UNIVERSITY OF MINNESOTA

This is to certify that I have examined this copy of a doctoral thesis by

Murali Haran

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Luke Tierney/Brad Carlin

Name of Faculty Adviser(s)

Signature of Faculty Adviser(s)

Date

GRADUATE SCHOOL

Efficient Perfect and MCMC Sampling Methods for Bayesian Spatial and Components of Variance Models

A THESIS
SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL
OF THE UNIVERSITY OF MINNESOTA
BY

MURALI HARAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

Luke Tierney, Adviser
Brad Carlin, Adviser

June 2003

Acknowledgments

I am very grateful to the faculty of the School of Statistics for providing me with a very nice environment for learning and research. I have always felt at home and encouraged, and over the years have become good friends with several faculty members. My thesis committee members, Luke Tierney, Brad Carlin, Glen Meeden and Charlie Geyer, have all had (in very different ways) an enormous, positive impact on my life as a graduate student.

I thank Luke Tierney for being an excellent advisor. He has been a great source of ideas and insights for my thesis, and his laid-back style has encouraged me to take my time to learn and mature as a researcher. In particular, I thank him for suggesting key ideas related to producing approximations for marginal posterior distributions, as well as for pointing me to the Møller and Nicholls paper on perfect tempering. His careful reading and comments on my thesis were most helpful. I am also very grateful to him for providing financial support during my studies here.

Brad Carlin has been a terrific advisor — not only has he provided me with the main inspiration related to working on computing for spatial models, which has formed the basis of a lot of my dissertation research, but he has also encouraged me to write papers and to present my work at conferences. In addition to learning about spatial modeling and computation from Brad, I have also learnt about working with people from other departments (while working on a project on environmental data), and received invaluable guidance on writing and presentation. He has always been generous with his time, and has provided me with technical help, along with lots of career advice. I have also downed numerous beers at conferences at his expense, for

which I will always be grateful.

Glen Meeden was a great Director of Graduate Studies during my early years here. He has often had plenty of words of wisdom for me, but more importantly, I have always come away from my meetings with him feeling like there was never any *real* reason to worry or take anything too seriously. I have enjoyed lots of good conversation with him, ranging from weighty discussions on politics, music, film, and research, to humorous ramblings on just about anything else. The department parties at Glen's home have always been fun as well. I would also like to thank Gary Oehlert for having been a supportive D.G.S. during the latter half of my tenure, and for hosting numerous parties where I have shamelessly eaten vast quantities of good food.

Charlie Geyer has been a fantastic resource to me from the very beginning of my graduate studies here. It has always been a great comfort to know that his door has always been open when I have needed help with anything from computing to real analysis to getting basic ideas about Markov chains straightened out. His knowledge of many different statistical areas, and his strong opinions about "the Right Thing to do" and his unique way of thinking about the big picture have all had an enormous impact on the way I think about statistics. I have also acquired a new appreciation for jazz by digging into his vast music collection, and we have enjoyed numerous live jazz and Indian classical music performances together over the years. Charlie has been a great teacher and friend to me.

I would like to thank Jim Hodges for providing a lot of the ideas that went into the work in Chapter 2. I have had fun collaborating with him.

I am grateful to Galin Jones for providing me with insights on regenera-

tion in MCMC, suggestions for interesting papers to read, and general advice on research and job search strategies. Galin has been a good friend and a terrific addition to the department in general, and it was great to have him here during the last two years of my Ph.D.

Many thanks to the staff of the School of Statistics who have been extremely helpful to me. Jane Sell, Dana Tinsley, Mary Hildre and Lavone Johnson have all been wonderful — I cannot think of a single mistake with any of the thousands of documents and paperwork they have had to deal with on my account (though I do recall several occasions when they have rescued me from my own absent-mindedness). Thanks also to Seth Mayotte and Marisa Riviere for a largely problem-free computing environment.

The people who have had the biggest impact on my graduate school life have been my fellow students. An incomplete list of friends who have made the whole graduate school experience enriching and fun is: Vera Bulaevskaya (my classmate through the "thick and thin" of core course work and everything else), Laura Pontiggia, Matt Gregas, Francisca Winston, Jennifer Sartorius, Maurizio Tiso, Jessica Stoering, Todd Watts, Liqiang Ni, Rong Yang, Lexin Li, Chris Cook, Ron Neath, Despina Stefan, Tom Azeredo, Michael Peascoe, Lou Sherfesee, Christopher Wiggenhorn and Iain Pardoe.

Above all, I value the love and encouragement of my parents, E.G.P. and Indira, and sister, Meera, and my grandparents. In spite of being thousands of kilometres/miles away, they have always found a way to provide me with wonderful (and crucial) family support.

I dedicate this thesis to the memory of my grandmother, Thathi.

Abstract

Bayesian hierarchical models give rise to complicated posterior distributions. Monte Carlo and Markov chain Monte Carlo (MCMC) methods are used to estimate expectations with respect to these complicated distributions. When there is enough structure in the problem, it is often possible to design algorithms that are much more efficient than "off-the-shelf" MCMC algorithms. This thesis investigates and develops efficient Monte Carlo and MCMC algorithms for inference for some important Bayesian hierarchical models.

A major focus of the work presented here is on studying computation for Bayesian models used for modeling areal (spatially aggregated) data via spatial Poisson models (Besag, York and Mollie 1991). These models use Gaussian Markov random fields to model spatial correlation in disease mapping applications. The usual Gibbs sampling and univariate-update Markov chain Monte Carlo methods exhibit very poor convergence and mixing properties for these models. The work here describes various systematic MCMC block sampling techniques to solve this problem.

While block sampling methods are often effective, the theoretical work needed to rigorously assess the accuracy of MCMC based estimates is prohibitively difficult for most realistic problems, and hence ad hoc "convergence diagnostics" techniques are typically used in practice. If the samples drawn are i.i.d. rather than dependent, these issues are easily resolved; however, i.i.d. or exact sampling methods are generally not considered to be practical for complicated, multivariate continuous distributions. This thesis proposes a systematic method for producing heavy tailed proposal distributions that can be used in exact sampling schemes for Bayesian disease mapping models.

These proposal distributions allow the implementation of the rejection sampling and perfect tempering (Moller and Nicholls, 1999) algorithms to draw i.i.d. samples from the posterior distributions of interest. Exact simulation algorithms are also studied for some Bayesian variance components models, and their application to the widely used Bayesian one-way ANOVA model is successfully demonstrated.

