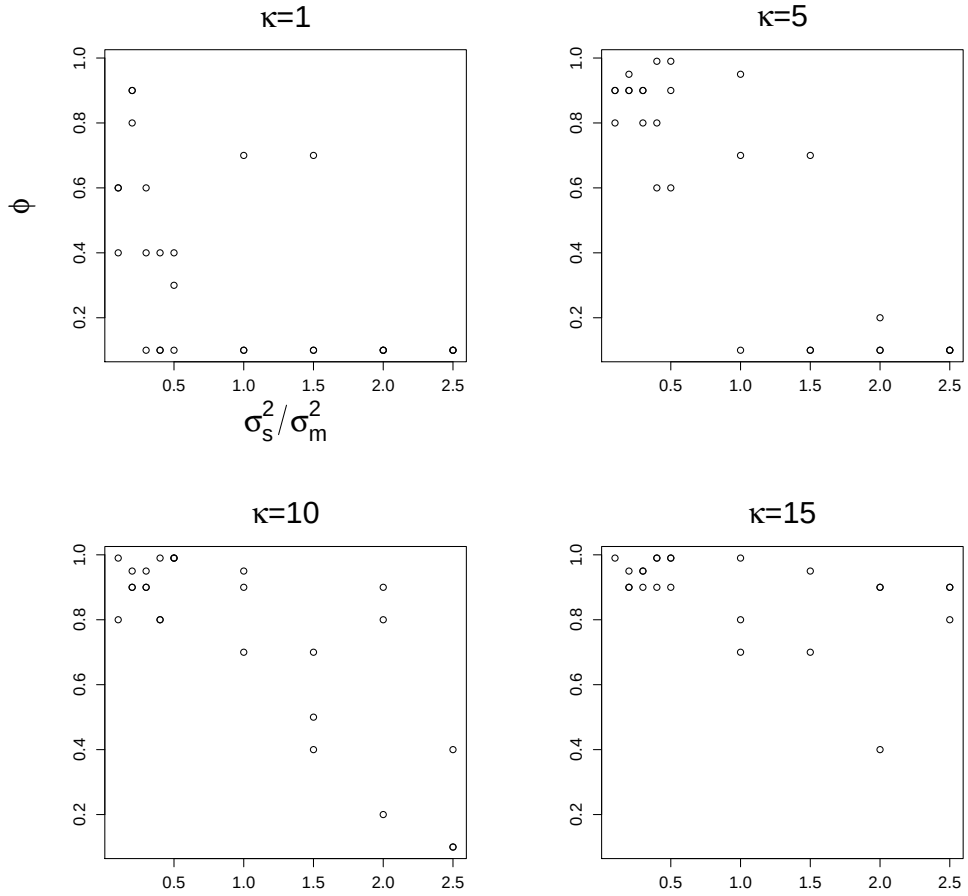
example, when $\kappa = 15$, the true lag-1 autocorrelation in the data must be at least 0.8 to identify M_{101} as the true model, regardless of how large the signal-to-noise ratio is.

We use a prior of the form given by equations (6.27) and (6.28) with $\kappa = 1$ for the remainder of this chapter, since this prior seems reasonable given the distributions of the MLEs of fixed parameters examined in Section 6.3.2 and, given the results in Figures 6.17 and 6.18, provides the best chance of identifying a true model amongst M_{011}, M_{101}, and M_{001}. Briefly, setting $\kappa = 1$ assumes a prior standard deviation of 100 for the regression intercept and 15 for the regression slope. MLEs for β from fitting these dynamic regression models to time series from voxels taken from the word recognition data set, displayed via two-dimensional kernel density estimates in Figure 6.11, appear to be within 2 prior standard deviations (with $\kappa = 1$) of the average MLE of their respective regions or clusters displayed in Tables 6.4 and 6.5. Similarly, most MLEs for ϕ, σ_s^2, and σ_m^2 (not shown) fall within two standard deviations of their respective region or cluster averages.

6.4.4 Comparing posterior model probabilities using simulated fMRI data

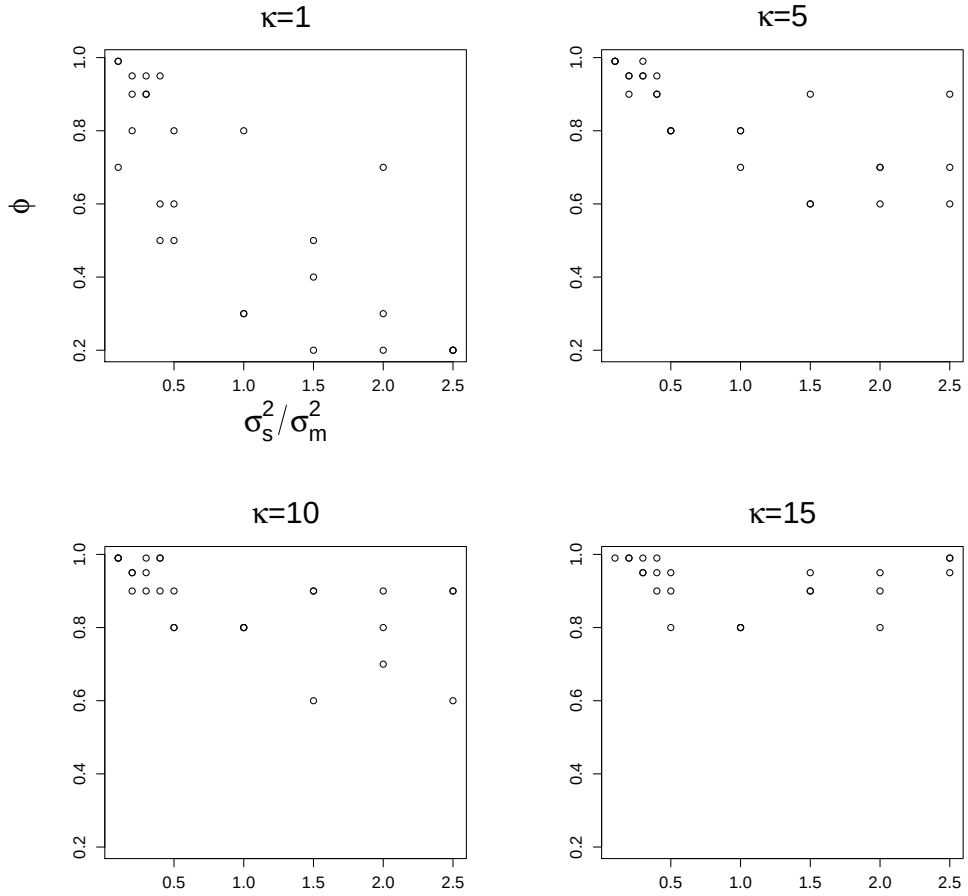
Our final analysis using simulated data is aimed at determining how many particles are needed when running the PL algorithm to accurately estimate relative

Figure 6.17: Distinguishing the true dynamic slope model M_{011} from the dynamic intercept and simple linear regression models with increasing prior variance



Minimum values of $\phi \in \{0.1, 0.2, \ldots, 0.9, 0.95, 0.99\}$ (y-axis) for which the log marginal likelihood estimates under M_{011} (obtained from three independent PL runs with 500 particles on data simulated from M_{011}) with $\beta = (750, 15)'$, $\sigma_m^2 = 10$, and fixed $\sigma_s^2 \in \{0.1, 0.2, \ldots, 0.5, 1.0, 1.5, 2.0, 2.5\}$ (x-axis) each exceed all the log marginal likelihood estimates from the three PL runs on the same data under M_{101}, and also exceed the analytical estimates of the log marginal likelihood under M_{001}. Each panel corresponds to one of four increasing κ values (where κ partially determines the prior hyperparameters under M_{011} - as described further in Section 6.4.2 and equation (6.29)). Results are shown for three separate sets of simulations from M_{011}, where each set consists of simulated time series under all combinations of the aforementioned fixed parameter values. Prior hyperparameters under M_{101} and M_{001} were set as described in Section 6.4.2 with $\kappa = 1$. If there exists no value of ϕ for which M_{011} is distinguished from M_{101} and M_{001} for fixed σ_s^2/σ_m^2 within a given set of simulations, no point is plotted for that value of σ_s^2/σ_m^2.

