Improving Exploration of Posterior Distributions in Spatial Models - A Markov Chain Monte Carlo Approach

Bridette Anne-Marie Hayes

Thesis submitted for the degree of Doctor of Philosophy

Trinity College Dublin
Department of Statistics

July 2005

Declaration

This thesis is entirely my own work. The contents of this thesis have not been submitted as an exercise for a degree at this or any other University. The contents of this thesis may be lent or copied by the Library at Trinity College Dublin.

Bridette Anne-Marie Hayes

Summary

A Markov chain Monte Carlo (MCMC) algorithm is proposed for the evaluation of a posterior distribution. The posterior distribution is from a model that has a spatial structure and exhibits many characterisics which are typically cumbersome to MCMC algorithms. The algorithm is construct with the purpose of conquering or significantly reducing these difficulties. The performance of this algorithm is then investigated for a diversity of circumstances.

Acknowledgements

This research was carried out with financial support from the Enterprise Ireland "Basic research grant" program; under the grant title "Bayesian statistical methods for the analysis of data of different spatial and/or temporal support", and the supervision of Dr. Simon P. Wilson of the Department of Statistics.

I would like to express my sincere thanks to Dr. Simon Wilson for his endless patience and guidance throughout the exhaustive process of researching and writing this PhD thesis. The members of the statistics department have been most generous with their time, assistance and support, of which I am greatly appreciative. Within the walls of Trinity, this thesis among other things has allowed me the wonderful company and friendship of my fellow postgrads; Caroline and Eleisa.

Naturally, I need to thank my parents Ann and Michael for their continued encouragement in my endeavours through the years. Also, a special thanks to Harry and Clement Spidergrace for their insights, inspiring peculiarities and entertainment when required. Sue and J also played an indispensable part in concluding this thesis, with their much pestering and utter resolution in conducting me to the library. And lastly, I am indebted to my dear friends; Siobhan, Paul, Conor, Luce and Malx, whose individual support at various stages has contributed to the completion of this thesis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Latent spatial process models are useful in many applications of spatial data. Bayesian inference, and indeed other forms of likelihood-based inference, must often be implemented by MCMC, but latent spatial process models combine features that make traditional MCMC methods perform poorly. This inadequate performance is manifest as poor mixing within the posterior distribution of parameters of the model. Addressing this issue within MCMC methods and incorporating possible solutions is a natural step.

In this thesis we propose an MCMC scheme for exploring the posterior distribution of a spatial model that involves two principal ideas to combat the mixing problems caused by high dimensionality and strong posterior correlation between parameters. The scheme combines the approach of coupling and blocking, using coarse and fine scale MCMC chains. The algorithm is applied to a latent spatial Gaussian model, within a Bayesian framework and under various experimental conditions. Particular attention is given to the efficiency with which the posterior distribution is explored.

1.1 Review of Spatial Models

Chapter 2 gives a broad background to spatial modeling, where the more commonly used approaches are described in some detail. There is a diversity of potential approaches, from variograms to spatial autoregressive models. The first half of the chapter is concerned with defining some standard terminology and introducing the approaches which have dominated spatial analysis until recent times. The latter part is dedicated to the modeling based approaches which have become more prevalent with the advent of widespread usage of Bayesian methods. The models to be used in Chapters 4 and 5 are also described here, as are their methods of simulation.

1.2 Statistical Methodology

It is the Bayesian framework that is embraced in this dissertation, the background and principals of which are outlined in Chapter 3. The Bayesian method and the previously mentioned model-based approaches of Chapter 2 have only in relatively recent times gained popularity. This is primarily due to advances in computing power, which have made Bayesian models more tractable. Computation of the posterior distribution, and useful functions of it such as expectations, is achieved through sampling from it. In order to sample from the posterior distribution of a Bayesian model, Markov Chain Monte Carlo (MCMC) is used. The particulars of this method are described and an emphasis has been given to its application to spatial models. The last section provides specifics on measuring an MCMC algorithm's efficiency and general diagnostics for measuring the success with which the algorithm explores the posterior distribution.

1.3 Blocking Algorithm

An MCMC algorithm for inference with a spatial model is proposed in Chapter 4. This model assumes that observations are a function of a latent Gaussian field. In this thesis the observations are count data, but the latent idea is easily extended to other data types. Two versions of the algorithm are compared to a standard algorithm. The main difference between these approaches is a blocking scheme, that samples blocks of variables jointly, rather than each variable singly. The proposed algorithms have two alternative blocking schemes, where the original method of Diggle et al. (1998) does not. There is also a useful proposal function used within the blocking scheme to enhance its efficiency. The behaviour of all three algorithms is investigated under a number of circumstances using four datasets with different characteristics.

1.4 Coupling Algorithm

The spatial setting has been extended to require a more complex model where the observations are multivariate count data, that display both within and between location correlations. The model is taken to be a function of a number of latent processes. With increased complexity of model comes the need for a more sophisticated MCMC approach in order to successfully sample from its posterior distribution. The approach chosen is that of a coupling technique. There are many factors which may affect the performance of such an algorithm and it is the influence of these factors that are investigated in Chapter 5. This approach was greatly influenced by a palaeoclimatology dataset with a complex correlation and aggregation structure.

1.5 Overall Framework and Major Contributions

This research is carried out within the Bayesian paradigm and its focus of interest is the development of efficient MCMC algorithms in the context of spatial models and with specific attention given to the effective exploration of a target distribution. The main contribution is the extension of two MCMC techniques, with an investigation of their properties under a selection of conditions. Specifically, the following are the main contributions made by this research:

- In Chapter 4, a detailed comparison of three MCMC schemes for sampling from the posterior distribution in the case of univariate spatial count data.
- The development of an MCMC method, based on the ideas of aggregation and coupling, for sampling from the posterior distribution in the complex case of multivariate spatial count data, where both within and between location correlations are modeled.
- A study of the performance of this coupled chain approach with respect to various factors that affect its ability to explore the posterior distribution.

Chapter 2

Spatial Models

There are a large number of spatial models; see Ripley (1988), Cressie (2001), Møller and Waagerpetersen (2003). Here, we concentrate on one type, the latent spatial Gaussian model.

2.1 Spatial Data

As the name suggests, spatial data has the property that each observation is associated with a geographical region or spatial location. Data points that are closer together in space are often likely to have more similar attributes than those that are far apart. By taking note of their spatial location and calculating the distance between points, a spatial model incorporates this relationship (or local variation). But depending on the setting, an alternative measure may be more sensible. Spatial data may be more formally described as a form of stochastic process.

Definition 2.1 (Stochastic Process). *A stochastic process is a set of random variables* $\{Y(s) : s \in S\}$*, where* S *is referred to as the indexing set and* $Y(s) \in D$*, where* D *is called the state space.*

A spatial stochastic process has indexing set S representing a set of locations. In our context we will have $S \subset \mathbb{R}^2$. We will always use the usual Euclidean metric when talking about distances between points in S.

The spatial variation associated with this process is described by its covariance structure. Let C be a covariance function and ρ be the correlation between $Y(s_1)$ and $Y(s_2)$ for two points $s_1, s_2 \in S$, where

$$C(s_1, s_2) = E[(Y(s_1) - E(Y(s_1)))(Y(s_2) - E(Y(s_2)))],$$

and

$$\rho(s_1, s_2) = \frac{C(s_1, s_2)}{\sqrt{C(s_1, s_1)C(s_2, s_2)}}.$$

This covariance and correlation structure can be modelled in many ways.

2.1.1 Correlation Function

The correlation function controls the smoothness and the extent of dependence in the spatial process. Some of the more commonly used correlation functions arise from the Matérn Family:

$$\rho(u) = \left(\frac{u}{\phi}\right)^{\kappa} \frac{1}{2^{\kappa-1}\Gamma(\kappa)} K_{\kappa}\left(\frac{u}{\phi}\right),$$

where $u = d(s_1, s_2)$ is distance between two points, $\kappa > 0$ and $\phi > 0$ are parameters, and K_κ denotes a Bessel function of order κ. Two well known members of the family are the exponential correlation function

$$\rho(u) = \exp\left(-\frac{u}{\phi}\right),$$

setting κ to 0.5 and the Gaussian correlation function

$$\rho(u) = \exp\left(-\left(\frac{u}{\hat{\phi}}\right)^2\right),$$

as $\kappa \to \infty$ and $\hat{\phi} = 2\sqrt{u+1}\phi$. Another correlation function family is the powered exponential:

$$\rho(u) = \exp\left(-\left(\frac{u}{\phi}\right)^{\kappa}\right),$$

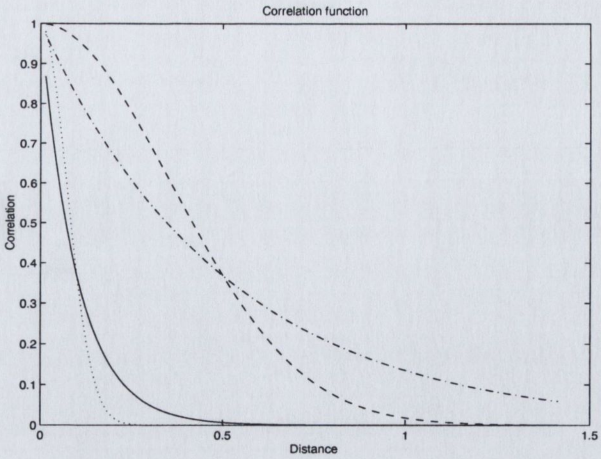


Figure 2.1: The powered exponential correlation function on a unit square with (ϕ, κ) variable. Here $(\phi, \kappa) = (0.1, 1)$ is the solid line, $(0.1, 2)$ is the dotted line, $(0.5, 1)$ is the dot-dash line and $(0.5, 2)$ is the long dash line.

where $\phi > 0$ and $\kappa < 2$ (these values are discussed further in chapter 4). For these functions ϕ can be interpreted as a scaling (or range) parameter for dependence between points, and the κ parameter can be viewed as a smoothness parameter or a parameter which describes the relative rate of change in the correlation between points; small κ implies higher spatial correlation at larger distances.

Two other important features in describing a spatial process are homogeneity and isotropy.

Definition 2.2 (Homogeneity). *A homogeneous process is one where $E(Y(s))$ and var$(Y(s))$ are constant in s and that C and ρ only depend on the vector h from s_1 to s_2, that is they are independent of absolute location.*

Definition 2.3 (Isotropy). *An isotropic process is one where C and ρ are only dependent on $d(s_1, s_2)$, where d is distance between the points s_1 and s_2.*

The natural extension of the latter definition is that if the spatial correlation between

$Y(s_1)$ and $Y(s_2)$ depends not only on the length of the separation vector h, but also on the its direction, then Y is said to be *anisotropic.* Similarly, if the mean or variance of the covariance structure changes over the space, then the process is said to exhibit *heterogeneity* or *non-stationarity.*

The correlation functions given above are examples of isotropic processes. Observe that they are valid definitions for a correlation function since the covariance matrix of any set of points so defined will be positive definite, see Matern (1986).

2.2 Spatial Processes

Spatial data can have various attributes and take a number of different forms:

- Discrete or continuous;
- Individual points in space or spatially aggregated into regions;
- Located at regular or irregular points in space;
- Randomly distributed or clustered/patterned locations;
- Individual measures or measure taken repeatedly over time.

An extensive framework for categorizing and modelling spatial data is given by Cressie (2001). Using the notation given above, we will outline the general classifications and approaches he suggests. Let $s \in \mathbb{R}^d$ be a location in d-dimensional space, and $Y(s)$ be a data value (or possibly a vector) observed at s. The full dataset can then be modeled as the multivariate random process

$$\{Y(s) : s \in S\},$$

where s varies over an index set $S \subset \mathbb{R}^d$. Cressie (2001) then categorizes spatial data into three cases.

- *geostatistical data* (also sometimes referred to as *point source data*), where $Y(s)$ is a random vector at a location s which varies continuously over S, a fixed subset of $\mathbb{R}^d$ that contains a d-dimensional rectangle of positive volume;

- *lattice data* (or alternatively named *regional summary data*), where S is again a fixed subset (of regular or irregular shape), but now containing only a countable number of sites in $\mathbb{R}^d$, normally supplemented by neighbour information.

- *point pattern data*, where now S is itself random; its index set gives the locations of random events that are the spatial point pattern. $Y(s)$ itself can simply equal 1 for all $s \in S$ (indicating occurrence of the event), or possibly give some additional covariate information (producing a *marked point pattern process*).

An example of the first case might be a collection of measured oil reserves at various fixed source points s, with a primary goal being to predict the reserve available at some unobserved location s^*. For the second category, observations may correspond to pixels making up an image, each of which has four neighbors (above, below, left and right). Point pattern data often arise as locations of disease occurrence or existence of a certain species of plant (supplemented with rainfall, temperature or other covariate measurements to produce a marked point pattern).

Our interest is in geostatistical data or rather *model based geostatistics*, a term which was introduced by Diggle et al. (1998), who combine traditional geostatistical methods with those of generalised linear models. Naturally, depending on the type of spatial data, different methods may be used for modeling the events or measurements. Some of the techniques commonly used for estimation, inference and prediction are given below.

2.3 Spatial Modeling

Two key ideas have dominated the analysis of spatial data (or at least geostatistics) until recent times. These are the *variogram* (and its associated measures) and *kriging*. With the advent of widespread usage of Bayesian methods, this has changed somewhat and more strongly modeling based approaches have come to the forefront.

2.3.1 Variograms, Covariograms and Correlograms

Suppose $Y(s)$ satisfies the following definition:

Definition 2.4 (Second-Order Stationarity). *Y(s) is a second-order (or weak) stationary process if*

$$\begin{aligned} E[Y(s)] &= \mu, \\ C[Y(s_1), Y(s_2)] &= C_o(s_1 - s_2) < \infty. \end{aligned}$$

for some function C_o.

That is firstly, the expectation at s does not depend upon s or alternatively viewed, the expectation is invariant over the study area. Secondly, the covariance depends only on the separation vector; if the covariance is dependent on the separation distance $u = |s_1 - s_2|$ alone, then the resulting measure is said to be isotropic, as defined earlier. When the stochastic process is Gaussian, second-order stationarity and homogeneity are equivalent properties of teh process, see Cressie (2001). C_o is called the covariogram (or stationary covariance function) and is analogous to the autocovariance function in time series analysis. Suppose then that

$$\text{var}(Y(s_1) - Y(s_2)) = 2\gamma(s_1 - s_2) \;\; \forall s_1, s_2 \; \in T$$

$$2\gamma(h) = 2(C_o(0) - C_o(h))$$

for some function γ. The quantity $2\gamma(\cdot)$, which is a function of the separation vector h is called a *variogram* (and $\gamma(\cdot)$ a *semivariogram*) by Cressie (2001) and previously by Matheron, but has also had many appearances under different names. Lastly, a correlogram is defined as

$$\rho_o(h) = \frac{C_o(h)}{C_o(0)} = 1 - \frac{\gamma(h)}{C_o(0)},$$

which is referred to as an autocorrelation function by time series analysts. The classical estimator of the variogram proposed by Matheron is

$$2\hat{\gamma}(h) = \frac{1}{|N(h)|} \sum_{s \in N(h)} (Y(s_1) - Y(s_2))^2, \quad h \in \mathbb{R}^2$$

letting $N(h) = \{s = (s_1, s_2) : s_1 - s_2 = h\}$, where $|N(h)|$ is the number of distinct pairs of $N(h)$. This is in fact not a particularly robust estimator, but there are many others, such as the linear, spherical and exponential variograms, which are more suitable in various settings.

Measures defined on the variogram are the *sill*, which is $2C_o(0)$, where if $C_o(h) \to 0$ then

$$\lim_{h \to \infty} 2\gamma(h) = 2C_o(0).$$

The other measure is the *nugget*, which is $\lim_{h \to 0} 2\gamma(h)$. The latter need not be zero (i.e. the variogram may have a discontinuity at the origin) due to microscale variability or measurement error. For a monotonic variogram that reaches its sill exactly, the distance at which the sill is reached is called the *range*. If the sill reaches zero, then observations further away than the range are uncorrelated. If the sill is reached asymptotically and is less than 0.05, the distance such that $\rho(s_1, s_2) = 0.05$ is called the *effective range*. Also it is noted that, $\gamma(h) = \gamma(-h)$ and that $\gamma(0) = 0$.

In statistical inference, the idea is to search for a valid variogram that, as a measure of spatial dependence, is closest to that given by the data. Variogram estimators cannot be used directly for spatial prediction (*kriging*), but are commonly used in spatial analysis.

2.3.2 Kriging

Kriging refers to a method of spatial prediction for the process at a point s^*, given data $y = \{y(s_1), \ldots, y(s_n)\}$. *Ordinary kriging* is the prediction of $Y(s^*)$, under the following assumptions.

Model:

$$Y(s) = \mu + \delta(s), \quad \mu \in \mathbb{R}, \ \mu \ unknown$$

where $\delta(s) \overset{iid}{\sim} N(0, \sigma^2)$ or a spatial zero-mean process.

Predictor:

$$\hat{Y}(s^*) = \sum_{i=1}^{n} \lambda_i y(s_i), \quad \text{for} \sum_{i=1}^{n} \lambda_i = 1.$$

This latter condition, that the coefficients of the linear predictor sum to 1, guarantees uniform unbiasedness:

$$E(\hat{Y}(s^*)) = \mu = E(Y(s^*)), \quad \forall\, \mu \in \mathbb{R}.$$

The optimal linear unbiased predictor for kriging is generally (at least in a classical setting) taken to be that which minimizes the mean-squared prediction error

$$\sigma_e^2 = E\left([Y(s^*) - \hat{Y}(s^*)]^2\right)$$

over $\lambda_1, \ldots, \lambda_n$. This optimization problem is solved using a series of equation, involving Lagrange multipliers (to ensure the coefficient constraints above are fulfilled) and variogram estimates. Methods for appropriately choosing kriging predictors and solving them are detailed in Chapter 3 of Cressie (2001).

There are several variants of this:

- *Simple kriging* is the case in which μ is known and the coefficients are not constrained to sum to 1;

- *Robust kriging*, which deals with situations where the data are not normal and there are isolated outliers;

- *Universal kriging*, which is probably more useful in real settings. Universal kriging includes the use of explanatory variables, and assumes a model for $Y(s)$ of the form:

  $$Y(s) = \sum_{j=1}^{k} \beta_j X_j(s) + \delta(s),$$

  and a predictor for $Y(s^*)$ of the form

  $$\hat{Y}(s^*) = \sum_{i=1}^{n} \lambda_i y(s_i), \quad \text{for } \lambda' X(s) = x',$$

  where $X(s)$ are known functions and $x = X(s^*)$. Again this latter condition is necessary for a uniformly unbiased predictor, that is $E(\hat{Y}(s^*)) = E(\lambda' Y) = \lambda' X(s)\beta$.

- *Bayesian kriging*, where Bayesian principles (as discussed formally in Chapter 3) are used to model an unknown, deterministic process, by way of a random process. For a non-stationary mean, the model is

  $$Y(s) = \mu(s) + \delta(s)$$

  where $\delta(.)$ is a zero-mean stationary random process. This model is useful for analysing physical processes that are spatially heterogeneous. Assuming that $\mu(.)$ is a random process with parameter θ with prior $\pi(\theta)$, one could estimate the parameters of the random process $\mu(.)$ based on the marginal distributions of $y(s) = (y(s_1), \ldots, y(s_n))$. Given the marginal posterior distribution of the parameters and the joint distribution of $\pi(Y(s), Y(s^*))$, by marginalizing over the parameters of the conditional distribution $\pi(Y(s^*)|y(s))$, the predictive distribution is given by

$$\pi(Y(s^*)|y(s)) \quad = \quad \int \pi(Y(s^*)|y, \mu, \delta, \theta)\pi(\mu, \delta, \theta|y)\; d\mu\; d\delta\; d\theta.$$

Typically this integral is evaluated by Monte Carlo integration (see Chapter 3, Section 3.2.3). If $\mu^{(n)}, \delta^{(n)}$ and $\theta^{(n)}$ are samples from the posterior distribution then:

$$\pi(Y(s^*)|y(s)) \approx \frac{1}{N}\sum_{n=1}^{N} \pi(Y(s^*)|y(s), \mu^{(n)}, \delta^{(n)}, \theta^{(n)}).$$

That is, substitute the posterior draws into the conditional distribution of the target values $Y(s^*)$. A Bayesian implementation of kriging is presented by Diggle et al. (1998).

2.3.3 Spatial Autoregressive Models

The previous spatial methods which have been described lend themselves to geostatistical settings. Autoregressive models are used to characterize the spatial dependencies observed in lattice and regional data, that is they model each location as a linear combination of its neighbouring locations.

There are two principle specifications of these models: *conditional autoregressive (CAR) models* and *simultaneous autoregressive (SAR) models*, that is CAR models are defined by full conditionals, whereas SAR are defined by autoregressive equations. These autoregressive models are derived from those in time series, defined as

$$X_t = aX_{t-1} + \epsilon_t, \quad \epsilon_t \sim N(0, \sigma^2) \text{ independent}$$

or equivalently

$$E(X_t|\text{past values}) = aX_{t-1}, \quad \text{var}(X_t|\text{past values}) = \sigma^2$$

where the X_t is assumed to have a Gaussian distribution. The extension of these to allow symmetry of dependency and further dependencies gives the CAR and SAR models. The CAR and SAR Gaussian processes can be specified as follows.

CAR models:

$$P(x_i|x_j, j \neq i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2\sigma^2}\left(x_i - (\mu_i + \sum_{j=1}^{N} \beta_{ij}(x_j - \mu_j))\right)^2\right)$$

Letting $\mathbf{X}$ be the $N \times 1$ matrix of variables X_i, μ the $N \times 1$ matrix of means μ_i, and $\mathbf{B}_c$ the $N \times N$ matrix whose diagonal elements are zero, and whose element (i, j) is β_{ij} and the X_i are Gaussian. The random field is then given by:

$$\mathbf{X} = \mu + \mathbf{B}_c(\mathbf{X} - \mu) + \eta,$$

where η is called the noise vector, is normal and satisfies

$$cov(\eta) = \sigma^2(\mathbf{I} - \mathbf{B}_c) \quad \text{and} \quad cov(\eta, \mathbf{X}) = \sigma^2\mathbf{I}.$$

The necessary and sufficient conditions for a valid formulation are that $(\mathbf{I} - \mathbf{B}_c)$ be symmetric and positive definite, see Ripley (1981).

SAR models: Consider the process defined by the set of N simultaneous autoregressive equations

$$X_i = \mu_i + \sum_{j=1}^{N} \beta_{ij}(X_j - \mu_j) + \varepsilon_i, \quad i = 1, \ldots, N,$$

where the noise sequence ε is Gaussian and

$$cov(\varepsilon) = \sigma^2\mathbf{I} \quad \text{and} \quad cov(\varepsilon, \mathbf{X}) = \sigma^2(\mathbf{I} - \mathbf{B}_s^T)^{-1},$$

where $\mathbf{B}_s = \beta_{ij}$ for $i \neq j$ and has zero diagonal elements. Then the necessary and sufficient condition for this to exist is that $(\mathbf{I} - \mathbf{B}_s)$ be non-singular, see Ripley (1981).

The joint probability density function for these Gaussian processes is then given by

$$P(\mathbf{X}) = (2\pi\sigma^2)^{-\frac{N}{2}} |\mathbf{Q}|^{\frac{1}{2}} \exp\left[-\frac{1}{2\sigma^2}(\mathbf{X} - \mu)^T \mathbf{Q}(\mathbf{X} - \mu)\right],$$

where $|\mathbf{Q}|$ denotes the determinant of $\mathbf{Q}$. That is, $\mathbf{X}$ has a multivariate normal distribution with mean vector μ, and covariance matrix $\sigma^2\mathbf{Q}^{-1}$. For a CAR model $\mathbf{Q} = (\mathbf{I} - \mathbf{B}_c)$ and for a SAR model $\mathbf{Q} = (\mathbf{I} - \mathbf{B}_s^T)(\mathbf{I} - \mathbf{B}_s)$. With the CAR scheme the covariance matrix for X_i determines $\mathbf{B}_c$, whereas with a SAR process many $\mathbf{B}_s$ can give X_i's covariance matrix, see Ripley (1981). Also, it is to be noted is that any SAR process is a CAR process with $\mathbf{B}_c = \mathbf{B}_s + \mathbf{B}_s^T - \mathbf{B}_s^T\mathbf{B}_s$. The reverse is also true, taking $\mathbf{B}_s = (\mathbf{I} - \mathbf{L}^T)$ where $\mathbf{L}\mathbf{L}^T$ is the Cholesky decomposition of $(\mathbf{I} - \mathbf{B}_c)$. This form of decomposition is discussed later.

2.3.4 Latent Gaussian Models

The data in which we are interested in modeling is geostatistical in nature. Often with geostatistical data the phenomenon itself is not directly observable or may be controlled by unobservable variables. Hence, we will assume a model where there exists an unobserved latent stochastic process $X(s)$ and that a relationship exists between $Y(s)$ (the observed process) and $X(s)$. The joint distribution of the observed and latent processes is the distribution of interest and can be specified by the marginal distribution of $X = (X(s_1), \ldots, X(s_n))$ and the conditional distribution of $Y = (Y(s_1), \ldots, Y(s_n))$ given X;

$$\pi(Y, X) = \pi(Y|X)\pi(X),$$

We need to model $\pi(Y|X)$ and $\pi(X)$. To do this there are a few assumptions commonly made. The first is that X is a Gaussian process and the second is that conditional on X, the Y's are independent. Given a Gaussian process $X \sim N(\mu, \Sigma)$ with correlation function $\rho(u)$ as described in Section 2.1.1, then modeling a Y process conditional on a latent Gaussian process X is commonly presented as Gaussian, ie

$$Y|X \sim N(X, \sigma^2)$$

or

$$Y(s_i) = X(s_i) + \epsilon_i,$$

where $i = (1, \ldots, n)$ indicate locations and $\epsilon_1, \ldots, \epsilon_n$ are iid $N(0, \sigma^2)$. The marginal distribution of Y is the multivariate Gaussian,

$$Y \sim N(\mu 1, R),$$

where 1 is a row vector of size n of ones, R is an $n \times n$ matrix with $R = \Sigma + \sigma^2 I$ and $\Sigma_{ij} = \text{var}(x)\rho(\|x_i - x_j\|)$, thus

$$\rho(y_i, y_j) = \frac{\text{var}(x)}{\text{var}(x) + \text{var}(\epsilon)} \rho(x_i, x_j).$$

Other models which extend from this view Y as a function of X and some other known covariates.

2.3.5 Latent Spatial Model for Univariate Count Data

In many cases, such as those mentioned in Section 2.2, one is dealing with count data, which are assumed to follow a Poisson distribution;

$$Y_i | X \sim \text{Poisson}(\lambda(s_i))$$

where $\lambda(s_i) = \exp(X(s_i))$ or $\lambda(s_i) = c_i \exp(X(s_i))$. The additional location specific parameter c_i may be useful in accounting for varying amounts of time in measuring Y (as in the Rongelap Island data considered later) or proportional population size for an area.

In the univariate model, data have a mean that is a function of a Gaussian field. The model can take many forms. The one presented here is quite common in disease oriented applications, due to its count nature.

Let $s_1, \ldots, s_n$ be spatial locations and the data $(y_1, \ldots, y_n)$ be observed at $s_1, \ldots, s_n$. Assume that

$$Y(s_i) \sim \text{Poisson}(\exp(\beta + X(s_i))),$$

for $i = 1, \ldots, n$. The latent variable $X = (X(s_1), \ldots, X(s_n))$ is multivariate normal

$$X \sim \text{MVN}(0, \Sigma(\theta)),$$

where $\theta = (\alpha, \sigma^2, \delta)$, and

$$\Sigma_{ij}(\theta) \quad = \quad \sigma^2 \exp(-(\alpha d_{ij})^\delta), \tag{2.1}$$

for distance d_{ij} between s_i and s_j. In our model the distance has been scaled by the maximum distance over the region so that $0 < d_{ij} < 1$. The correlation matrix for this model could be parameterized to have any of a number of forms, for example a simplified version is:

$$\rho_{ij}(\theta) = \exp(-\alpha d_{ij})$$

which is achieved by letting $\delta = 1$, alternatively any of the functions mentioned in Section 2.1.1. Here α is equal to $\frac{1}{\phi}$ for the functions described in Section 2.1.1.

In the above model there may be a number of quantities of interest. If we are concerned with disease mapping, then the underlying distribution of the field is our key concern, but in other circumstances evaluation of any one of the individual parameters or the nature of the spatial correlation may be the structure of interest.

When conducting Bayesian inference the evaluation of the posterior is the element of import. The joint posterior for the univariate model above is given as:

$$\begin{aligned} p(X, \beta, \theta | y) &\propto p(y|X, \beta) p(X|\theta) p(\theta) p(\beta) \\ &\propto \exp(\sum y_i(\beta + X_i) - \sum e^{(\beta + X_i)}) \\ &\quad \times |\Sigma(\theta)|^{-\frac{1}{2}} \exp(-\frac{1}{2} X^T \Sigma(\theta)^{-1} X) p(\theta) p(\beta). \end{aligned}$$

We assume uniform priors. For β the range is $[-100, 1000]$, for α the range is $[0, 100]$, for σ^2 the range is $[0, 100^2]$, while for δ it is $[0, 2]$; δ must lie in this range, see Chapter 4, Section 4.1.1 for details. The ranges for β and α are very large and in practice their values are always well within them. In particular circumstances one would often have prior information that could lead to a more informative prior being used. In particular, one often uses inverse-gamma or lognormal for σ^2

2.3.6 Latent Spatial Model for Multivariate Count Data

Another form of data which we are interested in examining is multivariate. Here the data are observed at n spatial locations, whose mean is a function of a linear combination of T independent latent spatial Gaussian processes.

Let $s_1, \ldots, s_n$, be spatial locations. We assume that

$$Y_j(s_i) \sim \text{Poisson}(\exp(\lambda_{ij})), \quad j = 1, \ldots, r$$

are observed at each location s_i, for $i = 1, \ldots, n$, i.e. there are r response variables. Further, we assume

$$\lambda_{ij} = \beta_{j0} + \sum_{t=1}^{T} \beta_{jt} X_{it}$$

where X_t are independent latent Gaussian processes

$$X_t \sim \text{MVN}(0, \Sigma(\theta_t)),$$

with $\theta_t = (\alpha_t, \sigma_t^2)$ and

$$\Sigma_{ik}(\theta_t) = \sigma_t^2 \exp(-(\alpha_t d_{ik})),$$

where d_{ik} is the distance between s_i and s_k. This is the multivariate extension of the model described by Diggle et al (1998). It allows correlations both between and within locations. We fix σ^2 to avoid problems of identifiability with β_{jt}; see Chapter 5.

To conduct Bayesian inference with such a model, we must compute the posterior distribution:

$$\begin{aligned} p(X,\beta,\theta|Y) &\propto p(Y|X,\beta).p(X|\theta).p(\theta).p(\beta) \\ &\propto \prod_{j=1}^{r}\left[\prod_{i=1}^{n}\exp^{Y_{ij}\lambda_{ij}}(\exp^{-\exp^{(\lambda_{ij})}})\right]p(\beta_j) \\ &\times \prod_{t=1}^{T}|\Sigma(\theta_t)|^{-\frac{1}{2}}\exp(-\frac{1}{2}X_t^T\Sigma(\theta_t)^{-1}X_t)\,p(\theta_t). \end{aligned}$$

where $X = (X_1, \ldots, X_T)$, $\beta = (\beta_1, \ldots, \beta_r)$, $\theta = (\theta_1, \ldots, \theta_T)$ and $\lambda_{ij} = \beta_{j0} + \sum_{t=1}^{T}\beta_{jt}X_{it}$. See Appendix A.2.1 for details on the priors for this model.

2.3.7 Simulation of a Gaussian Process

In what follows we make extensive use of the simulation of a multivariate normal distribution. To simulate an n-dimensional Gaussian process, whose joint probability density function is MVN(m, Σ), the following properties are useful. Let

$$\zeta = (\zeta_1, \zeta_2, \ldots, \zeta_n)$$

be independent $N(0,1)$ random variables and thus $\zeta \sim MVN(0, I)$. By properties of the Gaussian distribution, $L\zeta \sim MVN(0, LL^T)$, for any matrix L. Thus, one can simulate X by finding a matrix L such that,

$$LL^T = \Sigma$$

then $L\zeta \sim MVN(0, LL^T)$ and so $m + L\zeta \sim MVN(m, \Sigma)$. This L can be found using *Cholesky decomposition.* Cholesky decomposition states that there is a lower triangular matrix L with $LL^T = \Sigma$, if Σ is symmetric positive definite, see Press et al. (1986) for details. The probability density function of the Gaussian process is,

$$\begin{aligned} \pi(X) &\propto \exp(-\frac{1}{2}X^T\Sigma^{-1}X) \\ &= \exp(-\frac{1}{2}X^T(LL^T)^{-1}X) \\ &= \exp(-\frac{1}{2}(L^{-1}X)^T I(L^{-1}X)) \end{aligned}$$

and the standard multivariate Gaussian density function is,

$$\pi(\zeta) \propto \exp(-\frac{1}{2}\zeta^T I \zeta)$$

where I is the identity matrix. Hence, $L^{-1}X$ is standard Gaussian, so we can sample from X using the relationship $L^{-1}X = \zeta$ and thus $X = L\zeta$.

As regards computation time, this process is $O(n^3)$, see Press et al. (1986), or in the case of Markov random fields, if Σ is spare then it will be $O(n^2)$, see Rue (2001); further computational approaches of this variety are outlined by Rue (2001). These techniques are found in most numerical linear algebra texts. This decomposition also turns out to be one of the dominant characteristics utilised in calculations of Chapters 4 and 5.