The thesis concludes with a discussion of ideas for further automating the exact simulation methods developed here, along with possible extensions of exact sampling to more complicated models.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Monte Carlo

Monte Carlo integration is a general and powerful technique for approximating intractable expectations with respect to a known distribution. The method estimates these expectations by using appropriate averages of computer simulated random variates with this distribution. Consider the problem of estimating properties of a probability distribution, F on a space $\mathcal{X}$. In other words, the problem is to to calculate $E(g(x))$ w.r.t. distribution F.

$$E(g(x)) = \int_{\mathcal{X}} g(x) dF(x) dx$$

for some integrable function g. There are several numerical or analytical approximations available for such integrals, but the focus of this thesis is on Monte-Carlo methods. Suppose it is possible to draw pseudo-random samples $X_1, \cdots, X_n$ from F. By the strong law of large numbers, with probability 1,

$$\bar{g}_n = \frac{\sum_{i=1}^{n} g(X_i)}{n} \to E(g) \text{ as } n \to \infty.$$

If $E(g^2)$ is finite, there is a central limit theorem for $\bar{g}_n$, i.e.,

$$\sqrt{n}(\bar{g}_n - E(g)) \xrightarrow{d} N(0, \sigma^2)$$

and the usual sample variance is a consistent estimate of σ^2.

The difficulty with Monte Carlo methods lies in developing efficient algorithms for generating samples with distribution F when F is complicated. Algorithms such as rejection (accept-reject) sampling produce independent, identically distributed random variates, and work well for a large number of problems. Rejection sampling is generally believed to be limited in its ability to handle high dimensional, continuous distributions, though in Chapters 4 and 5 some important situations are described where rejection sampling can be practical. However, in practice, most people resort to Markov chain based algorithms for simulating draws from such complicated distributions.

1.2 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) methodology originated in the statistical physics literature over forty years ago, and has been used in spatial statistics and image analysis for two decades. However, only in the last fifteen years has MCMC methodology penetrated mainstream statistical practice, thereby opening up whole new areas of realistic statistical modeling by allowing for the simulation of random variables from complicated distributions. While the general framework for the Metropolis-Hastings algorithm for constructing Markov chains has been around since Metropolis et al. (1953) and Hastings (1970), it was not until the papers by Geman and Geman (1984),

Tanner and Wong (1987) and Gelfand and Smith (1990), that MCMC methods became widely used in realistic Bayesian modeling problems. Of course, since MCMC is a technique for computing probabilities when analytical procedures fail, the applications are not confined to Bayesian computations. Instances of non-Bayesian applications include hypothesis testing and likelihood inference. For an introduction to these applications, see Geyer (1995) and the references therein.

A brief description of the Metropolis-Hastings algorithm, which forms the basis for all MCMC algorithms, is provided here. The widely used Gibbs sampler and the Metropolis algorithm are both special cases of the Metropolis-Hastings algorithm. The algorithm is a way to generate a Markov chain with F as its stationary distribution (as before, F is the distribution of interest, and let F be dominated by a measure ν). The algorithm requires a proposal distribution, Q (with respect to the same dominating measure ν), and this proposal could depend on the current value of the chain. The Metropolis-Hastings algorithm to generate a Markov chain, $\{X_0, X_1, X_2, \dots\}$ with stationary density F, using a proposal density Q, proceeds as follows.

Metropolis-Hastings Algorithm

1. Set the state of the Markov chain, X_0 to any initial value in the support of the distribution F. Set $i = 0$.

2. Draw $X^* \sim Q(X_i, X)$ where X_i is the current state of the chain.

3. Compute the ratio

$$r(X_i, X^*) = \frac{F(X^*)Q(X^*, X_i)}{F(X)Q(X_i, X^*)},$$

and set $\alpha(X_i, X^*) = \min(r, 1)$.

4. Accept the proposed draw with probability α. This is done by drawing $U \sim \text{Unif}(0, 1)$. If $U < \alpha(X_i, X^*)$, set $X_{i+1} = X^*$, else set $X_{i+1} = X_i$.

5. Set i to $i + 1$ and return to Step 2.

Under mild conditions, $X_i \to X \sim F$. For a proof of this, see Tierney (1994).

Since each new state of the Markov chain can be a vector of values, each update can be obtained by proposing a new value for either a single component of the vector, or for several components of the vector at the same time (in "blocks"), or for the entire vector at once. If the updates are done for just some of the components of the vector at a time, several terms in the ratio r will cancel, and r will then depend on the posterior distributions conditional on the components of the vector that remain fixed for that update. If these conditional distributions happen to be known distributions that are easy to sample from, the corresponding proposal for the components of the vector can then simply be this known distribution. It is easy to see then that the ratio r will be 1 for such updates, i.e., these proposals are always accepted. This specific case of the Metropolis-Hastings algorithm is refered to as a **Gibbs sampler**. Of course, it is also possible, in some simple examples, to have fully blocked Gibbs samplers. In other words, the joint distribution of the vector is a known distribution, and each proposal for the entire vector is always accepted. The **Metropolis algorithm** is simply the Metropolis-Hastings algorithm with a symmetric proposal (so $Q(X_i, X^*) = Q(X^*, X_i)$) so that the acceptance ratio simplifies to $r(X_i, X^*) = F(X^*)/F(X)$.

1.3 Block Updating Schemes

Samples are often drawn from posterior distributions via standard procedures such as univariate MCMC and Gibbs sampling. Univariate MCMC methods involve updating the state of the Markov chain by changing a single parameter at a time. When the distributions are of high dimensions and have highly correlated parameters, as is often the case with spatial models, the univariate MCMC algorithms become highly inefficient. This inefficiency is because the Markov chains do not explore the state space rapidly enough, thereby producing highly autocorrelated dependent states (samples). This is also closely related to the fact that such Markov chains typically converge towards the stationary distribution very slowly. Accelerating or improving the "mixing" of such chains is an area that has generated a tremendous amount of research in the past decade, and there have been many techniques proposed to achieve improved mixing. Block sampling, or the changing of several parameters of the Markov chain simultaneously, is a well known method for improving the mixing of Markov chains. The difficulty of implementing block updating schemes is that one needs a reasonably good understanding of the correlation among the parameters within each block; without a good estimate of this, it becomes very difficult to get the Markov chains to accept any proposed moves to new states. It is therefore hard to find systematic methods for producing practical block sampling algorithms for hierarchical models. In Chapter 2, a variety of systematic block sampling algorithms for a Bayesian spatial model are investigated, including varying block sizes, updating some parameters more often than others, and so on. In addition to studying how efficiently the Markov chain is mixing, the amount of computing time taken

up by the algorithms is also taken into account. This is an important practical consideration, since sophisticated algorithms may produce Markov chains that mix rapidly, but the price in terms of computational time may be too heavy to make them useful.