Figure 6.18: Distinguishing the true dynamic intercept model M_{101} from the dynamic slope and simple linear regression models with increasing prior variance



Minimum values of $\phi \in \{0.1, 0.2, \ldots, 0.9, 0.95, 0.99\}$ (y-axis) for which the log marginal likelihood estimates under M_{101} (obtained from three independent PL runs with 500 particles on data simulated from M_{101}) with $\beta = (750, 15)'$, $\sigma_m^2 = 10$, and fixed $\sigma_s^2 \in \{0.1, 0.2, \ldots, 0.5, 1.0, 1.5, 2.0, 2.5\}$ (x-axis) each exceed all the log marginal likelihood estimates from the three PL runs on the same data under M_{011}, and also exceed the analytical estimates of the log marginal likelihood under M_{001}. Each panel corresponds to one of four increasing κ values (where κ partially determines the prior hyperparameters under M_{101} - as described further in Section 6.4.2 and equation (6.29)). Results are shown for three separate sets of simulations from M_{101}, where each set consists of simulated time series under all combinations of the aforementioned fixed parameter values. Prior hyperparameters under M_{011} and M_{001} were set as described in Section 6.4.2 with $\kappa = 1$. If there exists no value of ϕ for which M_{101} is distinguished from M_{011} and M_{001} for fixed σ_s^2/σ_m^2 within a given set of simulations, no point is plotted for that value of σ_s^2/σ_m^2.

posterior model probabilities among M_{011}, M_{101}, and M_{001}. This should depend on how different the marginal likelihoods are among the three models. For instance, if the marginal likelihood of the data under one of the models is large relative to the marginal likelihoods under the others, the posterior probability is likely to be 1 for that model and 0 for the others, even if the estimate of the marginal likelihood is highly variable. For this reason, we consider a "worst" case scenario, i.e. time series simulated from each of M_{011} and M_{101} with true fixed parameter values set such that it is difficult to distinguish the true model from among M_{011}, M_{101}, and M_{001}. Using Figures 6.15 and 6.16 as a guide, we simulate time series from each of M_{011} and M_{101} using the following true fixed parameter values:

$$M_{011} : \beta = (750, 15)' \quad \phi = 0.3 \quad \sigma_s^2 = 1 \quad \sigma_m^2 = 10 \tag{6.31}$$

$$M_{101} : \beta = (750, 15)' \quad \phi = 0.5 \quad \sigma_s^2 = 1 \quad \sigma_m^2 = 10 \tag{6.32}$$

The PL algorithm was run twenty times under both M_{011} and M_{101} on each simulated time series using 500, 1000, 5000, and 10000 particles. Again, we specify prior hyperparameters based on the MLEs as described in Sections 6.4.2 and 6.4.3 with $\kappa = 1$. By grouping together a single log marginal likelihood approximation (based on a single PL run) under M_{101}, a single log marginal likelihood approximation under M_{011}, and an exact marginal likelihood under M_{001} calculated according to equation (6.25), a set of approximate posterior probabilities among the three models can be calculated according to equation (3.37) with prior model probabilities equal to 1/3 for each model. For each of the given number

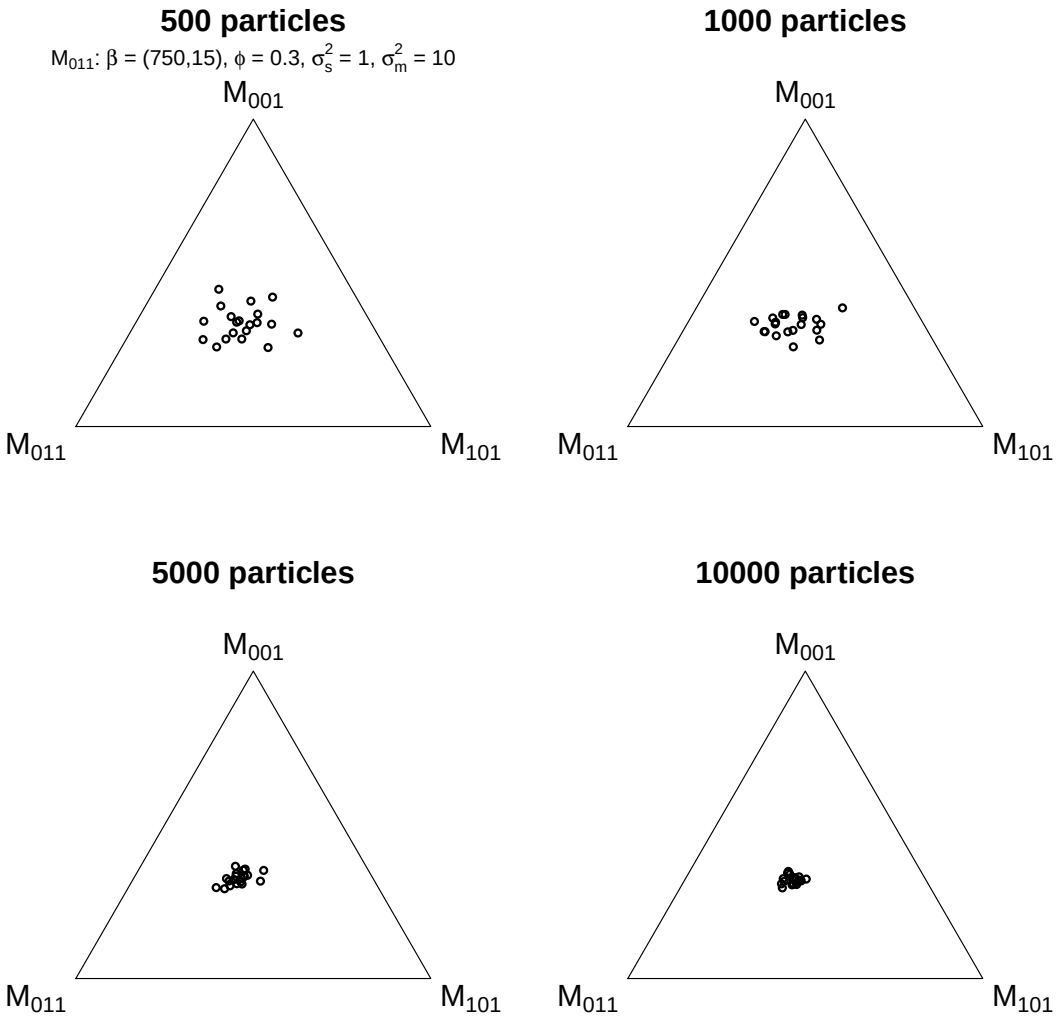
of particles, twenty such sets of approximate posterior model probabilities were calculated using the log marginal likelihood approximations based on the twenty PL runs under each model.

The results are displayed in Figure 6.19 using compositional plots. Notice that the estimates of the posterior model probabilities become less variable with increasing number of particles used in the PL, evidenced by the points in the compositional plots in Figure 6.19 clustering together in panels with higher number of particles. In addition, the points cluster near the middle of the ternary diagrams, suggesting that the three models considered are equally likely given the data. This is to be expected, since we purposely chose true values of the fixed parameters (particularly ϕ) for simulation such that correctly identifying M_{011} as the true model would be difficult. Based on these figures, we suggest that at least 5000 particles be used when running the PL under these models in order to obtain stable estimates of posterior model probabilities.