Chapter 3

Statistical Methodology

3.1 Statistics

Statistical inference is concerned with drawing conclusions about unknown quantities of interest from data and other information. Usually, the data are not sufficient to determine the unknown quantities exactly, or are themselves observed with uncertainty leading to uncertainty in the values of the unknowns. Statistical inference quantifies this uncertainty by probability. Inference from data takes one of two main approaches: the Frequentist (or Classical) approach or the Bayesian approach. These differ in the way in which they interpret probability as the measure of uncertainty:

Long-run frequency probability: Frequentist inference interprets the probability of an event as the proportion of the time it would occur in a long sequence of observations (i.e. as the number of trials tends to infinity).

Subjective probability: Bayesian inference has the probability of an event as a number between 0 and 1 that measures a particular person's subjective opinion as to how likely that event is to occur.

The Frequentist approach is based solely on the observation of the occurrence of events, i.e. utilising the former definition of probability. Bayesian statistics uses probability subjectively, and can incorporate prior knowledge about the event, that may change as more information becomes available. Thus the posterior becomes the new prior when new information becomes arises. Motivations for utilising the Bayesian framework are that it is a conceptually simple method, it has a strong axiomatic foundation, the interpretation of its conclusions are intuitive and it lends itself to complex probability models, which allow for more realistic modelling. It is the approach that we follow in this dissertation. It does have drawbacks, notably computational complexity. This dissertation concerns itself with tracking one aspect of these difficulties in the context of spatial models.

3.2 The Bayesian Framework

Bayesian inference is founded on the notion that probability, interpreted subjectively, is the only way to describe uncertainty. In practical terms, this can be thought of in a similar way to carrying out a survey or experiment. Before the experiment is carried out, there is usually some prior knowledge or degree of belief about an unknown quantity (or random variable) of interest, denoted θ. This can be expressed in the form of a probability statement. Lets us refer to the knowledge or background information as H. For example, if we were to do a survey on the number of students who wear reading glasses, we might believe it to be in a certain range. This belief may derive from exposure to the population of interest, a previous study of a subset population, eg. maths students, or some other useful observation. A Bayesian will quantify this uncertainty about θ using subjective probability. This will be a function of two arguments: the unknown θ and the known H. This probability $P(\theta|H)$, as a function of θ must obey the three *Laws of Probability*:

1. **Convexity**: $0 \leq P(\theta|H) \leq 1$, where if an event is certain $P(\theta) = 1$ and if an event is impossible $P(\theta) = 0$.

2. **Additivity**: $P(\theta_1 \text{ or } \theta_2|H) = P(\theta_1|H)+P(\theta_2|H)$, if θ_1 and θ_2 are mutually exclusive.

3. **Multiplicativity**: $P(\theta_1 \text{ and } \theta_2|H) = P(\theta_1|H).P(\theta_2|\theta_1, H)$

The $P(\theta|H)$ may vary from person to person, given that their experience of H may vary. Given that some amount of prior knowledge often exists, assigning prior probability is the most reasonable approach to employ. In the case where there is no prior knowledge available, a uniform or flat prior maybe used ,i.e. $\pi(\theta) = 1$. The formal approach and implications in applying this idea are given below. For convenience, we do not write H in any further probability statement.

3.2.1 Bayes Theorem

This theorem was established by Reverend Thomas Bayes, an English minister and part-time mathematician. Bayes theory of probability was the first to invert the probability statement ("inverse probability" or Bayesian inference). That is obtain probability statements about θ, the parameter of interest, given the observed data y. Stigler (1986) describes the historical development of inverse probability, as does Dale (1991) and many others. Bayes famous paper was published posthumously in 1763 in the "Philosophical Transactions of the Royal Society", and was entitled "Essay towards solving a problem in the doctrine of chances". Suppose $y = (y_1, \ldots y_n)$ is a vector of n observation, whose probability density $\pi(y|\theta)$ depends on the value of k parameters $\theta = (\theta_1, \ldots \theta_k)$. Suppose also that θ itself has a probability density $\pi(\theta)$. Then by the multiplicativity law

$$\pi(y|\theta)\pi(\theta) = \pi(y, \theta) = \pi(\theta|y)\pi(y),$$

hence

$$\pi(\theta|y) = \frac{\pi(y|\theta)\pi(\theta)}{\pi(y)},$$

where

$$\pi(y) = E[\pi(y|\theta)] = \begin{cases} \sum \pi(y|\theta)\pi(\theta), & \theta \text{ discrete}, \\ \int \pi(y|\theta)\pi(\theta)\ d\theta, & \theta \text{ continuous}, \end{cases}$$

is taken over the admissible range of θ. Given that the variable is θ and y is a known constant, the above equation can be written in its more familiar form:

$$\pi(\theta|y) \propto \pi(y|\theta)\pi(\theta).$$

There are three components to Bayes theorem.

Prior

The $\pi(\theta)$ is referred to as the prior distribution of θ and reflects the knowledge known about θ *a priori*, before the data are observed.

Likelihood

The $\pi(y|\theta)$ may be regarded as a function not of y, but of θ, in which case it is referred to as the "likelihood" (or sometimes the Model) for θ given y and can be written $l(\theta|y)$. Note that the likelihood is a conditional probability statement, as to how likely it is for y to be observed if the parameters take the value θ.

Posterior Distribution

In Bayesian analysis, it is the conditional distribution of θ given the data (y) which is of interest, i.e. $\pi(\theta|y)$. This is called the *posterior* distribution, thus Bayes Theorem can be written less formally as:

$$\textit{posterior distribution} \propto \textit{likelihood} \times \textit{prior distribution}.$$

This distribution describes the state of knowledge about θ having observed y.

3.2.2 Posterior Expectations

All measures of interest within Bayesian inference are functions of the posterior distribution. Posterior expectations for functions of θ is the way in which most quantities of interest are expressed, for example, mean values

$$\begin{aligned} E[f(\theta)|y] &= \frac{\int f(\theta)\pi(y|\theta).\pi(\theta)d\theta}{\int \pi(y|\theta).\pi(\theta)d\theta} \\ &= \int f(\theta)\pi(\theta|y)d\theta \end{aligned} \tag{3.1}$$

or regions of highest density. To compute the normalising constant, find marginals or calculate posterior expectations and thus integration is required. This integration is the source of most practical difficulties for Bayesian inference, especially in high dimensional problems.

3.2.3 Methods of Evaluation of the Posterior Distribution

The principal obstacle to implementing Bayesian inference, particularly for complex models, is the evaluation of integrals. There are several approaches:

1. Analytical evaluation, although this is possible for only a few models and in low dimensions of θ. Conjugate priors may be used to assist in enabling analytical or partially analytical solutions to integrands such as the denominator of expression 3.1; (see Carlin and Louis, 1996, Chapter 2 for elaboration of this approach).

2. Numerical evaluation (also called numerical integration or quadrature) methods such as Simpson's Rule and Gaussian quadrature. These tend to be difficult to apply and inaccurate for high dimensional (> 3) problems, see Press et al. (1986).

3. Asymptotic methods include the Laplace approximation, normal approximation (Gelman et al., 1995a) and Monte Carlo integration. Asymptotic methods rely on results obtained when the sample size n gets large.

The Laplace Method and Normal Approximation To evaluate the integral in Equation 3.1, the *Laplace Method* involves expressing the integrand in the form $\exp\left[\log\left(f(\theta)\pi(\theta|y)\right)\right]$, then expanding $\log[f(\theta)\pi(\theta|y)]$ as a function of θ in a quadratic Taylor series around the mode. The resulting approximation is proportional to a normal density in θ and its integral is:

$$f(\theta_o)\pi(\theta_o|y)(2\pi)^{\frac{1}{2}}|-u^{''}(\theta_o)|^{-\frac{1}{2}},$$

where d is the dimension of θ, $u(\theta)=\log(f(\theta)\pi(\theta|y))$ and θ_o is the point at which $u(\theta)$ is maximised e.g. the maximum of the integrand of 3.1, which we can find by maximising $\log(f(\theta)\pi(y|\theta)\pi(\theta))$, from Gelman et al. (1995a) Chapter 10. This is an approximation to the integrand of 3.1. The *Normal approximation* is a more basic version of the Laplace method that performs poorly in multimodal or asymmetrical situations; (see Carlin and Louis, 1996, Chapter 5 for a description of these methods applied to posterior distribution estimation) .

Monte Carlo integration Monte Carlo integration is the method of approximating integrals using samples from a probability distribution. Having drawn samples from the required distribution, it then forms sample averages to approximate expectations.

$$\begin{aligned}\int f(x)\,dx &= \int \frac{f(x)}{p(x)}p(x)\,dx\\ &= E\left(\frac{f(x)}{p(x)}\right)\\ &\approx \frac{1}{N}\sum_{i=1}^{N}\frac{f(x_i)}{p(x_i)}\end{aligned}$$

where x_i is a sample from the probability density function $p(x)$. Monte Carlo integration uses samples that have been obtained by a Monte Carlo method. These methods divide into two categories: non-iterative and iterative methods. On the non-iterative side there are methods such as rejection sampling and importance sampling, while the iterative approach generally refers to a group of methods collectively known as Markov Chain Monte Carlo (MCMC).

Monte Carlo integration is the most common approach to take in evaluating the above integrals. Both iterative and non-iterative Monte Carlo methods are discussed in the next section. It is however iterative methods that are the primary source of interest in this dissertation and as such are examined in greater detail. The methods are introduced by way of some historical background.

3.3 Monte Carlo Methods

In the past, complex data have often been modelled using overly simple models in order that the inference could be implemented, which was not always entirely satisfactory. While Bayesian methods are theoretically reasonably simple, being the application of the laws of probability, they require evaluation of complex integrals, such as constants of proportionality and expectations with the form of those mentioned above. Only in the most rudimentary of cases are these integrals analytically tractable. It is with the advent of modern numerical techniques and advanced computing power that these problems have become accessible. MCMC (Markov chain Monte Carlo) in particular has provided a method which allows for inference with much more complex models.

The method is named after the city of the same name in Monaco, due to its association with gambling and specifically roulette, the roulette wheel itself being a simple random number generator. The name and the method's systematic development date from 1944, but many isolated incidences of its informal use exist prior to that (Comte de Buffon, Lord Kelvin and Student (1908) to name but a few).

The origins of the method as a research tool stems from work on the Manhattan project during World War II. It was also at around this time, that the first electronic computer (the ENIAC) was completed. Ulam and von Neumann suggested its use in simulation of the probabilistic problems concerned with random neutron diffusion in fissile materials, via random samples. Metropolis and Ulam published a paper in 1949 detailing the idea behind this and thus opening up a new field of research (Metropolis and Ulam, 1949).

The Monte Carlo method is, in general terms, any technique used for obtaining solutions to (deterministic) problems using random numbers. Presented below are some of the more popular approaches of the method. The approaches have been subdivided into the categories of iterative and non-iterative.

Non-iterative Monte Carlo For *Importance Sampling*, to approximate the posterior expectation given by Equation 3.1, let

$$g(\theta) \approx c\pi(y|\theta)\pi(\theta)$$

for some easily sampled density $g(\theta)$ and the normalizing constant c. Then defining a *weight function* $w(\theta) = \pi(y|\theta)\pi(\theta)/g(\theta)$,

$$\begin{aligned} E(f(\theta)|y) &= \frac{\int \frac{f(\theta)\pi(y|\theta)g(\theta)}{g(\theta)} d\theta}{\int \frac{\pi(y|\theta)\pi(\theta)g(\theta)}{g(\theta)} d\theta} && (3.2) \\ &= \frac{\int f(\theta)w(\theta)g(\theta)\ d\theta}{\int w(\theta)g(\theta)\ d\theta} \\ &\approx \frac{\frac{1}{N}\sum_{i=1}^{N} f(\theta_i)w(\theta_i)}{\frac{1}{N}\sum_{i=1}^{N} w(\theta_i)} && (3.3) \end{aligned}$$

where $\theta_i \overset{iid}{\sim} g(\theta)$ and $g(\theta)$ is know as the *importance function.* How closely $g(\theta)$ resembles $c\pi(y|\theta)\pi(\theta)$ determines how good the approximation given by Equation 3.2 is.

For the method of *Rejection Sampling*, instead of trying to approximate the normalized posterior

$$\pi(\theta|y) = \frac{\pi(y|\theta)\pi(\theta)}{\int \pi(y|\theta)\pi(\theta)\, d\theta},$$

a probability density function $g(\theta)$ is introduced. Suppose then there exists a constant $M > 0$, such that $\pi(y|\theta)\pi(\theta) < Mg(\theta)$ for all θ. The rejection method then proceeds as follows:

(a) Generate $\theta_i \sim g(\theta)$.

(b) Generate $U \sim \text{Uniform}(0, 1)$.

(c) If $MUg(\theta_i) < \pi(y|\theta_i)\pi(\theta_i)$, accept θ_i, otherwise reject θ_i.

(d) Return to step (a) and repeat until the desired sample of θ_i's has been obtained.

The accepted θ_i's are random variables from $\pi(\theta|y)$. M should be chosen such that as few as possible samples are unnecessarily rejected.

Iterative Monte Carlo Markov chain Monte Carlo is Monte Carlo integration combined with the use of Markov chains. MCMC draws samples from a Markov chain whose stationary distribution is the distribution of interest (also referred to as the target distribution). The distribution of interest is the posterior distribution in the case of Bayesian statistics. When the chain has reached its stationary distribution, an adequate sample from the support of the distribution can then be obtained. The reason for using Markov chains is that with Monte Carlo integration, when the target distribution $\pi(x)$ is not a standard one, it may be difficult to draw samples from it directly, i.e. it may not have a closed form and importance or rejection sampling can be difficult. Some underlying theory as to why MCMC works is more formally detailed in the next two sections, as are its primary algorithms.

3.3.1 Markov Chains

The idea of Markov dependence is a concept attributed to a Russian mathematician Andrei Andreivich Markov. At the start of the 20th century, he investigated the sequence of vowels and consonants in the poem "Onegin" by Puskin. He developed a probabilistic model where successive letters depended on all their predecessors only through the immediate predecessor. The model allowed him to obtain good estimates of the relative frequency of vowels in the poem. The French mathematician Henri Poincare studied sequences of random variables that were in fact Markov chains, at around the same time.

For a stochastic process as given by Definition 2.1, we think of T as discrete time, then the stochastic process X_t can be thought of as the path of a particle moving randomly in (state) space D, observed at discrete times and its position at time t being X_t.

Definition 3.1 (Markov Chain). *A Markov chain is a stochastic process $\{X_t : t \in \mathbb{N}\}$, for which the conditional distribution of X_{t+1} is independent of $X_1, \ldots, X_{t-1}$ given X_t. That is $\forall\ \tau_1, \tau_2, \ldots, \tau_{t+1} \in D$:*

$$\mathrm{P}(X_{t+1} = \tau_{t+1} | X_t = \tau_t, X_{t-1} = \tau_{t-1}, \ldots, X_0 = \tau_0) = \mathrm{P}(X_{t+1} = \tau_{t+1} | X_t = \tau_t).$$

This is referred to as the Markov property.

Definition 3.2 (Stationary in time). *A Markov chain X_t is said to be stationary (or homogeneous) in time if the conditional probabilities are independent of t. That is $\forall\, i, j \in D$ and $\forall\, t \in \{0, 1, \ldots\}$,*

$$\mathrm{P}(X_{t+1} = j | X_t = i) = \mathrm{P}(X_1 = j | X_0 = i).$$

A matrix of these probabilities is known as a *transition matrix* for the discrete finite case (or *transition kernel*, in the continuous case) and is denoted P^i_j, i.e the probability of transition from one state i to another state j. For simplicity the definitions that follow are given with regard to the discrete case. The entries in the matrix are in $[0, 1]$ and

$\sum_j \mathrm{P}^i_j = 1$ since:

$$\begin{aligned}\sum_j \mathrm{P}^i_j &= \sum_j \mathrm{P}(X_t = j | X_{t-1} = i), \\ &= \mathrm{P}(X_t \in D | X_{t-1} = i), \\ &= 1 \qquad\qquad \text{for a finite statespace.}\end{aligned}$$

Key concepts, especially in the context of simulation, are that of a stationary distribution ϕ and the asymptotic behaviour of the chain as the number of steps or iterations $t \to \infty$.

Definition 3.3 (Stationary Distribution). *A distribution ϕ on S is said to be a stationary distribution of a chain with transition probability* P *if:*

$$\phi = \phi \mathrm{P}$$

Once a chain reaches a stage where ϕ is the distribution of the chain, the chain retains this distribution ϕ for all subsequent stages. This distribution is also known as the "invariant" or "equilibrium" distribution. It is the existence of this stationary distribution in a Markov chain that allows us to sample from a target distribution π. The conditions necessary to orchestrate such a chain with the required stationary distribution are outlined in the next section, however there are some further constraints on the chain which need to be mentioned here. For a Markov chain to converge in distribution to a unique stationary distribution, it is sufficient to exhibit the following properties: irreducibility, recurrence and aperiodicity.

Definition 3.4 (Irreducibility). *Let $C \subseteq D$, $(i,j) \in C$ and $i \neq j$. Then C is said to be irreducible if $\exists\ n < \infty$ such that:*

$$(\mathrm{P}^i_j)^n > 0, \quad \forall\ (i,j) \in C$$

where $(\mathrm{P}^i_j)^n = \mathrm{P}(X_{t+n} = j | X_t = i)$.

Definition 3.5 (Recurrence). *A state $j \in D$ is said to be recurrent if the Markov chain starting in position j returns to j with probability 1. (Or positive recurrent if in addition the mean time to return is finite.)*

Definition 3.6 (Aperiodic). *The period d_j of a state j is the largest common divisor of the set $\{n \geq 1 : (\mathrm{P}_j^j)^n \geq 0\}$. A state j is aperiodic if $d_j = 1$.*

Definition 3.7 (Ergodic). *If all states of a chain are positive recurrent and aperiodic, then it is said to be ergodic.*

Definition 3.8 (Limiting Distribution). *If the stationary distribution ϕ exists and*

$$\lim_{n\to\infty} \mathrm{P}^n \phi_0 = \phi,$$

independent of the initial distribution ϕ_0 of the chain, $\phi^n (= P^n \phi_0)$ will approach ϕ as $n \to \infty$. This is referred to as the Limiting Distribution.

Theorem 3.1 (Ergodic). *For an irreducible ergodic Markov chain, a limiting distribution $\phi(j) = \lim_{n\to\infty}(\mathrm{P}_j^i)^n$ exists such that:*

$$\phi(j) = \sum_{i=0}^{\infty} \phi \mathrm{P}_j^i$$

i.e. $\phi(j)$ is the unique limiting distribution, which is the chain's stationary distribution.

A proof for this is found in many places, for example Grimmett and Stirzaker (1982).

3.3.2 Metropolis-Hastings Algorithm

Many methods have been proposed to construct Markov chains having a given stationary distribution, but all of them are special cases of the Metropolis et al. (1953) and Hastings (1970) general framework. This was proposed by Hastings (1970) as a general form of the Metropolis algorithm. The algorithm works as follows to produce a Markov chain whose stationary distribution is $\pi(s)$.

1. Pick a starting value $x_0 \in D$.

2. For $t = 0, 1, \ldots$ choose a candidate point $x^{'}$ from a distribution $q(x|x_t)$, where q is called the proposal distribution. The proposal distribution is arbitrarily chosen, but generally depends on the current point x_t, e.g. $q(.|x_t) = N(x_t, \sigma^2)$, where σ^2 is fixed.

3. The computation of the acceptance probability for $x^{'}$ is given by $\alpha(x_t, x^{'})$ where

$$\alpha(x_t, x^{'}) = \min\left(1, \frac{\pi(x^{'})q(x_t|x^{'})}{\pi(x_t)q(x^{'}|x_t)}\right)$$

4. The next state of the chain is then:

$$x_{t+1} = \begin{cases} x^{'} & \text{with probability } \alpha(x_t, x^{'}), \\ x_t & \text{otherwise,} \end{cases}$$

i.e. $x_{t+1} = x^{'}$ if the candidate point is accepted, or the chain maintains its current value, if the candidate point is rejected.

Justification Suppose that the chain has already reached equilibrium at iteration $t-1$, i.e. $\phi_{t-1} = \pi$. Then we need to choose a transition probability $P(x_t, x^{'})$ to maintain the equilibrium distribution, i.e. $\phi_t = \pi$. Consider moving between any two states x_t and $x^{'}$. To go from x_t to $x^{'}$ the transition probability is:

$$P(x_t, x^{'}) \propto \pi(x^{'})q(x^{'}|x_t)\alpha(x_t, x^{'})$$

and conversely if going from $x^{'}$ to x_t. To maintain equilibrium, it is sufficient that these densities be equal:

$$\pi(x^{'})q(x^{'}|x_t)\alpha(x_t, x^{'}) = \pi(x_t)q(x_t|x^{'})\alpha(x^{'}, x_t)$$

a condition that is referred to as "detailed balance". To see this, taking $P(x_t, x^{'})$ as the elements of P, given that detailed balance exists and $\phi_{t-1} \propto \pi$ we can verify that equilibrium is preserved, since

$$\begin{aligned}\phi_t(x_t) &= \phi_{t-1}(x_t)\left[1-\int_S q(x_t|x^{'})\alpha(x_t,x^{'})\mathrm{d}x^{'}\right]+\int_S \phi_{t-1}(x^{'})q(x^{'}|x_t)\alpha(x^{'},x_t)\mathrm{d}x^{'}\\ &= \phi_{t-1}(x_t)+\int_S\left[\phi_{t-1}(x^{'})q(x^{'}|x_t)\alpha(x^{'},x_t)-\phi_{t-1}(x_t)q(x_t|x^{'})\alpha(x_t,x^{'})\right]\mathrm{d}x^{'}\\ &= \phi_{t-1}(x_t) \qquad \text{(due to detailed balance);}\end{aligned}$$

see Gilks et al. (1996), Chapter 1, Section 1.3 for details. Given that π is fixed and $q(.|.)$ is chosen arbitrarily, the acceptance probability must be the element which allows us to control the distribution of the chain. There are in fact many acceptance functions which may be chosen to ensure the correct stationary distribution of the chain. According to the accept probability given above, the expected number of moves will be the same in each direction. This is the optimal accept probability with respect to reaching equilibrium as quickly as possible and traversing the distribution, as shown by Peskun (1973).

Metropolis Algorithm

The Metropolis algorithm is a special case where the proposals are symmetric, i.e. they have the form $q(x^{'}|x)=q(x|x^{'})$. Hence the accept probability reduces to:

$$\alpha(x,x^{'})=\min\left(1,\frac{\pi(x^{'})}{\pi(x)}\right).$$

Metropolis-within-Gibbs

The Single-Component Metropolis is a variation of the Metropolis algorithm, and is more commonly referred to as "Metropolis-within-Gibbs". Instead of updating the whole of X together, X can be divided up into a number of components $X_1,\ldots,X_k$, then these components are updated individually. It samples as follows:

1. Let $X_{-i}=(X_1,\ldots,X_{i-1},X_{i+1},\ldots,X_k)$, so that there are $k-1$ components (the size of each component need not be the same).

2. Let x_i^t denote the state of X_i at the end of iteration t and $q_i(.|.)$ proposes a point for the i^{th} component only.

3. For $\pi(x_i|x_{-i})$, (this is called the full-conditional distribution of X_i), the accept probability for $x_i^{'}$ is then:

$$\alpha(x_i^t x_i^{'}) = \min\left(1, \frac{\pi(x_i^{'}|x_{-i})q_i(x_i^t|x_i^{'}, x_{-i})}{\pi(x_i^t|x_{-i})q_i(x_i^{'}|x_i^t, x_{-i})}\right)$$

.

The idea is each updating step produces a move in the direction only of the element in X_i (if the candidate is accepted).

Gibbs Sampler

The Gibbs Sampler is a special case of Metropolis-within-Gibbs. The Gibbs sampler, as introduced in a statistical inference context by Geman and Geman (1984), and popularized by Gelfand and Smith (1990), is largely responsible for the introduction of MCMC to Bayesian statistics and for the increased popularity of Bayesian statistics. In turn this has lead to the increased popularity of Bayesian statistics.

For the Gibbs sampler, the proposal distribution for X_i is its full conditional distribution (see below for details). This leads to an accept probability of 1, i.e. the proposal value is always accepted. An iteration of the algorithm is:

$$\begin{aligned}
&X_1^{(t)} \sim \pi(x_1|x_{-1}^{t-1}) \\
&X_2^{(t)} \sim \pi(x_2|x_1^t, x_3^{t-1}, \ldots, x_n^{t-1}) \\
&\vdots \\
&X_n^{(t)} \sim \pi(x_n|x_{-n}^t)
\end{aligned}$$

Note: This requires that the full conditionals are available for sampling.

Full Conditional By the term full conditional of X_i is meant

$$\pi(x_i|x_{-i}) = \frac{\pi(x_i, x_{-i})}{\int \pi(x_i, x_{-i})\, dx_i} \propto \pi(x_i, x_{-i}), \tag{3.4}$$

where x_{-i} denotes a vector x without the i^{th} component and in the present context $\pi(x)$ is our stationary distribution, partitioned into n components. The full conditional is derived from the joint distribution of the variables. In the case of the posterior distribution, where Y is observed, the joint posterior distribution for X is:

$$\pi(x) = \pi(x|y) = \frac{\pi(x,y)}{\int \pi(x,y)\ dx}. \tag{3.5}$$

From Equations 3.4 and 3.5, the full conditional for X_i can be found by noticing that:

$$\begin{aligned} \pi(x) &= \frac{\pi(x_i, x_{-i}|y)}{\pi(x_{-i}|y)}, \\ &= \frac{\pi(x_i, x_{-i}, y)}{\pi(x_{-i}, y)}, \\ &\propto \pi(x_i, x_{-i}, y), \end{aligned} \tag{3.6}$$

since the denominator of expression 3.6 does not depend on X_i. Thus to construct a full conditional for X_i, we are only required to take the terms in the joint (posterior) distribution that involve X_i.

3.4 MCMC Mixing

For a Monte Carlo method to work well, it is important that it produces a good representative sample from the target distribution. If this has not happened then the approximations computed using the sample may not be reliable. When one uses MCMC, this property is called *mixing*.

3.4.1 Why is Mixing Important?

Mixing is a property of a Markov chain that has attained its stationary distribution. It refers to the speed with which the chain explores the support of the stationary distribution. It is a qualitative concept, but "good mixing" occurs when the simulated chain traverses the entire parameter space rapidly, spending short periods of time in the extremes of the distribution and being predominantly in the body of the distribution. This is desirable because rapid mixing means fewer iterations of the Markov chain are required to have a good representative sample of π. One thing to note is that sometimes there is a trade-off between rapid mixing and speed with which a sample is obtained. For example, if a very sophisticated algorithm mixes better per iteration than a simple one, but its CPU time per iteration is much larger, then it may not be a practical improvement.

Slow mixing obviously means that we have to run the chain for longer to get a reliable sample. One of the main causes of slow mixing is strong correlation between the variables. An example of a badly mixing Markov chain is the Gibbs sampler for a bivariate normal density with strong correlation. That is, since the target distribution $\pi(.)$ is concentrated around a diagonal, the proposal X_{t+1} will be concentrated close to X_t, hence the chain will move slowly. Such a situation is illustrated in Figure 3.1 (a). Multimodality within $\pi(.)$ is also a cause of slow mixing.

3.4.2 Why is Mixing a Problem in Spatial Models?

Spatial models tend to have a combination of attributes which cause problems for mixing. They are usually of high dimension, with many of the parameters being strongly correlated. The posterior distribution may also be multimodal, especially in the case of mapping disease incidence. They are a difficult class of problems, but the techniques below may help in finding suitable schemes with which to implement them.

3.4.3 Approaches to Improving Mixing

Careful Choice of Proposals

Take for example a random walk proposal:

$$q(X, X^{'}) = q(|X - X^{'}|).$$

It may take the form $X' \sim N(X, \sigma^2)$, where it is indexed by a scale parameter σ, which needs to be chosen carefully:

- A small σ creates a conservative proposal distribution that generates small steps. In this case X' will generally have a high acceptance rate, but will move slowly through the support. An example of this undesirable behaviour is shown in Figure 3.2 (c).

- A large σ generates large steps and will propose moves from the body to the tails of the distribution. This gives a low accept rate, i.e. the chain will frequently not move, also resulting in slow mixing, as seen in Figure 3.2 (b).

Theoretical justification for aiming to have the proportion of times that a proposal is accepted to be in the range $[0.15, 0.5]$ is provided by Gelman et al. (1995b).

Reparameterization

If there is strong correlation among the X_i's, appropriate reparameterization should reduce it and improve mixing. This involves transforming X to new variable Y, so that there is less correlation between the components. In the case of the Metropolis algorithm, another strategy would be to transform the proposals, which is equivalent to reparameterizing.

Coupling

When multimodality is the cause of slow mixing, reparameterization will not help much. A better solution might be Metropolis-Coupled MCMC (Geyer, 1991) or a similar hybrid MCMC. MCMCMC requires the running of m MCMC chains in parallel, with different stationary distributions $\pi_i(x)$, for $i = 1, \ldots, m$, where $\pi_1(x) = \pi(x)$ and $\{\pi_i(x), i > 1\}$ are chosen to improve mixing, for example:

$$\pi_i(x) \propto \pi(x)^{\frac{1}{1+\lambda(i-1)}}, \lambda > 0$$

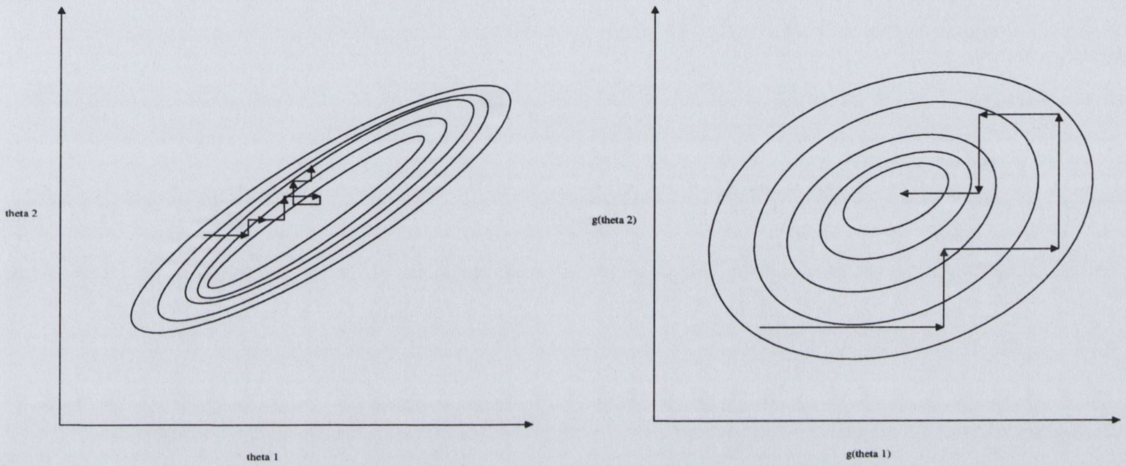


Figure 3.1: Plot(a) depicts the contour lines of a bivariate posterior density with components that are highly correlated. Also shown is a possible chain trajectory illustrating slow movement within the distribution. In plot (b), a transformation of the parameters has been performed, which reduces the dependence between the components and allows the posterior to be explored much more efficiently.

After each iteration an attempt is made to swap the states of two of the chains using a Metropolis-Hastings step. Heuristically, swapping states between chains will confer some of the rapid mixing of the modified chains to the unmodified chain. If $\pi(x)$ is multimodal and if a modified chain moves freely among these modes, the swap will hopefully result in the unmodified chain changing modes, thus improving mixing. Proposed swaps will seldom be accepted if there is much of a gradient between chains, thus it is necessary to have many chains which differ only gradually with i.

The obvious disadvantage of MCMCMC is that while you have run m chains, only the output from one is used. Other methods with similar motivation that improve slow mixing due to multimodality are: simulated tempering, (Geyer and Thompson, 1995) and the Langevin-Hastings algorithm, (Roberts and Tweedie, 1996). Simulated tempering is closely related to MCMCMC. It uses one chain, switching distributions within that chain rather than between several parallel chains. The Langevin-Hastings algorithm proposes points based on local properties, such as the gradient of $\pi(.)$, thus the chain is encouraged to move in the directions of the local modes. A more recent example of a coupling algorithm comes from Higdon et al. (2002), where there are two chains run in parallel, one a version that uses a smaller dataset, that is a simpler version of the original, and hence it is hoped that the chain mixes better for Bayesian inference. Potential proposals for each chain are constructed from the other, allowing greater traversing of the original chain, especially in multimodal situations.

Blocking

If Gibbs or Metropolis-within-Gibbs are being used and some of the components are highly correlated (i.e. between chain correlation is high) in the stationary distribution $\phi(x)$, then the mixing tends to be slow. One way of reducing this correlation and hence improving the mixing is to update some of the highly correlated parameters in a block. This means that the values for the correlated parameters are chosen not conditional on each other.

Updating Order

Although typically the components are updated in a fixed order, this is not necessary. They could be updated in a random fashion and furthermore each component need not be updated on each iteration. The components to be updated on each iteration could be selected randomly. It has been suggested that to improve mixing, it might be appropriate to update highly correlated parameters more frequently than the others (Zeger and Karim, 1991). If the probability with which the parameters are updated is not fixed, but depends on X_t, then the accept probability has to be altered. Specifically, let $S(i)$ be the probability with which component i is updated, then

$$\alpha(x_{-i}, x_i, x_i^{'}) = \min\left(1, \frac{\pi(x_i^{'}|x_i)S(x_i|x_i^{'}, x_{-i})q(x_i|x_i^{'}, x_{-i})}{\pi(x_i|x_i^{'})S(x_i^{'}|x_i, x_{-i})q(x_i^{'}|x_i, x_{-i})}\right)$$

Given the current values of x_i, the probability with which a component is updated may change from iteration to iteration, i.e. $S(i|.)$. Random updating also has some good theoretical properties in terms of convergence. More recently, optimal ordering for efficiency and convergence of MCMC algorithms has been investigated by Mira (2001).