While block sampling MCMC techniques are often effective, one still has to worry about MCMC convergence and mixing diagnostics when selecting a sampler that appears to provide the most accurate results. This is a difficult problem, with a variety of solutions proposed - the most rigorous theoretical solutions prove to be impractical for anything but the simplest situations, while for more difficult situations, where good diagnostics are much more important, fairly ad hoc methods are generally used. Cowles and Carlin (1996) point out that all of these proposed methods can fail to detect non-convergence problems, even for relatively simple problems. In fact, with all MCMC methods, one always needs to worry about important issues that do not arise with ordinary Monte Carlo methods:

- How long before the chain is sufficiently close to stationarity ? The length of the chain that is discarded before the chain is judged to be reasonably close to stationarity is usually called burn-in, so this issue is generally referred to as the problem of determining burn-in.

- What is the best way to assess the accuracy of the estimates based on the samples after burn-in ?

These are difficult theoretical questions. For a nice description of the kind of theoretical work necessary to attempt to answer these questions, see Jones and Hobert (2001). If the samples drawn are i.i.d., the first problem is

completely avoided, and the second problem is easily solved. It is therefore of interest to see if exact simulation algorithms, algorithms that draw i.i.d. samples from the distribution of interest, are practical in some important situations.

1.4 Exact Simulation

A large part of this thesis focuses on methods that produce i.i.d. samples from posterior distributions for some important hierarchical models, thereby avoiding the diagnostics issues presented by MCMC methods. The techniques explored are described in Chapter 3, which includes a discussion of the classic Monte Carlo algorithm of rejection sampling (Section 3.1.1), and a recently developed method called perfect sampling (Section 3.2).

Perfect sampling, which was introduced in a paper by Propp and Wilson (1996), is a method which uses a clever idea to obtain i.i.d. draws from the exact stationary distribution of interest by using Markov chains. In general, to obtain a perfect sampler, one must design an algorithm whose updates satisfy many restrictive conditions. Unfortunately, this typically implies that perfect sampling algorithms are least likely to be available in exactly those situations where they can have the most impact. In other words, problems where it is hardest to assess whether or not Markov chain Monte Carlo methods are performing well also happen to be problems where perfect samplers are usually very hard to implement. Møller and Nicholls (1999) describe one family of algorithms, which they call perfect tempering algorithms, which attempts to produce perfect samplers for difficult problems. For the models considered

in this thesis, Bayesian disease mapping and a hierarchical Bayesian random effects model, it is still difficult to satisfy the conditions that are required for Moller and Nicholls algorithm. Chapters 4 and 5 describe methods by which one can satisfy these conditions , while still allowing the algorithm to be easily used. There have been a few attempts at i.i.d. samplers for the kind of Bayesian random effects models considered in Chapter 5, but there have apparently been no attempts to produce i.i.d. draws from the kind of multivariate, continuous distributions that are studied in Chapter 4. Chapters 4 and 5 show how the exact simulation algorithms can be fairly efficient and practical when applied to real data sets.

Chapter 2

Block Sampling MCMC Approaches for Spatial Models

Structured Markov Chain Monte Carlo (SMCMC) was introduced by Sargent, Hodges and Carlin (2000) as a general method for Bayesian computing in richly-parameterized models. Here, "richly parameterized" refers to hierarchical and other multilevel models. SMCMC provides a simple, general and flexible framework for accelerating convergence in an MCMC sampler by providing a systematic way to update groups of similar parameters in blocks while taking full advantage of the posterior correlation structure induced by the model and data. Sargent et al. (2000) apply SMCMC to several different models, including a hierarchical linear model with normal errors and a hierarchical Cox proportional hazards model.

Blocking, i.e., simultaneously updating multivariate blocks of (typically highly correlated) parameters, is a general approach to accelerating MCMC convergence. Liu (1994) and Liu et al. (1994) confirm its good performance

for a broad class of models, though Liu et al. (1994, Sec.5) and Roberts and Sahu (1997, Sec 2.4) give examples where blocking slows a sampler's convergence. This chapter shows how spatial models of the kind proposed by Besag, York and Mollie (1991) using non-stationary "intrinsic autoregressions" are richly parameterized and lend themselves to the SMCMC algorithm. Bayesian inference via MCMC for these models has generally used single parameter updating algorithms with often poor convergence and mixing properties. There have been some recent attempts to use blocking schemes for similar models. Cowles (2002, 2003) uses SMCMC blocking strategies for geostatistical and areal data models with normal likelihoods, while Knorr-Held and Rue (2002) implement blocking schemes using algorithms that exploit the sparse matrices that arise out of the areal data model by using appropriate fast sampling techniques described in Rue (2001).

This chapter describes several different strategies for block-sampling parameters in the posterior distribution when the likelihood is Poisson. Among the SMCMC strategies considered here are blocking using different-sized blocks (grouping by geographical region), updating jointly with and without model hyperparameters, "oversampling" some of the model parameters, reparameterization via hierarchical centering, and "pilot adaptation" of the transition kernel. Our results suggest that the techniques will generally be far more accurate (produce less correlated samples) and often more efficient (produce more effective samples per second) than univariate sampling procedures.

The remainder of this chapter is organized as follows. Section 2.1 provides the details of the spatial models for areal data. The rest of Section 2.1.2

then lays out the basics of the "constraint case" formulation which is a framework for rewriting the class of spatial models in a manner so the SMCMC approach can be used to advantage. A variety of different SMCMC algorithms are outlined in Section 2.2, including a description of the features and potential advantages and disadvantages of each. Section 2.3 then presents the datasets (both related to cancer control in the counties of the state of Minnesota), followed by detailed tabular and graphical results concerning the efficiency of the SMCMC algorithms relative to each other and to the univariate updating algorithm. The datasets are quite different in size and character, and are suggestive of general situations where SMCMC may or may not be expected to pay significant dividends. Finally, Section 2.4 discusses the findings, briefly compares them with other similar approaches for areal data, and offers directions for future applied and methodologic research in this area.