6.4.5 Comparing models for word recognition data using particle learning

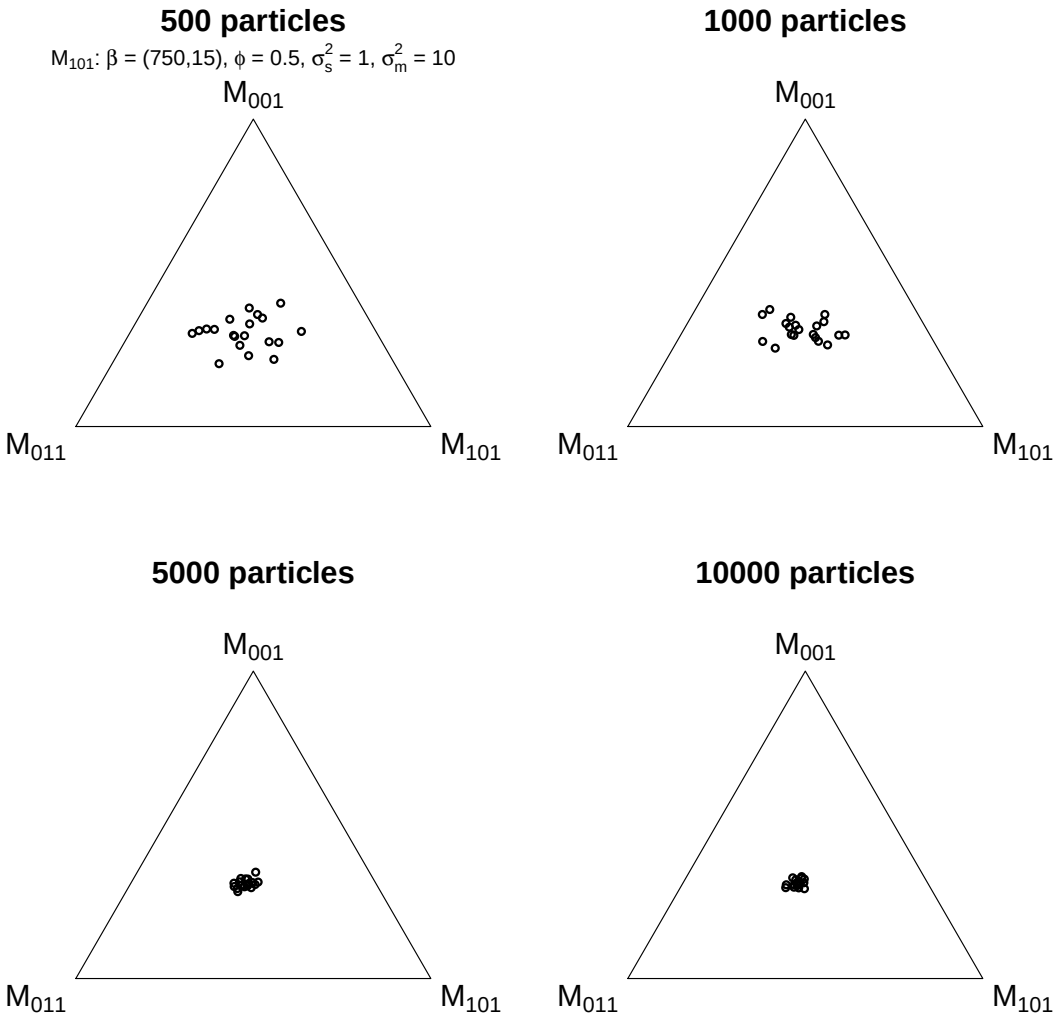
In this section, we examine results from running the PL algorithm on actual fMRI data generated from the word recognition experiment described in Section 6.1.4. We ran the PL algorithm using 5000 particles under each of M_{011} and M_{101} on time series from every voxel in our study (750 total). As in Sections 6.4.2,

Figure 6.19: Ternary diagrams of posterior model probabilities for simulated fMRI data from dynamic slope model



Posterior model probabilities among M_{011}, M_{101}, and M_{001} (corners of triangles) estimated for each of twenty runs of the PL under each model for increasing number of particles (plot panels) on data simulated from M_{011} with $\beta = (750, 15)'$, $\phi = 0.3$, $\sigma_s^2 = 1$, and $\sigma_m^2 = 10$. Each point represents a set of posterior probabilities (one for each model), and the proximity of the point to a particular corner of the triangle represents the posterior probability of the model in that corner relative to the other models. The prior distribution $p(x_0, \theta)$ used in the PL runs is given by equations (6.27), (6.28), and (6.29) with $\kappa = 1$ and b_0, φ_0, a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} set based on the MLEs as described in Section 6.4.2.

Figure 6.20: Ternary diagrams of posterior model probabilities for simulated fMRI data from dynamic intercept model



Posterior model probabilities among M_{011}, M_{101}, and M_{001} (corners of triangles) estimated for each of twenty runs of the PL under each of the models for increasing number of particles (plot panels) on data simulated from M_{101} with $\beta = (750, 15)'$, $\phi = 0.5$, $\sigma_s^2 = 1$, and $\sigma_m^2 = 10$. Each point represents a set of posterior probabilities (one for each model), and the proximity of the point to a particular corner of the triangle represents the posterior probability of the model in that corner relative to the other models. The prior distribution $p(x_0, \theta)$ used in the PL runs is given by equations (6.27), (6.28), and (6.29) with $\kappa = 1$ and b_0, φ_0, a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} set based on the MLEs as described in Section 6.4.2.

6.4.3, and 6.4.4, the prior distribution $p(x_0, \theta)$ was specified by equations (6.27), (6.28), and (6.29) with $\kappa = 1$. However, prior means b_0 and φ_0 were set to the average MLEs (under the corresponding model for which the PL is to be run) of β and ϕ, respectively, among voxels in the brain region (or cluster for SV-left and SV-right) from which the voxel being analyzed came from (MLEs were calculated as described in Section 6.3.2, summarized in Tables 6.4 and 6.5). The inverse-gamma hyperparameters a_{m_0}, b_{m_0}, a_{s_0}, and b_{s_0} were set such that the prior means for each of σ_s^2 and σ_m^2 were equal to their respective regional or cluster average MLEs, and such that each prior variance was equal to 500.

For each voxel-specific time series, we estimated the marginal likelihood of the data under each of M_{011} and M_{101} from the output of the PL algorithm run under each model. We also computed the exact marginal likelihood under M_{001} for time series from each voxel using equations (6.25) and (6.26) (with prior hyperparameters specified as in the previous paragraph). We then used the estimated and exact marginal likelihoods to compute approximate relative posterior probabilities among the three models for each voxel, according to equation (3.37), with prior model probabilities equal to 1/3 for each model. The model with the highest posterior probability for a given voxel was determined to be the "preferred" model for that voxel. The proportion of voxels that prefer each of the models across the six different brain regions are displayed in Table 6.7.

Table 6.7: Proportion of voxels favoring different regression models

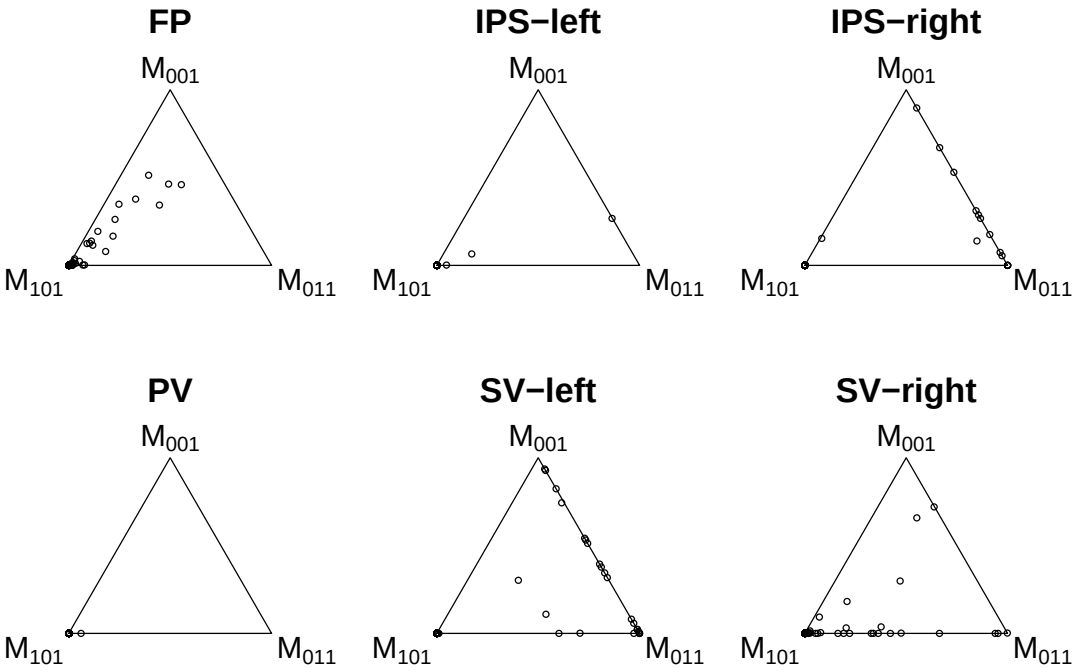
Region	M_{101}	M_{011}	M_{001}
FP	0.976	0.000	0.024
IPS-left	0.992	0.008	0.000
IPS-right	0.880	0.096	0.024
PV	1.000	0.000	0.000
SV-left	0.800	0.144	0.056
SV-right	0.952	0.032	0.016

Proportion of voxels in each brain region (rows) with highest posterior model probability for each of M_{101}, M_{011}, and M_{001} (columns). For M_{101} and M_{011}, posterior model probabilities were approximated using the PL with 5000 particles. For M_{001}, the true posterior probability was calculated analytically according to equation (6.25). The prior distribution, $p(x_0, \theta)$, assumed for each of the three models is as described at the beginning of Section 6.4.5.

From Table 6.7, it is clear that a vast majority of the voxels prefer M_{101}. The compositional plots in Figure 6.21 affirm this as well, with a majority of points from each brain region concentrated near the corner of the ternary diagram represented by M_{101}. Models that account for temporal autocorrelation in fMRI time series using a constant slope and an autocorrelated error structure, such as M_{101}, have been standard in fMRI studies, and these results provide further support for that standard.