All of the above ideas for improving mixing are those most widely used. There are many adaptations of these and research in this area is still active, see Green (1995), Kendal (1997), Rue (2001) and Higdon et al. (2002) for recent examples.

3.5 Diagnostics and Measures of Efficiency

Diagnostics are the methods employed to monitor the appropriateness of a chosen analytical approach, given the output produced by the approach. Again, there are a number of problems associated with MCMC, particularly assessment of convergence and the number of samples needed beyond this to gain a reliable estimate of the target distribution and summaries there of. However, our interest lies in assessing the mixing and detecting structures in the data which may be causing poor mixing, i.e.

- High correlation between the components of the target distribution;
- Multimodality.

Some of the approaches to assess mixing and detecting its associated causes are outlined below. For a more formal development and extensive review of basic MCMC diagnostic and ideas for variance reduction, the reader is refered to Neal (1993) and Ripley (1987).

3.5.1 Basic Diagnostics

The most basic diagnostic approach to take when running an MCMC chain is to monitor the output visually. This can be done via trace plots (sequence of x^t against t) or even just numbers to the screen (to confirm that none of the components are stuck at one value). The latter three trace plots seen in Figure 3.2 illustrate some of the more likely problems to be encountered when examining the output. In high dimensional cases, it is not possible to do this for all of the components, so one chooses any fixed effect type parameters and perhaps a few random effects - say locations on a map or individuals in a population.

The next step is to look at histograms for the parameters of interest. If for example one of the parameters had a very skewed distribution, it may need to be transformed in some way. If the user is satisfied to move forward at this point, one would look at the autocorrelation (acf) plots of the parameters being monitored.

Autocorrelation Function An autocorrelation plot consists of a plot of values for $\hat{\rho}(s)$ versus s, for $s = 1, 2, \ldots$, where $\rho(s)$ is correlation coefficient at lag s. The autocorrelation function is given as:

$$\rho(s) \quad = \quad \frac{\varphi(s)}{\varphi(0)}$$

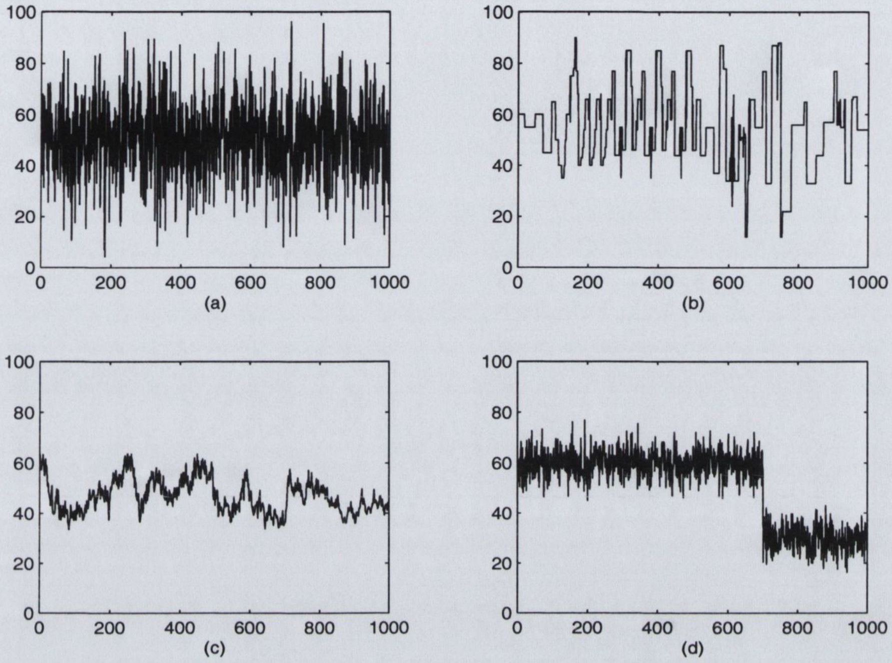


Figure 3.2: Plot(a) shows an ideal trace plot, i.e. one which appears to be mixing well. Plot(b) has many moves rejected, it exhibits poor mixing qualities and the movement of the chain is stunted and may be missing entire parts of the distribution. Plot(c) shows a chain which is traversing the distribution too slowly. Plot(d) presents a chain that has not converged and a model which may be over-parameterized, possibly with an identifiability problem or the plot may be indicative of multimodality.

and the autocovariance $\varphi(s)$ is

$$\varphi(s) = E\left[(x^{(t)} - \bar{x})(x^{(t+s)} - \bar{x})\right].$$

A popular summary measure related to this plot is lag 1 autocorrelation for each parameter. If the acf plot is not reducing "rapidly" or the lag 1 autocorrelation is high, this is indicative of strong within-chain correlation. By within-chain correlation is meant that for a particular parameter in the model, its chain exhibits correlation from one move to the next. If there is high within-chain correlation, then this is often associated with high between-chain correlation. High between chain correlation is when the behaviour of one component (or parameter) in the chain is influenced by the behaviour of another component, for example if a large value is taken by one of the parameters, then it may be followed by a small value being accepted by another parameter. High between-chain correlation would need to be reviewed using some cross-correlation plots or even just scatter plots for any parameters suspected of such. Reparameterization or blocking may need to be implemented to overcome this.

Kernel Density Estimate In the Bayesian framework a kernel density estimate is usually applied to estimate a posterior density for a parameter of interest, from samples from the posterior. More generally, density estimation entails the construction of an estimate of the density function from a set of observed data points, assumed to be a sample from an unknown probability density function. Formally, if $X = (X_1, \ldots, X_n) \in \mathbb{R}$ is a sample, then the density estimate is given as:

$$\hat{f}(x) = \frac{1}{nh}\sum_{s=1}^{n} K(\frac{x - x_s}{h}),$$

where h is a smoothing parameter (also refered to as a bandwidth or window width) and K is called the kernel function. The kernel function will satisfy:

$$\int_{-\infty}^{\infty} K(x)\, dx = 1.$$

Usually K will be a symmetric probability density function, such as the normal density, but this is not necessary. The normal kernel function is given as:

$$K(x) = \frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2).$$

The kernel density estimate could be likened to the histogram, the kernel estimate is the sum of a series of densities placed at each of the observations, where a histogram is the sum of observations between fixed points and put in a series of boxes. The estimate $\hat{f}$ is of course a density, since the kernel is a probability density function. Choice of kernel and size of smoothing parameter are discussed in Silverman (1986). Kernel density estimation is a standard method of examining posterior densities of parameters and functions of the parameters.

It has to be admitted that most users do not use anything more sophisticated than the above procedures, when assessing convergence and mixing of the MCMC algorithm. However, more sophisticated diagnostics are available.

3.5.2 Reviews of Specialized Diagnostics

There is a wealth of specialized MCMC diagnostics available. It is noted at this point that most of the available diagnostics investigate the rate at which the chain converges or whether the chain has reached convergence. There are also a number of methods that are not tailored towards convergence but attempt to measure the performance of the sampler. These methods can be used in themselves or in conjunction with convergence oriented diagnostics.

There have been a number of comparative reviews of MCMC diagnostics — Cowles and Carlin (1996), Brooks and Roberts (1998) and Mengersen et al. (1999) — each with slightly different emphasis. Cowles and Carlin (1996) compare the performance of several convergence diagnostics in an applied setting, also giving guidance on implementation.

They conclude that many MCMC diagnostics proposed in the statistical literature are fairly difficult to use, often requiring problem-specific coding and perhaps analytical work. Also, although many of the diagnostics often succeed at detecting the flaws they are designed to identify, they can also fail in this role, even in idealized cases. Hence, Cowles and Carlin (1996) advocate the use of a variety of diagnostic tools rather than any single plot or statistic. Brooks and Roberts (1998) offer a similar review and conclusions, focusing on the mathematical characteristics of the various approaches.

3.5.3 Measures for Mixing Performance

Given these comparative reviews and our particular field of interest, i.e. mixing, this greatly reduces the number of diagnostics which would be of practical interest. The two we consider are the Gelfand and Rubin statistic (Gelfand and Rubin, 1992) and Yu and Mykland's cusum path plots (Yu and Mykland, 1998), with a supplemental quantitative statistic based on the cusum by Brooks (1998). Both of these approaches effectively measure mixing of the chains.

Gelfand and Rubin Statistic

The Gelfand and Rubin statistic is one of the most popularly used in MCMC diagnostics. The statistic is a measure of how well the target distribution has been traversed, i.e. when the statistic has a value close to 1, the target distribution has been fully explored. The emphasis of the approach is in detecting slow mixing and hence reducing bias in the estimation. The method is as follows:

- Run a small number (m) of parallel chains with different starting points. These chains must be initialised in an overdispersed fashion with respect to the true posterior. Run the chains for $2n$ iterations.

- The "shrink factor" for the parameter of interest is:

$$\sqrt{\hat{R}} = \sqrt{\left(\frac{n-1}{n} + \frac{m+1}{m}\frac{B}{W}\right)\left(\frac{df}{df-2}\right)}$$

  where B is the variance between the means from the m parallel chains, W is the average of the m within-chain variances and df is the degrees of freedom of the approximate t density to the posterior distribution. The authors suggest estimating the degrees of freedom by the method of moments,

$$\mathrm{df} = 2\frac{\hat{V}^2}{\hat{\mathrm{var}}(V)}$$

  where $\hat{V} = (\frac{n-1}{n} + \frac{m+1}{mn}B)$, but many similar competing estimates for the degrees of freedom can be devised.

- The authors show that for a Markov chain with stationary distribution, $\sqrt{R} \to 1$ as $n \to \infty$. The difference from 1 for the $\hat{R}$ of the chain of interest is then a measure of convergence.

- This value should be close to 1 for all the parameters. Slow mixing samplers will initially have much larger B than W. This is because the chain starting points are overdispersed relative to the target density.

- Values close to 1 show good convergence. A large GR statistics may arise as a result of slow mixing or multimodality.

The obvious disadvantage of this method is the requirement to find overdispersed distributions to start with, in order to account for the possibility of multiple modes. However the method does not detect the existence of unexplored modes in the target distribution.

Cusum Path Plots

A quite simple and potentially useful method for measuring the quality of the mixing of the chain was proposed by Yu and Mykland (1998). It is a graphical procedure based on cusum path plots, applied to a univariate summary statistic or a single component of the chain. They suggest that the speed with which the chain mixes is indicated by the smoothness of the plot. It takes the following form:

- Take n iterations (after burn-in) of the parameter of interest, say θ.

- Let $\hat{\mu}$ be an estimate of the mean of θ, from $\theta^{(1)}, \ldots, \theta^{(n)}$.

- Then the observed cusum (or partial sum) is

$$\hat{S}_t = \sum_{j=1}^{t} (\theta^{(j)} - \hat{\mu}), \ t = 1, \ldots, n.$$

- The cusum path plot is a plot of $\hat{S}_t$ verses t, connecting successive points.

- Smoother plots, wandering further away from zero indicate slower mixing chains, while jagged plots which cross back and forth about zero regularly, indicate fast mixing chains.

- Yu and Mykland suggest comparing the plots to a benchmark plot, got from an *iid* variate generated from a normal distribution with its mean and variance matched to those of the sample iterates, to reduce the subjectivity of the method.

Brooks (1998) proposes an additional measure that could be implemented in conjunction with these cusum plots, which allows for a more objective interpretation of the plots. It involves ascribing some formal measure to the terms used to describe the plots, i.e. "smooth" and "hairy". Simply, a completely smooth plot would remain travelling in the same direction, while a completely hairy plot will alternate direction. Thus an index is created by counting the number of times the cusum plot changes its direction.

Define

$$d_t = \begin{cases} 1, & \text{if } S_t < \min(S_{t-1}, S_{t+1}), \\ & \text{or } S_t > \max(S_{t-1}, S_{t+1}), \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$D_n = \frac{1}{n}\sum_{2}^{n-1} d_t.$$

D_n takes values between 0 and 1, 0 indicating total smoothness and exceptionally poor mixing and 1 indicating maximum hairiness and much movement of the in the chain, but not necessarily rapid mixing. When n is large, D_n should lie within the bounds:

$$\frac{1}{2} \pm z_{\frac{\alpha}{2}}\sqrt{\frac{1}{4n}}$$

(by the Law of Large Numbers) for the behaviour of the plot to be considered reasonable. The above two approaches are the ones that we consider most useful, beyond those that are normally used.

There are however a number of statistics which approach the diagnosis from a rigorous mathematical perspective. The convergence rate of an MCMC algorithm is dictated by how close to 1 the absolute value of the second largest eigenvalue of its transition kernel (or transition matrix) is. Roberts (1992) and Garren and Smith (2000) propose estimating this eigenvalue from the sample output and thus comment on the rate of convergence. In practice, it is usually too difficult to obtain a useful estimate of this value (Roberts, 1995).

Effective Sample Size

Another way of thinking about the property of mixing is in terms of the way in which it affects the number of samples required to represent the distribution of interest. The number of samples required depends on the level of error in the Monte Carlo estimate obtained from the data generated by the MCMC method. Assessing this error leads to quantifying the quality of the MCMC sample data. To assess this, recall the basic Monte

Carlo estimation formula:

$$\begin{aligned} E(f(X)) &= \int f(x)\pi(x)\mathrm{d}x \\ &\approx \frac{1}{N}\sum_{t=1}^{N} f(x^{(t)}) = \bar{F} \end{aligned}$$

where the $x^{(t)}$ are samples from $\pi(x)$. The above average is an unbiased estimate of $f(X)$. If the $x^{(t)}$ are independent, the Law of Large Numbers guarantees that the average converges almost surely to $E(f(X))$, as N increases. A Markov chain simulation at its stationary distribution produces a series of dependent values;

$$x^{(1)}, x^{(2)}, \dots, x^{(N)},$$

but $\bar{f}$ will still be an unbiased estimate of $E(f(x))$, but because of the dependence of these values,

$$Var(\bar{f}) \neq \frac{\sigma^2}{N}$$

To account for this dependence, we need to calculate how much it affects the variance and hence the number of samples we require to equate with a corresponding number of iid samples. The samples which are generated can be thought of as having a smaller relative value to iid samples. This relative measure is particularly useful when comparing MCMC methods. The theory for quantifying this relative measure (or error assessment) as outlined by Ripley (1987) and Neal (1993) is as follows. An estimate for $\sigma^2 = Var(f(X))$ is given by;

$$\sigma^2 \approx \frac{1}{N-1}\sum_{t=1}^{N}(f(x^t) - \bar{f})^2.$$

On substitution, an estimate for the variance of the Monte Carlo estimate of $E(f(x))$ is then obtained,

$$Var(\bar{f}) \approx \frac{1}{N(N-1)}\sum_{t=1}^{N}(f(x^{(t)}) - \bar{f})^2.$$

The variance of $\bar{f}$ can also be expressed in terms of the autocovariance function, defined as:

$$\gamma(s) = E\left[(f(x^{(t)}) - E(x(X)))(f(x^{(t+s)}) - E(f(X)))\right]$$

and the autocorrelation function $\rho(s)$, defined as $\rho(s) = \frac{\gamma(s)}{\sigma^2}$, where $\rho(0) = \sigma^2$. The variance of $\bar{f}$ is then given by:

$$\begin{aligned} Var(\bar{f}) &= E[(\bar{f} - E(f))^2] = E\left[\left(\frac{1}{N}\sum_{t=1}^{N}(f(x^{(t)}) - \bar{f})\right)^2\right] \\ &= \frac{1}{N^2}\sum_{t,t'=1}^{N} E\left[(f(x^{(t)}) - \bar{f})(f(x^{(t')}) - \bar{f})\right] \\ &= \frac{1}{N^2}\sum_{t,t'=1}^{N} \gamma(t' - t) \\ &= \frac{1}{N}\sum_{-N<s<N} (1 - |s|/N)\,\gamma(s). \end{aligned}$$

For N large,

$$Var(\bar{f}) = \frac{1}{N}\left[\sigma^2 + 2\sum_{s=1}^{\infty}\gamma(s)\right] = \frac{\sigma^2}{N/\tau}$$

where $\tau = 1 + 2\sum \rho(s)$, and $\rho(s)$ is the autocorrelation function at lag s. The value τ can be thought of as the number of dependent sample points from the Markov chain that are required to give the equivalent to one independent sample. Another way of viewing this measurement is that the "effective sample size" (ESS) generated by the simulation is N/τ. This is going to be less than N when $\sum \rho(s) > 0$, which will generally be the case. Methods (such as those mentioned in Section 3.4.3) reduce $Var(\bar{f})$ and hence increase the ESS, thus essentially improving mixing and contributing to the efficiency of the algorithm.

The diagnostics which we have embraced are: trace plots, autocorrelation plots, kernel density estimates, and effective sample size.

Chapter 4

Blocking Algorithm for Univariate Latent Spatial Models

Although many inference methods have been proposed for spatial data (Cressie, 2001), we will concentrate on the Bayesian approach. In this chapter we look at the performance of MCMC methods for implementing Bayesian inference for a univariate Poisson model with a latent Gaussian field. The joint posterior distribution for the latent spatial Gaussian model already presented in Chapter 2, Section 2.3.5 is given by:

$$\begin{aligned} p(x,\beta,\theta|y) &\propto p(y|x,\beta)p(x|\theta)p(\theta)p(\beta) \\ &\propto \exp(\sum y_i(\beta + x_i) - \sum e^{(\beta+x_i)}) \\ & \quad |\Sigma(\theta)|^{-\frac{1}{2}} \exp(-\frac{1}{2}x^T\Sigma(\theta)^{-1}x)p(\theta)p(\beta). \end{aligned}$$

The priors for θ and β that we choose are proper uniform priors, the ranges used are detailed in Chapter 2, Section 2.3.5. The details for each of the algorithms given below will be with respect to this model. Here we present three MCMC schemes for simulating from the above posterior:

- Sequential Update, Diggle et al. (1998);

- Partial Block Update, Rue (2001);

- Total Block Update.

The first two use Metropolis-within-Gibbs (see Chapter 3, Section 3.3.2) and the last one uses a Metropolis update. From the latter sections of Chapter 3 it may be apparent that a comparison of these algorithms with regard to the efficiency with which they explore the target distribution is of interest. In what follows, we use the powered exponential correlation function as given in Chapter 2, Section 2.3.5, Equation 2.1.

4.1 MCMC Algorithms for the Univariate Latent Spatial Model

4.1.1 Sequential Update

The sequential update, as proposed by Diggle et al. (1998), samples each of the parameters individually. The α, δ, σ^2 and β are sampled from their full conditionals, using random walk proposals. A random walk proposal is one which proposes a new point dependent on the previous point, with equal probability of being larger or smaller than the previous point, i.e. $q(x, x^{'}) = p(|x - x^{'}|)$. This proposal may be uniform in an interval centred at the current point, normal with mean as the current point or any other symmetric distribution. It may also be either an additive or multiplicative function of the current value; multiplicative moves are achieved by working on a log scale. The amount by which the proposed point differs from the previous point depends on a scale parameter, which is carefully chosen, as discussed in Section 3.4.3. The X_i's are also sampled from their full conditionals using a conditional univariate Gaussian proposal. Details of the proposal and the acceptance probabilities are given in Appendix A, Section A.1.1. The computational efficiency in calculation of the conditional univariate Gaussians is improved by using Cholesky decomposition in combination with partitioning properties of the conditional-variance, (Whittaker, 1990).

We note that an alternative sampling approach that could be used is to define $Z_i = X_i + \beta$, then proceed to sample β and θ given Z_i. This approach may be applied to all the updating schemes presented and makes the sampling of β simpler, see Gammerman (1997). We do not adopt this approach here, because this does not generalise straightforwardly to the multivariate case of Chapter 5, i.e. $Y_{ij} \sim \text{Poisson}(\exp(\beta_{j0} + \sum_{t=1}^{T} \beta_{jt} X_{it}))$ and we want to allow for as close as possible a comparison of the results here with those of Chapter 5.

The values of both α and σ^2 must be strictly positive. In theory, zero is an *absorbing state* of the Markov chain for each parameter, (Besag et al., 1991). At the boundaries $\delta = 0$ and $\delta = 2$ the resulting correlation matrix becomes positive semi-definite and hence singular, so δ will be constrained to take values $0 < \delta < 2$.

4.1.2 Partial Block Update

In the partial block update, the set of parameters is partitioned into three blocks x, θ and β, where $\theta = (\alpha, \delta, \sigma^2)$. All the parameters in each block are updated jointly. The parameters within each block are strongly correlated, hence the choice of blocks. There can also be problems with mixing as a result of correlation between the hyperparameters and the latent variable X, to lessen this between chain correlation a complete block update is applied, as describes in the next section. The partial block MCMC scheme takes the form below:

- Update θ (the parameters in Σ) from its full conditional,

$$p(\theta|x) \propto |\Sigma(\theta)|^{-\frac{1}{2}}.\exp(-\frac{1}{2}x^T\Sigma(\theta)^{-1}x),$$

  using three independent random walk proposals, i.e. α, σ^2 and δ are updated simultaneously.

- Update β, from its full-conditional:

$$p(\beta|y,x) \propto \prod \exp(-e^{\beta+x_i}).\exp(\beta+x_i)^{y_i}$$

  as for the sequential method of the previous section.

- Update $X = \{X_1, X_2, \dots, X_n\}$ in one block, from its full conditional:

$$p(x|y,\beta,\theta) \propto \prod \exp(-e^{\beta+x_i} + y_i x_i).\exp(-\frac{1}{2}x^T\Sigma(\theta)^{-1}x)$$

  using the multivariate Gaussian proposal distribution:

$$p(x^{'}|\theta,\beta,y) \propto \exp(-\frac{1}{2}x^{'T}(\Sigma(\theta)^{-1}+C)x^{'} + (y-B)^T x^{'}).$$

  This distribution arises as a result of making an approximation (Rue, 2001) to the joint posterior at the current point X_i. The approximation takes the form:

  - $C = e^{\beta}\text{diag}(C_1, \dots, C_n)$;
  - $B = e^{\beta}(B_1, \dots, B_n)^t$.

  where B_i and C_i are obtained from the approximation

$$\exp(x_i^{'}) \approx A_i + B_i x_i^{'} + \frac{1}{2}C_i x_i^{'2},$$

By replacing $\exp(x)$ in the full conditional of x with this quadratic approximation, we produce the multivariate normal distribution given above. Samples for x are drawn from this Gaussian proposal distribution and thus the block update of the x_i's becomes very natural. The approximation could be made using a Taylor expansion, but Rue (2001) suggests a more global and also accurate approach to improve the accept probabilities. He proposes that an "overall" good fit to the full conditional for X in the region where $X^{'}$ is expected to be located is more important than a precise fit around X. The approximation is given by:

$$(A_i, B_i, C_i) = \text{arg min}\left[\int_{x_i-\Delta}^{x_i+\Delta} \{\exp(x_i^*) - (A_i + B_i x_i^* + \frac{1}{2}C_i x_i^{*2})\}^2 \, \mathrm{d}x_i^*\right]$$

where Δ is a crude guess of the step length of x_i to $x_i^{'}$. Full details of the calculations are given in Appendix A, Section A.1.2.

4.1.3 Total Block Update

This is a Metropolis-Hastings algorithm that uses Rue's approximation to the X_i's proposal function. The total block update algorithm works much the same as that above, except everything is updated concurrently. As before, propose new $(\beta^{'}, \alpha^{'}, \sigma^{'2}, \delta^{'})$ using some random walk proposals. Then using the proposed $\theta^{'}$ and $\beta^{'}$, propose a new vector of $x_i^{'}$ using Rue's method, such that:

$$p(x^{'}|\theta, \beta, y) \propto \exp(-\frac{1}{2}x^{'T}(\Sigma(\theta)^{'-1} + C)x^{'} + (y - B)^T x^{'}),$$

where

- $C = e^{\beta^{'}} diag(C_1, \ldots, C_n)$;
- $B = e^{\beta^{'}}(B_1, \ldots, B_n)^t$,

as defined before. The $(\theta^{'}, \beta^{'}, x^{'})$ are accepted or rejected all together; see Appendix A, Section A.1.3 for details of the acceptance probability.

4.2 Mixing Properties of Algorithm

There are two issues that have arisen in the previous sections:

- the effect of blocking on mixing;

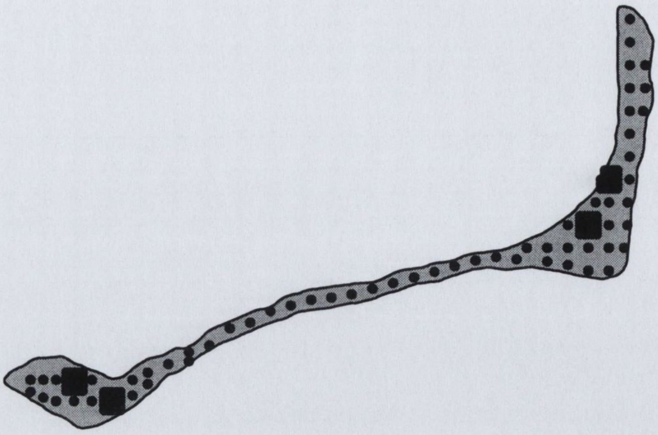


Figure 4.1: A map of Rongelap island and the locations of the measurements taken.

- the effectiveness of Rue's approximation (as a proposal for X) as a method to improve acceptance probabilities within a block update and hence improve the overall mixing.

"Blocking" has been shown to be a useful approach in aiding MCMC convergence by Liu (1994) and Liu et al. (1994), but they have also given counterexamples. The performance of Rue's approximation (Rue, 2001) combined with the above two blocking schemes, relative to the basic sequential update method (Diggle et al., 1998) is viewed here using a number of datasets, each with different attributes. The chosen attributes are those characteristics of the data which may affect the performance of the algorithms, namely:

- correlation level within the data;

- size of β.

The schemes are applied to a popular historical set of spatial data, the "Rongelap Island" data and three simulated datasets, which exhibit varying levels of these properties. The performance is viewed using a number of visual representations and effective sample size. Effective sample size as described in Section 3.5.3 can be thought of as a response variable.

4.3 Description of Data

The Rongelap island dataset consists of radioactive counts taken from 157 locations on Rongelap island. The measures are taken at point locations, for a stated period of time. An illustration of the island and the location at which the measurement were taken are given in Figure 4.1. The study arose as a result of the island being evacuated, due to experiencing contamination from the fall-out from the Bikini Atoll nuclear weapons testing programme during the 1950's. The levels of contamination have since been under investigation to establish risks involved for possible re-habitation.

The three simulated datasets comprise of 200 locations that are randomly positioned on a square. The maximum distance between points is taken to be 1. Two of the datasets have points that are very strongly correlated, the first with a mean of e^3 and the second with mean 1. The third dataset has much less correlation. This is quite close to being iid and has a mean value of e^3. The exact values of the parameters used to simulate the data are given in Table 4.1. The behaviour of the three schemes on the four datasets has been monitored and the findings outlined below.

4.4 Results

All the algorithms have been programmed in C and run on Unix machines. In the case of simulated datasets 2 and 4 the algorithms were initialised to their true values and run for $100,000$ iterations. For dataset 3, two further replicate datasets (i.e. with the same parameterizations) were created and the algorithms run on each of them to check for posterior bias. Also, the runs on dataset 3 and its replicates used various initial values for each of the parameters and were run for $100,000$ iterations post burn-in. Similarly, the Rongelap island dataset was run for $100,000$ iterations after burn-in. For each of the algorithms and datasets, the samples from the runs have been thinned to include every

100^{th} iteration, so our actual sample size is $1,000$.

The speed with which the blocking algorithms produce an iteration is much faster than that of the sequential algorithm, it is of the order of 4 times faster for the size of datasets considered here. The algorithms are order $O(n^3)$; the Cholesky decomposition is the most expensive operation. The algorithms were run on a variety of UNIX machines and the actual length of the run time for each of the algorithms varied and depended on what other processes were being run on the processor. Also, the total block update and the partial block update show substantially better mixing properties then the sequential update. Using effective sample size as a measure of mixing efficiency, the improvement in effective sample obtained is between 20 and 1.2 times that achieved by the sequential, depending on the dataset and parameters viewed.

As a separate experiment, we have run the partial block update algorithm on a number of datasets of varying size to assess how increasing dataset size affects run time. The algorithm was run on a Intel Pentium 4 with 3GHz CPU and 1GB RAM. The results are as follows: $n = 50$ took 0.005 sec/iter, $n = 100$ took 0.018 sec/iter, $n = 200$ took 0.09 sec/iter, $n = 400$ took 0.85 sec/iter and $n = 800$ took 31.4 sec/iter. From this we observe that the relationship between sample size and iterations per seconds is not a linear one. As sample size increases the speed of the algorithm decreases disproportionately and so there may be some limitations with regard to the sample size for which this type of algorithm maybe considered suitable. There have been some ideas put forward for dealing with larger datasets such as those found in geostatistics, see Rue and Tjelmeland (2002) and Whiley and Wilson (2004). Rue and Tjelmeland (2002) advocates the discretisation of the values of the unknown parameters in the covariance structure and approximate the Gaussian field by a Gaussian Markov random field while Whiley and Wilson (2004) suggests a variety of parallelising routines for computations on the covariance matrix.

Acceptance Rates For the sequential update each of the parameters are updated individually and so each has an acceptance rate. The acceptance rates for the sequential updates are as follows: the X_i's have an average acceptance rate of $0.16 - 0.31$; β is between $0.12 - 0.17$; α is between $0.06 - 0.12$; δ is between $0.02 - 0.08$ and σ^2 is between $0.06 - 0.11$. These acceptance rates are lower than may be considered desireable (see Gelman et al. (1995b)), and could be improved by reducing the variance of the random walks used. A number of scaling for the random walk were tried for this scheme, some with more favourable acceptance rates but all of which showed a slow traversing of the chain. The partial block update proposes updates of all the X_i's together, next the β and then the remaining θ's. The acceptance rates for the partial block updates are as follows: the X_i's are between $0.55 - 0.73$; the β is between $0.14 - 0.28$ and the θ's are between $0.09 - 23$. The total block update simply updates all of the parameters together and the acceptance rate for this is between $0.22 - 0.47$. The acceptance rates for the sequential algorithm are a little low but are within a reasonable range. The acceptance rates are a little more favourable for the blocking algorithms.

4.4.1 Rongelap Island Data

In the output from the sequential update as seen in Figure 4.2, the trace plots show strong evidence of correlation between the parameters and quite slow mixing of the chain. This is especially the case for α and σ^2, i.e. if α takes a very low value close to 1, then σ^2 will increase in value to compensate. It is the presence of this correlation between the parameters that is causing such poor mixing within the sequential update. There is also quite high correlation present within the chains, as seen in the autocorrelation plots, which further indicates the poor quality of mixing within the chains.

The two block updating schemes give very similar results for this dataset. The trace plots show the chains to be mixing very well, with no apparent between chain correlation present, i.e. the blocking has removed this undesirable property. The within chain correlation has more or less died away by the 10^{th} effective sample, with the exception of the σ^2 parameter in the partial block update. On inspection of a plot of σ^2 kernel density (see Figure 4.6), the density estimates are very similar for each of the parameters, with the exception of the α's. The sequential method in particular gives a slightly different density estimate for α. This reflects the evidence produced by the other diagnostics. Lastly, the effective sample size gives very conclusive evidence that the blocking scheme improves mixing many fold, see Table 4.3. The estimates found for parameters are not the same as those given by Diggle et al. (1998), but are consistent with those obtained by colleagues, Whiley and Wilson (2004).

4.4.2 Dataset 2

This dataset has quite high spatial correlation, see Table 4.1. The two blocking schemes show worse mixing, compared with the previous dataset (particularly obvious in the ess) and there is also evidence of some between-chain correlation. The acf plots in Figure 4.3 show that the within-chain correlation is quite high, particularly for α and σ^2, which is why the mixing of the chains is suffering so much. The effective sample sizes associated with the total block update are substantially better than those for the sequential and the partial update. The effective sample size for the partial and sequential updates vary with respect to which parameters behave best within each scheme.

The sequential update struggles a great deal with these data. After an extensive number of runs of the algorithm, it was observed to regularly get "stuck" and only sometimes returns to exploring the posterior distribution at all. From the trace plot shown, it can also be seen that there is high correlation between the α and σ^2 parameters. The acf plots are indicative of the extremely high correlation present within the parameters, again particularly within α and σ^2, hence the exceptionally poor mixing. This dataset illustrates that although blocking is capable of greatly reducing the effects of correlation, it is not immune to its effects in extreme cases.

Parameters	Datasets		
	2	3	4
β	3.0	0.0	3.0
α	10.0	10.0	20.0
δ	0.8	0.8	0.8
σ^2	0.75	0.75	0.75

Table 4.1: Parameter values used in the construction of the three simulated datasets.

4.4.3 Dataset 3

This dataset has the same high level of correlation as dataset 2, but now the mean count is 1. The sequential update suffers from much the same difficulties as it did for the previous dataset, due to extremely high correlation present in the data. High correlation is no longer the main issue for the blocking schemes. The performance of the blocking schemes relies on the accuracy of Rue's normal approximation to the Poisson likelihood. This approximation worsens as the Poisson mean gets small (where the Poisson distribution is more skewed) and the number of points increase. So, with a Poisson mean of 1, Rue's approximation breaks down and the proposals are never accepted. This obstacle may be overcome in a number of ways: an alternative proposal function could be used – Rue et al. (2004) has put forward some ideas on this – the mean of the Poisson could be artificially increased to allow the current proposal for the X's to be usable or a sampling method such as that described by Gammerman (1997) could be applied. Rue et al. (2004) uses a sequential representation of the latent variables, then constructs univariate approximations at each location and joins them together to sample from the posterior. The univariate approximations are made by using log-quadratic splines, but the authors also suggest some other approaches for getting the univariate approximations. This approach appears to be very efficient and accurate when the posterior density is unimodal and less accurate when it is not. We have chosen to increase the Poisson mean and thus continue with the current blocking proposals.