2.1 Applying Structured MCMC to Areal Data

2.1.1 Spatial Modeling of Areal Data

Besag, York and Mollie (1991) describe a spatial model for areal data (i.e., data arising as sums or averages over geographic regions). The number of disease events in region i, Y_i, is modeled as a Poisson random variable with mean $E_i \exp(\mu_i)$. E_i is the expected number of events in region i while μ_i is the log-relative risk of disease. So $Y_i \sim \text{Poi}(E_i e^{\mu_i})$, with μ_i modeled linearly as

$$\mu_i = \theta_i + \phi_i \ , \ i = 1, ..., N,$$

where N is the total number of regions, and $\{\theta_1, .., \theta_N\}, \{\phi_1, .., \phi_N\}$ are vectors of random effects. The θ_i's are independent and identically distributed Gaussian normal variables, while the ϕ_i's are conditionally autoregressive and assumed to follow a Gaussian Markov random field (GMRF). In this way, each θ_i captures the ith region's extra-Poisson variability due to area-wide heterogeneity, while each ϕ_i captures the ith region's excess variability attributable to regional clustering. These distributions are specified as follows:

$$\theta_i|\tau_h \sim N(0, 1/\tau_h), \text{ and } \phi_i|\phi_{j\neq i} \sim N(\mu_{\phi_i}, \sigma^2_{\phi_i}), i = 1, ..., N \ ,$$

$$\text{where } \mu_{\phi_i} = \frac{\sum_{j\neq i} w_{ij}\phi_j}{\sum_{j\neq i} w_{ij}} \text{ and } \sigma^2_{\phi_i} = \frac{1}{\tau_c \sum_{j\neq i} w_{ij}}.$$

The μ_{ϕ_i} for a region i is thus a weighted average of the clustering parameters in other regions. Most commonly, w_{ij} is taken as 0 unless regions i and j are immediate neighbors. If the regions i and j are adjacent, w_{ij} is set to 1, although since they are merely weights, other forms of the w_{ij} are also possible. Note that the prior on ϕ_i leaves the overall level of the GMRF unspecified; the prior is therefore improper due to translation invariance.

A question of epidemiological interest is whether more of the variability of the observations in different regions is captured by heterogeneity (corresponding to global variability) or by clustering (corresponding to local variability). Thus the variance components of the θ_i and the ϕ_i are of interest in their own right (Best et al., 1999), suggesting the need for hyperpriors on the τ_h and τ_c. A "fair" specification here is complicated by the fact that τ_h is an *unconditional* prior precision, while τ_c is part of a *conditional* prior precision. Conjugate Gamma hyperpriors are placed on the precision parameters, namely $\tau_h \sim G(\alpha_h, \beta_h)$ and $\tau_c \sim G(\alpha_c, \beta_c)$ with $\alpha_h = 1.0$, $\beta_h = 100.0$,

$\alpha_c = 1.0$ and $\beta_c = 50.0$ (these hyperpriors have means of 100 and 50, and standard deviations of 10000 and 2500 respectively, a specification recommended by Bernardinelli et al., 1995). See Eberly and Carlin (2000) for more discussion of "fair but vague" priors for τ_h and τ_c.

2.1.2 SMCMC Algorithm Basics

Following Hodges (1998), a hierarchical model can be expressed in the general form

$$\left[\begin{array}{c} y \\ \hline 0 \\ \hline M \end{array}\right] = \left[\begin{array}{c|c} X_1 & 0 \\ \hline H_1 & H_2 \\ \hline G_1 & G_2 \end{array}\right] \left[\begin{array}{c} \theta_1 \\ \hline \theta_2 \end{array}\right] + \left[\begin{array}{c} \epsilon \\ \hline \delta \\ \hline \xi \end{array}\right] . \tag{2.1}$$

The first set of rows of this layout correspond to the "data cases," or the terms in the joint posterior into which the response, the data y, enters directly. The second set of rows (corresponding to the H_i) are called "constraint cases" since they place stochastic constraints on possible values of θ_1 and θ_2. The third set of rows, the "prior cases" for the model parameters, have known (specified) error variances for these parameters. Equation (2.1) can be expressed as $Y = \mathrm{X}\mathbf{\Theta} + \mathrm{E}$, where X and Y are known, $\mathbf{\Theta}$ is unknown, and E is an error term with block diagonal covariance matrix $\Gamma = \mathrm{Diag}(\mathrm{Cov}(\epsilon), \mathrm{Cov}(\delta), \mathrm{Cov}(\xi))$. If the error structure for the data is normal, i.e., if the ϵ vector in the constraint case formulation (2.1) is normally distributed, then the conditional posterior density of $\mathbf{\Theta}$ is

$$\Theta | Y, \Gamma \sim N((X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}Y)\ ,\ (X^T\Gamma^{-1}X)^{-1})\ . \tag{2.2}$$

The basic SMCMC algorithm is then nothing but the following two-block Gibbs sampler :

(1) Sample $\mathbf{\Theta}$ as a single block from the above normal distribution, using the current value of Γ.

(2) Update Γ using the conditional distribution of the variance components with the current value of $\mathbf{\Theta}$.

In the spatial model setting, the errors are not normally distributed, so the normal density described above is not the correct conditional posterior distribution for $\mathbf{\Theta}$. Still, a SMCMC algorithm with a Metropolis-Hastings implementation can be used, with the normal density in (2.2) taken as the candidate density.

2.1.3 Application to Spatial Modeling

Consider a data set of N regions with C pairs of adjacent neighbors. Thus, there are $2N + 2$ model parameters: $\{\theta_i : \ i = 1, \ldots N\}$, $\{\phi_i : \ i = 1, \ldots N\}$, τ_h and τ_c. The SMCMC algorithm requires transforming the Y_i data points to $\hat{\mu}_i = \log(Y_i/E_i)$, which can be conveniently viewed as the response since they should be roughly linear in the model parameters (the θ_i's and ϕ_i's). For the constraint case formulation, the different levels of the model are written down case by case. The data cases are $\hat{\mu}_i$, $i = 1, \ \ldots, N$. The constraint cases for the θ_i's are $\theta_i \sim N(0, 1/\tau_h)$, $i = 1, \ \ldots, N$. For the constraint cases involving the ϕ_i's, the differences between the neighboring ϕ_i's can be used to get an unconditional distribution for the ϕ_i's using pairwise differences (Besag et al. 1995). Thus the constraint cases can be written as

$$(\phi_i - \phi_j)|\tau_c \sim N(0, 1/\tau_c) \text{ for each } i, j \text{ that are adjacent regions.} \qquad (2.3)$$

To obtain an estimate of Γ, estimates are needed for the variance-covariance matrix corresponding to the $\hat{\mu}_i$'s (the data cases) and initial estimates of the variance-covariance matrix for the constraint cases (the rows corresponding to the θ_i's and ϕ_i's). The delta method can be used to obtain an approximation as follows : assume $Y_i \sim N(E_i e^{\mu_i}, E_i e^{\mu_i})$ (roughly), so invoking the delta method shows that $\text{Var}(\log(Y_i/E_i))$ is approximately $1/Y_i$. A reasonably good starting value is particularly important here since these variance estimates are never updated (the data variance section of Γ stays the same throughout the algorithm). For initial estimates of the variance components corresponding to the θ_i's and the ϕ_i's, the means of the hyperprior densities on τ_h and τ_c can be substituted into Γ.