However, there are a few brain regions for which a small percentage of voxels prefer M_{011} or M_{001}. Most notably, about 10% of voxels in IPS-right and close to 15% of voxels in SV-left prefer M_{011}. To examine this further, we have plotted 95% credible intervals for the filtered distributions of the dynamic slope, $p(\beta_1 + x_t | y_{1:t})$ at each time t, based on samples generated from the PL runs under M_{011}. We also display 95% credible intervals for ϕ and σ_s^2/σ_m^2 for each voxel based on samples

Figure 6.21: Posterior probabilities of dynamic regression models for real fMRI data



Posterior model probabilities among dynamic regression models (corners of triangles) calculated according to equation (3.37) and represented via compositional plots for 125 voxels (black dots) in each of 6 brain regions (plot panels). Marginal likelihoods for calculating posterior model probabilities were calculated analytically using equation (6.25) for M_{001}, and approximated using the PL with 5000 particles for each of M_{011} and M_{101}. The prior distribution $p(x_0, \theta)$ assumed for each model is as described at the beginning of Section 6.4.5. Each point represents a set of posterior probabilities (one for each model), and the proximity of the point to a particular corner of the triangle represents the posterior probability of the model in that corner relative to the other models.

from these PL runs generated at time $t = T = 245$ (i.e. conditional on all the data). Figures 6.22 and 6.23 display these intervals for 5 by 5 slices of voxels in IPS-right and SV-left, respectively. In addition, in Figure 6.23 for SV-left, we have color coded the lines representing the sequential credible intervals according to whether the corresponding voxel falls into the low or high activation cluster (there is only one cluster in IPS-right, hence only one line color throughout Figure 6.22). Colored bars have been inserted along the top of the plots to provide a visualization of the relative posterior model probabilities for each voxel. A legend lists the models and corresponding colors.

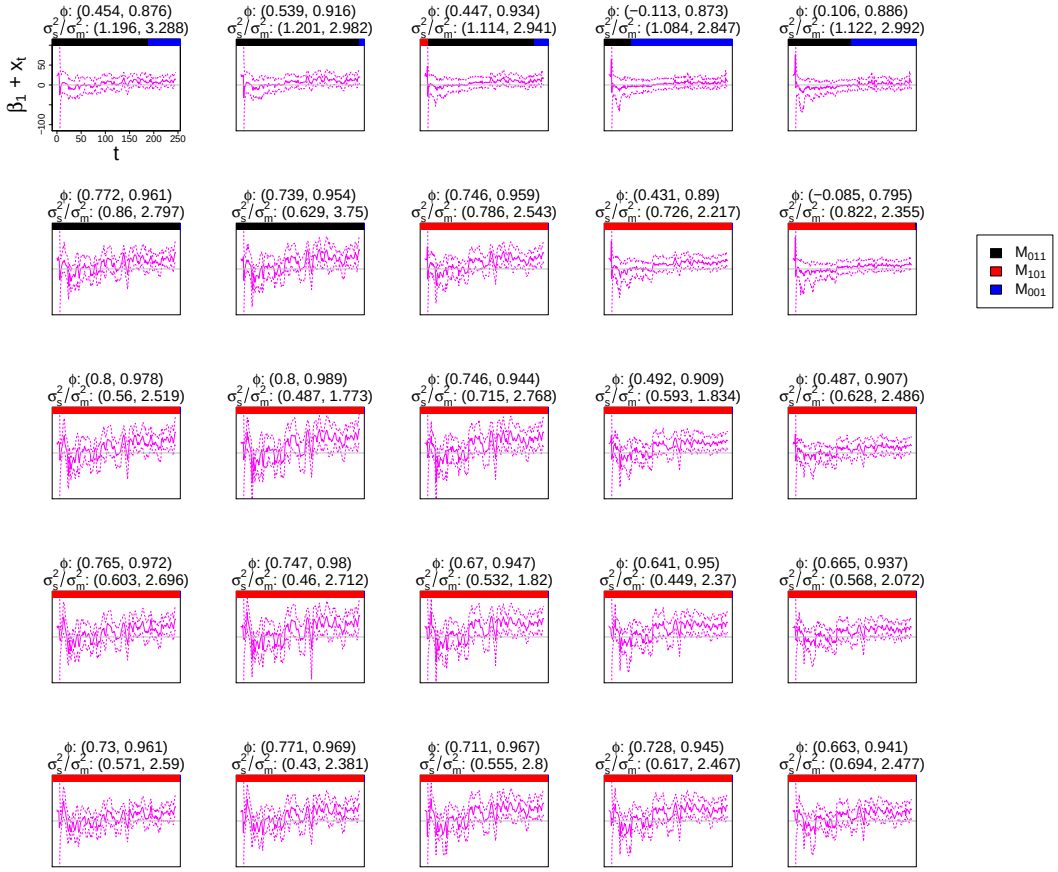
These figures for both slices of voxels reveal spatial patterns in the voxel-specific model preferences. For example, a cluster of voxels in the top two rows of Figure 6.22 (corresponding to neighboring voxels from the slice from IPS-right) prefer M_{011} or M_{001}, while the rest prefer M_{101}. In SV-left 6.23, a cluster of voxels in the top left portion of the slice prefer M_{011} or M_{001}, with the rest preferring M_{101}. In both brain regions, it appears that the voxels which prefer M_{011} or M_{001} tend to have lower values of the dynamic slope throughout time than do the voxels which prefer M_{101}. This is more pronounced in SV-left, where the k-means clustering method separated voxels into low and high activation clusters. In addition, voxels in SV-left which prefer M_{101} tend to have 95% credible intervals for ϕ which contain larger values than do 95% credible intervals for voxels which prefer M_{011}, and in some cases the upper bounds of these intervals extend near or

beyond the stationary region (i.e. close to or greater than 1). The behavior of the dynamic slopes for these voxels appears to be nonstationary, with a rising trend over the course of the experiment.

The results shown in Figures 6.22 and 6.23 seem to run counter to our hypothesis that the dynamic slope model would be better suited than the dynamic intercept model to model changes in brain activation over the course of the experiment. Specifically, voxels in IPS-right and SV-left with dynamic slopes that change the most over time tend to prefer the dynamic intercept model. One explanation for this could be that an increase in neural activity over the course of the experiment is also accompanied by an increase in sources of autocorrelation, such as heartbeat or respiration, that are better accounted for by the dynamic intercept model, but nonetheless manifest themselves in terms of an increasing dynamic slope when fit by M_{011}.

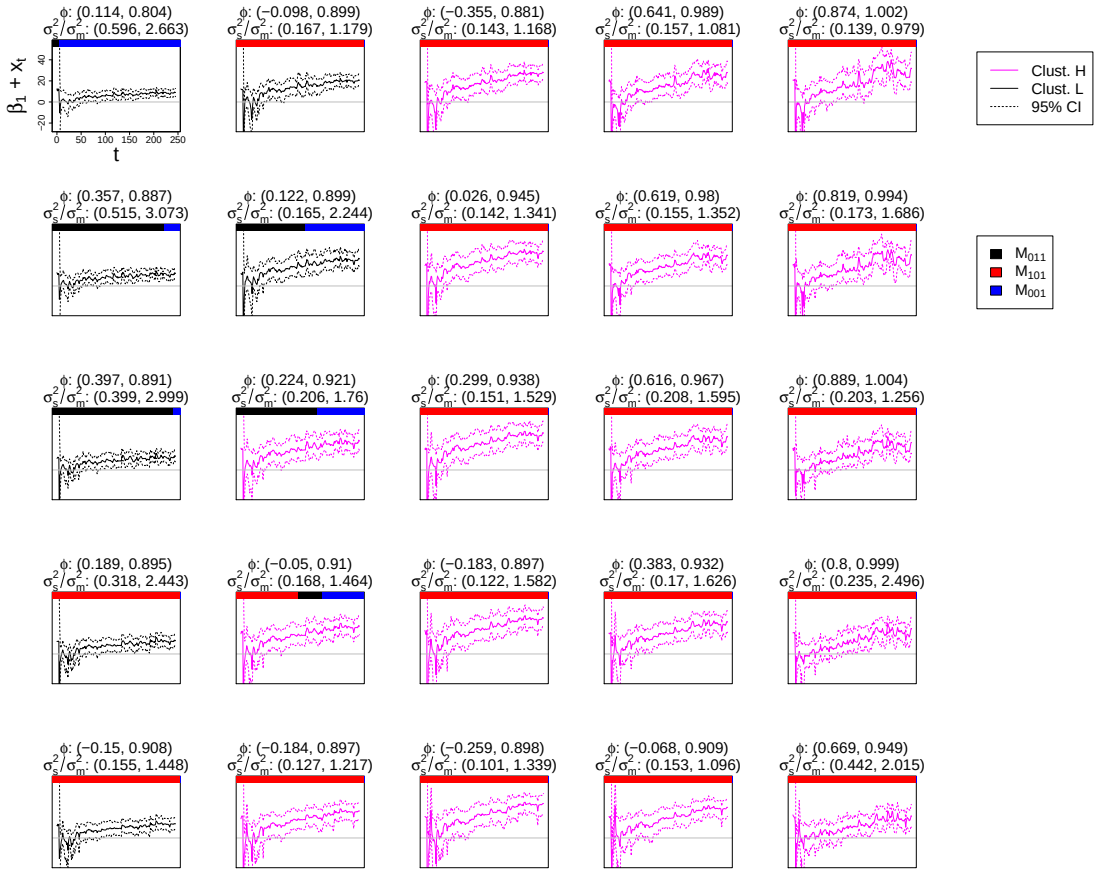
An alternate explanation for these results could be that the increase in the dynamic slope over time is not due to increased neural activity, but rather to misspecification of the hrf. The errors between the observed and expected BOLD responses in voxels for which the canonical hrf used in this analysis is inaccurate will be temporally autocorrelated, and they could be modeled according to a first-order autoregressive process as in M_{101}. The dynamic slope model could be accounting for inaccuracies in the hrf through an increasing slope, while the dynamic intercept model more suitably captures these inaccuracies through a