Using an inflated Poisson mean ($\beta = log(10)$) the two blocking schemes traverse the support of the posterior distribution, but do intermittently get stuck, as is seen in the trace plots of Figures 4.4 and 4.7. Visually the mixing would not appear to be as good as that seen for the Rongelap data or dataset 4; the autocorrelation function in Figure 4.4 and the effective sample sizes given in Table 4.3 reflect the less than perfect mixing for both of the blocking schemes. The total block shows better ess values than either the sequential or the partial block schemes. The sequential scheme showed quite poor mixing in the trace plots of Figures 4.4 and 4.7, this is reiterated in the associated acf plots and ess values. The performance of each of the algorithms is similar to that for dataset 2, which had the same quite high level of spatial correlation and hence much the same reasoning applies as to why each of the schemes perform poorly or at less well.

Posterior bias Two replicate datasets have been produced with the same properties as those of dataset 3. Each of the algorithms have been run on these replicate datasets to check for posterior bias. There would appear to be some posterior bias when using the sequential update, particularly in the alpha parameter estimate, but to some extent in all of the parameters, see Table 4.2 and Figure 4.7. The partial block algorithm exhibit a small to negligible amount of bias and this appears to disappear in the total block update, see Table 4.2 and Figure 4.7. Knorr-Held and Rue (2002) show similar findings using a variety of blocking schemes. They claim that estimates based on single-site algorithms or even blocks of parameters without the hyperparameters maybe biased, even for very long runs. They also note however that such bias was not present for all of the datasets examined and thus the results may be data dependent.

Convergence To check for the convergence properties of the algorithms, each of the algorithms have been initialised from a number of different starting points, while being run on dataset 3 and its replicate datasets. Each of the algorithms appears to have succeeded in finding convergence, however the length of burn-in required varies. In the case of the sequential algorithm often convergence is slow, given that only one in every hundred iterations is kept. Given the poor mixing that is observed in Figures 4.4 and 4.7 is unsurprising. The rate of convergence is much faster for the blocking algorithms, see Figure 4.7.

4.4.4 Dataset 4

The last dataset has a log-mean of 3.0 and quite low correlation. It can be seen from the trace plots in Figure 4.5 and Table 4.3 that all three algorithms perform much better than in the previous datasets. The characteristics of this dataset (low correlation and high Poisson mean) are relatively suitable for MCMC methods.

The performance of the two blocking schemes is very similar, but the trace plot for the sequential scheme shows that its mixing is not quite as rapid. Also, there is evidence in the sequential trace that there is correlation between α and σ^2, which as mentioned previously is one of the causes of the slower mixing in this method. The acf plots support these conclusions. The within-chain correlation dies away rapidly using the block updating schemes, but not to the same extent for the sequential method. This indicates that the mixing for the blocking schemes is better than that for the sequential update. Again this is supported quantitatively by the effective sample size, which is much smaller (better) for the blocking schemes. The consistent difference between the total and partial schemes could be attributed to less good mixing in X's of the partial scheme, which itself maybe caused by poorer mixing of its α and δ parameters.

4.5 Conclusions

Our main interest in these algorithms is the efficiency with which they explore the target distribution, with the obvious constraint being the time it takes them to do so. After

Model Parameters	Algorithm	Dataset 3			
		True	Run 1 (μ, σ)	Run 2 (μ, σ)	Run 3(μ, σ)
β	Sequential	0.0	(0.03,0.19)	(0.05,0.07)	(0.13,0.17)
	Partial	log(10.0)	(2.30,0.16)	(2.26,0.23)	(2.17,0.33)
	Total	log(10.0)	(2.31,0.21)	(2.28,0.27)	(2.26,0.19)
α	Sequential	10.0	(7.62,2.95)	(7.69,3.80)	(6.84,2.42)
	Partial		(9.68,3.42)	(9.49,3.17)	(9.65,3.57)
	Total		(9.95,2.74)	(10.06,2.97)	(9.71,2.89)
δ	Sequential	0.8	(0.89,0.19)	(0.88,0.32)	(0.92,0.15)
	Partial		(0.83,0.20)	(0.84,0.17)	(0.84,0.21)
	Total		(0.82,0.15)	(0.80,0.17)	(0.82,0.18)
σ^2	Sequential	0.75	(0.70,0.25)	(0.69,0.21)	(0.57,0.24)
	Partial		(0.72,0.09)	(0.75,0.11)	(0.70,0.13)
	Total		(0.74,0.08)	(0.76,0.06)	(0.72,0.10)

Table 4.2: These are the mean and standard deviation for each of the parameters in the model, for each algorithm and for each of the dataset 3 runs. Also given is the true parameter values for the model, the closer the mean is to this value the less biased the estimate.

Parameter	Algorithm	Datasets			
		Rongelap	Dataset 2	Dataset 3	Dataset 4
β	Sequential	12.41	31.51	44.91	39.59
	Partial	2.57	42.82	31.31	6.46
	Total	4.49	19.68	22.36	5.28
α	Sequential	57.15	58.33	46.32	23.78
	Partial	5.09	48.11	45.67	2.11
	Total	2.35	37.20	33.64	1.57
δ	Sequential	49.38	24.41	55.32	13.80
	Partial	16.91	30.69	30.79	2.01
	Total	2.62	20.95	26.81	1.96
σ^2	Sequential	55.34	63.59	73.65	15.62
	Partial	8.86	52.97	52.96	2.91
	Total	2.19	32.46	34.97	1.21

Table 4.3: These are the τ values of the effective sample size (ess $= N/\tau$, where N is sample size) for the parameters of the model, for each of the three algorithms. The smaller the τ value the better the algorithm's performance.

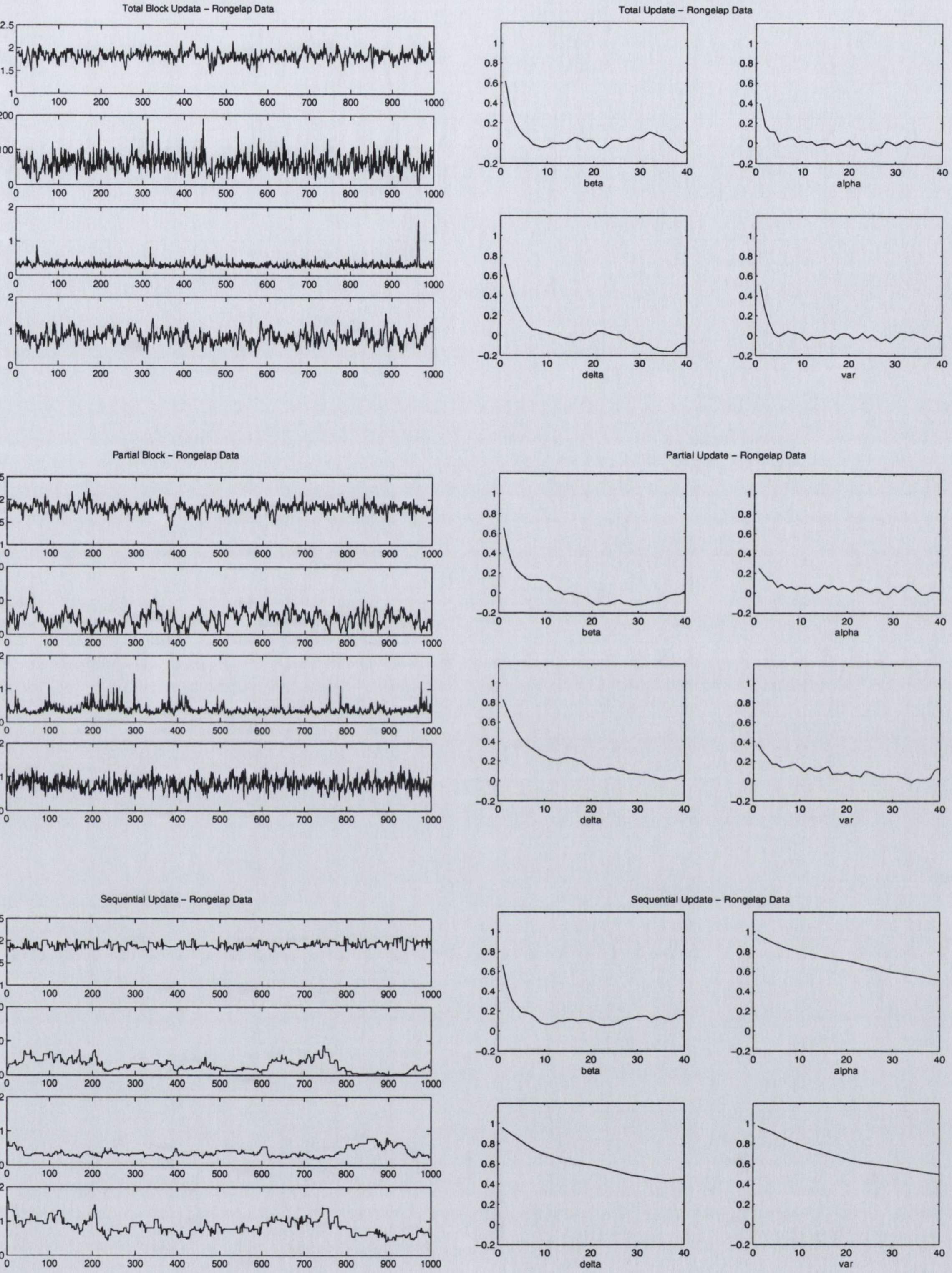


Figure 4.2: Trace plots and autocorrelation plots for the Rongelap Island data.

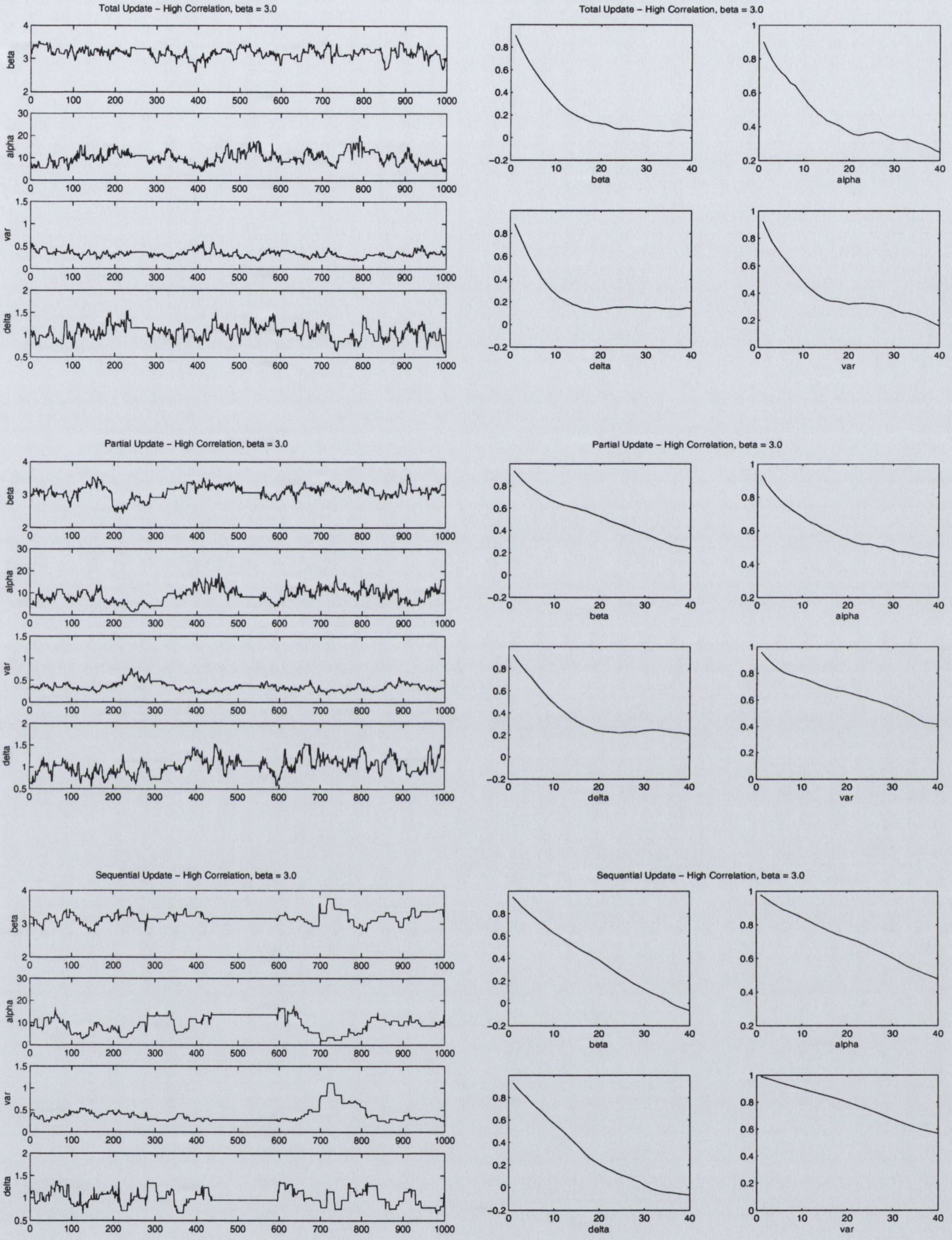


Figure 4.3: Trace plots and autocorrelation plots for the high correlation data, $\beta = 3.0$.

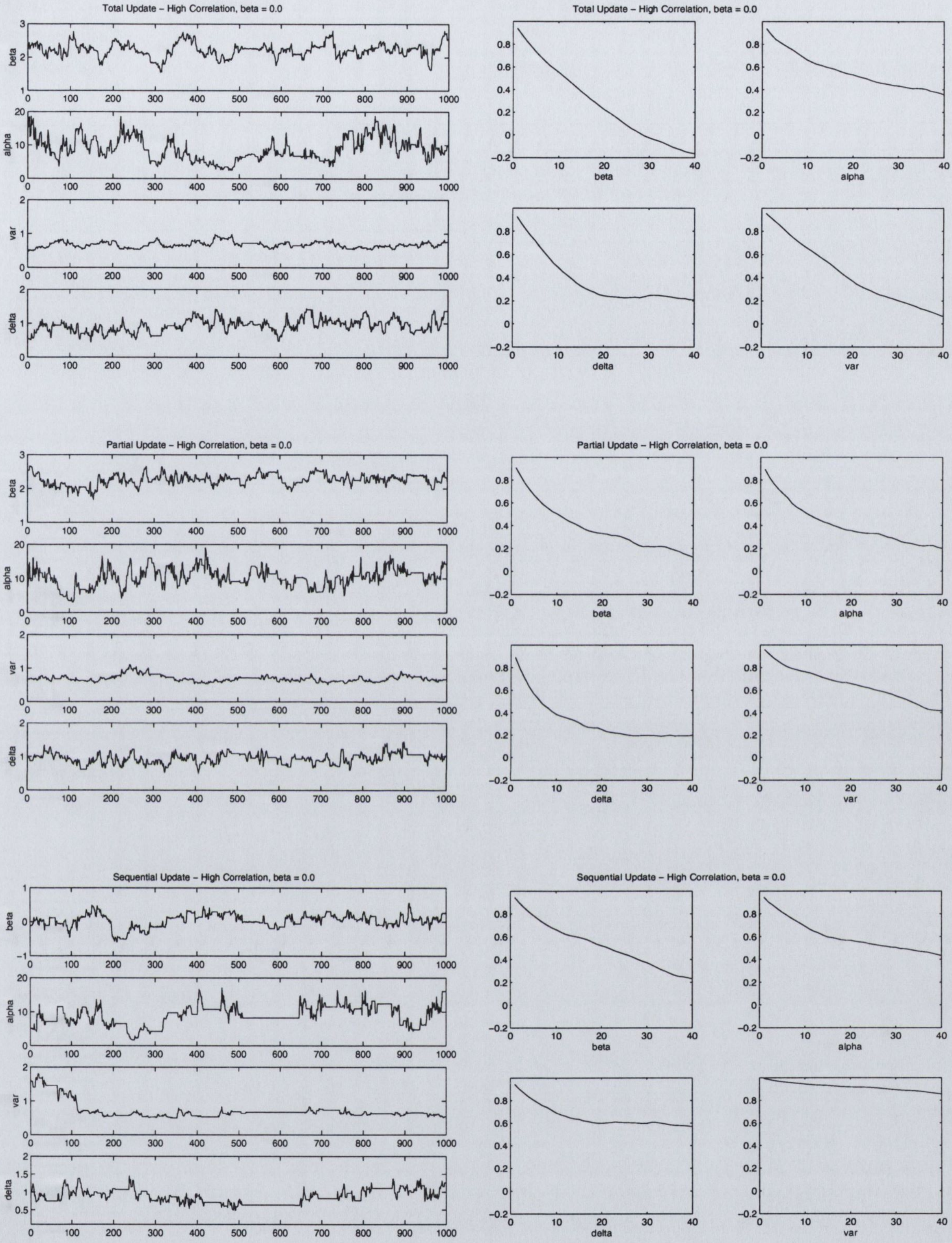


Figure 4.4: Trace plots and autocorrelation plots for the high correlation data, $\beta = 0.0$ for the sequential algorithm and $\beta = \log(10.0)$ for the blocking algorithms.

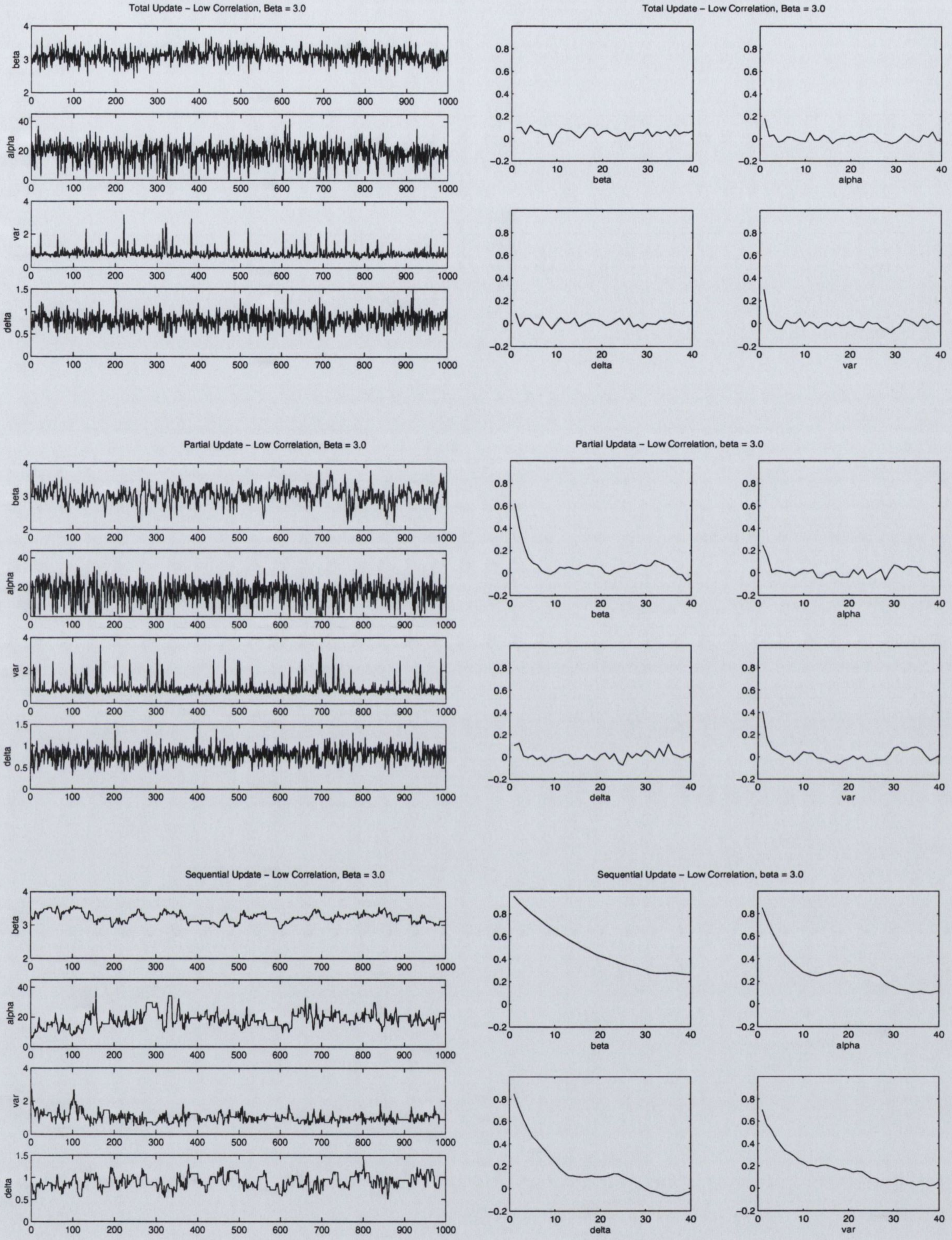


Figure 4.5: Trace plots and autocorrelation plots for the low correlation data, $\beta = 3.0$.

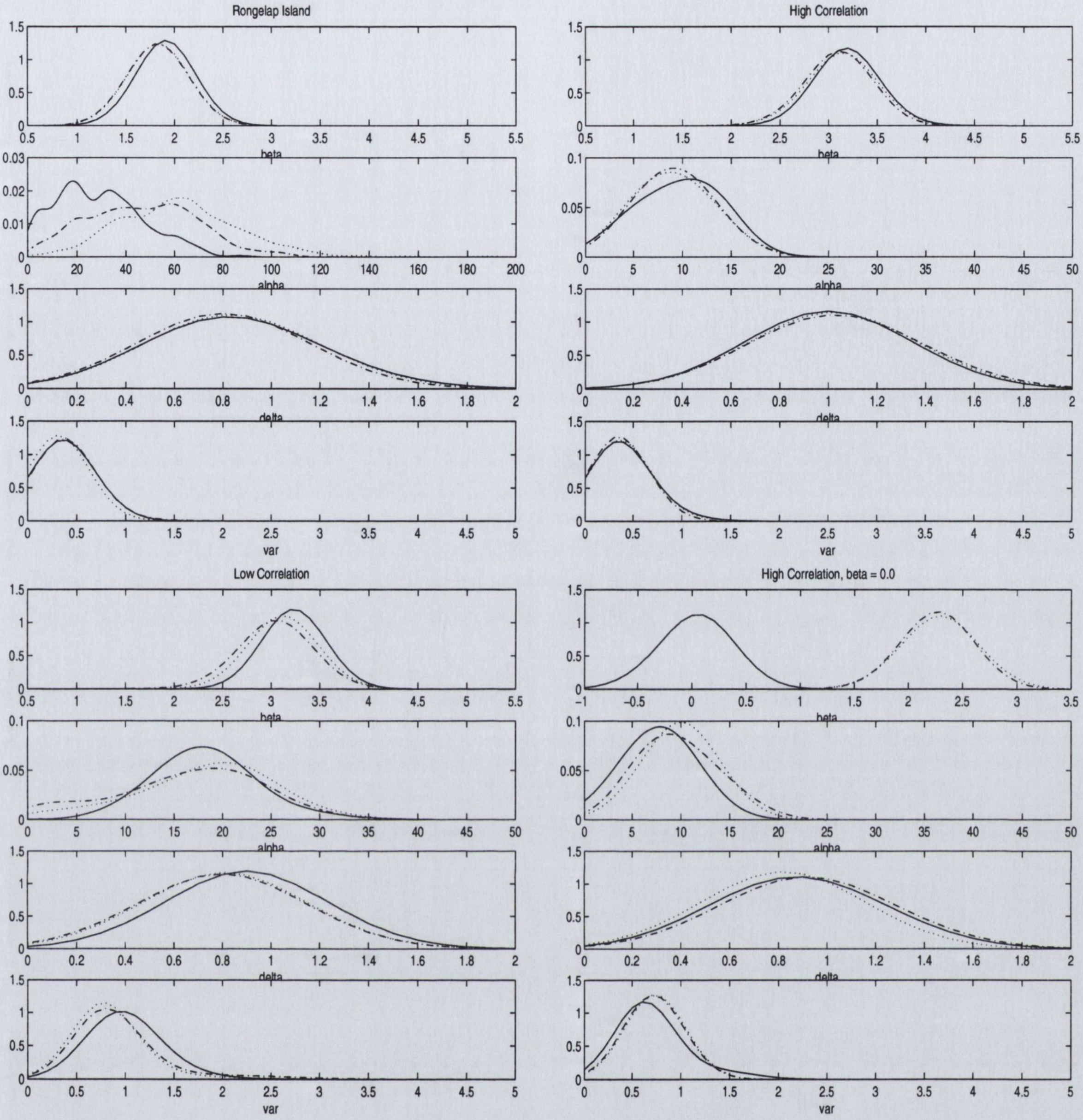


Figure 4.6: Kernel density plots for the Rongelap Island data (top left), high correlation data, $\beta = 3.0$ (top right), low correlation data (bottom left) and high correlation, $\beta = 0.0$ (bottom right). The solid line represents the sequential output, the dash-dot line the partial update output and the dotted line the total update output. The estimates of β in the bottom right dataset differ due to the use of an inflated Poisson of $\beta = \log 10.0$ for the two blocking algorithms and a $\beta = 0.0$ for the sequential algorithm. A Gaussian kernel and optimal bandwidth have been used in the construction of these kernel density plots.

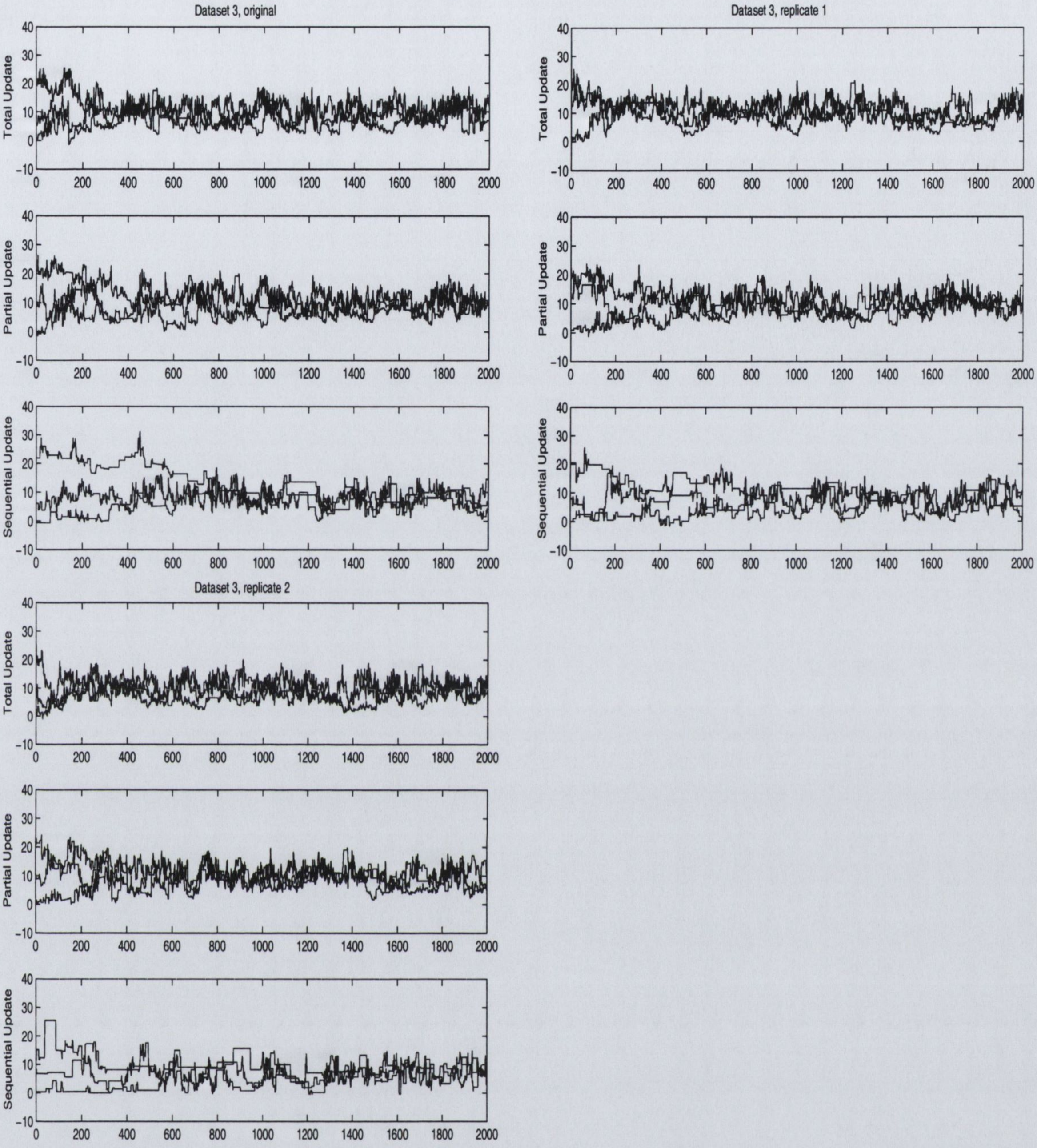


Figure 4.7: Trace plots of the α variable with various initializing values for each of the algorithms (total update, partial update and sequential). They are run on dataset3 (high correlation, $\beta = 0.0$) and two additional replicate datasets with the same parameterizations. The results for the original dataset3 are given in top left, replicate1 dataset top right and replicate2 dataset bottom left.

assessing their performances on the given datasets, the main conclusions are:

1. The speed per iteration with which the blocking algorithms run is much faster than that of the sequential algorithm. They complete an iteration in approximately a quarter of the time it take the sequential algorithm.

2. The main drawback of the blocking schemes is that the coding required is more complex and thus there is an initial investment of time. This complexity is mostly due to the details in Rue's approximation of the X_i's.

3. Mixing is generally better with blocking schemes than with a sequential scheme. In particular, mixing will improve using blocking when between-chain correlation is high, and will improve many fold when the between-chain correlation is average to low. However, blocking will not make the associated difficulties disappear entirely, as can be seen in their effective sample size measures.

4. Rue's approximation for the X_i's appears to work well (i.e. quite high acceptance probabilities were observed) under certain conditions. Rue's method requires that the Poisson distribution be well approximated by the normal, if this is not the case, other measure need to be take such as using an artificially increased Poisson mean or an alterative proposal function for the X_i's.

5. There is some bias present in the estimates attained from the sequential algorithm. This is substantially reduced using the partial block algorithm and completely eliminated with the total block algorithm. All of the algorithms succeed in converging when initialised from various starting values.

6. The blocking algorithms work well on the dataset sizes considered here, however were the dataset size to increase dramatically, there would be practical issues with regard to their speed of computation and hence the time required to obtain an adequate sample from the support of the posterior distribution.

In summary, the blocking schemes are more efficient in their traversing of the target distribution, but their coding requires an investment of time at the outset.

Chapter 5

Coupling Algorithm for Multivariate Latent Spatial Models

The model adopted in this chapter is a multivariate extension of the one utilized in Chapter 4. The main difference is that there are now r response variables (y_i) at each of the s_i locations, rather than just one, to allow correlation both within and between locations. These correlations are modelled by increasing the number of latent processes to T. While a univariate latent field will model spatial correlation, it cannot induce cross-correlations between observations at a single location. By introducing more than one latent field, cross-correlation can be modelled. This poses the question; how many latent fields T should one have? For identifiability, $T < r$. In this work we restrict ourselves to $T = 2$. For ease of computation, T should be as small as possible and yet allow sufficient fitting of the correlation structure. Usual model choice methods, such as Bayes factors, can be used to select the best value of T. A fully Bayesian treatment would allow T to be a random variable, and its value to be inferred. This could be implemented by dimension-changing

Metropolis methods, such as reversible jump (Green, 1995).

The model construction is described in detail in Chapter 2, Section 2.3.6. Using this model each latent process is given $\mathrm{var}(x_{it}) = 1$ to avoid overparameterisation. The joint posterior distribution for the model is given by:

$$\begin{aligned} p(x,\beta,\theta|y) &\propto p(y|x,\beta)p(x|\theta)p(\theta)p(\beta) \\ &\propto \prod_{j=1}^{r}\left[\prod_{i=1}^{n} e^{\lambda_{ij}y_{ij}}\, e^{-\exp(\lambda_{ij})}\right] p(\beta_j) \\ &\quad \times \prod_{t=1}^{T} |\Sigma(\theta_t)|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}x_t^T\Sigma(\theta_t)^{-1}x_t\right) p(\theta_t), \end{aligned}$$

where $y = (y_1,\ldots,y_r)$, $x = (x_1,\ldots,x_T)$, $\beta = (\beta_1,\ldots,\beta_r)$, $\theta = (\theta_1,\ldots,\theta_T)$, $\beta_j = (\beta_{j0},\ldots,\beta_{jT})$ and $\lambda_{ij} = \beta_{j0} + \sum_{t=1}^{T}(\beta_{jt}x_{it})$.