To be able to compute the Hastings ratio, the distribution of the ϕ_i's is rewritten in the joint pairwise difference form (Besag et al. , 1995), with the appropriate exponent for τ_c (Hodges and Carlin, 2001):

$$p(\phi_1, \phi_2, ..., \phi_N|\tau_c) \propto \tau_c^{(N-1)/2} \exp\left\{-\frac{\tau_c}{2}\sum_{i\sim j}(\phi_i - \phi_j)^2\right\}, \tag{2.4}$$

where $i \sim j$ if i and j are neighboring regions. Finally, the joint distribution of the θ_i's is given by

$$p(\theta_1, \theta_2, ..., \theta_N|\tau_h) \propto \tau_h^{N/2} \exp\left\{-\frac{\tau_h}{2}\sum_{i=1}^{N}\theta_i^2\right\}. \tag{2.5}$$

The $(2N + C) \times 2N$ design matrix for the spatial model is defined by :

$$X = \left[\begin{array}{c|c} I_{N\times N} & I_{N\times N} \\ \hline -I_{N\times N} & 0_{N\times N} \\ \hline 0_{C\times N} & A_{C\times N} \end{array}\right]. \tag{2.6}$$

The design matrix is divided into two halves, the left half corresponding to the N θ_i's and the right half referring to the N ϕ_i's. The top section of this design matrix is an $N \times 2N$ matrix relating $\hat{\mu}_i$ to the model parameters θ_i and ϕ_i. In the ith row, a 1 appears in the ith and $(N+i)$th columns while 0s appear elsewhere. Thus the ith row corresponds to $\mu_i = \theta_i + \phi_i$. The middle section of the design matrix is an $N \times 2N$ matrix which imposes a stochastic constraint on each θ_i separately (θ_i's are i.i.d normal). The bottom section of the design matrix is a $C \times 2N$ matrix with each row having a -1 and 1 in the $(N+k)$th and $(N+l)$th columns respectively, corresponding to a stochastic constraint being imposed on $\phi_l - \phi_k$ (using the pairwise difference form of the prior on the ϕ_i's as described in (2.3) with regions l and k being neighbors). The variance-covariance matrix Γ is a diagonal matrix with the top left section corresponding to the variances of the data cases, i.e., the $\hat{\mu}_i$'s. Using the variance approximations described above, the $(2N+C)\times(2N+C)$ block diagonal variance-covariance matrix is

$$\Gamma = \left[\begin{array}{c|c|c} \mathrm{Diag}(1/Y_1, 1/Y_2, \ldots, 1/Y_N) & 0_{N\times N} & 0_{N\times C} \\ \hline 0_{N\times N} & \frac{1}{\tau_h} I_{N\times N} & 0_{N\times C} \\ \hline 0_{C\times N} & 0_{C\times N} & \frac{1}{\tau_c} I_{C\times C} \end{array}\right]. \qquad (2.7)$$

The SMCMC candidate generating distribution is thus of the form (2.2),

$$\Theta|\hat{\boldsymbol{\mu}}, \Gamma \sim N((X^T\Gamma^{-1}X)^{-1}X^T\Gamma^{-1}\hat{\boldsymbol{\eta}}\ ,\ (X^T\Gamma^{-1}X)^{-1}), \qquad (2.8)$$

where $\boldsymbol{\Theta}$=$\{\theta_1, \cdots \theta_N, \phi_1, \cdots \phi_N\}^T$, $\hat{\boldsymbol{\mu}}^T = \{\hat{\mu}_1, \cdots, \hat{\mu}_N\}$, $\hat{\boldsymbol{\eta}}^T = \{\hat{\boldsymbol{\mu}}^T, \mathbf{0}_{N+C}\}^T$.

Now,

$$X^T\Gamma^{-1}\hat{\boldsymbol{\eta}} = \begin{bmatrix} V^{-1}\hat{\boldsymbol{\mu}} \\ V^{-1}\hat{\boldsymbol{\mu}} \end{bmatrix}.$$

Thus $X^T\Gamma^{-1}\hat{\boldsymbol{\eta}}$ needs to be computed only once at the beginning of the algorithm since it does not depend on the sampled value of τ_h, τ_c.

Note that the exponent on τ_c in (2.4) would actually be $C/2$ (instead of $(N-1)/2$) if obtained by taking the product of the terms in (2.3). Thus, (2.3) is merely a form used to describe the distribution of the ϕ_is for the constraint case specification. The formal way to incorporate the distribution of the ϕ_is in the constraint case formulation is by using an alternate specification of the joint distribution of the ϕ_is, as described in Besag and Kooperberg (1995). This form is like an $N \times N$ Gaussian density, but with a *singular* precision matrix, Q,

$$p(\phi_1, \phi_2, ..., \phi_N|\tau_c) \propto \exp\left(-\frac{\tau_c}{2}\boldsymbol{\phi}^T Q \boldsymbol{\phi}\right), \text{ where } \boldsymbol{\phi}^T = (\phi_1, \phi_2, ..., \phi_N),$$

and

$$Q_{ij} = \begin{cases} c & \text{if } i = j \text{ where } c = \text{ number of neighbors of region } i \\ 0 & \text{if } i \text{ is not adjacent to } j \\ -1 & \text{if } i \text{ is not adjacent to } j \end{cases}$$

However, it is possible to show that this alternate formulation (using the corresponding design and Γ matrices) results in the same SMCMC candidate mean and covariance matrix for $\boldsymbol{\Theta}$ given τ_h and τ_c as the one described in (2.2); see the appendix for details. This formulation becomes much more important in the context of Chapter 4.

2.2 Algorithmic Schemes

Univariate MCMC (UMCMC): For the purpose of comparison with the different blocking schemes, the first sampler considered was a univariate

(updating one variable at a time) sampler. This was done by sampling τ_h and τ_c from their gamma full conditional distributions, and then, for each i, sampling each θ_i and ϕ_i from its full conditional distribution. The latter used a Metropolis step with univariate Gaussian random walk proposals, the variances of which were tuned to produce acceptance rates between 30% and 70%.