Figure 6.22: Filtered dynamic slopes and posterior model probabilities for data from IPS-right



Sequential 95% credible intervals for dynamic slopes (lines) and 95% credible intervals for $\phi|y_{1:T}$ and $\frac{\sigma_s^2}{\sigma_m^2}|y_{1:T}$ (the latter two fixed parameter credible intervals are written in text above each plot panel). Each interval was obtained by running PL under M_{011} with 5000 particles on time series from each voxel in a 5 by 5 slice in the y-z plane of the left intraparietal sulcus. Pink lines represent results based on all 125 IPS-right voxels, since clustering was not applied in this brain region. The proportion of the solid bar along the top of a particular plot panel colored for a specific model (as indicated in the model legend) represents the posterior probability of that model, relative to the other models, given data from the corresponding voxel for that plot panel (also represented by Figure 6.21). The prior distribution $p(x_0, \theta)$ assumed for each model is as described at the beginning of Section 6.4.5.

Figure 6.23: Filtered dynamic slopes and posterior model probabilities for data from SV-left



Sequential 95% credible intervals for dynamic slopes (lines) and 95% credible intervals for $\phi|y_{1:T}$ and $\frac{\sigma_s^2}{\sigma_m^2}|y_{1:T}$ (the latter two fixed parameter credible intervals are written in text above each plot panel). Each interval was obtained by running PL under M_{011} with 5000 particles on time series from each voxel in a 5 by 5 slice in the y-z plane of secondary visual left. Line colors correspond to whether a voxel was classified into the low (black lines) or high (pink lines) activation cluster according to the k-means clustering algorithm applied to the MLEs for θ obtained in Section 6.3.2. The proportion of the solid bar along the top of a particular plot panel colored for a specific model (as indicated in the model legend) represents the posterior probability of that model, relative to the other models, given data from the corresponding voxel for that plot panel (also represented by Figure 6.21). The prior distribution $p(x_0, \theta)$ assumed for each model is as described at the beginning of Section 6.4.5.

dynamic intercept. Voxels which prefer M_{011} or M_{001} could have BOLD responses that are more accurately characterized by the canonical hrf, and the gradual increase in the dynamic slope for these voxels could reflect small changes in neural activation or other sources of autocorrelation that mirror the stimulus pattern.

Model identification results from Sections 6.3.1 and 6.4.2 suggest that our fMRI model comparison results using PL should be interpreted with caution. Specifically, the true ϕ and σ_s^2/σ_m^2 need to be sufficiently large in order to be able to identify the true model among M_{011}, M_{101}, M_{001} using an approximation of the marginal likelihood. In addition, the marginal likelihood can be sensitive to specific prior distributions on the unknown states and fixed parameters. For example, if the prior distribution of states and fixed parameters used in this Section is closer to the posterior under M_{101} than it is to the posterior under M_{011}, model comparison results based on the marginal likelihood of the data could be biased toward M_{101}. The average maximum likelihood estimates of ϕ and σ_s^2 for IPS-right and SV-left shown in Tables 6.4 and 6.5, respectively, appear to be high enough to believe that the results shown in Figures 6.22 and 6.23 are not a fluke to misidentification. However, the 95% credible interval estimates for ϕ and σ_s^2/σ_m^2 obtained from running the PL, displayed above the plots in Figures 6.22 and 6.23, indicate that there is a large degree of uncertainty in these parameter estimates.

6.5 Discussion

In this chapter, we present an analysis of fMRI data collected from an episodic word recognition task, focusing specifically on voxels from six different brain regions. In Section 6.2.1, we fit regression models with ARMA errors via maximum likelihood to data from randomly selected voxels within 5 by 5 by 5 voxel cubes from each brain region, and we found that AIC and AICC prefer models with an ARMA(3,3) error structure while BIC tends to prefer AR(1) or ARMA(1,1) errors. We showed via simulation that testing for significant brain activation in fMRI time series using a standard OLS regression technique leads to an inflation of the false positive rate of concluding significant brain activation. In addition, in the presence of highly autocorrelated time series, we showed that a method for adjusting the degrees of freedom in the t-test for significant brain activation can lower the false positive rate while decreasing the power of the test. We also illustrated using a simulated example that comparing autocorrelation estimation algorithms by examining the independence of model residuals can give misleading results.

We proposed models for accounting for autocorrelation in fMRI data that contain a dynamic intercept, dynamic slope, or both. Using simulated fMRI data from each model, we explored parameter settings under which the distribution of maximum likelihood estimates appear normally distributed and centered at the true values. We concluded that it is easiest to find parameter settings under

which true parameter values in the dynamic slope model can be identified through examining the distribution of these maximum likelihood estimates, while identification of the model with both a dynamic slope and dynamic intercept is the most difficult. We fit the dynamic slope, dynamic intercept, and ordinary regression models to the word recognition data set using maximum likelihood, and identified clusters of high and low activation in the secondary visual cortex.

Lastly, we introduced a model comparison strategy based on estimating the marginal likelihood of data under different models using PL. We showed using simulated data that sufficiently high lag-1 autocorrelation and signal-to-noise ratios need to be present in the data in order to correctly select the true model amongst the dynamic slope or dynamic intercept models. Using the fMRI data from the word recognition experiment, we estimated relative posterior probabilities among the dynamic slope, dynamic intercept, and ordinary regression models and found that a vast majority of voxels prefer the dynamic intercept model, while the region with the highest percentage of voxels that prefer the dynamic slope model is the left secondary visual cortex ($\approx 15\%$).

It is conceivable that the most appropriate model for these data might be one with both a dynamic intercept and a dynamic slope. For instance, it could be the case that, for most voxels, variation in the data due to sources that may be better captured by the dynamic intercept model, such as physiological processes or misspecification of the hrf, overwhelm variation in the data that could be

captured by a dynamic slope, such as learning. The small clusters of voxels that prefer the dynamic slope model could be one of the few areas where the dynamic slope component accounts for more of the variation in the data. The fact that a significant portion of the relative posterior model probability in these voxels belongs to M_{001} (i.e. visible blue bars in Figures 6.22 and 6.23) further supports the notion that the dynamic slope accounts for very little of the autocorrelation in the data relative to the dynamic intercept.

Our results in Section 6.3.1 suggest that fMRI experiments similar to the episodic word recognition task described in Section 6.1.4 do not generate time series long enough to adequately estimate fixed parameters in the model with both a dynamic slope and dynamic intercept. Spatio-temporal modeling approaches that borrow information from neighboring voxels could possibly alleviate this problem and open up the possibility of correctly analyzing models with multiple autoregressive components. Several more recent studies have used spatio-temporal models that incorporate time-varying regression slopes to study dynamic brain connectivity (Ho et al.; 2005; Bhattacharya and Maitra; 2011).

The results from this section support the use of a dynamic intercept model, i.e. a model with a constant slope and autocorrelated error structure, which has been the norm in voxel-wise analysis of fMRI time series using the GLM. The dynamic slope model, despite being less suitable for the word recognition data, is perhaps a more interpretable model and, as shown in Section 6.4.2, can be more

easily identified when it is the true model. The use of the PL algorithm for model comparison provides insight into the relative appropriateness of these models for describing the behavior of neural activation in specific brain regions of interest and provides motivation for parameterizing future models for fMRI data in terms of a dynamic slope.