Motivation This model has been motivated by fossil pollen data that has been observed at many locations throughout Europe in lakebed sediment. The underlying concept attached to the data is that ecological behaviour is directly related to climate. Thus, if there is knowledge about the ecology of an area, as inferred from pollen counts, then inference can be made with regard to its climate. Furthermore, knowing how present day ecology (pollen counts) relates to climate, it is then possible to predict what past climate might have been like given the fossil pollen. The data comprise of pollen counts $y_j(s_i)$ for $j = 1,\ldots,r$ different species of plant at each of $i = 1,\ldots,n$ locations. These pollen samples are taken from cores of lake sediment, spanning a time period of $10,000$ years (i.e. back to the last ice age). For each location s_i, the pollen samples are taken from a sediment sequence at intervals, that are irregular in time due to variation in sediment accumulation rate. Radiocarbon dating determinations are made on samples from the sediment sequence, and interpolation is used to estimate the remainder. An important feature of the data is that the counts display spatial correlation and across species correlation at the same location. Difficulties attached to the data are that there is an indeterminate amount of both spatial and temporal aggregation. A further issue is that climate during the last glacial age would have been much more extreme, and that climatic conditions over the past $10,000$ years would have been much more variable than present day conditions. This mismatch of information makes it difficult to model past ecology, Whiley et al. (to appear) have given an approach for these data.

There are two attributes of the pollen data which have been instrumental in our approach: spatial correlation and aggregation. The model which we have chosen is based on these data but without the temporal element, hence it uses a spatial correlation structure. Also, the natural occurrence of spatial aggregation has been key in inspiring the concept of utilizing its benefits via artificially aggregating the data, as is seen by the method given later in this chapter.

5.1 Coupling

MCMC algorithms can be very slow both to converge and explore the target distribution thereafter; potential reasons for this are discussed in Chapter 3, Section 3.4. Many different methods have been proposed for the improvement of their rate of convergence and mixing, see Chapter 3, Section 3.4.3 in this text and also Gilks and Roberts (1996). The approach which we have taken in what follows of this chapter comes under the category of coupling; some details of this have already been discussed in Chapter 3, Section 3.4.3. Essentially, a coupled MCMC method runs several chains in parallel that are allowed to exchange information. At least one of the chains, but not necessarily all of them, has stationary distribution that is the target. The nature of this exchange is such that the chain or chains whose stationary distribution is the target are able to be explored quicker. We combine the ideas of coupling and blocking to create a new algorithm, whose aim is to improve the exploration of a target distribution.

Coupling algorithms have many purposes and come in many forms. That is to say, there are a number of MCMC techniques which fall under the umbrella of coupling and depending on the problem being resolved the approach will differ. Here we give a brief description of some of those which have been presented.

The first method to introduce an alternative update of MCMC was the Swendsen and Wang (1987) algorithm for the Ising model (this was not strictly coupling, but more augmentation), this lead to the introduction of many other algorithms of this type. A more general algorithm was proposed by Geyer (1991) – Metropolis-coupled MCMC (MCMCMC). Details of this algorithm have already been given in Chapter 3, Section 3.4.3. Many MCMC methods have arisen from or were derived from ideas in physics and statistical physics, particularly multi-resolution problems which are discussed at the end of this section. One example is due to Geyer and Thompson (1995) referred to as "simulated tempering", which is based on simulated annealing, an optimization technique from physics.

Both MCMCMC and tempering use a one parameter family of probability distributions $(h_i(x), i = 1, \ldots, m)$, indexed by a parameter i called temperature, ranging from the distribution of interest, which is the coldest temperature $(h_1(x))$, to the hottest distribution $(h_m(x))$, which is easy to simulate. If $h(x)$ is the unnormalised density for the distribution of interest, then $h(x)^{1/\beta}$ for $\beta > 1$ are the "heated" unnormalised densities, where β generally includes a scaling effect of the temperature. The $h_i(x)$ is known as an "energy" function and the form of movement between states is referred to as "powering up" in simulated annealing. Unlike annealing, simulated tempering does not impose a monotonically decreasing schedule of temperatures, but rather it moves in a random walk. The stationary distribution of the sampler is proportional to $h_i(x)\pi(i)$, where the $\pi(i)$ are artificial weighting terms chosen in advance. These are intended to approximately equalize the time spent at each temperature. The algorithm takes the form:

1. Update x using Metropolis-Hastings or Gibbs update for h_i.

2. Set $j = i \pm 1$, where $q(1,2) = q(m, m-1) = 1$ and $q(i, i+1) = q(i, i-1) = \frac{1}{2}, 1 < i < m$.

3. Accept transition with probability $\min(1, r)$, where

$$r = \frac{h_j(x)\pi(j)q(j,i)}{h_i(x)\pi(i)q(i,j)}.$$

Tempering has the advantage over MCMCMC that it keeps only one copy of state x rather than m copies, so the chain uses less storage and mixes better. The disadvantage is that it requires good choice of $\pi(i)$. Details of the number of distributions to use and the choice of $\pi(i)$ are given in Geyer and Thompson (1995).

Other developments have come from Basseville et al. (1992). They examine "multi-scale stationarity" and fusion of data from different resolutions, with application in signal and image processing. Frantz et al. (1990) also introduced a method of coupling, which proposed jumping (called J-walking) between low and high temperature random walks, using Boltzmann distributions to allow full exploration of a region. Their motivation is to avoid quasi-ergodicity and reduce the time required in running Metropolis algorithms. Propp and Wilson (1996) tackle the problem of identifying when the Markov chain has reached the target distribution using coupling. They present an algorithm, which uses monotone coupling from the past and samples exactly from the distribution of interest rather than approximately, which is the case with standard MCMC. The coupled chains, rather than running from the present into the future, run from a point in the past until the present. The distance into the past that the algorithm needs to go is determined during the running of the algorithm itself. The algorithm is however not particularly universal, as it relies on the Markov chain having a special structure.

More recent contributions to the area have come from: Barone et al. (2002), Pinto and Neal (2001), Holloman et al. (2002) and Higdon et al. (2002). Barone et al. (2002) present an algorithm which combines ideas from coupled Markov chain methods and from existing algorithms based on over-relaxation. Over-relaxation is a technique used to help speed the chains progress through the parameter space by ensuring that each new value in the chain is negatively correlated with its predecessor. Adler (1981) provides a standard over-relaxation method when the full conditionals are Gaussian, this approach has been generalised by many authors. Barone et al. (2002) present examples in which the proposed algorithm converges faster than the existing over-relaxation algorithm and the Gibbs sampler. They also look at the efficiency of the algorithm by viewing the asymptotic variance for various parameters. The algorithm essentially has two different two parameter families (g_y and f_x) of algorithms, whose output converges to a random vector from the product density $g_y(y)f_x(x)$. Both chains (X' and Y') are updated using a linear combination of all the current values from each of the random vectors X and Y. This approach differs from that given by Geyer (1991), where the updating of the chains is followed by swapping between chains. With respect to the asymptotic variance, they conclude that once equilibrium has been reached, passing information from X to Y is no longer an advantage.

Similarly, Pinto and Neal (2001) propose passing information between two chains, X and Y. Here, Y the chain of interest, samples from the posterior and X is a chain that samples from a Gaussian approximation of the posterior. These chains will be highly correlated and Pinto and Neal (2001) utilize this correlation to constructing a more accurate MCMC estimate for posterior expectations.

Holloman et al. (2002) extend the Genetic algorithm (a maximization technique for functions defined on multi-dimensional space) by making use of related solutions of different dimensions. Again this approach utilises the transfer of information between scales of varying resolution to obtain more accurate results.

5.2 A Coupling Algorithm for Multivariate Latent Spatial Models

Some issues with MCMC for simulating from the posterior distribution of this type of model are:

- multimodality in the posterior, causing chains to "get stuck" at one mode and not fully exploring the distribution;
- mixing;
- large scale problems tend to be computationally very demanding.

The latter issue may be aided to some extent by blocking, such as that implemented in Chapter 4, given that blocking algorithms generally run more quickly. However the improvement will not be huge since the main computational issue is that Cholesky factorization is $O(n^3)$. Blocking can also be used to combat multimodality, if the spatial locations were blocked into regions with similar attributes. In the multivariate case, this would not be a trivial task, so multimodality can pose a real problem with spatial data. A chain may even appear to have converged, when in fact it is just exploring an area around a mode. Multimodality has not been detected with the data used in this thesis.

One possible solution to both of these issues would be to reduce the size of the data set, by aggregating it. Aggregating the data or using a coarsened version of the data would allow faster runs and reduce the probability of chains being trapped in local maxima. Naturally, this approach would come at a price, the probable cost of aggregation being:

- loss of information;

- and "ecological fallacy" i.e. conclusions on relationships between variables at coarse scale not necessarily true at original scale.

One possible way in which the benefits of aggregation may be enjoyed, without the ill effects, might be to use coupled chains. Coupled MCMC chains as discussed earlier are not a new MCMC method, but combining the idea with coarse and fine-scaling of the chains is a relatively recent development, Higdon et al. (2002).

With the model described in Chapter 2, Section 2.3.6 in mind, an implementation of coupled MCMC, with coarse and fine scale chains could take the form given below. The notation will be as before, letting y be the observed count data, x represents the latent variables and θ the hyperparameters in the model. A tilde (eg $\tilde{x}$) above a variable indicating that it is aggregate data or that it is a variable associated with the coarse chain. The intended effect of the coupled MCMC algorithm is that the coarse chain mixes better; then swapping information with the fine chain allows the fine chain to mix better as well. The coupled MCMC algorithm proceeds as follows. Let C be a coarsening operation, such that

$$Cx = \tilde{x} = (\tilde{x}_1, \ldots, \tilde{x}_{\tilde{n}}),\ \tilde{n} < n.$$

We have chosen C to be a summation of counts within certain regions, see Section 5.4 for details. Then the original and the coarse chain alternate (to some degree) between each being updated with their usual MCMC proposals, to being updated letting their proposal depending on the current value(s) of the other chain.

$$\begin{tikzcd}
(x,\theta)^{(1)} \arrow[r, "MCMC"] & (x,\theta)^{(2)} & & (x,\theta)^{(3)} \arrow[r, "MCMC"] & (x,\theta)^{(4)} & \ldots & (x,\theta)^{(N)} \\
& {} \arrow[rr, "SWAP"] & & {} & & & \\
(\tilde{x},\tilde{\theta})^{(1)} \arrow[r, "MCMC"] & (\tilde{x},\tilde{\theta})^{(2)} & & (\tilde{x},\tilde{\theta})^{(3)} \arrow[r, "MCMC"] & (\tilde{x},\tilde{\theta})^{(4)} & \ldots & (\tilde{x},\tilde{\theta})^{(N)}
\end{tikzcd}$$

where N indicates the number of iterations, $\overset{MCMC}{\longrightarrow}$ represents a regular MCMC iteration and $\overset{SWAP}{\longrightarrow}$ represents a swapping of information between the two chains. The next thing to consider is the order of updating or proposal for the parameters and how to propose a swap between the chains. The order of updating the parameters is: first the latent variables, then the θ's. In the case of the swapping proposals, all the parameters in both chains are updated concurrently (i.e. $X_1, \ldots, X_T$ and the θ's), and the proposals are made in the same order as the standard MCMC updates.

5.2.1 Fine to Coarse Swapping Proposal

The swapping proposal kernel for the coarse-scale, generated from the fine is:

$$q((x, \theta) \rightarrow (\tilde{x}', \tilde{\theta}')) = I[\tilde{x}' = C(x)] \times I[\tilde{\theta}' = h(\theta)]$$

or

$$q((x, \theta) \rightarrow (\tilde{x}', \tilde{\theta}')) = I[\tilde{x}' = C(x)] \times \pi(\tilde{\theta}' | \tilde{x}', \tilde{y})$$

where $I[.]$ is the indicator function and $h()$ is some deterministic function. That is, the proposal for $\tilde{x}$ is a deterministic function of x. In this case, we use the coarsening function C on x to get a suitable proposal $\tilde{x}'$, the $\tilde{x}$ are given the same location as their associated $\tilde{y}$. The candidate value for $\tilde{\theta}'$, given the newly proposed $\tilde{x}'$, could be generated from some deterministic function of θ or could be simulated from its full conditional, under the coarse posterior distribution. When choosing a function $h(.)$ to propose $\tilde{\theta}'$, it too will reflect the degree of coarsening in the aggregate data. Depending on which of the $\tilde{\theta}'$'s is being proposed, $h(.)$ may be an empirical guess or could be aided by the use of a correlogram, see Chapter 2, Section 2.3.1. We have chosen the deterministic approach, the selected $h(\theta)$ is based on empirical observations, the value taken depends on both the specific θ parameter concerned and the coarse dataset involved. For example, for the $1-2$ coarsened dataset, the $\tilde{\alpha}' = h(\tilde{\alpha}) \simeq 6\tilde{\alpha}$.

5.2.2 Coarse to Fine Swapping Proposal

The proposal kernel that generates a fine-scale proposal from the coarse chain is more problematic, and takes the form

$$q((\tilde{x},\tilde{\theta}) \rightarrow (x^{'},\theta^{'})) = \pi(x^{'}|\theta^{\dagger},\tilde{x} = C(x^{'})).\pi(\theta^{'}|x^{'},y),$$

where $\theta^{\dagger}$ is generated using some deterministic function of $\tilde{\theta}$. Then $x^{'}$ is simulated such that $Cx^{'} = \tilde{x}$. $x^{'}$ has a distribution with $(n - \tilde{n})$ degrees of freedom. Thus, if we generate $x^{'}_{n-\tilde{n}}$ from the marginal distribution of x, the remaining $\tilde{n}$ values of $x^{'}$ can be found using the coarsened data, like so:

$$x^{'}_{-k} = \tilde{x}_k - \sum x^{'}_{k}, \text{where } k \text{ indicates a region } k,\ \ k = (1,\ldots,\tilde{n}).$$

Hence, we have an $x^{'}$ with the desired distribution. Then either simulate $\theta^{'}$ from its full conditional given $x^{'}$ or use an appropriate deterministic proposal function $h(.)$, i.e. the reciprocal of the function used in the fine to coarse proposal for $\tilde{\theta}^{'}$. We have taken $h(\theta) = c\tilde{\theta}$, for a some constant c that depends on the amount of aggregation used in the coarse chain. The c is chosen from empirically observation, for example $\alpha^{'} = \frac{1}{6}\tilde{\alpha}$. The non-swapping, MCMC updates can take any form, such as the blocking schemes described in Chapter 4.

By exchanging information between the fine-scale parameter space and the coarse-scale parameter space, this coupled chain has stationary distribution $\pi(x,\theta|y) \times \tilde{\pi}(\tilde{x},\tilde{\theta}|\tilde{y})$, i.e. all T spatial field are updated jointly. Then by taking the fine-scale realisations, we have a chain with stationary distribution $\pi(x,\theta|y)$. Higdon et. al. (2002) proposed a similar implementation to this for a Markov random field model, with univariate observations.

5.3 Investigation of Algorithms' Properties

Some of the factors which affect the algorithm's performance are:

- Data size;
- Amount of correlation present in the data;
- Levels of coarsening;
- Number of coarse chains;
- Rate of swapping between coarse and fine scales;
- Relative number of coarse and fine scale iterations;
- Choice of $h(\theta)$;
- Size of β's.

The idea is to evaluate the performance of the coupled MCMC algorithm under various conditions, such as those listed above. Concerning the above factors, there are three which we consider to be paramount in investigating the behaviour of the coupling algorithm: coarsening, correlation and rate of swap. Each of these elements and the form of the experiment as a whole, is elaborated upon below.

5.3.1 Latin Square Design

Blocking (in the sense of experimental design) in an experiment is a way to reduce residual error, by removing the variability due to a known variable. Blocking was first introduced by R.A.Fisher in the 1940's during the experiments at Rothamsted, as a method to counteract spatial variability. A Latin square is a particularly efficient block design. As the name suggests, it is a square design. A Latin square for 3 factors each with p levels (a $p \times p$ Latin square) is a square containing p rows, p columns and p^2 cells. Usually, the variables rows and columns of the square are regarded as nuisance parameters, which may otherwise lead to variability in assessing the effects of the variable of interest. The rows and columns of the square are both orthogonal to the third factor. Table 5.1 gives an example of a Latin

	Variable (a)				
Variable (b)	1	2	3	4	5
1	A	B	C	D	E
2	B	C	D	E	A
3	C	D	E	A	B
4	D	E	A	B	C
5	E	A	B	C	D

Table 5.1: This is an example of a 5x5 Latin square, but there are many other possible arrangements which could be used.

square design. Note, each of the variables needs to have the same number of levels, also, each of the letters occurs once only in each row and column.

The statistical model for a Latin square with 3 factors is:

$$y_{ijk} = \mu + \tau_i + \beta_j + \eta_k + \epsilon_{ijk}$$

where there is an underlying mean value of μ, then τ_i, β_j and η_k represent the three factors (for a three factor Latin square) at levels i, j and k respectively and ϵ_{ijk} represents the error. Using analysis of variance, an appropriate statistic to test the various hypothesis of interest is an F-test, i.e. to test for a difference in effect between any levels of the factors. One possible disadvantage of a Latin square design is it provides few degrees of freedom for error in a small square, eg in a 3×3 square, there are 2 error degrees of freedom. So replication of the square is desirable.

Given the number of factors and levels of each that we wish to investigate, the Latin square is an appropriate and efficient experimental design to employ. The factors that we have selected for investigation are those which we feel are most likely to influence the behaviour of this type of coupling approach. The factors which we have chosen are: level of coarsening applied to data, frequency of proposed swaps and amount of correlation present in the data. Although correlation level in the data is not directly related to the

	High Correlation	Medium Correlation	Low Correlation
Coarsening Level 1	swap 1	swap 2	swap 3
Coarsening Level 2	swap 2	swap 3	swap 1
Coarsening Level 3	swap 3	swap 1	swap 2

Table 5.2: The Latin Square design employed by the experiment.

process of coupling, it has proved to be an important factor in the blocking experiment of Chapter 4. It also introduces the possibility of investigating possible interaction effects between correlation levels in the data and the degree to which the data is coarsened. Each of the factors will have three levels:

- Levels of correlation will be high, medium and low;
- Coarsening factors are $1-2, 1-4$ and $1-6$;
- Swapping rates are $1-10, 1-25$ and $1-100$.

The Latin square to investigate if these factors have an effect on the outcome of our MCMC is given in Table 5.2. Our measurement of interest (or response) is *effective sample size.* This Latin square allows us to examine if any of these variables significantly affect the effective sample size. There are two further hypothesis which interest us, these being:

- Is there a relationship between correlation level and degree of coarsening?
- Is there a relationship between degree of coarsening and swapping rate?

To investigate these hypotheses, the model needs to be extended. The statistical model for this is then given as:

$$y_{ijk} = \mu + \tau_i + \beta_j + \eta_k + (\tau\beta)_{ij} + (\tau\eta)_{ik} + (\beta\eta)_{jk} + \epsilon_{ijk}$$

where $\mu, \tau_i, \beta_j, \eta_k$ and ϵ_{ijk} are as previously given. The $(\tau\beta)_{ij}, (\tau\eta)_{ik}$ and $(\beta\eta)_{jk}$ represent the interactions between the variables; coarsening-by-correlation, coarsening-by-swapping

	High Correlation	Medium Correlation	Low Correlation
Coarsening Level 1	swap 3	swap 1	swap 2
Coarsening Level 2	swap 1	swap 2	swap 3
Coarsening Level 3	swap 2	swap 3	swap 1

Table 5.3: A replicate Latin square to that of Table 5.2 to allow for examination of interactions.

	High Correlation	Medium Correlation	Low Correlation
Coarsening Level 1	swap 2	swap 3	swap 1
Coarsening Level 2	swap 3	swap 1	swap 2
Coarsening Level 3	swap 1	swap 2	swap 3

Table 5.4: A replicate Latin square to that of Table 5.2 and 5.3 to allow examination of interactions.

and correlation-by-swapping respectively. To investigate this model, some further replication in the experiment is required. The replication needs to be chosen carefully, so as to allow the interactions to be examined. The specifics of the replication for the additional two replicate Latin squares is given in Tables 5.3 and 5.4.

5.4 Description of Data

To investigate the factors mentioned above, we have three datasets, which are simulated to have the desired properties. Each dataset comprises of three response variables and two latent processes. Level of correlation is the only factor in the experiment governed by the data. The levels of correlation chosen for the three datasets are:

- High ($\alpha_1 = 5$, $\alpha_2 = 10$);
- Medium ($\alpha_1 = 30$, $\alpha_2 = 40$);
- Low ($\alpha_1 = 50$, $\alpha_2 = 100$).

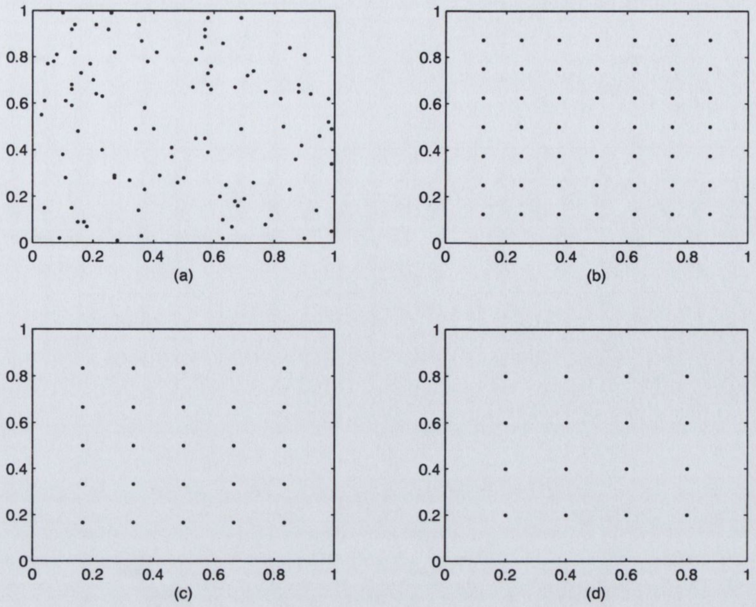


Figure 5.1: Plot (a) shows the distribution of the fine points on the unit square. Plots (b), (c) and (d) show the coarse points derived from the fine, in the ratio of 1-2, 1-4, 1-6 respectively.

where α_1 and α_2 are the levels of correlation associated with the first and second latent variables respectively, for each of the three datasets. The values taken by β are small, but not small enough to cause difficulty with mixing such as that described in Chapter 4. Our experience with this model is that mixing performance is more sensitive to the α_i and hence its selection as a experimental factor. The data are randomly located on a square map. There are many methods of generating spatial locations, some such as the Strauss process would give a smoother, more even dispersal of locations (Møller and Waagerpetersen, 2003). The approach taken to produce coarsened data is as follows.

1. Taking the outermost locations of each side as the boundary of the area, then divide this into regions of equal size.

   - The number of regions will approximately relate to a factor of coarsening.

- For example, when $n = 100$, coarsening by 4, we create a 5×5 grid and expect $\tilde{n} = 25$.

2. Sum the Poisson counts within each region.

3. Use center point of each region as the location reference for coarse data.

Naturally, there are many alternative coarsenings.

5.5 Results

We have not directly compared our algorithm with the uncoupled alternatives. This is because one can run the fine and coarse chains in parallel on separate processors, handing over to one of the processors for swap moves. Thus the coupled algorithm run on two processors, will be almost as fast as any single chain MCMC method on the fine chain only. We have run the algorithm in sequence. Our expectation of the algorithm is that by introducing a coarsened dataset (which we have coarsened by summing), its posterior will be less "peaked", although possibly still somewhat multimodal. Exchanging information between the potentially less modal coarsened process and the ordinary process may thus create better mixing in the latter.

For each experiment, the algorithm was run for $500,000$ iterations. The fine chains were initialized to their true values, hence burn-in was generally not required. To check that convergence was not an issue, a subset of the algorithms were run from various starting values. Posterior bias was also considered, the same subset of algorithms were run on a number of replicate datasets. These subsets were run for $100,000$ iterations. The coarse chain was run for $10,000$ iterations, allowing it to converge prior to beginning the coupling algorithm. This assists in reducing the overall algorithm run time, as the coarse chain runs substantially faster than the fine. Also, it is easier to choose suitable swapping proposals, once both chains have reached convergence. The samples have been thinned to include every 100^{th} iteration, so the actual sample size is $5,000$ for the main experiment or $1,000$ for the subset of checking algorithms. All the algorithms have been programmed in C and run on UNIX machines. The algorithms take a comparable amount of time to run, taking an average of 0.0632 sec/iter. The main source of variation in run time arises from the number of other tasks assigned to the processor. In Chapter 4, Section 4.4 we detailed a sample size experiment, where as the sample size increased, so too did the time per iteration, but disporportionately and by a far greater amount than the sample size. Given that this algorithm has the same underlying mechanism as the partial blocking algorithm of Chapter 4, it would be affected in a similar way by increasing the sample size. We would thus expect the time per iteration to increase substantially if the sample size were increased.

The diagnostics used to monitor the results are as before: trace plots, histograms, autocorrelation plots, kernel density plots and effective sample size (as defined in Chapter 3, Section 3.5.3). The visual diagnostics were all examined and all showed satisfactory results. However, the effective sample size measure proves to be more useful in distinguishing differences within the results. Given the additional parameters involved in this model and the large number of experiments, it would not be constructive to present all of the plots or estimates. We have chosen a selection of trace plots to view general behaviour of the algorithms. A concise but full representation of the results is given using main effects plots (Figures 5.4 - 5.7) and their associated p-values (Table 5.6). These diagnostics use the response variable τ, i.e. where effective sample size $= N/\tau$, and N is number of samples obtained. Further diagnostics such as the interaction plots are given in Appendix B.1.

Posterior bias Two replicate datasets have been produced with the same properties as those of dataset 3, i.e. high levels of correlation, where $\alpha_1 = 5.0$ and $\alpha_2 = 10.0$. A subset of the algorithms has been run on these replicate datasets to check for posterior bias. The subsets considered are the following combination of factors: coarsening level 1 with swapping rate 1, coarsening level 1 with swapping rate 3, coarsening level 3 with swapping rate 1 and coarsening level 3 with swapping rate 3. There appears to be very little bias in the algorithms. Coarsening level 3 with either of the swapping rates exhibits a small to negligible amount of bias in the β parameters. The mean and standard deviation are given for a selection of the parameters for each of the runs and each of the algorithms in Table 5.5, with traces for the α_1 parameter given in Figure 5.2.

Convergence To check the convergence properties of the algorithms, we have taken the same subset of the algorithms as those mentioned with respect to the posterior bias check. Each of the algorithms has been initialised from a number of different starting points, while being run on the high correlation dataset (dataset 3) and its replicate datasets. Each of the algorithms are run for $100,000$ iterations, keeping one in every 100 iterations. All of the algorithms converge to the same values. The length of burn-in required was minimal. Given the large number of parameters involved, we have selected the trace plots of α_1 to give an indication of the behaviour of the algorithms, see Figure 5.2. Since the model is invariant to relabelling, results for each of the two fields and their associated parameters should be the same if the algorithm has converged, e.g. β_{11} and β_{12} should be the same and so on. Looking at each of Figures 5.5-5.7 in turn, we see that the β parameters associated with the latent fields (2^{nd} and 3^{rd} plot in each figure) are broadly the same, except for Figure 5.5, where there appears to be some difference.

Acceptance rates The acceptance probabilities for the parameters are similar for all of the algorithms. The α_1 and α_2 have an acceptance rates in the region of $0.42 - 0.58$, the β's which are all updated together have an accept rate between $0.13 - 0.18$ and the latent variables have an acceptance rates of $0.78 - 0.87$. So, the acceptance rates for each of the parameters is quite reasonable. There is also the proposed swaps between coarse and fine chain, the acceptance rates for these vary. For a proposed swap of 1 in 10 the acceptance rate was $0.00016 - 0.00022$, for a proposed swap of 1 in 25 the acceptance rate was $0.0004 - 0.0005$ and for a proposed swap of 1 in 100 the acceptance rate was $0.001 - 0.0016$. These actual number of swaps between the coarse and fine algorithms works out at $8 - 11$, $8 - 10$ and $5 - 8$ for proposed rates of 1 in 10, 1 in 25 and 1 in 100 respectively.

For ease of notation we use the terms $\beta_1, \ldots, \beta_9$ to represent β_{10}, β_{11}, β_{12}, β_{20}, β_{21}, β_{22}, β_{30}, β_{31}, β_{32} respectively for graphs and tables in this section and its associated appendix.

5.5.1 Degree of Coarsening

The level of coarsening appears to be a significant factor, as indicated by the p-values from an analysis of variance test. Specifically, the effective sample size for the β parameters is affected by the degree of coarsening imposed. This is not so much the case for the θ parameters; even viewing early trace and autocorrelation plots, the θ parameters would appear much less impacted by change. From the main effects plots given in Figures 5.5, 5.6 and 5.7, it can be seen that generally levels 1 and 3 provide better results than level 2, in terms of the ESS values for the β's. This is complicated somewhat by the fact that the level of coarsening strongly interacts with correlation level, as seen by the relevant plots shown in Appendix B.1 and confirmed by the p-values given in Table 5.6. The opposite is more or less the case for the θ's, but it is not significant in terms of its p-value.

5.5.2 Level of Correlation

Correlation did not appear to have a significant effect on the effective sample size for the β's, but from observed significance levels the θ's would appear to be somewhat more sensitive to the level of correlation present in the data. This may also be the most likely

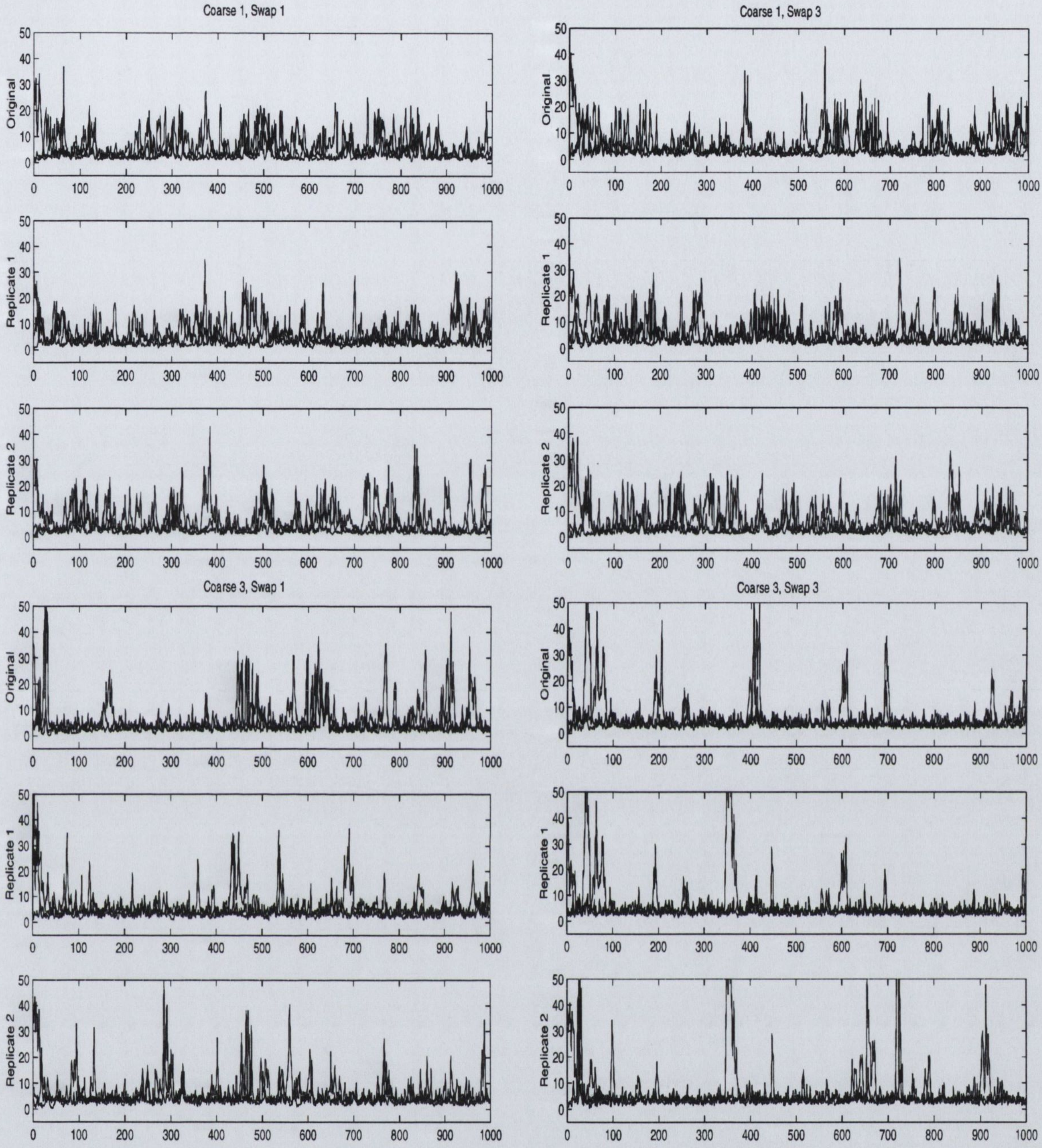


Figure 5.2: These are trace plots of the α_1 variable with various initializing values for two levels or coarsening and two rates of swapping, i.e. level 1 and swap 1, level 1 and swap 3, level 3 and swap 1, level 3 and swap 3. They are run on dataset3 (high correlation) and two additional replicate datasets with the same parameterizations. The results for the coarse level 1 and swap 1 are given in top left, coarse level 1, swap 3 are top right, coarse level 3, swap 1 are bottom left and coarse level 3, swap 3 are bottom right.