Reparameterized Univariate MCMC (RUMCMC): The parameters $(\theta_1, \ldots, \theta_N, \phi_1, \ldots, \phi_N)$ can be reparameterized as $(\mu_1, \ldots, \mu_N, \phi_1, \ldots, \phi_N)$, where $\mu_i = \theta_i + \phi_i$. The (new) model parameters and the precision parameters were sampled in a similar manner as for UMCMC. This "hierarchical centering" was suggested by Besag et al. (1995) and Waller et al. (1997) for the spatial model, and discussed in general by Gelfand et al. (1995).

Structured MCMC (SMCMC): Initially, a pilot adaptation strategy was studied, which involved sampling (τ_h, τ_c) from their gamma full conditionals, updating the Γ matrix using the averaged (τ_h, τ_c) sampled so far, updating the SMCMC candidate covariance matrix and mean vector using the Γ matrix, and then sampling $(\boldsymbol{\theta}, \boldsymbol{\phi})$ using the SMCMC candidate in a Metropolis-Hastings step. After running the above steps for a "tuning" period, the SMCMC candidate mean and covariance were fixed, (τ_h, τ_c) were sampled as before, and $(\boldsymbol{\theta}, \boldsymbol{\phi})$ were sampled via the Metropolis-Hastings algorithm using SMCMC proposals. Some related strategies studied included adaptation of the Γ matrix more or less frequently, adaptation over shorter and longer periods of time, and pilot adaptation while blocking on groups of regions.

The results with pilot adaptation schemes indicated that a single pro-

posal, regardless of adaptation period length, will probably be unable to provide a reasonable acceptance rate for the many different values of (τ_h, τ_c) that will be drawn in realistic problems. As such, a far superior strategy involved oversampling $\boldsymbol{\Theta}$ relative to (τ_h, τ_c); that is, the SMCMC proposal is always based on the current (τ_h, τ_c) value. In this algorithm, τ_h and τ_c are sampled from their gamma full conditionals, and the SMCMC proposal are computed based on the Γ matrix using the generated τ_h and τ_c. A Hastings independence subchain is run for each (τ_h, τ_c) pair, by sampling a sequence of length 100 (say) of $\boldsymbol{\Theta}$'s using the SMCMC proposal. Further implementational details for this algorithm are given in the appendix.

Blocking by geographical proximity: This scheme is identical to SMCMC, except that smaller blocks of parameters are used. That is, instead of sampling $\boldsymbol{\Theta}$ in one large block, $\boldsymbol{\Theta}$ is now sampled by breaking it up into smaller blocks of parameters, grouped on the basis of geographical proximity. The proposals for the smaller blocks were then obtained by 'reading off' the corresponding marginal distributions from the SMCMC proposal for the entire block. For instance, if $\theta_1, \theta_4, \theta_7$ were selected as a block, then the proposal of the block would be a normal distribution with mean consisting of the first, fourth and seventh elements of the mean vector and the corresponding covariance submatrix described in (2.8). This scheme was tried for several different groupings and block sizes; while it often did better than univariate schemes, the full-block SMCMC scheme above was always faster and more efficient. The slight increase in acceptance rates for smaller individual blocks did not seem to compensate for the poor mixing relative to the SMCMC algorithm. Details of this algorithm are given in the appendix.

Blocking with precision components: Knorr-Held and Rue (2002) suggest a version of the following algorithm that samples $\boldsymbol{\Theta}$ along with the precision components in a single block.

1. Sample τ_h and τ_c from some proposal distribution. For instance, the proposal used for this algorithm (suggested by Knorr-Held and Rue (2000)) generates a τ_h candidate by multiplying the current value of τ_h by a Uniform on $(1/f, f)$ where f is a tuning constant. A Gamma proposal could also be used but it appears to be difficult to tune the parameters to obtain good acceptance rates for the large block.

2. Using the generated value of τ_h and τ_c, compute the SMCMC proposal mean and covariance matrix.

3. Sample $\boldsymbol{\Theta}$ using the SMCMC proposal.

4. Accept-reject (τ_h, τ_c, $\boldsymbol{\Theta}$) as one large block using a Metropolis-Hastings ratio.

This fully blocked SMCMC algorithm was found to be somewhat costly in terms of computation time, since a new mean and covariance matrix needs to be computed at every iteration of the algorithm (for each new sampled value of τ_h, τ_c). Also, the above algorithm did not even always produce the least auto-correlated samples. For the purpose of speeding up the fully blocked algorithm, Knorr-Held and Rue (2002) use clever sparse matrix algorithms effectively (described in detail in Rue, 2001). These methods are explored further in the exact simulation algorithms of Chapter 4.

Reparameterized Structured MCMC (RSMCMC): This algorithm is the SMCMC analogue of the reparameterizated univariate algorithm (RUM-

CMC). The algorithm follows exactly the same steps as the SMCMC algorithm, with the only difference being that $\boldsymbol{\Theta}$ is now $(\boldsymbol{\mu}, \boldsymbol{\phi})$ instead of $(\boldsymbol{\theta}, \boldsymbol{\phi})$, and the proposal distribution is adjusted according to the new parameterization. It is obtained in the same manner as before. The algorithm is described in detail in the appendix.

2.3 Results

2.3.1 Description of Datasets

Minnesota Cancer Detection Data

The first dataset is taken from the Minnesota Cancer Surveillance System (MCSS), a cancer registry maintained by the Minnesota Department of Health. The MCSS is population-based for the state of Minnesota, and collects information on geographic location and stage at detection for colorectal, prostate, lung, and female breast cancers. An external audit (Cancer Surveillance and Control Program, 1997) performed in June 1996 estimated that MCSS hospital-based case finding was 99.6% complete for microscopically confirmed cancers, and 99.1% complete for all cancers. We may thus think of these data as an essentially complete picture of all cancers that occur in Minnesota.

The computational approaches are illustrated by analyzing the MCSS data for two of the cancers, breast and colorectal. Each of the 87 counties in the data set has associated with it the total number of cancer cases recorded between 1995 and 1997, and the number of these detected late. The expected number of late detections for that county can be taken to be the number of

cancer cases for that county multiplied by the statewide rate of late detections. The question of interest is whether there are clusters of counties in the state of Minnesota with much higher than expected late detection rates for either cancer. The spatial model provides smoothed estimates of the relative risk of cancer cases being detected late in each county. Counties (or clusters of counties) emerging with higher smoothed late detection rate may be targets for aggressive screening efforts, such as the deployment of mobile mammography units.