Chapter 7

Future work

In Chapters 4, 5, and 6, we implemented a variety of SMC methods to estimate unobserved states and unknown fixed parameters in state-space models. Alternative methods exist that can perform more efficiently under certain model settings. For instance, Rao-Blackwellization (Doucet et al.; 2000) could have been implemented within the PL algorithm in order to marginalize states and track only state sufficient statistics. This would lead to more efficient estimates of the filtered distributions of unknown states. Future work on tracking epidemic outbreaks using SMC methods could incorporate Rao-Blackwellization to estimate unobserved disease states in a population more efficiently.

In Chapter 5, we found that the PL algorithm proved to be more efficient than the RM or KDPF for analyzing data from the local level DLM described in Section 2.3.1. However, PL can only be applied to models for which the distributions

$p(x_{t+1}|y_{t+1}, x_t, \theta)$, $p(y_{t+1}|x_t, \theta)$, and $p(\theta|y_{1:t}, x_{0:t})$ are analytically tractable. In many cases, only some of these distributions may be available, and it is also possible that only some elements of θ admit distributions that can be tracked using sufficient statistics. In this case, a strategy such as one described in Dukic et al. (2012) that combines the approaches described in Section 3.2 could be implemented to optimize efficiency by sampling states and fixed parameters from known distributions when possible and from approximations, as in Liu and West (2001), when not. This strategy could be used for the dynamic regression models described in Section 2.3.3 when stationarity of the state process is desired.

While SMC methods have an apparent advantage over MCMC by being able to sequentially update the estimate of the filtered distribution of the current state of a system, there remain many situations when an MCMC analysis, or a combination of approaches, might be preferred. For instance, the performance of SMC algorithms degrades if run over a long period of time, and SMC methods cannot operate on models where prior distributions on states or fixed parameters are too diffuse. To address these problems, MCMC and SMC methods could be used in conjunction with one another. For instance, MCMC could be run prior to running a particle filter in order to find a reasonable range of values over which prior distributions can be defined (Chapter 5 Petris et al.; 2009). In addition, a possible strategy for continuously monitoring incoming streams of syndromic surveillance

data may consist of restarting SMC runs daily using posterior samples from an MCMC run overnight.

SMC methods also have the ability to provide direct approximations of the marginal likelihood under each model, and thus compare alternative models. In Chapter 6, we used particle learning in this way to compare alternative models for fMRI data. However, our approach required separate particle filter runs under each model in order to obtain competing estimates of the marginal likelihood. In addition, a single PL run using 5000 particles on time series consisting of 245 TRs took about 45 minutes to complete. Thus, this model comparison strategy is only feasible for analyzing small brain regions of interest. There exist methods that can compare models within a single particle filter run by allowing particles to jump between models (Berzuini and Gilks; 2001; Zhou et al.; 2013). These approaches open up the possibility of comparing models of fMRI data from a larger portion of the brain within reasonable computing time.

It is likely that more complicated models for tracking an epidemic (Shaman and Karspeck; 2012; Bhadra et al.; 2011) and analyzing fMRI data (Buxton et al.; 1998) than what we presented in this thesis are needed to more accurately describe the data-generating mechanisms. For example, results from Chapter 6 indicate that a regression model with both a dynamic intercept and a dynamic slope may be appropriate for fMRI time series. The increase in the dimension of the parameter space associated with larger models makes estimation more challenging, as

demonstrated with the epidemic model analyzed in Section 4.8 and the dynamic regression models for fMRI times series in 6.3.1. While an MCMC approach such as PMCMC may perform better than SMC methods in some high-dimensional settings, both approaches are limited by the amount of data available. For this reason, spatio-temporal modeling approaches that borrow information from neighboring infected areas or brain regions are a promising direction for these fields.

Bibliography

Aguirre, G. K., Zarahn, E. and D'esposito, M. (1998). The variability of human, BOLD hemodynamic responses, *NeuroImage* **8**(4): 360–369.

Alicia, Q., Diez, R. M. and Gamerman, D. (2010). Bayesian spatiotemporal model of fMRI data, *NeuroImage* **49**(1): 442–456.

Alspach, D. L. and Sorenson, H. W. (1972). Nonlinear Bayesian estimation using Gaussian sum approximations, *IEEE Transactions on Automatic Control* **AC-17**(4): 439–448.

Anderson, R. M., Fraser, C. and Ghani, A. C. (2004). Epidemiology transmission dynamics and control of SARS: the 2002-2003 epidemic, *Philosophical Transactions of the Royal Society B Biological Sciences* **359**: 1091–1104.

Andersson, J. L., Hutton, C., Ashburner, J., Turner, R. and Friston, K. (2001). Modeling geometric deformations in EPI time series, *NeuroImage* **13**: 903–919.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations, *The Annals of Statistics* **37**(2): 697–725.

Andrieu, C., Doucet, A. and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods, *Journal of the Royal Statistical Society, Series B: Methodological* **72**(3): 269–342.

Ashburner, J. and J., F. K. (1999). Nonlinear spatial normalization using basis functions, *Human Brain Mapping* **7**: 254–266.

Ashburner, J., Neelin, P., Collins, D. L., Evans, A. and Friston, K. (1997). Incorporating prior knowledge into image registration, *NeuroImage* **6**: 344–352.

Ashby, F. G. (2011). *Statistical Analysis of fMRI Data*, The MIT Press, Cambridge, Massachusetts and London, England.

Beckmann, C. F. and Smith, S. M. (2004). Probabilisitc independent component analysis for functional magnetic resonance imaging, *Medical Imaging, IEEE Transactions on* **23**(2): 137–152.

Bedard, M. (2008). Optimal acceptance rates for Metropolis algorithms: moving beyond 0.234, *Stochastic Processes and their Applications* **118**(12): 2198–2222.

Bennett, C. M. and Miller, M. B. (2013). fmri reliability: influences of task and experimental design, *Cognitive, Affective, and Behavioral Neuroscience* **13**(4): 690–702.

Berzuini, C. and Gilks, W. (2001). Resample-move filtering with cross-model jumps, *Sequential Monte Carlo Methods in Practice*, Springer, pp. 117–138.

Bhadra, A., Ionides, E. L., Laneri, K., Pascual, M., Bouma, M. and Dhiman, R. C. (2011). Malaria in northwest india: Data analysis via partially observed stochastic differential equation models driven by Levy noise, *Journal of the American Statistical Association* **106**(494): 440–451.

Bhattacharya, S. and Maitra, R. (2011). A nonstationary nonparametric bayesian approach to dynamically modeling effective connectivity in functional magnetic resonance imaging experiments, *The Annals of Applied Statistics* **5**(2B): 1183–1206.

Bowman, F. D., Caffo, B., Basset, S. S. and Kilts, C. (2008). A Bayesian hierarchical framework for spatial modeling of fMRI data, *NeuroImage* **39**: 146–156.

Boynton, G. M., Engel, S. A., Glover, G. H. and Heeger, D. J. (1996). Linear systems analysis of functional magnetic resonance imaging in human V1, *Journal of Neuroscience* **16**: 4207–4221.

Bullmore, E., Brammer, M., Williams, S., 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, *Magn. reson. Med.*

Burock, M. A. and Dale, A. M. (2000). Estimation and detection of event-related fMRI signals with temporally correlated noise: A statistically efficient and unbiased approach, *Human Brain Mapping* **11**: 249–260.

Buxton, R. B., Wong, E. C. and Frank, L. R. (1998). Dynamics of blood flow and oxygenation changes during brain activation: the balloon model, *Magnetic Resonance in Medicine* **39**(6): 855–864.

Cahn, B. R. and Polich, J. (2006). Meditation states and traits: EEG, ERP, and neuroimaging studies, *Psychological Bulletin* **132**: 180–211.

Cappé, O., Godsill, S. J. and Moulines, E. (2007). An overview of existing methods and recent advances in sequential Monte Carlo, *Proceedings of the IEEE* **95**(5): 899–924.

Cappé, O., Moulines, E. and Rydén, T. (2005). *Inference in hidden Markov models*, Springer Science+ Business Media.

Carter, C. K. and Kohn, R. (1994). On Gibbs sampling for state space mdoels, *Biometrika* **81**: 541–553.

Carvalho, C., Johannes, M., Lopes, H. and Polson, N. (2010). Particle learning and smoothing, *Statistical Science* **25**(1): 88–106.

Casella, G. and Berger, R. L. (2002). *Statistical Inference*, 2 edn, Duxbury: Thomas Learning.

Chew, C. and Eysenbach, G. (2010). Pandemics in the age of Twitter: Content analysis of tweets during the 2009 H1N1 outbreak, *PLoS One* **5**(11): e14118.

Chib, S. and Greenberg, E. (1994). Bayes inference in regression models with arma(p,q) errors, *Journal of Econometrics* **64**: 183–206.