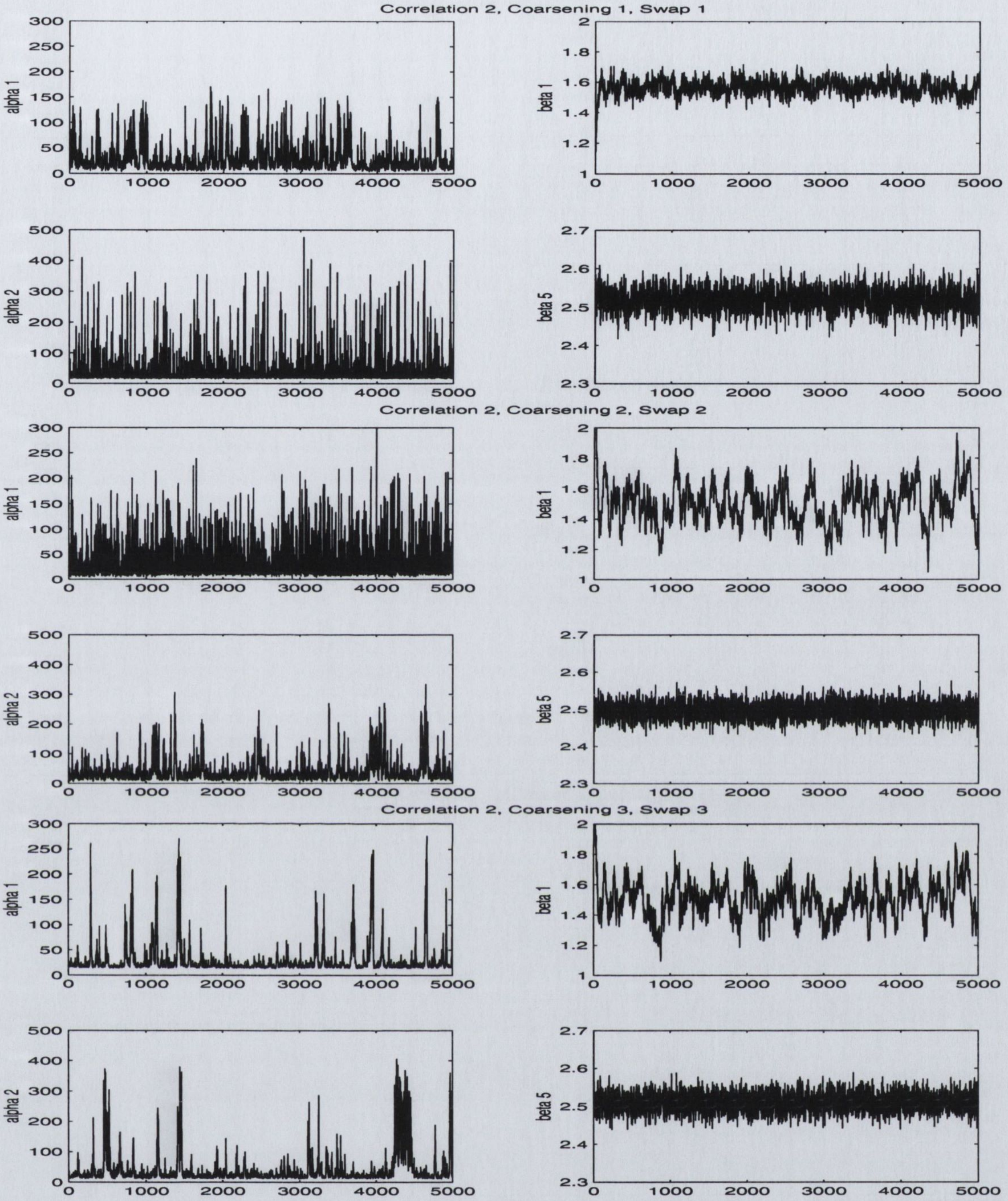


Figure 5.3: These are trace plots of some parameters for runs on the dataset with correlation level 2 and varying levels of coarsening and proposed swaps. The α parameters are on the right-handside and some β parameters are on the left-handside. The top two rows are from coarsening 1, swap 1, the middle two rows are from coarsening 2, swap 2 and the last two rows are from coarsening 3, swap 3.

Model		Dataset 3			
Parameters	Algorithm	True	Run 1 (μ,σ)	Run 2 (μ,σ)	Run 3(μ,σ)
α_1	C1,S1	5.0	(4.98,4.04)	(5.04,3.98)	(4.83,4.10)
	C1,S3		(5.02,4.49)	(5.00,3.87)	(5.09,4.19)
	C3,S1		(5.12,4.83)	(4.96,4.06)	(4.99,3.10)
	C3,S3		(4.99,5.32)	(5.01,5.16)	(5.01,6.30)
α_2	C1,S1	10.0	(10.33,4.35)	(9.35,3.93)	(9.67,4.62)
	C1,S3		(9.68,4.61)	(9.77,3.17)	(10.05,3.57)
	C3,S1		(9.45,3.94)	(9.86,4.47)	(10.31,5.09)
	C3,S3		(10.25,4.64)	(9.66,7.97)	(10.43,4.79)
β_1	C1,S1	1.5	(1.45,0.09)	(1.49,0.05)	(1.53,0.04)
	C1,S3		(1.43,0.05)	(1.54,0.07)	(1.58,0.03)
	C3,S1		(1.62,0.15)	(1.61,0.17)	(1.62,0.13)
	C3,S3		(1.42,0.15)	(1.61,0.14)	(1.47,0.18)
β_2	C1,S1	2.5	(2.59,0.09)	(2.58,0.07)	(2.42,0.05)
	C1,S3		(2.43,0.03)	(2.54,0.07)	(2.54,0.06)
	C3,S1		(2.52,0.05)	(2.58,0.07)	(2.62,0.08)
	C3,S3		(2.62,0.05)	(2.51,0.03)	(2.63,0.07)
β_3	C1,S1	3.5	(3.59,0.09)	(3.52,0.04)	(3.47,0.05)
	C1,S3		(3.43,0.05)	(3.74,0.12)	(3.54,0.07)
	C3,S1		(3.62,0.10)	(3.71,0.07)	(3.52,0.08)
	C3,S3		(3.52,0.12)	(3.61,0.11)	(3.52,0.08)
β_7	C1,S1	1.5	(1.59,0.09)	(1.48,0.06)	(1.62,0.08)
	C1,S3		(1.53,0.04)	(1.64,0.07)	(1.44,0.07)
	C3,S1		(1.62,0.15)	(1.51,0.17)	(1.52,0.18)
	C3,S3		(1.52,0.09)	(1.61,0.12)	(1.52,0.13)
β_9	C1,S1	3.5	(3.59,0.10)	(3.48,0.12)	(3.62,0.15)
	C1,S3		(3.49,0.09)	(3.64,0.12)	(3.51,0.14)
	C3,S1		(3.52,0.15)	(3.61,0.19)	(3.55,0.18)
	C3,S3		(3.52,0.23)	(3.71,0.21)	(3.52,0.18)

Table 5.5: These are the means and standard deviations for a selection of the parameters in the model, run on a combination of coarse level 1 and 3 and proposed swap rate 1 and 3 and for each of the replications of dataset 3. C1,S3 indicates coarse level 1 and proposed swapping rate 3, the other abbreviations in the table are constructed in the same way. Also given is the true parameter values for the model, the closer the mean is to this value the less biased the estimate.

place to find an effect with correlation, as α is the variable which is used to control correlation in the data or to explain the correlation in the data. This is also consistent with the results found in Chapter 4, i.e. the α parameter was the most sensitive to correlation levels. Although it is not significant, as a broad observation, correlation level 2 did appear to generate the best mixing properties. Also, as already mentioned in Section 5.5.1, there is an interaction between the amount of correlation present in the dataset and the degree to which the coarse chain is aggregated. This reflects the non-linear relationship between correlation and aggregation in the data. The other possible interaction here is the level of correlation with the rate of proposed swaps between chains. This interaction does not have a particularly intuitive interpretation, thus it is not of immense interest, and conveniently it is not at all significant.

5.5.3 Rate of Proposed Swapping

The proposed rate of swapping between chains does not appear to be significant for either the β's or the θ's. There was however a significant (or close to significant) interaction effect, between the proposed rate of swapping and the level of coarsening for the θ parameters, see Table 5.6.

5.5.4 Interaction: Degree of Coarsening by Level of Correlation

The main observations here (wrt the β parameters) are that level 2 coarsening generally gives the poorest performance, but there is an interaction, whereby at correlation level 2 it performs best or at least comparably to the other coarse levels, see Appendix B. That is, at a medium level of coarsening $(1-4)$ and with a medium level of correlation in the data, we get the best results. Generally, coarsening 1 behaves better than coarsening level 3, but there is also some interaction here. With regard to the θ's, these again are not actually significantly affected, see Table 5.6, but it is observed that coarsening 1 and 2 behave similarly, i.e. both behave best at correlation level 2, where as coarsening 3 behaves worst

Variable	Coarsening	Correlation	Swap	Coarsening x Correlation	Coarsening x Swap	Correlation x Swap
α_1	0.174	0.005	0.837	0.480	0.088	0.156
α_2	0.115	0.104	0.201	0.725	0.055	0.092
β_1	0.004	0.200	0.492	0.054	0.424	0.383
β_2	0.019	0.795	0.370	0.774	0.679	0.833
β_3	0.328	0.961	0.889	0.861	0.805	0.596
β_4	0.017	0.140	0.151	0.026	0.529	0.646
β_5	0.007	0.038	0.055	0.013	0.210	0.094
β_6	0.182	0.082	0.253	0.051	0.219	0.423
β_7	0.030	0.152	0.127	0.065	0.209	0.894
β_8	0.063	0.324	0.317	0.101	0.482	0.917
β_9	0.082	0.202	0.150	0.145	0.239	0.935

Table 5.6: The p-values for the effective sample size of each of the main parameters in the model, given the factors of interest and their interactions.

at correlation 2.

5.5.5 Interaction: Rate of Proposed Swapping by Degree of Coarsening

With respect to the θ's, level 1 and 2 coarsening behave best or comparably at swapping rate 2, whereas level 3 coarsening sees its worst results at swapping level 2, see Appendix B. However, given that neither coarsening level or swapping rate have shown themselves to be significant for the θ's, this observation has little bearing on the findings, see Table 5.6.

5.5.6 Interaction: Rate of Proposed Swapping by Level of Correlation

There was no significant interaction effect present between the level of correlation in the dataset and the proposed rate of swapping between coarse and fine chains for either the θ or β parameters, see Table 5.6.

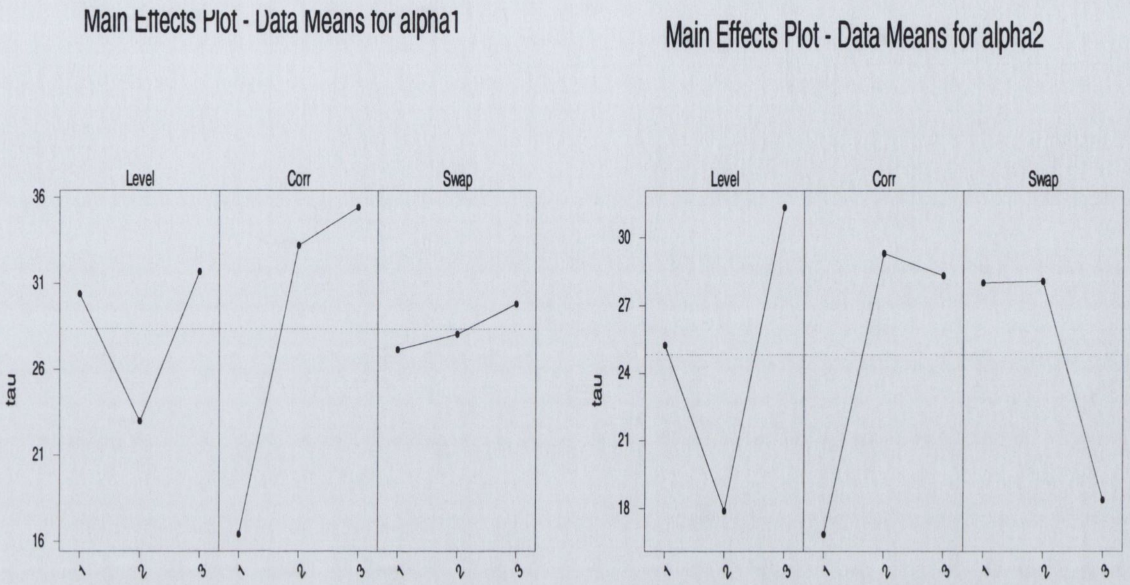


Figure 5.4: These are main effects plot with τ as the response variable, i.e. they show the maximum likelihood estimates of τ, where ess $= N/\tau$ and N is sample size. The smaller the τ value the better the estimate. The plots are for the θ parameters, i.e. α_1 on the left-hand side and α_2 on the right-hand side. Positions $1, 2$ and 3 on each of the graphs indicate increasing levels of coarsening or proposed rate of swap and decreasing levels of correlation.

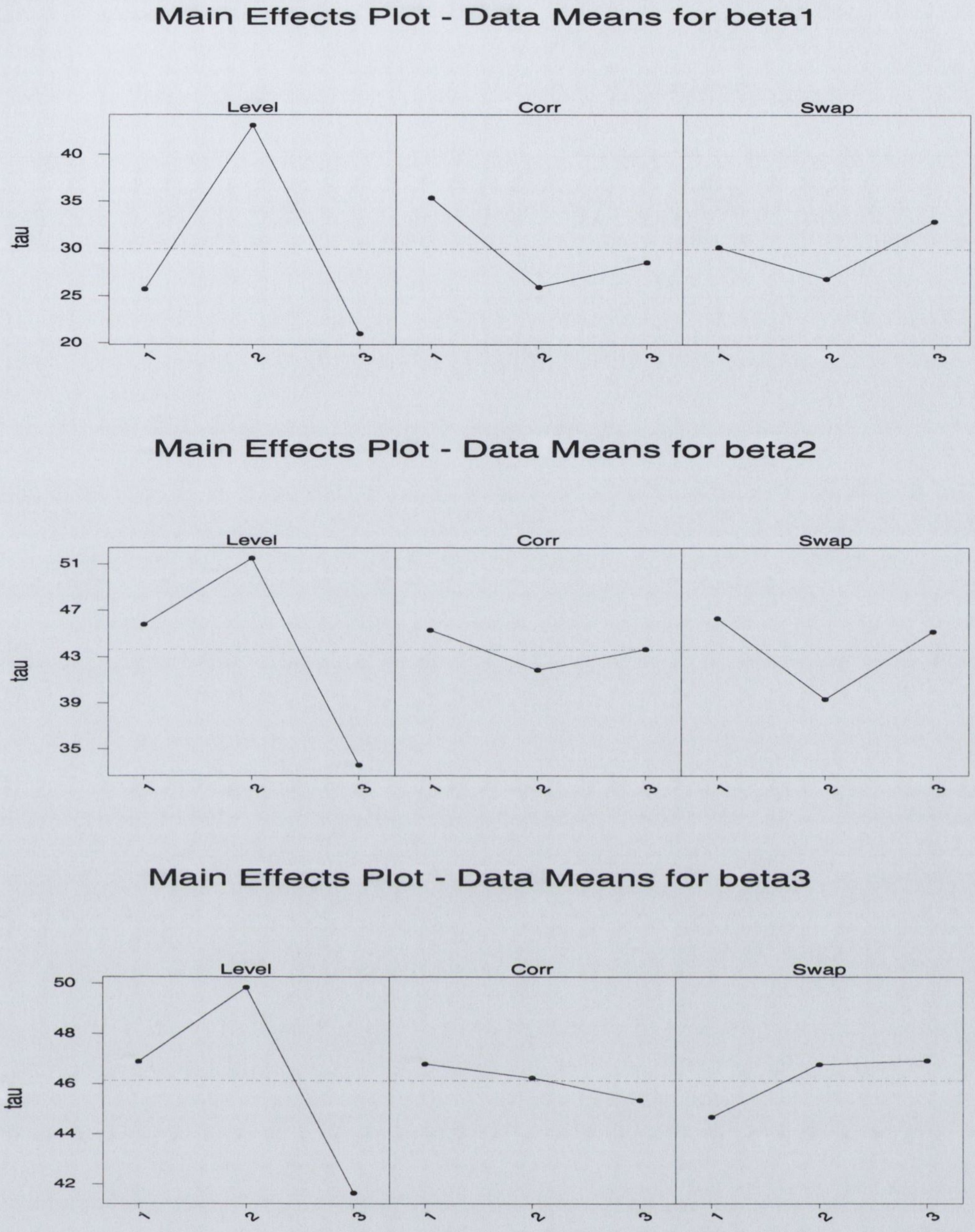


Figure 5.5: These are main effects plot with τ as the response variable, i.e. they show the maximum likelihood estimates of τ, where ess $= N/\tau$ and N is sample size. The smaller the τ value the better the estimate. The plots are for the β parameters, i.e. β_1 on top, β_2 in the middle and β_3 on the bottom. Positions $1, 2$ and 3 on each of the graphs indicate increasing levels of coarsening or proposed rate of swap and decreasing levels of correlation.

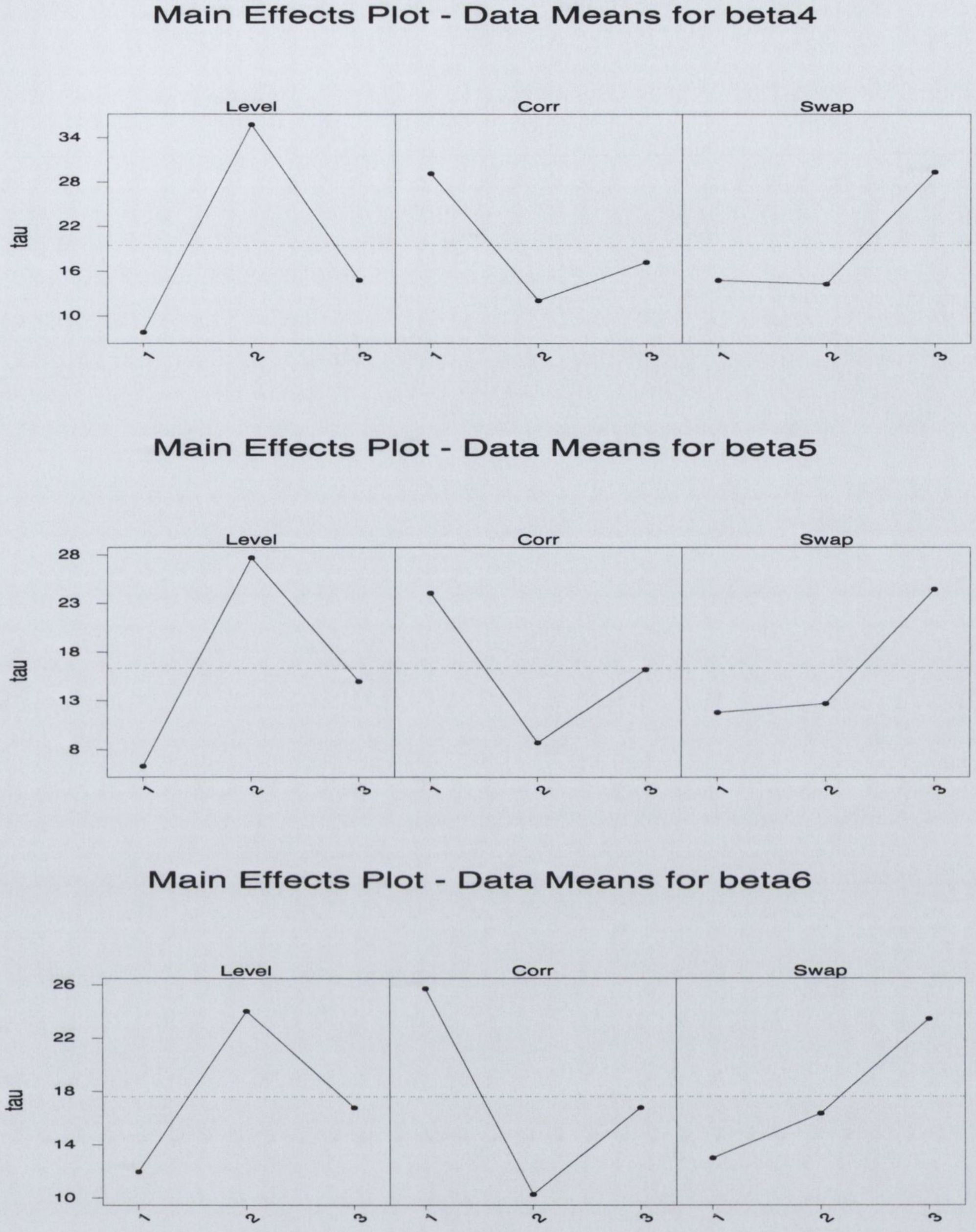
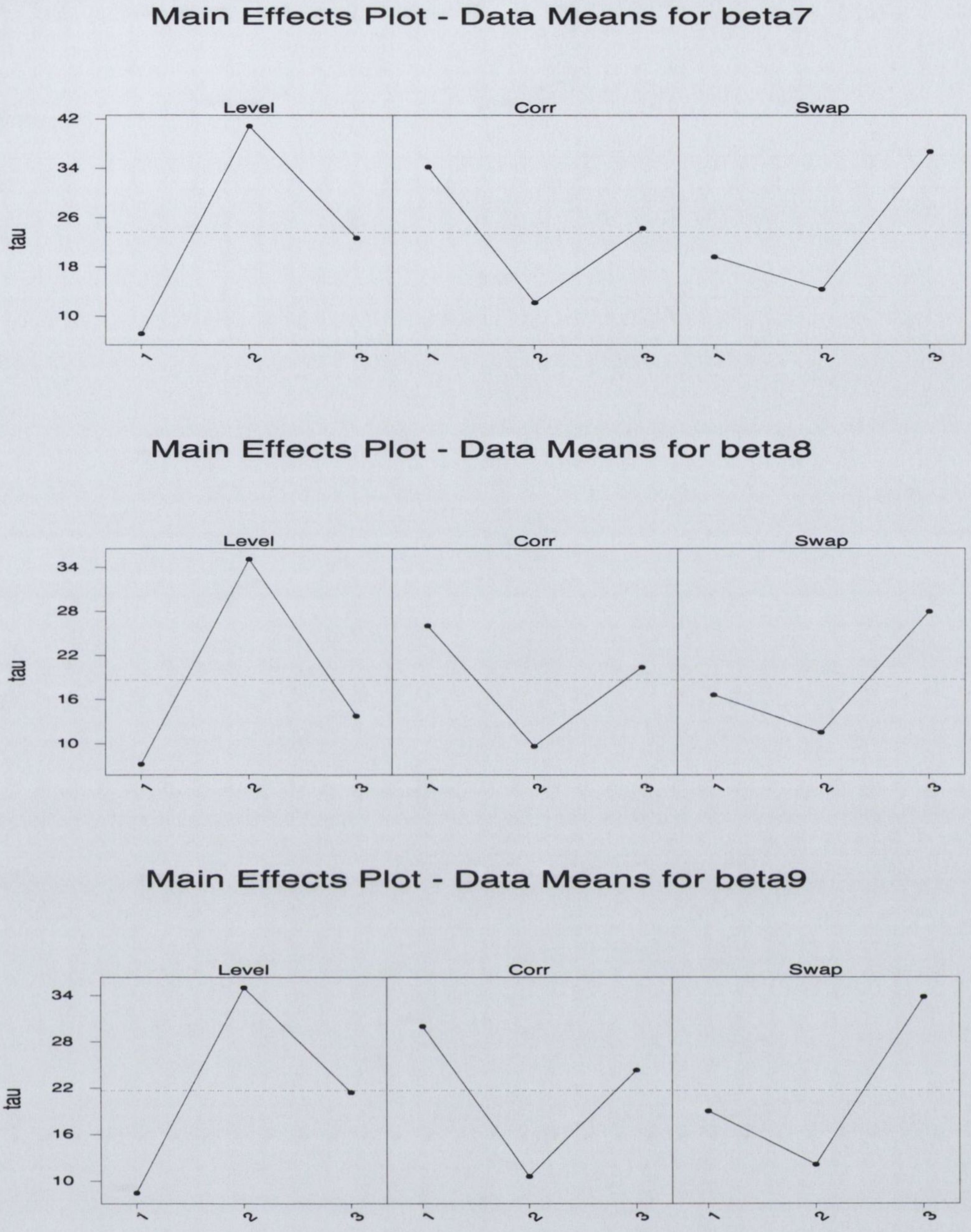


Figure 5.6: These are main effects plot with τ as the response variable, i.e. they show the maximum likelihood estimates of τ, where ess $= N/\tau$ and N is sample size. The smaller the τ value the better the estimate. The plots are for the β parameters, i.e. β_4 on top, β_5 in the middle and β_6 on the bottom. Positions $1, 2$ and 3 on each of the graphs indicate increasing levels of coarsening or proposed rate of swap and decreasing levels of correlation.


Main Effects Plot - Data Means for beta7

Main Effects Plot - Data Means for beta8

Main Effects Plot - Data Means for beta9

Figure 5.7: These are main effects plot with τ as the response variable, i.e. they show the maximum likelihood estimates of τ, where ess $= N/\tau$ and N is sample size. The smaller the τ value the better the estimate. The plots are for the β parameters, i.e. β_7 on top, β_8 in the middle and β_9 on the bottom. Positions 1, 2 and 3 on each of the graphs indicate increasing levels of coarsening or proposed rate of swap and decreasing levels of correlation.

5.6 Conclusions

Coarsening It is difficult to draw exceptionally clear conclusions from the results. In theory, we would have expected improved mixing properties as the level of coarsening in the data increased. One reason for a diminished effect or a confusing result may be that the relationship between correlation in the coarse data relative to the fine data is different for each dataset, therefore the effect of coarsening is not linear for different correlations. If this association is not linear, then the effective sample size may not be a linear function of the coarsening either. Also, there may be a kind of trade-off for increased aggregation, where the coarser the data the better the mixing, but also the less relevant or related it may be to the fine data.

Correlation Correlation has not shown itself to be of significance. This is not too surprising, as the blocking scheme with Rue's approximation would have eliminated many of the mixing problems due to correlation. There was interaction with coarsening level, which again may relate to the non-linear relationship between these two variables. Also, at level 3 correlation (the lowest level), the data is very close to being iid in nature, so one would not expect to see an effect from the coupling. That is to say, when there is no correlation in the data, then coarsening will have no effect.

Swapping Rate of proposed swapping between the fine scale and the coarse scale chain did not appear to have an effect. The reason for this is that, the proposed rate of swap and the actual rate of swap are different. The actual rate of swapping has remained approximately the same (adjusted itself to being almost constant) within the series of experiments, that is, the rate of swaps would be the same for each experiment and thus not a real factor.

Additional observations Whether significant or not, from inspection of the main effects and interaction plots (see Figures 5.5 - 5.7 and Appendix B.1), the θ's generally behave best in the situations where the β's struggled most and vice versa. A possible explanation for this may be, that if the θ's are mixing or moving very efficiently, the β's in some sense may be finding it difficult to "keep up" and similarly when the situation is reversed. The β's are updated in a block and so exhibit very similar behaviour to each other; as is the case for the θ_1 and θ_2.

Chapter 6

Summary and Conclusions

6.1 Summary of Blocking Algorithm

The performance of an MCMC method for implementing Bayesian inference for a univariate spatial Poisson model has been examined. The data that the model are applied to utilizes a latent Gaussian field. The MCMC method presented is a blocking scheme, which also introduced a proposal for the latent variables (X_i), using a Gaussian approximation of the full conditional of the X_i's. This proposal allows very favourable acceptance probabilities for the X_i's. Two versions of the method are compared with a standard algorithm on various datasets. Evaluation of its performance is with respect to mixing and measured by effective sample size. Levels of correlation in the datasets appear to substantially affect the output.

6.2 Summary of Coupling Algorithm

A new MCMC method is introduced, primarily based on a coupling technique. This method utilises data with varying levels of resolution. The model upon which the MCMC method is applied is a multivariate spatial Poisson model, which incorporates a number of latent Gaussian fields. The data is spatial, but now has a number of responses at each location.

The MCMC method utilises the benefits of aggregating data without loss of information. There are two phases to the algorithm; a standard MCMC mode, which is quite similar to the previously investigated algorithm, and a swapping mode, which attempts to exchange information between the aggregated and fine Markov chains. The method is examined under a number of circumstances which would be expected to effect its performance. The experiment is carried out using a Latin square design.

6.3 Conclusions

Time is a constraint in running MCMC algorithms, and is a motivation for improving their efficiency. For the blocking algorithm, the time taken per iteration on the size of dataset considered is substantially less than that of a corresponding sequential approach. Efficiency with respect to mixing is also significantly improved using the blocking scheme. The approximation used in the latent variable proposal contributes to this efficiency, but is time consuming to program initially.

The combination of the coupling and blocking has been seen to resolve many of the associated difficulties when we move to the more complex multivariate count model. Blocking also improves mixing in the multivariate model. We have tried to further improve mixing by running coupled chains. The performance of the coupling as a function of various factors was investigated. However, the relationship between the factors involved is not always straight-forward or even intuitive. Factors have an effect, but it is complex and almost certainly problem dependent. The degree of aggregation applied was the main influencing factor. Aggregation appeared to have a non-linear relationship with the level of correlation present in the data, which manifest itself correspondingly with the measure of efficiency. The effect of aggregating data has long been one of interest, and would certainly lend itself to further investigation within this framework.

6.4 Further Work

There are many elements of the MCMC algorithms which have been presented that could be extended or further investigated, a few of these are described below.

6.4.1 Parallel Algorithm

The current structure of the coupling algorithm is a fine MCMC chain run for a number of iterations, followed by a coarsened chain running for the same number of iteration. This structure lends itself entirely to a parallel program rather than the present sequential one. If the program was parallelised, the most obvious difference would be in its run time. Such a coupled algorithm would not be much slower than a simple non-coupled MCMC approach in terms of iterations/sec. It would also become much easier to change or control many of the other elements of the algorithm.

One possibility would be that there could be any number of chains run in parallel, with different levels of aggregation. This would allow a much more sophisticated exchange of information. It would also eliminate some very convoluted notation currently used in the coding of the coupling algorithm. Also, the more aggregated the dataset, the faster its cycle time. This may or may not be of advantage because of the trade-off between speed and relevance to the fine chain. If it were favourable for the more aggregated chains to be run for longer, then a parallel environment would be advantageous, and conversely there would be no drawback.

6.4.2 Other Factors

There are many other factors which are likely to effect the performance of the coupling algorithm and hence would be desirable to examine. MCMC algorithms can be sensitive to data size, but specific to the coupling algorithm would be the number of chains, the relative number of chain iterations between swaps and choice of proposals for swaps ($h(\theta)$).

6.4.3 Swapping Rate

One of the difficulties encountered during the coupling experiment was that the proposed swapping rate did not appear to have an effect. Effectively the accept rates for the proposed swaps neutralised swapping as a factor, i.e. swapping was constant regardless of the proposed swap rate. Pursuing this factor and establishing its real effect, especially its relationship with aggregation level, would be of interest.

6.4.4 Other Algorithms

Although we have already seen that in many circumstances the blocking algorithm outperforms other standard algorithms, it may be worthwhile comparing the coupling method to other such algorithms. In particular, it may be useful to compare the coupling algorithm with the blocking algorithm, for a number of datasets with varying degrees of correlation.

6.4.5 More Complex Models

There is the possibility of increasing the number of latent processes, which may lend itself to better modeling of the response variables. There could also be a more sophisticated correlation structure used in the model, to more accurately describe the relationship with the data.

Appendix A

Acceptance Probabilities

A.1 Univariate Count Model Accept Probabilities

Let $P(.)$ denote an accept probability, $\pi(.)$ and $\pi(.|.)$ denote marginal and conditional distributions respectively and $q(.|.)$ denote a proposal distribution. Let $y = (y_1, \ldots, y_n)$ be the observed values, y_i being observed at location s_i, where $i = 1, \ldots, n$ and Y_i be Poisson distributed with mean $\lambda_i = \exp(\beta + x_i)$. The X_i's are multivariate normal with covariance matrix $\Sigma(\theta)$, $\theta = (\alpha, \sigma^2, \delta)$, which has a structure as described in Chapter 2, Section 2.3.5.

A.1.1 Sequential Algorithm

In the sequential algorithm described in Chapter 4, Section 4.1.1, β is updated first, then α, σ^2, δ and each of the X_i's, to be consistent we give the acceptance probabilities for each of the parameters in accordance with the ordering of that in the main text.

Accept for β The accept probability for β is given as:

$$P(\beta', \beta) = \frac{\pi(\beta'|\theta, x, y)q(\beta|\beta')}{\pi(\beta|\theta, x, y)q(\beta'|\beta)} = \frac{\pi(\beta'|x, y)q(\beta|\beta')}{\pi(\beta|x, y)q(\beta'|\beta)}.$$

Then

$$\pi(\beta|x,y) \propto \pi(y|x,\beta)\pi(\beta), \quad \text{by Bayes Theorem,}$$

so

$$P(\beta^{'},\beta) = \frac{\pi(\beta^{'}|x,y)\pi(\beta^{'})q(\beta|\beta^{'})}{\pi(\beta|x,y)\pi(\beta)q(\beta^{'}|\beta)}.$$

Given that the same uniform prior is used, $\pi(\beta^{'})$ and $\pi(\beta)$ do not appear in the probability. Details of the range for this prior are given in Chapter 2, Section 2.3.5. The proposal probability $q(.|.)$ is an additive random walk, so this further reduces the acceptance probability to:

$$P(\beta^{'},\beta) = \frac{\pi(\beta^{'}|x,y)}{\pi(\beta|x,y)},$$

where

$$\pi(\beta^{'}|x,y) = \prod_{i=1}^{n}\left(-\frac{\exp(e^{\beta^{'}+x_i})\exp\left((\beta^{'}+x_i)y_i\right)}{y_i!}\right).$$

For ease of calculation and notation, we have used and will give the *log* accept probability:

$$\text{log}P(\beta^{'},\beta) = \sum_{i=1}^{n}\left(-\exp(\beta^{'}+x_i)+\exp(\beta+x_i)+\beta^{'}y_i-\beta y_i\right).$$

Alternatives to the above proposal probability are an independent Gaussian proposal that approximates $\pi(\beta|x,y)$ at the modes.