Minnesota Cancer Mortality Data

The second dataset comes from the Minnesota Department of Health's Center for Health Statistics, and consists of the age-adjusted cancer death rates r_i^* for each county i during the period 1991-1998. Deaths from cancer were determined by the ICD-9 codes on the death certificates of Minnesota residents. Census data from the same period were used to obtain an average population n_i for each county, thus determining an "age-adjusted cancer death total," $Y_i = n_i r_i^*$. Similar to the previous subsection, expected number of age-adjusted deaths for each county are specified as $E_i = n_i R$, where $R = (\sum_i Y_i)/(\sum n_i)$, the statewide age-adjusted cancer death rate. The model of Section 3 now applies as before. Again the substantive problem of interest is to determine overall spatial trends in cancer death, and whether or not counties with significantly elevated smoothed rates exist.

While the adjacency structure remains as in the above two data sets, the counts in this data set are appreciably higher, with a mean count around 700 (versus 34 and 51 for breast and colorectal cancer detection, respectively)

and a lowest count of 48 (versus 2 and 6). This dataset thus affords an opportunity to study the performance of the algorithms operating on a dataset with larger counts per region.

2.3.2 Summary of Results

Minnesota Cancer Detection Data

For the sake of brevity, results are displayed for only the four major algorithms described in Section 2.2, and for only a few parameters for each data set. To make a fair comparison among the various implementations, we can use the notion of effective sample size, or ESS (see Kass et al. , 1998, or Geweke, 1992). ESS is defined for each parameter as the number of MCMC samples drawn divided by the parameter's so-called autocorrelation time, $\kappa = 1 + 2\sum_{k=1}^{\infty} \rho(k)$, where $\rho(k)$ is the autocorrelation at lag k. The κ is estimated from the MCMC chain, using the initial monotone positive sequence estimator as given in Geyer (1992, pg.477). Let $\widehat{\Gamma}_m = \widehat{\gamma}_{2m} + \widehat{\gamma}_{2m+1}$ To obtain the estimated autocorrelation time, $\hat{\kappa}$, the first k empirical autocorrelations can be used,

$$\hat{\kappa} = 1 + \sum_{i=1}^{k} \hat{\gamma}_i$$

where k is the largest integer such that

$$\hat{\Gamma}_i > 0, \text{ and } \hat{\Gamma}_i \text{ is monotone}, i = 1, ..k.$$

This allows the window width (for calculation of $\hat{\kappa}$) to indirectly depend on n, the number of samples. This estimator is, to a large extent, protected against random noise, and under some mild assumptions, is a consistent estimator of autocorrelation time.

In what follows, ESS is used as a measure of algorithm accuracy (since less correlated samples lead to more accurate inference) and effective samples per second (ES/s) as a measure of algorithm efficiency. All algorithms were programmed in `C` and run on the same LINUX machines. The parameters presented in the table were selected on the basis of their ESS's. In particular, the selected parameters reflect the lowest, median, and highest differences in ESS between the UMCMC and SMCMC algorithm.

When experimenting with simulated data sets, reasonable acceptance rates (over 30%) were obtained using pilot adaptation for problems of dimension less than 30, but these acceptance rates quickly dropped as the dimension of the problem increased. This suggests that it would make sense to have a single normal proposal for *each* generated value of (τ_h, τ_c), so that the SMCMC proposal and the τ_h and τ_c values are "synchronized." This led to the oversampled SMCMC scheme described in the previous section. Fixing (τ_h, τ_c) and then immediately tuning the SMCMC proposal to produce an independence subchain of samples from the $\boldsymbol{\Theta}$ posterior helps the sampler achieve high acceptance rates and also good mixing properties (see Tables 2.1 and 2.2). For instance, in Table 2.1, the autocorrelations for the samples for all the model parameters are much lower for SMCMC than for the univariate sampler (UMCMC), and the corresponding ESS is much higher. The same is also true in Table 2.2, where the samples from the block samplers have practically no autocorrelations, and hence, have high ESS's. Even in terms of ES/s, SMCMC is pretty close to the corresponding univariate schemes for both data sets, with SMCMC doing better than UMCMC for around 50% of the parameters.

The results for the reparameterized versions of these two algorithms, RUMCMC and RSMCMC, suggest that reparameterization is not particularly effective for these two data sets. In fact, RUMCMC does worse than UMCMC for several of these parameters. The SMCMC version of the reparameterization algorithm (RSMCMC) does appear to do quite well, and produces comparable ESSs to the SMCMC algorithm. This suggests that block sampling in the manner suggested can improve a sampler even under a parameterization that performs poorly for univariate updating schemes.

In summary, for the (relatively small) breast cancer and colorectal cancer detection data sets, ESS is often higher. For both data sets, the block samplers beat the univariate samplers in terms of ESS for practically all the parameters. Even in terms of ES/s, the block samplers are comparable to the much faster univariate samplers, suggesting that the block sampler approaches may be generally preferable to the univariate algorithms.

Minnesota Cancer Mortality Data

To a significant extent, the performance of the SMCMC schemes hinges on the accuracy of the normal approximation to the Poisson likelihood. Since we know that this approximation is better for higher counts, the Minnesota cancer mortality data set described in Subsection 2.3.1 should result in better efficiencies than those observed in the previous subsection. The UMCMC, SMCMC, RUMCMC and RSMCMC schemes were run on this data set. As expected, UMCMC performs poorly for these large counts, though the reparameterized univariate algorithm (RUMCMC) does provide a significant improvement in this case. However, SMCMC and RSMCMC still

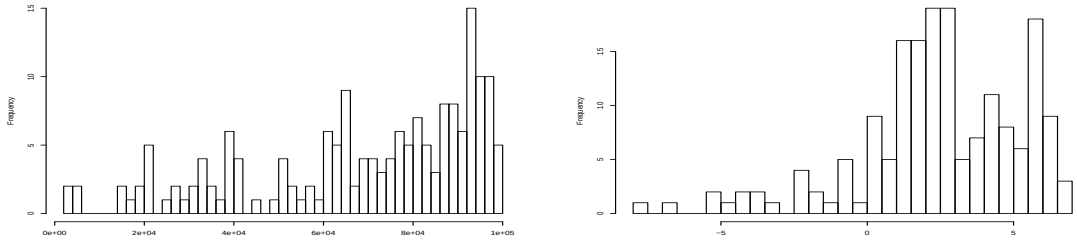
perform better than both univariate algorithms. For a representative set of parameters (selected in the same way as before), Table 2.3 shows that even when accounting for the amount of time taken by the SMCMC algorithm (in terms of ES/s), the SMCMC scheme results in a far more efficient sampler than the univariate algorithm; for some parameters (not shown here), SMCMC can produce as much as 64 times more effective samples per second.