Dangerfield, C. E., Ross, J. V. and Keeling, M. J. (2009). Integrating stochasticity and network structure into an epidemic model, *Journal of the Royal Society Interface* **6**(38): 761–774.

Dawdy, D. R. and Matalas, N. C. (1964). *Statistical and probability analysis of hydrologic data, part III: Analysis of variance, covariance and time series*, McGraw-Hill.

Dixon, M. and Wiener, G. (1993). TITAN: Thunderstorm identification, tracking, analysis and nowcasting-a radar-based methodology, *Journal of Atmospheric and Oceanic Technology* **10**(6): 785–797.

Doucet, A. and Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later, *Handbook of Nonlinear Filtering* **12**: 656–704.

Doucet, A., De Freitas, N. and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York.

Doucet, A., Godsill, S. and Andrieu, C. (2000). On sequential Monte Carlo sampling methods for Bayesian filtering, *Statistics and Computing* **10**(3): 197–208.

Dukic, V., Lopes, H. F. and Polson, N. G. (2012). Tracking epidemics with google flu trends data and a state-space seir model, *Journal of the American Statistical Association* **107**(500): 1410–1426.

Durbin, J. and Koopman, S. J. (2012). *Time series analysis of state space methods*, number 38, Oxford University Press.

Fearnhead, P. (2002). Markov chain Monte Carlo, sufficient statistics, and particle filters, *Journal of Computational and Graphical Statistics* **11**(4): 848–862.

Friston, K., Frith, C., Liddle, P. and Frackowiak, R. (1991). Comparing functional (PET) images: The assessment of significant change, *Journal of Cerebral Blood Flow and Metabolism* **11**: 690–699.

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

Friston, K., Holmes, A., Worsley, K., Poline, J., Frith, C. and Frackowiak, R. (1995b). Statistical parameteric maps in functional magnetic resonance imaging: A general linear approach, *Human Brain Mapping* **2**: 189–210.

Friston, K. J., Ashburner, J., Frith, C. D., B., P. J., Heather, J. D. and Frackowiak, R. S. (1995c). Spatial registration and normalization of images, *Human Brain Mapping* **2**: 165–189.

Friston, K. J., Penny, W., Phillips, C., Kiebel, S., Hinton, G. and Ashburner, J. (2002). Classical and bayesian inference in neuroimaging: Theory, *NeuroImage* **16**: 465–483.

Gardner, G., Harvey, A. C. and Phillips, G. D. A. (1980). An algorithm for exact maximum likelihood estimation of autoregressive-moving average models by means of Kalman filtering, *Applied Statistics* pp. 311–322.

Gardner, W. A. (1994). *Cyclostationarity in communications and signal processing*, Statistical Signal Processing Inc.

Gelfand, A. E. and Smith, A. F. M. (1990). Sampling-based approaches to calculating marginal densities, *Journal of the American Statistical Association* **85**: 398–409.

Geman, S. and Geman, D. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**: 721–741.

Gilks, W., Best, N. and Tan, K. (1995). Adaptive rejection metropolis sampling within Gibbs sampling, *Applied Statistics* pp. 455–472.

Gilks, W. R. and Berzuini, C. (2001). Following a moving target: Monte Carlo inference for dynamic Bayesian models, *Journal of the Royal Statistical Society, B* **63**: 127–146.

Ginsberg, J., Mohebbi, M. H., Patel, R. S., Brammer, L., Smolinski, M. S. and Brilliant, L. (2009). Detecting influenza epidemics using search engine query data, *Nature* **457**: 1012–1014.

Givens, G. H. and Hoeting, J. A. (2005). *Computational Statistics*, John Wiley and Sons.

Gordon, N. J., Salmond, D. J. and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation, *IEE Proceedings Part F: Communications, Radar and Signal Processing* **140**(2): 107–113.

Greene, J. D., Sommerville, R. B., Nystrom, L. E., Darley, J. M. and Cohen, J. D. (2001). An fMRI investigation of emotional engagement in moral judgement, *Science* **293**: 2105–2108.

Greicius, M. D., Krasnow, B., Reiss, A. L. and V, M. (2003). Functional connectivity in the resting brain: a network analysis of the default mode hypothesis, *Proceedings of the National Academy of Sciences* **100**(1): 253–258.

Greicius, M. D., Supekar, K., Menon, V. and Dougherty, R. F. (2009). Resting-state functional connectivity reflects structural connectivity in the default mode network, *Cerebral cortex* **19**(1): 72–78.

Hakenewerth, A. M., Waller, A. E., Ising, A. I. and Tintinalli, J. E. (2009). North Carolina Disease Event Tracking and Epidemiologic Collection Tool (NC DETECT) and the National Hospital Ambulatory Medical Care Survey (NHAMCS): comparison of emergency department data, *Academic Emergency Medicine* **16**(3): 261–269.

Hartigan, J. A. and Wong, M. A. (1978). A K-means clustering algorithm, *Applied Statistics* **28**: 100–108.

Harville, D. A. (1977). Maximum likelihood approaches to variance component estimation and to related problems, *Journal of the American Statistical Association* **72**(358): 320–338.

Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications, *Biometrika* **57**: 97–109.

Haykin, S. S. (2001). Kalman filtering and neural networks.

Heffernan, J. M., Smith, R. J. and Wahl, L. M. (2005). Perspectives on the basic reproductive ratio, *Journal of the Royal Society Interface* **2**(4): 281–293.

Henning, K. J. (2004). Overview of syndromic surveillance. What is syndromic surveillance, *MMWR Morb Mortal Wkly Rep* **53 (Suppl)**(Suppl): 5–11.

Ho, M.-H. R., Ombao, H. and Shumway, R. (2005). A state-space approach to modelling brain dynamics, *Statistica Sinica* **15**: 407–425.

Hodges, J. S. and Reich, B. J. (2010). Adding spatially-correlated errors can mess up the fixed effect you love, *The American Statistician* **64**(4): 325–334.

Hurvich, C. M. and Tsai, C. L. (1989). Regression and time series model selection in small samples, *Biometrika* **76**: 297–307.

Jones, R. H. (1980). Maximum likelihood fitting of ARMA models to time series with missing observations, *Technometrics* **22**(3): 389–395.

Kalman, R. (1960). A new approach to linear filtering and prediction problems, *Transactions of the ASME, Ser. D, Journal of Basic Engineering* **82**: 35–45.

Kiebel, S. J. and Holmes, A. P. (2007). The general linear model, *in* K. J. Friston, J. T. Ashburner, S. J. Kiebel, T. E. Nichols and W. D. Penny (eds), *Statistical parametric mapping: The analysis of functional brain images*, Academic Press, London.

King, A. A., Ionides, E. L., Breto, C., Ellner, S. P., Ferrari, M. J., Kendall, B. E., Lavine, M., Nguyen, D., Reuman, D. C., Wearing, H. and Wood, S. N. (2014). *Statistical inference for partially observed Markov processes*. R package version 0.49-2.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models, *Journal of Computational and Graphical Statistics* **5**(1): 1–25.

Leonski, B., Baxter, L. C., J., K. L., Maisog, J. and Debbins, J. (2008). On the performance of autocorrelation estiamtion algorithms for fMRI analysis, *IEEE Journal of Selected Topics in Signal Processing* **2**(6): 828–838.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering, *in* A. Doucet, J. F. G. De Freitas and N. J. Gordon (eds), *Sequential Monte Carlo Methods in Practice*, Springer-Verlag, New York, pp. 197–217.

Liu, J. S., Chen, R. and Wong, W. H. (1998). Rejection control and sequential importance sampling, *Journal of the American Statistical Association* **93**(443): 1022–1031.

Ljung, G. M. and Box, G. E. P. (1978). On a measure of lack of fit in time series models, *Biometrika* **65**: 297–303.

Lloyd, D. (2002). Functional MRI and the study of human consciousness, *Journal of Cognitive Neuroscience* **14**: 818–831.

Locasio, 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.

Logothetis, N. K. (2003). The underpinnings of the bold functional magnetic resonance imaging signal, *Journal of Neuroscience* **23**: 3963–3971.

Logothetis, N. K., Pauls, J., Augath, M., Trinath, T. and Oeltermann, A. (2001). Neurophysiological investigatiomn of the basis of the fMRI signal, *Nature* **412**: 150–157.

Ludkovski, M. and Niemi, J. (2010). Optimal dynamic policies for influenza management, *Statistical Communications in Infectious Diseases*.

Lund, T. E., Madsen, K. H., Sidaros, K., Luo, W.-L. and Nichols, T. E. (2006). Non-white noise in fMRI: Does modelling have an impact?, *Neuroimage* **29**: 54–66.