Accept for θ Similarly for θ, where α,σ^2 and δ are updated individually using random walk proposals and flat priors, details of these priors are given in Chapter 2, Section 2.3.5. Given the posterior distribution for θ:

$$\begin{aligned}\pi(\theta|x) &\propto \pi(x|\theta)\pi(\theta)\\ &\propto |\Sigma|^{-\frac{1}{2}}\exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)\end{aligned}$$

The log accepts for α,σ^2 and δ are then given by:

$$\text{log}P(.) = \frac{1}{2}(\text{log}|\Sigma| - \text{log}|\Sigma^{'}|) + \frac{1}{2}(x^T\Sigma^{-1}x - x^T\Sigma^{'-1}x)$$

where

$$\begin{aligned} \Sigma' &= \sigma^2 \exp(-(\alpha' d_{ij})^\delta) \\ &\text{or } \sigma'^2 \exp(-(\alpha d_{ij})^\delta) \\ &\text{or } \sigma^2 \exp(-(\alpha d_{ij})^{\delta'}) \end{aligned}$$

for $P(\alpha', \alpha), P(\sigma'^2, \sigma^2)$ and $P(\delta', \delta)$ respectively, and $\Sigma = \sigma^2 \exp(-(\alpha d_{ij})^\delta)$.

Accept for X_i Lastly the acceptance probability for the latent values (X_i's), each of which is updated individually is given by:

$$P(x_i', x_i) = \frac{\pi(x_i'|x_{-i}\theta)\pi(y_i|x_i'\beta)\pi(\theta)\pi(\beta)q(x_i|x_i')}{\pi(x_i|x_{-i}\theta)\pi(y_i|x_i\beta)\pi(\theta)\pi(\beta)q(x_i'|x_i)}$$

where $x' = (x_1, \ldots, x_i', \ldots, x_n)$ for X_i'. A proposal for X_i is generated from the conditional univariate Gaussian – MVN$(\mu_{i|-i}, \Sigma_{i|-i})$, where the conditional mean and variance are calculated using $\mu_{i|-i} = X_i - \Sigma_{i|-i}\left(\sum_{j=1}^{n}((\Sigma^{-1})_{ij}X_j)\right)$ and $\Sigma_{i|-i} = \frac{1}{(\Sigma^{-1})_{ii}}$ respectively, see Whittaker (1990) for details of these formulae. The posterior distribution for X_i is then given by:

$$\pi(x_i|x_{-i}\theta) = \frac{1}{(2\pi)^{\frac{1}{2}}|\Sigma|^{\frac{1}{2}}} \exp\left(-\frac{1}{2}x^T\Sigma^{-1}x\right)$$

and

$$\pi(y_i|x_i\beta) = \frac{\exp(-\lambda_i)\lambda_i^{y_i}}{y_i!}.$$

Given that the proposal for X_i is its conditional distribution, the proposal probabilities and the conditional Gaussian part of the posterior will cancel within the acceptance calculation. The log of the acceptance probability then reduces to:

$$\log P(x_i', x_i) = (\lambda_i - y_i \log\lambda_i) - \left(\lambda_i' - y_i \log\lambda_i'\right)$$

where λ_i is as defined earlier.

The computational cost for this method will be: low for the update of the β's, as this is just a vector multiplication; quite high for the θ's, i.e. there will be a matrix inversion for α, σ and δ; the x_i's are updated using one matrix inversion and some matrix multiplication between them, so a little less computationally intensive than the θ's.

A.1.2 Partial-Block Algorithm

These are the accept probabilities for the MCMC procedure of Chapter 4, Section 4.1.2.

Accept for X They are given in the form of their log accept probabilities for convenience. There is much cancellation in the accept probability for the latent variables, after which the below accept probability is arrived at:

$$\begin{aligned}\log P(x^{'},x) &= \frac{1}{2}x^{'T}Cx - \frac{1}{2}x^{T}C^{'}x + B^{T}x^{'} - B^{'T}x + \sum_{i=1}^{n}\exp(\beta + x_i)\\ &\quad -\sum_{i=1}^{n}\exp(\beta + x_i^{'}) - \frac{1}{2}\log|\Sigma^{-1} + C| + \frac{1}{2}\log|\Sigma^{-1} + C^{'}|\\ &\quad +\frac{1}{2}(y - B)^{T}(\Sigma^{-1} + C)^{-1}(y - B) - \frac{1}{2}(y - B^{'})^{T}(\Sigma^{-1} + C^{'})^{-1}(y - B^{'}),\end{aligned}$$

where $C^{'} = \exp^{\beta}\operatorname{diag}(C_1^{'},\ldots,C_n^{'})$, $B^{'} = \exp^{\beta}(B_1^{'},\ldots,B_n^{'})$. These are found by solving:

$$\begin{aligned}(A_i^{'},B_i^{'},C_i^{'}) &= \arg\min\left[\int_{x_i^{'}-\Delta}^{x_i^{'}+\Delta}\left(\exp(x_i) - (A_i^{'} + B_i^{'}x_i + \frac{1}{2}C_i^{'}x_i^2)\right)^2 \mathrm{d}x_i\right]\\ &= \arg\min\int_{x_i^{'}-\Delta}^{x_i^{'}+\Delta}\exp(2x_i) + A_i^{'2} + B_i^{'2}x_i^2 + \frac{1}{4}C_i^{'2}x_i^4 - 2A_i^{'}\exp(x_i)\\ &\quad -2B_i^{'}x_i\exp(x_i) - C_i^{'}x_i^2\exp(x_i) + 2A_i^{'}B_i^{'}x_i + A_i^{'}C_i^{'}x_i^2 + B_i^{'}C_i^{'}x_i^3\,\mathrm{d}x_i\\ &= \arg\min\frac{1}{2}\exp(2x_i) + A_i^{'2} + \frac{1}{3}B_i^{'2}x_i^3 + \frac{1}{20}C_i^{'2}x_i^5 - 2A_i^{'}\exp(x_i) + A_i^{'}B_i^{'}x_i^2\\ &\quad +\frac{1}{3}A_i^{'}C_i^{'}x_i^3 + \frac{1}{4}B_i^{'}C_i^{'}x_i^4 - 2B_i^{'}\exp(x_i)(x_i - 1) - C_i^{'}\exp(x_i)(x_i^2 - 2x_i + 2)\,\Big|_{x_i-\Delta}^{x_i+\Delta}\end{aligned}$$

Setting this equation equal to zero and differentiating with respect to A_i, B_i and C_i, then solving the resulting series of simultaneous equations gives the required minimised values for A_i, B_i and C_i. Only B_i and C_i are used in the proposal of x_i' and appear in the acceptance probability. The Δ parameter relates to the size of proposed movement between x and x'. .

Accept for β

$$\log P(\beta', \beta) = \left(\exp(\beta) - \exp(\beta')\right) \sum_{i=1}^{n} \exp(x_i) + (\beta' - \beta) \sum_{i=1}^{n} y_i,$$

Accept for θ

$$\log P(\theta', \theta) = \frac{1}{2}\left(-x^T \Sigma'^{-1} x + x^T \Sigma^{-1} x - \log|\Sigma'| + \log|\Sigma|\right)$$

where $\Sigma' = \sigma'^2 \exp(-(\alpha' d_{ij})^{\delta'})$. The log accept probabilities for the β's and θ's given, use independent random walk proposals and uniform priors, see Chapter 2, Section 2.3.5 for details.

A.1.3 Total-Block Algorithm

The accept probability for the MCMC algorithm of Chapter 4, Section 4.1.3 is:

$$P(x', \beta', \theta' | y, x, \beta, \theta) = \frac{\pi(y|x'\beta')\pi(x'|\theta')\pi(\beta')\pi(\theta')q(x|x')q(\beta|\beta')q(\theta|\theta')}{\pi(y|x\beta)\pi(x|\theta)\pi(\beta)\pi(\theta)q(x'|x)q(\beta'|\beta)q(\theta'|\theta)},$$

where

$$\pi(y|x', \beta') = \frac{\mathrm{e}^{-\exp(\beta' + x_i')} \exp(\beta' + x_i')^{y_i}}{y_i!},$$

$$\pi(x'|\theta') = \frac{1}{(2\pi)^{\frac{p}{2}} |\Sigma'|^{\frac{1}{2}}} \exp\left(-\frac{1}{2} x^T \Sigma'^{-1} x'\right)$$

and

$$\begin{aligned}
q(x^{'}|x) &\propto \exp\left(-\frac{1}{2}x^{'T}(\Sigma^{'-1}+C)x^{'}+(y+B)^{T}x^{'}\right)\\
&= \frac{|\Sigma^{'-1}+C|^{\frac{1}{2}}}{2\pi^{\frac{p}{2}}}\exp\left(-\frac{1}{2}x^{'T}(\Sigma^{'-1}+C)x^{'}+(y-B)^{T}x^{'}\right)\\
&\quad\times\exp\left(\frac{1}{2}(y-B)^{T}(\Sigma^{'-1}+C)^{-1}(y-B)\right).
\end{aligned}$$

Given that $\pi(\beta)$ and $\pi(\theta)$ are flat priors (see Chapter 2, Section 2.3.5 for details of their ranges), and the proposal functions $q(\beta^{'}|\beta)$ and $q(\theta^{'}|\theta)$ are random walks, these values will cancel out in the overall calculation. The log accept probability is then given as:

$$\begin{aligned}
\log(P(x^{'},\beta^{'},\theta^{'}|y,x,\beta,\theta)) &= \frac{1}{2}\left(-\log|\Sigma^{'}|+\log|\Sigma|-x^{'T}\Sigma^{'-1}x^{'}+x^{T}\Sigma^{-1}x\right)\\
&\quad+\sum_{i=1}^{n}\left(\exp(\beta+x_i)-\exp(\beta^{'}+x_i^{'})\right)+\sum_{i=1}^{n}\left(y_i(\beta^{'}+x_i^{'})-y_i(\beta+x_i)\right)\\
&\quad+\frac{1}{2}\left(-\log|\Sigma^{'-1}+C|+x^{'T}(\Sigma^{'-1}+C)x^{'}\right)\\
&\quad-\frac{1}{2}\left(-\log|\Sigma^{-1}+C^{'}|+x^{T}(\Sigma^{-1}+C^{'})x\right)\\
&\quad+\frac{1}{2}\left((y-B)^{T}(\Sigma^{'-1}+C)^{-1}(y-B)\right)-(y-B)^{T}x^{'}\\
&\quad-\frac{1}{2}\left((y-B^{'})^{T}(\Sigma^{-1}+C^{'})^{-1}(y-B^{'})\right)+(y-B^{'})^{T}x.
\end{aligned}$$

With some cancellation, this reduces to:

$$\begin{aligned}
\log(P(x^{'},\beta^{'},\theta^{'}|y,x,\beta,\theta)) &= \frac{1}{2}\left(-\log|\Sigma^{'}|+\log|\Sigma|\right)\\
&\quad+\sum_{i=1}^{n}\left(\exp(\beta+x_i)-\exp(\beta^{'}+x_i^{'})+y_i(\beta^{'}-\beta)\right)\\
&\quad+\frac{1}{2}\left(-\log|\Sigma^{'-1}C|+\log|\Sigma^{-1}+C^{'}|+x^{'T}Cx^{'}-x^{T}C^{'}x\right)\\
&\quad+\frac{1}{2}\left((y-B)^{T}(\Sigma^{'-1}+C)^{-1}(y-B)\right)+B^{T}x^{'}\\
&\quad-\frac{1}{2}\left((y-B^{'})^{T}(\Sigma^{-1}+C^{'})^{-1}(y-B^{'})\right)-B^{'T}x.
\end{aligned}$$

A.2 Multivariate Count Model Accept Probabilities

Let $y_{ij} = y_j(s_1), \ldots, y_j(s_n)$ be Poisson distributed, with means $\exp(\lambda_{ij})$, observed at locations $s_i = (s_1, \ldots, s_n)$. Then define λ_{ij} to be $\beta_{j0} + \sum_{t=1}^{T} \beta_{jt} x_{it}$ using t to index the latent variables x_t, for $t = 1, \ldots, T$. Also, $x_t \sim MVN(0, \Sigma_t)$ with $\Sigma_t(i,j) = \sigma_t^2 \exp(-\alpha_t d(i,j))$ for $\sigma_t^2 = 1$ and $d(i,j)$ is the distance between locations s_i and s_j. The accept probabilities for the multivariate latent Gaussian model described above and used in Chapter 5 are given below. Depending on the cycle of the program, the accept probability being calculated will take one of two forms, that of a block update for the model described or that of a swapping update. The calculation involved in the latter is substantially more complex. We shall give the accept probabilities for the block update applied to the fine and coarse chains first. Details of updating order and the algorithm as a whole are found in Chapter 5, Section 5.2.

A.2.1 Blocking

Accept for X_t Each latent variable $X_t = (x_{1t}, \ldots, x_{nt})$ is updated by conditioning on the other $T-1$ latent variables. The accept probability for each of the latent variables is:

$$P(x_t^{'}, x_t) = \min\left(1, \frac{\pi(x_t^{'}|x_{-t})q(x_t|x_t^{'})}{\pi(x_t|x_{-t})q(x_t^{'}|x_t)}\right)$$

where

$$\pi(x_t^{'}|x_t) \propto \pi(y|x_t^{'}, \beta)\pi(x_t^{'}|\theta_t)\pi(\beta)\pi(\theta_t).$$

The full conditional for y is:

$$\log(\pi(y|x_t, \beta)) = \sum_{j=1}^{r}\sum_{i=1}^{n}\left[-\exp(\beta^*_{(j,-t)} + \beta_{jt}x_{it}) + (\beta^*_{(j,-t)} + \beta_{jt}x_{it})y_{ij} - \sum_{f=1}^{y_{ij}}\log f\right]$$

and

$$\log(\pi(x_t|\theta)) = -\frac{1}{2}\log|\Sigma_t| - \frac{1}{2}x_t^T\Sigma_t^{-1}x_t.$$

The full conditional of x_t is:

$$\log\left(\pi(x_t|y,\beta,\theta_t)\right) = \sum_{j=1}^{r}\sum_{i=1}^{n}\left[\left(\beta^*_{(j,-t)} + \beta_{jt}x_{it}\right)y_{ji} - \exp\left(\beta^*_{(j,-t)} + \beta_{jt}x_t\right)\right] - \frac{1}{2}\left(x_t^T\Sigma_t^{-1}x_t\right),$$

for $\beta^*_{(j,-t)} = \left(\beta_{j0} + \sum_{-t=1}^{T}\beta_{(j,-t)}x_{(i,-t)}\right)$, where x_{-t} denotes vector x without component t, i.e. $-t = (1,\ldots,t-1,t+1,\ldots,T)$. The x_t's are proposed from their full conditional distribution, using the approximation $\exp(\beta_{jt}x_t) \approx A_j + B_j(\beta_{jt}x_t) + \frac{1}{2}C_j(\beta_{jt}x_t)^2$, and solving for A, B, C as described in Section A.1.2. So X_t is proposed from

$$\log(\pi(x_t|\beta,\theta_t)) = \sum_{j=1}^{r}\left[-\frac{1}{2}x_t^T(\Sigma_t^{-1} + C^*_{jt}\beta_{jt}^2)x_t + \sum_{i=1}^{n}\left(y_{ij} - B^*_{jt}\right)\beta_{jt}x_{it}\right],$$

where

- $C^*_{jt} = \exp(\beta^*_{(j,-t)})\mathrm{diag}C_j$;
- $B^*_{jt} = \exp(\beta^*_{(j,-t)})B_j$.

Then

$$\begin{aligned}\log P(x_k', x_t) \quad = \quad & \sum_{j=1}^{r}\sum_{i=1}^{n}\left[\beta_{jt}y_{ij}(x_{it} - x_{it}') + \exp(\beta^*_{(j,-t)})\exp(\beta_{jt}x_{it} - \beta_{jt}x_{it}')\right] \\ & + \frac{1}{2}\left(x_t'^T\Sigma_t^{-1}x_t' - x_t^T\Sigma_t^{-1}x_t + b_t^TQ_t^{-1}b_t - b_t'^TQ_t'^{-1}b_t'\right) \\ & + \frac{1}{2}\left(-\log|Q_t| + \log|Q_t'| + x_t'^TQ_tx_t' - x_t^TQ_t'x_t\right) + \sum_{i}^{n}\left(b_{ti}'^Tx_{it} - b_{ti}^Tx_{it}'\right)\end{aligned}$$

where

- $Q_t = \sum_{j=1}^{r}\left(\Sigma_t^{-1} + C^*_{jt}\beta_{jt}^2\right)$ and $Q_t' = \sum_{j=1}^{r}\left(\Sigma_t^{-1} - C'^*_{jt}\beta_{jt}^2\right)$;
- $b_{ti} = \sum_{j=1}^{r}(y_{ij} - B^*_{jt})\beta_{jt}$ and $b_{ti}' = \sum_{j=1}^{t}(y_{ij} - B'^*_{jt})\beta_{jt}$.

With some cancellation this reduces to:

$$\begin{aligned}\log P(x_t', x_t) \quad = \quad & \sum_{j=1}^{r}\sum_{i=1}^{n}\left[\exp(\beta^*_{(j,-t)})\left(\exp(\beta_{jt}x_{it}') - \exp(\beta_{jt}x_{it})\right)\right] \\ & + \frac{1}{2}\left(-\log|Q_t| + \log|Q_t'| + b_t^TQ_t^{-1}b_t - b_t'^TQ_t'^{-1}b_t'\right) \\ & + (B^*_{jt}\beta_{jt})^Tx_t' - (B'^*_{jt}\beta_{jt})^Tx_t - x_t'^T(C^*_{jt}\beta_{jt}^2)x_t' + x_t^T(C'^*_{jt}\beta_{jt}^2)x_t.\end{aligned}$$

Accept for β The accept probabilities for the β and θ parameters are very similar to those for Chapter 4, Section 4.1.2. The β's are all updated together, using random walk proposals and a normal priors. The priors for the β's are given as:

$$\pi(\beta_r) = \frac{1}{\sqrt{2\pi}\tilde{\sigma}_r} \exp\left(\frac{-(\beta_r - \tilde{\mu}_r)^2}{2\tilde{\sigma}_r^2}\right),$$

where $\tilde{\mu}_r$ and $\tilde{\sigma}_r^2$ are the sample mean and variance of $\log(Y_r)$. This is like an empirical Bayes approach. The variance of β should be small relative to $\tilde{\sigma}_r^2$ as it is the mean of the $\log(X_i)$. We also find that this improves the convergence for β over using a less informative prior. The log-accept probability is then given as:

$$\begin{aligned} \log P(\beta', \beta) &= \sum_{j=1}^{r}\sum_{i=1}^{n}\left(\beta_{j0} - \beta'_{j0}\right) y_{ij} + \left(\sum_{t=1}^{T}(\beta_{jt} - \beta'_{jt})x_{it}\right) y_{ij} + \exp\left(\beta'_{j0} + \sum_{t=1}^{T}\beta'_{jt}x_{it}\right) \\ &\quad - \exp\left(\beta_{j0} + \sum_{t=1}^{T}\beta_{jt}x_{it}\right) + \frac{1}{2\tilde{\sigma}_j^2}\left(\sum_{t=0}^{T}(\beta'_{jt} - \tilde{\mu}_j)^2 - (\beta_{jt} - \tilde{\mu}_j)^2\right). \end{aligned}$$

Accept for θ_t The θ_t's are updated separately using random walk proposal kernels and a uniform prior:

$$\pi(\theta_t) = \frac{1}{\theta_U - \theta_L}$$

where the range for the uniform prior is $(\theta_L, \theta_U) = (0, 500)$. The range for the θ_t priors is quite large and in practice their values fall well within them. In some circumstances, one may have a priori information that could lead to a more informative prior. The log-accepts for the θ_t's are then given as:

$$\log P(\theta'_t, \theta_t) = \frac{1}{2}\left(\log|\theta_t| - \log|\theta'_t| + x_t^T \theta_t^{-1} x_t - x_t^T \theta_t'^{-1} x_t\right),$$

where θ'_t is taken to mean $\sigma_t^2 \exp(-\alpha'_t d_{i,j})$ and $\sigma^2 = 1$.

A.2.2 Swap Between Coarse And Fine Chains

The proposal for the coarse chain and the fine chain is created by swapping their values. All of the values are proposed together. Notationally, the coarse chain parameters are differentiated from the fine chain parameters by use of a tilde over the parameter (eg $\tilde{x}$). The accept probability for this proposed swap of information is detailed below. Let

$$q((x,\theta,\tilde{x},\tilde{\theta}),(\tilde{x}^{'},\tilde{\theta}^{'},x^{'},\theta^{'})) = q((x,\theta),(\tilde{x}^{'},\tilde{\theta}^{'})) \times q((\tilde{x},\tilde{\theta}),(x^{'},\theta^{'})),$$

where $q((x,\theta),(\tilde{x}^{'},\tilde{\theta}^{'}))$ generates a coarse scale proposal $(\tilde{x}^{'},\tilde{\theta}^{'})$ from the current fine scale state (x,θ). Similarly, the kernel $q((\tilde{x},\tilde{\theta}),(x^{'},\theta^{'}))$ generates a fine scale proposal $(x^{'},\theta^{'})$ from the current coarse scale state $(\tilde{x},\tilde{\theta})$. Then denoting the posterior distributions for the coarse and fine chains as:

$$\pi(x,\theta|y) = \pi(y|x)\pi(x|\beta,\theta)\pi(\beta)\pi(\theta)$$

and

$$\tilde{\pi}(\tilde{x},\tilde{\theta}|\tilde{y}) = \tilde{\pi}(\tilde{y}|\tilde{x})\tilde{\pi}(\tilde{x}|\tilde{\beta},\tilde{\theta})\tilde{\pi}(\tilde{\beta})\tilde{\pi}(\tilde{\theta}).$$

The accept probability can then be written as:

$$P(x^{'},\theta^{'},\tilde{x}^{'},\tilde{\theta}^{'}|\tilde{y},\tilde{x},\tilde{\theta},y,x,\theta) = \frac{\pi(x^{'},\theta^{'}|y)\tilde{\pi}(\tilde{x}^{'},\tilde{\theta}^{'}|\tilde{y}) \times q((x,\theta,\tilde{x},\tilde{\theta}),(\tilde{x}^{'},\tilde{\theta}^{'},x^{'},\theta^{'}))}{\tilde{\pi}(\tilde{x},\tilde{\theta}|\tilde{y})\pi(x,\theta|y) \times q((x^{'},\theta^{'},\tilde{x}^{'},\tilde{\theta}^{'}),(\tilde{x},\tilde{\theta},x,\theta))}.$$

The proposal distributions q are all deterministic except for the proposal of the fine latent Gaussians from the coarse latent Gaussians, i.e. $q(\tilde{x},x^{'})$. The latent variables $x^{'}$ are generated from the marginal distribution $\pi(x^{'}|\theta^{\dagger})$, subject to the constraint that $Cx^{'} = \tilde{x}$, where $\theta^{\dagger}$ is a deterministic function of $\tilde{\theta}$. This is achieved by first generating $(n-\tilde{n})$ $x^{'}$'s from the $\pi(x^{'}|\theta^{\dagger})$, where $\tilde{n}$ is the number of coarse locations, then producing the remaining $\tilde{n}$ $x^{'}$ values such that

$$x^{'} = \tilde{x} - \sum_{l=1}^{L} x_{l}^{'},$$

where l indexes the fine values in each section of the coarse grid. This satisfies the constraint $Cx^{'} = \tilde{x}$. Also, the x's are proposed using the usual approximation within the marginal distribution, i.e. $\exp(\beta_{jt}x_t) \approx A_j + B_j(\beta_{jt}x_t) + \frac{1}{2}C_j(\beta_{jt}x_t)^2$.

Given that the entire update for both the coarse and fine chains is done in one step, the accept probability is then derived as follows:

$$\log(\pi(y|x^{'}\beta^{'})) = \sum_{j=1}^{r}\sum_{i=1}^{n}\left[-\exp(\beta^{'*}_{(j,-t)} + \beta^{'}_{jt}x^{'}_{it}) + (\beta^{'*}_{(j,-t)} + \beta^{'}_{jt}x^{'}_{it})y_{ij} - \sum_{f=1}^{y_{ij}}\log f\right]$$

$$\log(\pi(x^{'}|\theta^{'})) = \sum_{t=1}^{T}\left(-\frac{1}{2}\log|\Sigma^{'}_{t}| - \frac{1}{2}x_t^{'T}\Sigma_k^{'-1}x^{'}_t\right)$$

$$\log(\pi(\beta)) = \sum_{j=1}^{r}\left[-\frac{1}{2}\log(2\pi\tilde{\sigma}_j) - \frac{1}{2\tilde{\sigma}_j^2}\left(\sum_{t=0}^{T}(\beta_{jt} - \tilde{\mu}_j)^2\right)\right]$$

$$\log(\pi(\theta)) = -\log(\theta_U - \theta_L).$$

The β^* parameter is as given in the previous section, as are $\tilde{\sigma}_j^2$ and $\tilde{\mu}_j$. The probability distributions will take the same form for the coarse chain, and are indicated as such using a tilde. Then proposing x_t from its full conditional and using the approximation given in the previous section, the proposal probabilities are as follows:

$$\begin{aligned} q(x^{'}|x) &= \prod_{t=1}^{T}\prod_{j=1}^{r}\frac{|\Sigma_t^{\dagger-1} + C^{*}_{jt}\beta^{2}_{jt}|^{\frac{1}{2}}}{(2\pi)^{\frac{p}{2}}} \\ &\times\exp\left(-\frac{1}{2}x_{it}^{'T}\left(\Sigma_t^{\dagger-1} + C^{*}_{jt}\beta^{2}_{jt}\right)x^{'}_{it}\right) \\ &\times\exp\left(\left(y_{ij}\beta_{jt} - B^{*}_{jt}\beta_{jt}\right)^{T}x^{'}_{it}\right) \\ &\times\exp\left(\frac{1}{2}\left(y_{ij}\beta_{jt} - B^{*}_{jt}\right)^{T}\left(\Sigma_t^{\dagger-1} + C^{*}_{jt}\beta^{2}_{jt}\right)^{-1}\left(y_{ij}\beta_{jt} - B^{*}_{jt}\beta_{jt}\right)\right) \end{aligned}$$

and

$$\begin{aligned} q(x|x^{'}) &= \prod_{t=1}^{T}\prod_{j=1}^{r}\frac{|\Sigma_t^{-1} + C^{'*}_{jt}\beta^{2}_{jt}|^{\frac{1}{2}}}{(2\pi)^{\frac{p}{2}}} \\ &\times\exp\left(-\frac{1}{2}x_{it}^{T}\left(\Sigma_t^{-1} + C^{'*}_{jt}\beta^{2}_{jt}\right)x_{it}\right) \\ &\times\exp\left(\left(y_{ij}\beta_{jt} - B^{'*}_{jt}\beta_{jt}\right)^{T}x_{it}\right) \\ &\times\exp\left(\frac{1}{2}\left(y_{ij}\beta_{jt} - B^{'*}_{jt}\right)^{T}\left(\Sigma_t^{-1} + C^{'*}_{jt}\beta^{2}_{jt}\right)^{-1}\left(y_{ij}\beta_{jt} - B^{'*}_{jt}\beta_{jt}\right)\right), \end{aligned}$$

where

- $C^*_{jt} = \exp(\beta^*_{(j,-t)})\text{diag}(C_{1j}, \ldots, C_{nj})$ and $C^{'*}_{jt} = \exp(\beta^*_{(j,-t)})\text{diag}(C^{'}_{1j}, \ldots, C^{'}_{nj})$;
- $B^*_{jt} = \exp(\beta^*_{(j,-t)})(B_{1j}, \ldots, B_{nj})$ and $B^*_{jt} = \exp(\beta^*_{(j,-t)})(B^{'}_{1j}), \ldots, B^{'}_{nj})$.

Note that for the $q(.|.)$'s just given above, $i = (1, 2, \ldots, n - \tilde{n})$. To avoid confusion, let us use $\underline{i}$ to denote the reduced i index of the proposal function. Also, the order of the $\underline{i}$ index does not correspond to that of the original i index. The remaining $\tilde{n}$ latent values are generated deterministically, to fulfil the linear constraint $Cx^{'} = \tilde{x}$ as mentioned earlier. Then using the log of the proposal probabilities and reducing the notation by letting:

- $Q_t = \sum_{j=1}^{r} \left(\Sigma_t^{\dagger-1} + C^*_{jt}\beta^2_{jt}\right)$ and $Q^{'}_t = \sum_{j=1}^{r} \left(\Sigma_t^{-1} - C^{'*}_{jt}\beta^2_{jt}\right)$;
- $b_{ti} = \sum_{j=1}^{r}(y_{\underline{i}j} - B^*_{jt})\beta_{jt}$ and $b^{'}_{ti} = \sum_{j=1}^{r}(y_{\underline{i}j} - B^{'*}_{jt})\beta_{jt}$.

The proposal kernels for the β's and θ's are random walk proposals, so $\log(q(\beta|\beta^{'}) - q(\beta^{'}|\beta)) = 1$ and $\log(q(\theta^{\dagger}|\theta^{'}) - q(\theta^{'}|\theta^{\dagger})) = 1$. Another possibility is to use the conditional distribution of the field, i.e. conditional on the other x_i, rather than the marginal as we have used here.

The part of the log acceptance probability associated with the fine chain, using a proposal from the current coarse can then be written as:

$$\begin{aligned}
\log P(x^{'},\theta^{'}|y,\tilde{x},\tilde{\theta}) &= \frac{\pi(y|x^{'},\beta^{'})\pi(x^{'}|\theta^{'},\beta^{'}).\pi(\beta^{'})\pi(\theta^{'})q(\tilde{x},\theta^{\dagger}|x^{'},\theta^{'})}{\pi(y|x,\beta)\pi(x|\theta,\beta)\pi(\beta)\pi(\theta)q(x^{'},\theta^{'}|\tilde{x},\theta^{\dagger})} \\
&= \sum_{j=1}^{r}\sum_{i=1}^{n}\exp\left(\beta^{*}_{(j,-t)}+\beta_{jt}x_{it}\right)-\exp\left(\beta^{'*}_{(j,-t)}+\beta^{'}_{jt}x^{'}_{it}\right) \\
&\quad+\left(\beta^{'*}_{(j,-t)}+\beta^{'}_{jt}x^{'}_{it}\right)y_{ij}-\left(\beta^{*}_{(j,-t)}+\beta_{jt}x_{it}\right)y_{ij} \\
&\quad+\frac{1}{2\tilde{\sigma}^2_j}\left(\sum_{t=0}^{T}(\beta^{'}_{jt}-\tilde{\mu}_j)^2-(\beta_{jt}-\tilde{\mu}_j)^2\right) \\
&\quad+\sum_{t=1}^{T}\left(-\frac{1}{2}\left(\log|\Sigma^{'}_t|-\log|\Sigma_t|\right)-\frac{1}{2}\left(x^{'T}_{it}\Sigma^{'-1}_t x^{'}_{it}-x^{T}_{it}\Sigma^{-1}_t x_{it}\right)\right) \\
&\quad+\frac{1}{2}\left(\log|Q^{'}_t|-\log|Q_t|\right)+\left(\frac{1}{2}(x^{'T}_{t\underline{i}}Q_t x^{'}_{t\underline{i}})-\frac{1}{2}(x^{T}_{t\underline{i}}Q^{'}_t x_{t\underline{i}})\right) \\
&\quad+\left(b^{'}_{t\underline{i}}x_{t\underline{i}}-b_{t\underline{i}}x^{'}_{t\underline{i}}\right)+\frac{1}{2}\left(b^{'}_tQ^{'}_tb^{'}_t-b_tQ_tb_t\right).
\end{aligned}$$

There is little cancellation between the proposal and the probability distributions in this case. The proposal distribution arises from a marginal distribution of size $n-\tilde{n}$, whereas the probability distribution is of dimension n. The log accept probability associated with the coarse part of the chain is derived in the same fashion as the fine, except the proposals are deterministic. This part of the log accept is then given as:

$$\begin{aligned}
\log P(\tilde{x}^{'},\tilde{\theta}^{'}|y,x,\theta) &= \frac{\pi(\tilde{y}|\tilde{x},\tilde{\beta}^{'})\pi(\tilde{x}^{'}|\tilde{\theta}^{'},\tilde{\beta}^{'})\pi(\tilde{\beta}^{'})\pi(\tilde{\theta}^{'})q(x,\theta|\tilde{x}^{'},\tilde{\theta}^{'})}{\pi(y|\tilde{x},\tilde{\beta})\pi(\tilde{x}|\tilde{\theta},\tilde{\beta})\pi(\tilde{\beta})\pi(\tilde{\theta})q(\tilde{x}^{'},\tilde{\theta}^{'}|x,\theta)} \\
&= \sum_{j=1}^{r}\sum_{i=1}^{\tilde{n}}\exp\left(\tilde{\beta}^{*}_{(j,-t)}+\tilde{\beta}_{jt}\tilde{x}_{it}\right)-\exp\left(\tilde{\beta}^{'*}_{(j,-t)}+\tilde{\beta}^{'}_{jt}\tilde{x}^{'}_{it}\right) \\
&\quad+\left(\tilde{\beta}^{'*}_{(j,-t)}+\tilde{\beta}^{'}_{jt}\tilde{x}^{'}_{it}\right)\tilde{y}_{ij}-\left(\tilde{\beta}^{*}_{(j,-t)}+\tilde{\beta}_{jt}\tilde{x}_{it}\right)\tilde{y}_{ij} \\
&\quad+\frac{1}{2\tilde{\tilde{\sigma}}^2_j}\left(\sum_{t=0}^{T}(\tilde{\beta}^{'}_{jt}-\tilde{\tilde{\mu}}_j)^2-(\tilde{\beta}_{jt}-\tilde{\tilde{\mu}}_j)^2\right) \\
&\quad+\sum_{t=1}^{T}\left(-\frac{1}{2}\left(\log|\tilde{\Sigma}^{'}_t|-\log|\tilde{\Sigma}_t|\right)-\frac{1}{2}\left(\tilde{x}^{'T}_{it}\tilde{\Sigma}^{'-1}_t\tilde{x}^{'}_{it}-\tilde{x}^{T}_{it}\tilde{\Sigma}^{-1}_t\tilde{x}_{it}\right)\right).
\end{aligned}$$

Then the overall acceptance probability for the proposed swap of information between the coarse and fine chain will be the product of the probabilities just derived

$$\log(P(x^{'},\theta^{'},\tilde{x}^{'},\tilde{\theta}^{'}|y,x,\theta,\tilde{x},\tilde{\theta})) = \log P(x^{'},\theta^{'}|y,\tilde{x},\tilde{\theta},x,\theta) + \log P(\tilde{x}^{'},\tilde{\theta}^{'}|\tilde{y},x,\theta).$$

Appendix B

Additional Diagnostics

B.1 Interactions

In an experiment (such as that carried out in Chapter 5) the difference in response between levels of a factor may not be the same at all levels of the other factors in the experiment. When this occurs, there is said to be an "interaction" between factors. The graphs given below are useful in interpreting significant interactions. Note that when an interaction is large, the corresponding main effect have little practical meaning. Also, a significant interaction can often mask the significance of main effects.