Luo, W.-L. and Nichols, T. E. (2003). Diagnosis and exploration of massively univariate neuroimaging models, *NeuroImage* **19**: 1014–1032.

Martínez-Beneito, Conesa, D., López-Quílez, A. and López-Maside, A. (2008). Bayesian Markov switching models for the early detection of influenza epidemics, *Statistics in Medicine* **27**(22): 4455–4468.

Mazziotta, J. C., Toga, A. W., Evans, A., Fox, P. and Lancaster, J. (1995). A probabilistic atlas of the human brain: theory and rationale for its development. The International Consortium for Brain Mapping (ICBM), *NeuroImage* **2**: 89–101.

Merl, D., Johnson, L., Gramacy, R. and Mangel, M. (2009a). A statistical framework for the adaptive management of epidemiological interventions, *PLoS One* **4**(6): e5807.

Merl, D., Johnson, L. R., Gramacy, R. B. and Mangel, M. (2009b). A statistical framework for the adaptive management of epidemiological interventions, *PloS One* **4**(6): e5807.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. and Teller, E. (1953). Equation of state calculations by fast computing machines, *The Journal of Chemical Physics* **21**(6): 1087–1092.

Mills, C. E., Robins, J. M. and Lipsitch, M. (2004). Transmissibility of 1918 pandemic influenza, *Nature* **432**: 904–906.

Neill, D., Moore, A. and Cooper, G. (2006). A Bayesian spatial scan statistic, *in* Y. Weiss, B. Schölkopf and J. Platt (eds), *Advances in Neural Information Processing Systems 18*, MIT Press, Cambridge, MA, pp. 1003–1010.

Niemi, A. J. (2012). *smcUtils: Utility functions for sequential Monte Carlo*. R package version 0.2.2.

Nili, H., Wingfield, C., Walther, A., Su, L. and Marslen-Wilson, W. (2014). A toolbox for representational similarity analysis, *PLoS Computational Biology*.

Norman, K. A., Polyn, S. M., Detre, G. J. and Haxby, J. V. (2006). Beyond mind reading: multi-voxel pattern analysis of fMRI data, *Trends in cognitive sciences* **10**(9): 424–430.

Novozhilov, A. S. (2008). On the spread of epidemics in a closed heterogeneous population, *Mathematical Biosciences* **215**(2): 177–185.

O'Hagan, A. (1994). Bayesian inference, *Kendall's Advanced Theory of Statistics 2B*, 2 edn, Halsted.

Ovaskainen, O. and Meerson, B. (2010). Stochastic models of population extinction, *Trends in Ecology and Evolution* **25**(11): 643–652.

Pagan, A. (1979). Some identification and estimation results for regression models with stochastically varying coefficients, *Journal of Econometrics* **13**: 341–363.

Penny, W. D., Ashburner, J. T., Kiebel, S. J. and Nichols, T. E. (2011). *Statistical Parametric Mapping: The Analysis of Functional Brain Images*, Academic Press.

Petris, G., Petrone, S. and Campagnoli, P. (2009). *Dynamic linear models*, Springer.

Pinheiro, J. C. and Bates, D. M. (2000). *Mixed-Effects Models in S and S-Plus*, Springer-Verlag New York, LLC.

Pitt, M. K. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filters, *Journal of the American Statistical Association* **94**: 590–599.

Plummer, M. (2005). *Output analysis and diagnostics for MCMC*. R package version 0.10-3.

Poldrack, R. A., Mumford, J. A. and Nichols, T. E. (2011). *Handbook of Functional MRI Data Analysis*, Cambridge University Press.

Purdon, P. L. and Weisskoff, R. M. (1998). Effect of temporal autocorrelation due to physiological noise and stimulus paradigm on voxel-level false-positive rates in fMRI, *Human Brain Mapping* **6**: 239–249.

R Core Team (2013). *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0.

Randal, D., Cappé, O. and Moulines, E. (2005). Comparison of resampling schemes for particle filtering, *Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis*, pp. 64–69.

Richter, W. and Richter, M. (2003). The shape of the fMRI BOLD response in children and adults changes systematically with age, *NeuroImage* **20**: 1122–1131.

Robert, C. P. and Casella, G. (2004). *Monte Carlo Statistical Methods*, 2 edn, Springer Inc.

Roberts, G. O., Gelman, A. and Gilks, W. R. (1997). Weak convergence and optimal scaling of random walk Metropolis algorithms, *The Annals of Applied Probability* **7**(1): 110–120.

Sakamoto, Y., Ishiguro, M. and Kitagawa, G. (1986). Akaike information criterion statistics, *Dordrecht, The Netherlands: D. Reidel*.

Schwarz, G. (1980). Estimating the dimension of a model, *Annals of Statistics* **6**: 461–464.

Shaman, J. and Karspeck, A. (2012). Forecasting seasonal outbreaks of influenza, *Proceedings of the National Academy of Sciences* **109**(50): 20425–20430.

Sheinson, D. M., Niemi, J. and Meiring, W. (2014). Comparison of the performance of particle filter algorithms applied to tracking of a disease epidemic, *Mathematical Biosciences* **255**: 21–32.

Shumway, R. H. and Stoffer, D. S. (2006). *Time Series Analysis and Its Applications: With R Examples*, Springer Science+ Business Media.

Skvortsov, A. and Ristic, B. (2012). Monitoring and prediction of an epidemic outbreak using syndromic observations, *Mathematical Biosciences* **240**(1): 12–19.

Smith, M. A., Shneiderman, B., Milic-Frayling, N., Mendes Rodrigues, E., Barash, V., Dunne, C., Capone, T., Perer, A. and Gleave, E. (2009). Analyzing (social media) networks with NodeXL, *In Proceedings of the Fourth International Conference on Communities and Technologies* pp. 255–264.

Storvik, G. (2002). Particle filters in state space models with the presence of unknown static parameters, *IEEE Transactions on Signal Processing* **50**(2): 281–289.

Stroud, P. D., Sydroiak, S. J., Riese, J. M., Smith, J. P., Mniszewski, S. M. and Romero, P. R. (2006). Semi-empirical power-law scaling of new infection rate to model epidemic dynamics with inhomogeneous mixing, *Mathematical Biosciences* **203**: 301–318.

Sugiura, N. (1978). Further analysis of the data by Akaike's information criterion and the finite corrections, *Commun. Statist, A, Theory Methods* **7**: 13–26.

Unkel, S., Farrington, C., Garthwaite, P. H., Robertson, C. and Andrews, N. (2012). Statistical methods for the prospective detection of infectious disease outbreaks: a review, *Journal of the Royal Statistical Society: Series A (Statistics in Society)* **175**(1): 49–82.

van Herwaarden, O. A. and Grasman, J. (1995). Stochastic epidemics: Major outbreaks and the duration of the endemic period, *Journal of Mathematical Biology* **33**(4): 581–601.

Wagner, M., Moore, A. and Aryel, R. (2006). *Handbook of Biosurveillance*, Elsevier.

Wand, M. and Ripley, B. (2006). Kernsmooth: Functions for kernel smoothing for wand & jones (1995), *R package version* pp. 2–22.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*, 2nd edn, Springer-Verlag Inc, New York.

Wilson, A. G., Wilson, G. D. and Olwell, D. H. (2006). *Statistical Methods in Counterterrorism: Game Theory Modeling Syndromic Surveillance and Biometric Authentication*, Springer.

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

Woolrich, M. W., Jenkinson, M., Brady, J. M. and Smith, S. M. (2004). Fully bayesian spatio-temporal modeling of fmri data, *IEEE Transactions on Medical Imaging*.

Worsley, K. J. (1995). Estimating the number of peaks in a random field using the Hadwiger characteristic of excursion sets with applications to medical images, *Annals of Statistics* **23**: 640–669.

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

Worsley, K. J., Evans, A. C., Marrett, S. and Neelin, P. (1992). A three-dimensional statistical analysis for rCBF activation studies in human brain, *Journal of Cerebral Blood Flow and Metabolism* **12**: 900–918.

Worsley, K. J., Marrett, S., Neelin, P., Vandal, A. C., Friston, K. J. and Evans, A. C. (1996). A unified statistical approach for determining significant signals in images of cerebral activation, *Human Brain Mapping* **4**: 58–73.

Zhang, L., Guindani, M., Versace, F. and Vannucci, M. (2014). A spatio-temporal nonparametric Bayesian variable selection model of fmri data for clustering correlated time courses, *NeuroImage* **95**: 162–175.

Zhang, S. (2011). Estimating transmissibility of seasonal influenza virus by surveillance data, *Journal of Data Science* **9**: 44–64.

Zhou, Y., Johansen, A. M. and Aston, J. A. (2013). Towards automatic model comparison an adaptive sequential Monte Carlo approach, *arXiv preprint*.