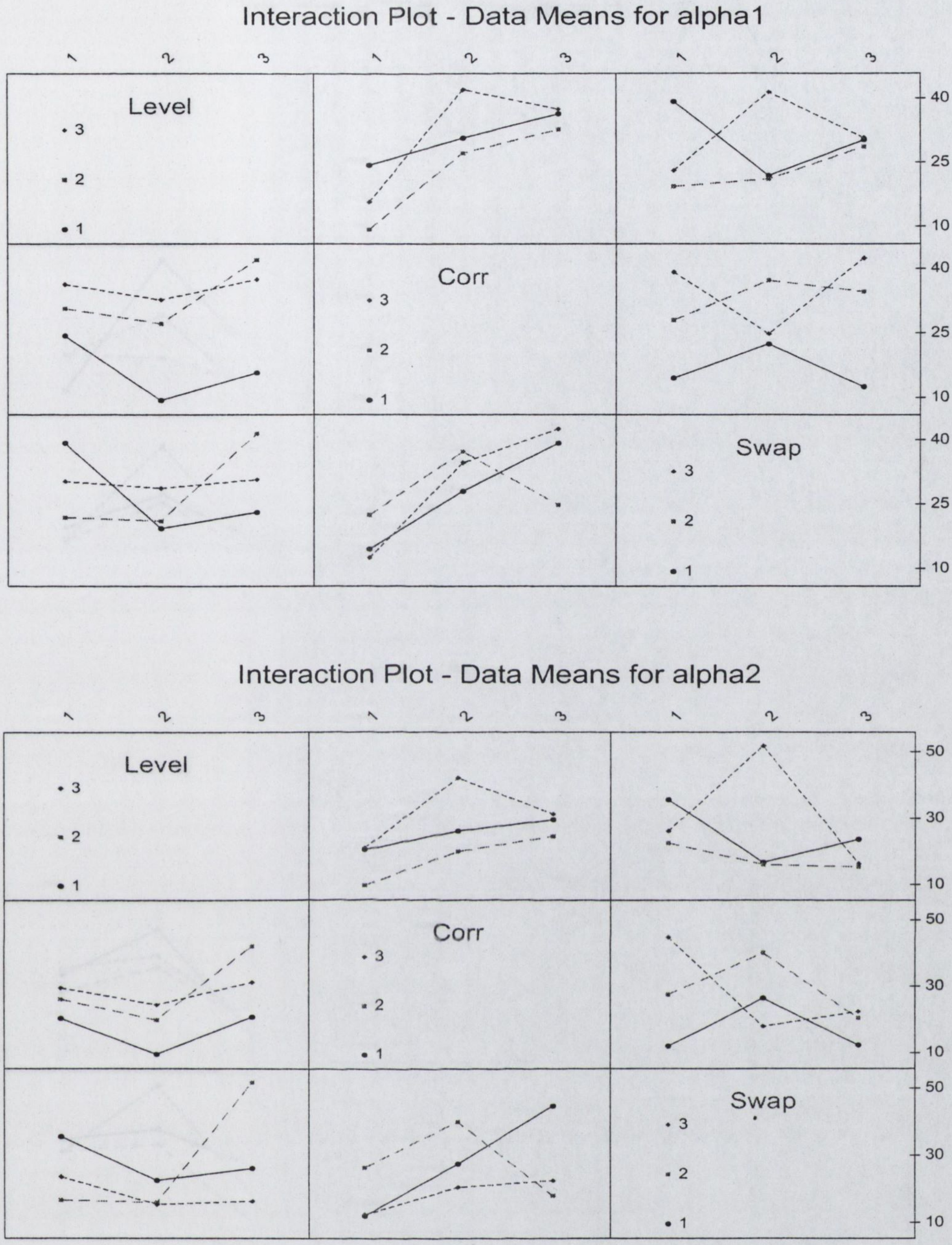
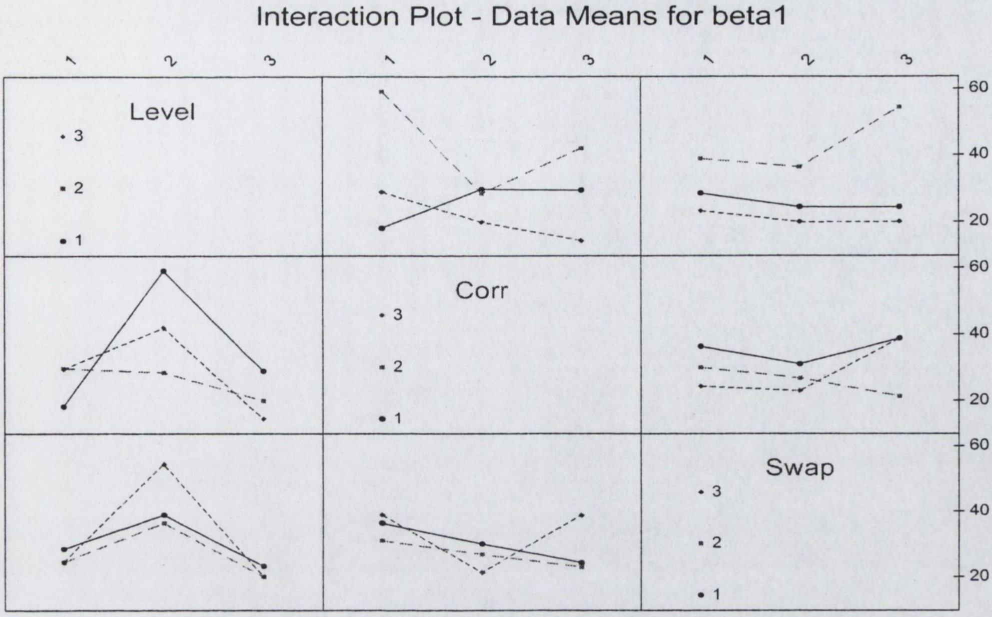


Figure B.1: Positions 1, 2 and 3 on each of the graphs indicating the various levels of coarsening, correlation and proposed rate of swap. The scale on the right-hand side is a maximum likelihood estimate for τ, i.e. effective sample size $= N/\tau$ and N is the number of sample taken. The smaller the value of τ the better.



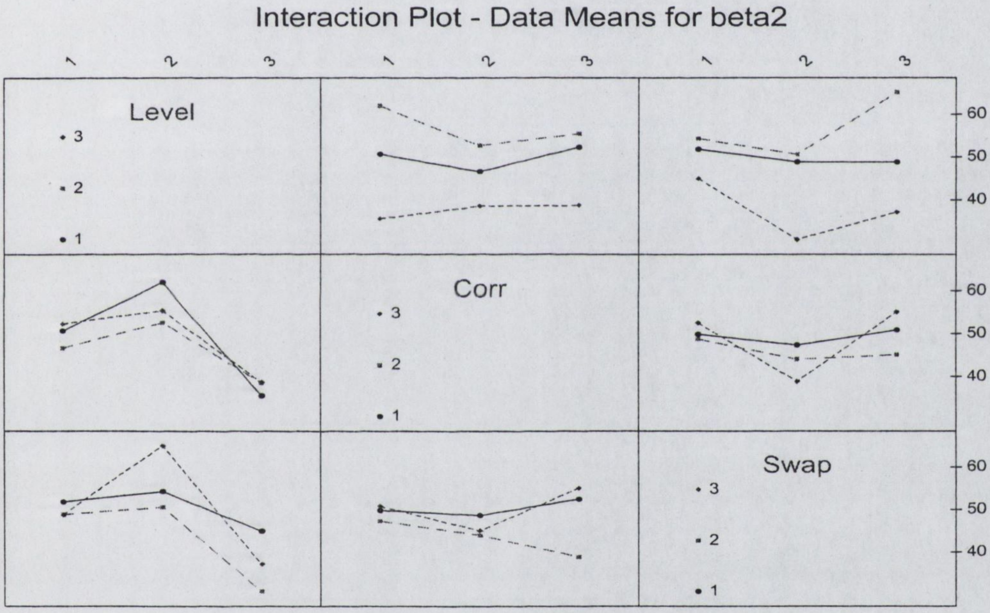
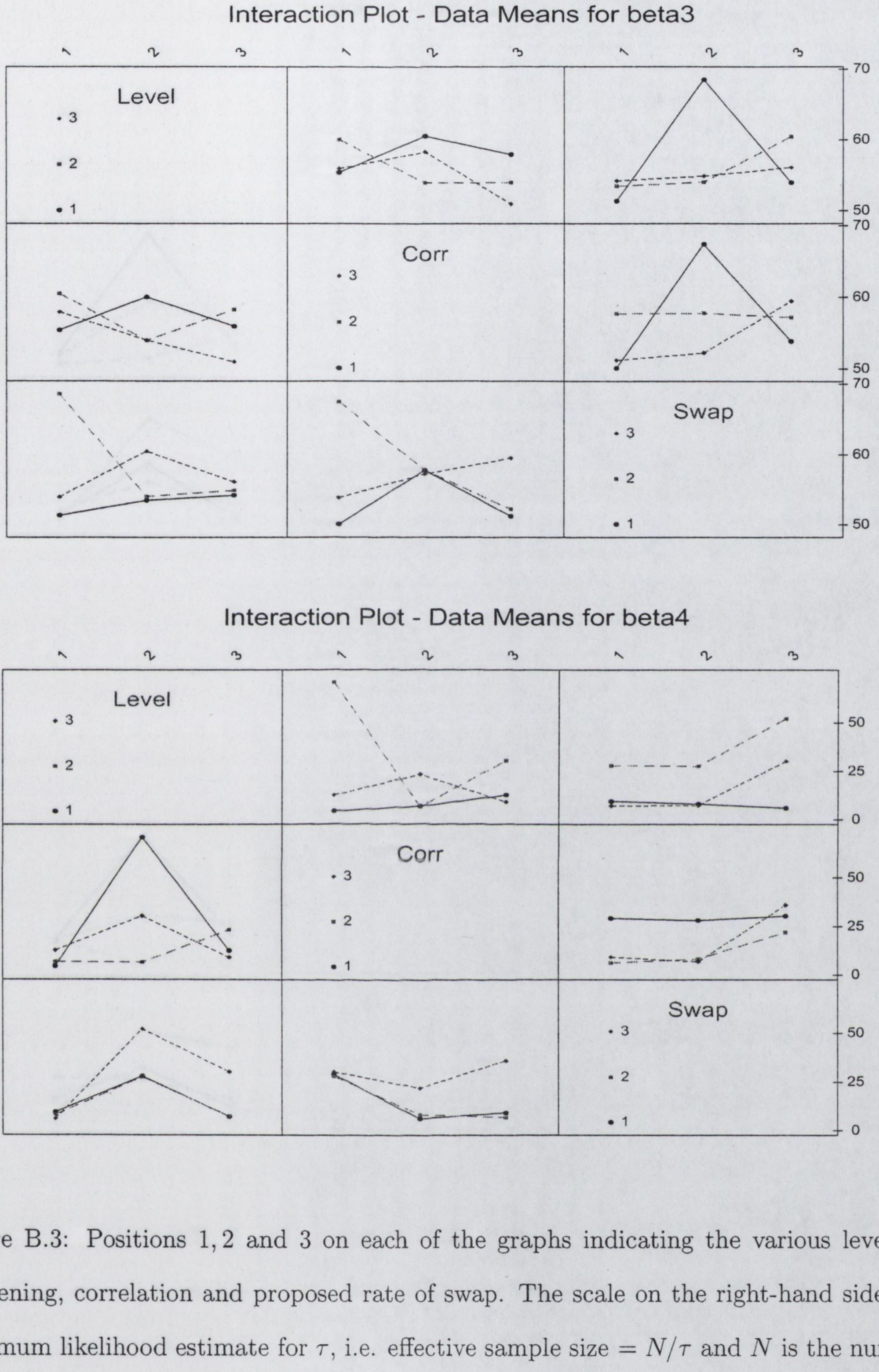


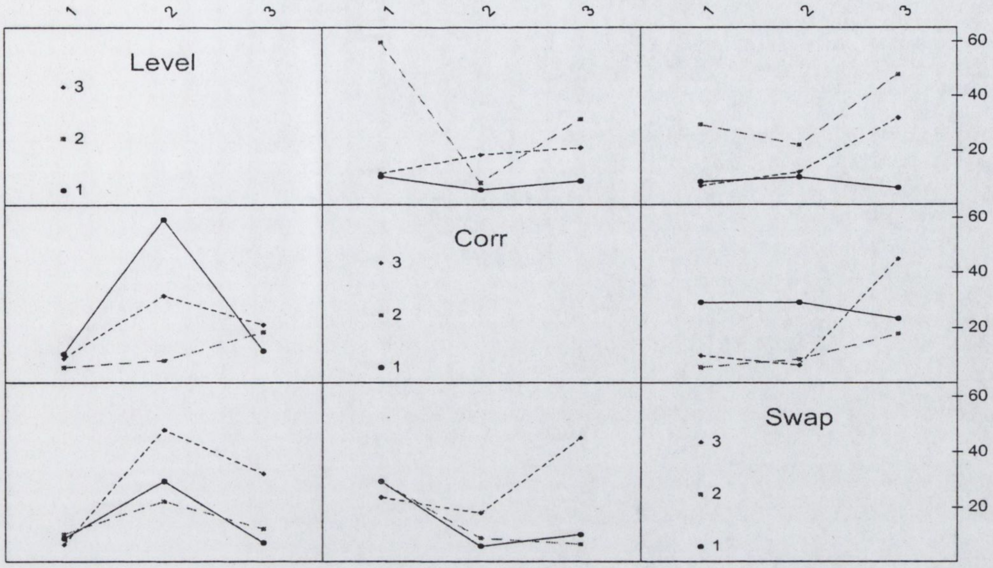
Figure B.2: Positions 1, 2 and 3 on each of the graphs indicating the various levels of coarsening, correlation and proposed rate of swap. The scale on the right-hand side is a maximum likelihood estimate for τ, i.e. effective sample size $= N/\tau$ and N is the number of sample taken. The smaller the value of τ the better.


Interaction Plot - Data Means for beta3

Interaction Plot - Data Means for beta4

Figure B.3: Positions 1, 2 and 3 on each of the graphs indicating the various levels of coarsening, correlation and proposed rate of swap. The scale on the right-hand side is a maximum likelihood estimate for τ, i.e. effective sample size $= N/\tau$ and N is the number of sample taken. The smaller the value of τ the better.

Interaction Plot - Data Means for beta5



Interaction Plot - Data Means for beta6

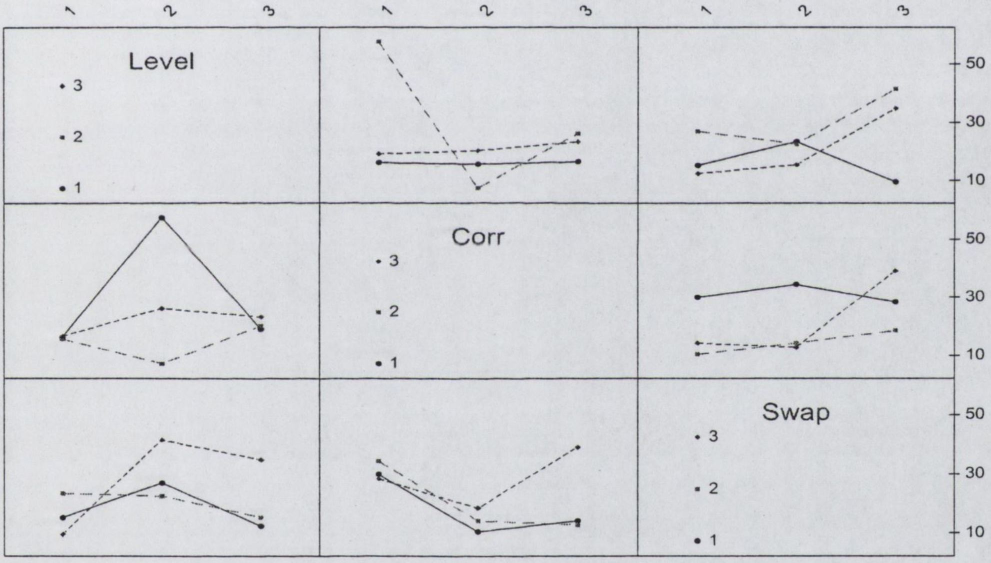


Figure B.4: Positions 1, 2 and 3 on each of the graphs indicating the various levels of coarsening, correlation and proposed rate of swap. The scale on the right-hand side is a maximum likelihood estimate for τ, i.e. effective sample size $= N/\tau$ and N is the number of sample taken. The smaller the value of τ the better.

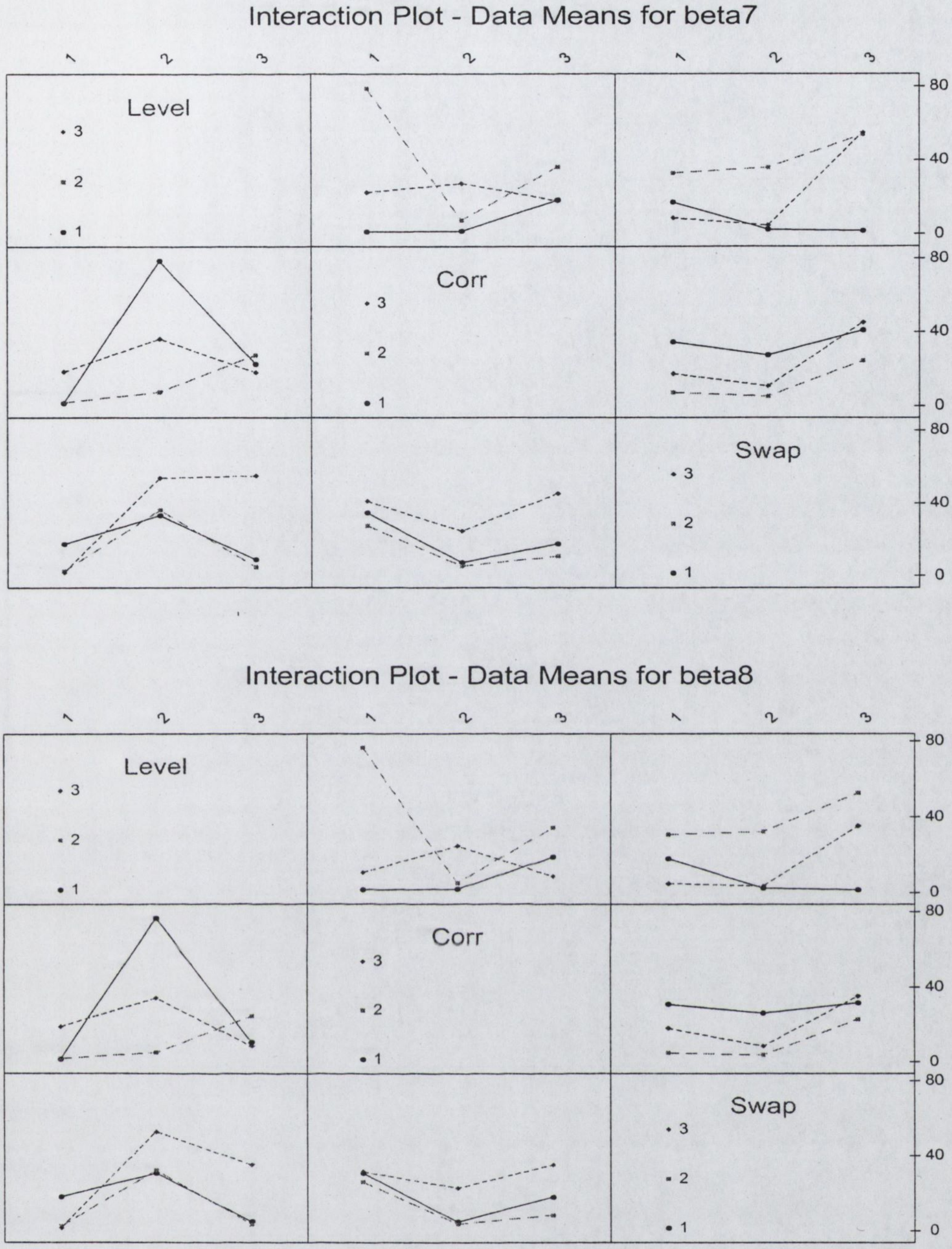


Figure B.5: Positions 1, 2 and 3 on each of the graphs indicating the various levels of coarsening, correlation and proposed rate of swap. The scale on the right-hand side is a maximum likelihood estimate for τ, i.e. effective sample size $= N/\tau$ and N is the number of sample taken. The smaller the value of τ the better.

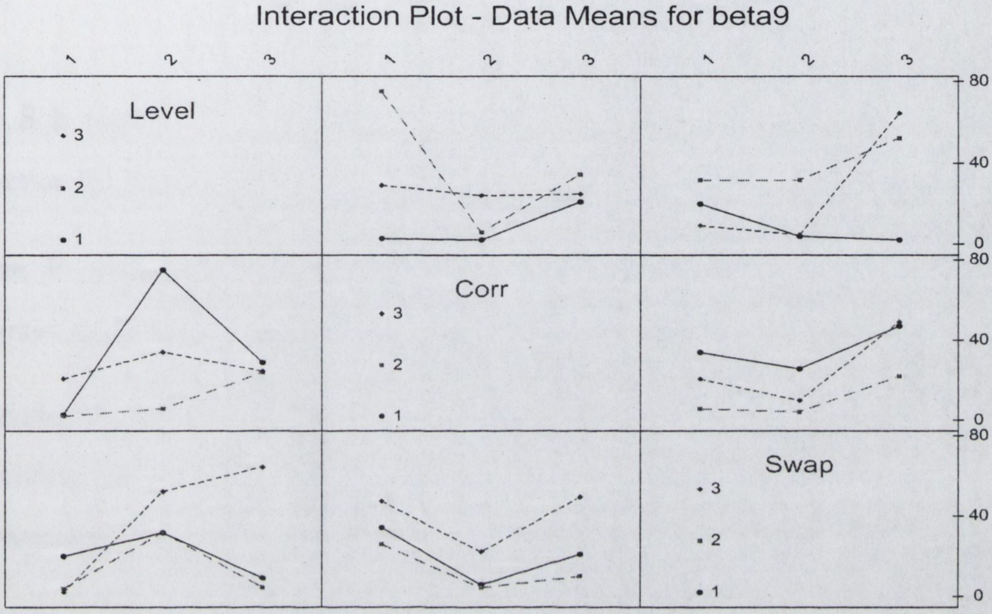


Figure B.6: Positions 1, 2 and 3 on each of the graphs indicating the various levels of coarsening, correlation and proposed rate of swap. The scale on the right-hand side is a maximum likelihood estimate for τ, i.e. effective sample size $= N/\tau$ and N is the number of sample taken. The smaller the value of τ the better.

Bibliography

Adler, S. L. (1981). Over-relaxation method for the Monte Carlo evaluation of the partition function for multiquadratic actions. *Physical Review D 23*, 2901–2904.

Barone, P., G.Sebastiani, and J. Stander (2002). Over-relaxation methods and coupled Markov chain for Monte Carlo simulation. *Statistics and Computing 12*, 17–26.

Basseville, M., A. Benveniste, K. Chou, S. Golden, R.Nikoukhan, and A. Willsky (1992). Modeling and estimation of multiresolution stochastic processes. *IEEE Transactions on Information Theory 38*, 766–784.

Besag, J., J. York, and A. Mollie (1991). Bayesian image restoration, with two applications in spatial statistics (with discussion). *Ann. Inst. Statist. Math. 43*, 1–59.

Brooks, S. (1998). Qualitative convergence diagnostics for MCMC via cusums. *Statistics and Computing. 8*, 267–274.

Brooks, S. and G. Roberts (1998). Convergence assessment techniques for Markov chain Monte Carlo. *Statistics and Computing. 8*, 319–335.

Carlin, B. P. and T. A. Louis (1996). *Bayes and emprirical Bayes methods for data analysis.* London: Chapman and Hall.

Cowles, M. and B. Carlin (1996). Markov chain Monte Carlo convergence diagnostics: A comparative review. *J. Amer. Statist. Assoc. 91*, 883–904.

Cressie, N. (2001). *Statistics for spatial data.* New York: Wiley.

Dale, A. M. (1991). *A history of inverse probability.* New York: Springer.

Diggle, P. J., J. A. Tawn, and R. A. Moyeed (1998). Model-based geostatistics (with discussion). *Applied Statistics 47*, 299–350.

Frantz, D., D. L. Freeman, and J.D.Doll (1990). Reducing quasi-ergodic behaviour in Monte Carlo simulations by J-walking: applications to atomic clusters. *J.Chem. Phys 93*, 2769–2784.

Gammerman, D. (1997). Sampling from posterior distribution in generalized mixed models. *Statistics and Computing 7*, 57–68.

Garren, S. and R. Smith (2000). Estimating the second largest eigenvalue of a Markov transition matrix. *Bernoulli 6*, 215–242.

Gelfand, A. E. and D. Rubin (1992). Inference for iterative simulation using multiple sequences. *Statist. Sci. 7*, 457–472.

Gelfand, A. E. and A. F. M. Smith (1990). Sampling based approaches to calculating marginal densities. *J. Am. Statist. Assoc. 85*, 398–409.

Gelman, A., J. B. Carlin, H. S. Stern, and D. B. Rubin (1995a). *Bayesian data analysis.* London: Chapman and Hall.

Gelman, A., G. Roberts, and W. Gilks (1995b). Efficient Metropolis jumping rules. In J. Bernardo, J. Berger, A. Dawid, and A. Smith (Eds.), *Bayesian Statistics 5.* Oxford University Press.

Geman, S. and D. Geman (1984). Stochastic relaxation, Gibb's distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Machine Intell. PAMI-6*, 721–741.

Geyer, C. (1991). MCMC maximum likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, Fairfax Station: Interface Foundation, pp. 156–163.

Geyer, C. J. and E. Thompson (1995). Annealing Markov chain Monte Carlo with applications to ancestral inference. *J. Amer. Statist. Assoc. 90*, 909–920.

Gilks, W. and G. Roberts (1996). Strategies for improving MCMC. In W. R. Gilks, S. Richardson, and D. J. Spiegelhalter (Eds.), *Markov chain Monte Carlo in practice*, Chapter 6, pp. 89–114. Chapman and Hall.

Gilks, W. R., S. Richardson, and D. J. Spiegelhalter (1996). *Markov chain Monte Carlo in practice.* London: Chapman and Hall.

Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika 82*, 711–732.

Grimmett, G. and D. Stirzaker (1982). *Probability and random processes.* New York: Oxford University Press.

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

Higdon, D., H. Lee, and Z. Bi (2002). A Bayesian approach to characterizing uncertainty in inverse problems using coarse and fine-scale information. *IEEE Transactions on Signal Processing 50*, 389–399.

Holloman, C., H. Lee, and D. Higdon (2002). Multi-resolution Genetic algorithms and Markov chain Monte Carlo. Technical report, Duke University, ISDS, Box 90251, Duke University, Durham, NC 27708, USA. Email: chris@stats.duke.edu.

Kendal, W. (1997). Perfect simulation for the area interaction point process. In C. Heyde and L. Accardi (Eds.), *Probability Towards 2000*, pp. 218–234. Springer.

Knorr-Held, L. and H. Rue (2002). On block updating in Markov random field models for disease mapping. *Scand. J. Stat. 29*, 597–614.

Liu, J. S. (1994). The collapsed Gibbs sampler in Bayesian computions with applications to a gene regulation problem. *Journal of the American Statistical Association 89*, 958–966.

Liu, J. S., W. H. Wong, and A. Kong (1994). Covariance structure of the Gibbs sampler with application to the comparisons of estimators and augmentation schemes. *Biometrika 81*, 27–40.

Matern, B. (1986). *Spatial Variation.* Heidelberg: Springer-Verlag.

Mengersen, K., C. Robert, and C. Guihenneuc-Jouyaux (1999). MCMC convergence diagnostics: A reviewww (with discussion). In J. Bernardo, J. Berger, A. Dawid, and A. Smith (Eds.), *Bayesian Statistics 6*, pp. 415–440. Oxford University Press.

Metropolis, N., A. Rosenbluth, M. N. Rosenbluth, A. Teller, and E. Teller (1953). Equations of state calculations by fast computing machines. *J. Chemical Physics 21*, 1087–1092.

Metropolis, N. and S. Ulam (1949). The Monte Carlo method. *J. Amer. Statist. Assoc. 44*, 335–341.

Mira, A. (2001). Ordering and improving the performance of MCMC. *Statistical Science 16*, 340–350.

Møller, J. and R. Waagerpetersen (2003). *Statistical inference and simulation for spatial point processes.* London: Chapman and Hall.

Neal, R. M. (1993). Probabilistic inference using Markov chain Monte Carlo methods. Technical Report CRG-TR-93-1, Department of Computer Science, Toronto, E-mail: radford@cs.toronto.edu.

Peskun, P. (1973). Optimum Monte Carlo sampling using Markov chains. *Biometrika 60*, 607–612.

Pinto, R. and R. Neal (2001). Improving Markov chain Monte Carlo estimators by coupling to an approximating chain. Technical Report 0101, University of Toronto, Department of Statisitcs, University of Toronto, Toronto, Canada, M5S 3G3. Email: ruxandra@utstat.toronto.edu.

Press, W. H., B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling (1986). *Numerical recipes: the art of scientific computing.* New York: Cambridge University Press.

Propp and Wilson (1996). Exact sampling with coupled markov chains and applications to statistical mechanics. *Random Structures and Algorithms 9*, 223–252.

Ripley, B. D. (1981). *Spatial statistics.* New York: Wiley.

Ripley, B. D. (1987). *Stochastic Simulation.* New York: Wiley.

Ripley, B. D. (1988). *Statistical inference for spatial processes.* New York: Cambridge University Press.

Roberts, G. (1992). Convergence diagnostics of the Gibbs sampler. In J. Bernardo, J. Berger, A. Dawid, and A. Smith (Eds.), *Bayesian Statistics 4*, pp. 775–782. Oxford University Press.

Roberts, G. (1995). Chapter 3. In W. Gilks, S. Richardson, and D. Spiegelhater (Eds.), *Markov chain Monte Carlo in Practice*, pp. 48–49. Chapman and Hall.

Roberts, G. and R. Tweedie (1996). Exponential convergence of Langevin diffusions and their discrete approximations. *Bernoulli 2*, 341–364.

Rue, H. (2001). Fast sampling of Gaussian Markov random fields. *J. R. Statist. Soc. B 63*, 325–338.

Rue, H., I. Steinsland, and S. Erland (2004). Approximating hidden Gaussian Markov random fields. *J. R. Statist. Soc. B 66*, 877–892.

Rue, H. and H. Tjelmeland (2002). Fitting Gaussian Markov random fields to Gaussian fields. *Scand. J. Statist. 29*, 30–48.

Silverman, B. (1986). *Density estimation for statistics and data analysis* (First ed.). New York: Chapman and Hall.

Stigler, S. M. (1986). *The history of statistics: The measurement of uncertainty before 1900.* Cambridge Massachusetts: Harvard University Press.

Student (1908). The probable error of a mean. *Biometrika 6*, 1–25.

Swendsen, R. and J. Wang (1987). Nonuniversal critical dynamics in Monte Carlo simulation. *Physical Rewiew Letters 58*, 86–88.

Whiley, M., J. Haslett, S. Bhattacharya, M. S. Townshend, S. Wilson, J. Allen, B. Huntley, and F. Mitchell (to appear). Bayesian palaeoclimate reconstruction. *J. Roy. Statist. Soc. A*.

Whiley, M. and S. P. Wilson (2004). Parallel algorithms for Markov chain Monte Carlo methods in latent spatial Gaussian models. *Statistics and Computing 14*, 171–179.

Whittaker, J. (1990). *Graphical Models in applied multivariate statistics.* New York: Wiley.

Yu, B. and P. Mykland (1998). Looking at Markov samplers through cusum path plots: A simple diagnostic idea. *Statist. Comp. 8*, 275–286.

Zeger, S. and M. Karim (1991). Generalized linear models with random effects: A Gibbs sampling approach. *J. Amer. Statist. Assoc. 86*, 79–86.