In fact, the improvement offered by SMCMC is even greater than suggested by Table 2.3. This can be seen in Figure 2.1, which shows the differences in ESS for all the parameters, along with the difference in ES/s. The UMCMC algorithm ESSs are always below 16,000 while over 79% of all the ESSs for the SMCMC algorithm are over 50,000, and SMCMC does better than UMCMC for *all* the parameters. Note that, even in terms of ES/s, the SMCMC algorithm improves on the univariate algorithm for all but 13% of the model parameters.

The RSMCMC algorithm generally outperforms RUMCMC in terms of ESS, as can be seen from Table 2.4 and Figure 2.2(a). However, Figure 2.2(b) shows that RUMCMC always has lower ES/s than RSMCMC. Thus, the high overhead of blocking seems to limit the cost-effectiveness of RSMCMC in this case.

2.4 Discussion

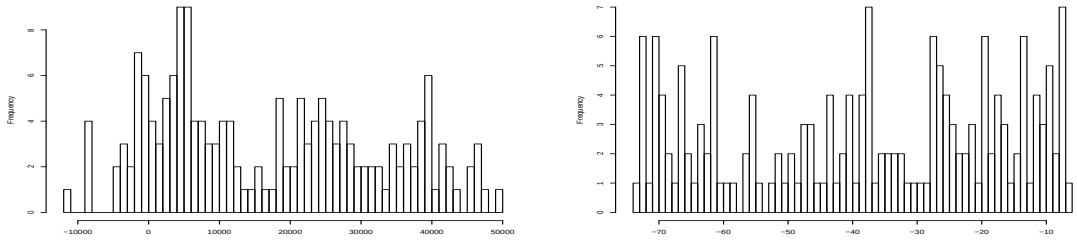
This chapter describes several block updating algorithms for analyzing Bayesian hierarchical models for disease mapping as described in Section 2.1. Among strategies considered were blocking on different size blocks, up-



(a) SMCMC ESS – UMCMC ESS

(b) SMCMC ES/s – UMCMC ES/s

Figure 2.1: ESS and ES/s for UMCMC versus SMCMC algorithms (positive values favor SMCMC)



(a) RSMCMC ESS – RUMCMC ESS

(b) RSMCMC ES/s – RUMCMC ES/s

Figure 2.2: ESS and ES/s for RUMCMC versus RSMCMC algorithms (positive values favor RSMCMC)

dating jointly with and without the hyperparameters, "oversampling" the parameters, hierarchical centering reparameterization, and "pilot adaptation" versus continuous tuning techniques for the proposal. A fixed Gaussian proposal even after a long tuning period does poorly; as the precision parameters change, a fixed proposal is unable to produce candidates that get accepted often enough for a good sample. Breaking the total area up into smaller blocks also does not appear to be a good solution since such schemes result

in samplers with poorer mixing properties than algorithms which sample in one large block.

Knorr-Held and Rue (2002) suggest that blocking the hyperparameters with the model parameters is perhaps the only way to ensure good mixing for such models. The schemes are easier to implement (they do not require use of specialized computer code), but at the same time can produce improvements in ESS *and* ES/s (the latter of which is not discussed by Knorr-Held and Rue). For the data sets considered here, the blocking scheme produces samples with good mixing even when the hyperparameters are sampled separately. In fact, sampling the τ_h and τ_c parameters separately and only at predetermined times resulted not only in much faster algorithms, but in posterior samples for the θ_is and ϕ_is having lower autocorrelations for all three data sets. The experience with these different algorithms thus leads us to prefer the SMCMC and RSMCMC algorithms as a systematic way for efficient sampling from such models.

The results of applying the algorithm to three Minnesota cancer data sets were described. While SMCMC and RSMCMC provide better mixing in the samplers for several parameters in the first two data sets (breast and colorectal cancer detection), the univariate algorithm (UMCMC) produced samples that were adequate for most parameters though RUMCMC did rather poorly in comparison. However, with the third data set (cancer mortality), the same univariate sampler performed very poorly, often producing effective sample sizes of only 100 to 200 for 100,000 samples from the chain. The reparameterized univariate algorithm (RUMCMC) did perform much better, but the SMCMC and RSMCMC algorithms still outperformed it in terms of ESS,

producing nearly uncorrelated samples, as well as reasonably high efficiency as measured by ES/s. Moreover, all of the results were based on only every tenth sample from the chain; the SMCMC algorithms would likely enjoy an even larger benefit had this thinned sample not been used.

These schemes were also tried on a few simulated data sets. θ_i and ϕ_i values were simulated from i.i.d. normal and CAR priors, respectively, given precision parameters $\tau_h = 250$ and $\tau_c = 100$, values roughly comparable to those observed in the posterior for the colorectal cancer data set. The data values Y_i were simulated from conditionally independent Poisson distributions having mean $E_i \exp(\theta_i + \phi_i)$, where the E_is were also as given in the colorectal cancer data. The SMCMC algorithms continued to outperform the univariate algorithms in much the same way as they did for the cancer detection and mortality data sets, with superior ESS and competitive ES/s results.

Overall, the experience with applying several SMCMC blocking schemes to real data sets suggests that SMCMC provides a standard, systematic technique for producing samplers with far superior mixing properties than simple univariate Metropolis-Hastings samplers. The SMCMC and RSMCMC schemes appear to be reliable ways of producing good ESSs, irrespective of the data sets and parameterizations. In many cases, the SMCMC algorithms are also competitive in terms of ES/s. On a more practical note, since the blocked SMCMC algorithms mix better, their convergence should be easier to diagnose and thus lead to final parameter estimates that are less biased. These estimates should also have smaller associated Monte Carlo variance estimates.

Results for the scheme using "full blocking" ($\boldsymbol{\Theta}$ and precision components) were generally promising in terms of fairly low autocorrelations for all the parameters. However, the ES/s values were never competitive, suggesting that for these data, much larger computation times are required. The larger computation times are due to the fact that a new SMCMC proposal mean and covariance needs to be computed for *every* update of the fully blocked Metropolis-Hastings algorithm, and obtaining this mean and covariance involves matrix inversions and choleski decompositions. There are also several other techniques that could perhaps be used in conjunction with some of the SMCMC techniques described here to produce further improvements in the properties of the samples produced. For instance, multi-chain annealing or tempering (Geyer and Thompson, 1995; Neal, 1996) and simulated sintering (Liu and Sabatti, 1999) are recent approaches to accelerate sampling for such models. Also, there may be ways to further speed up computation by using linear algebra techniques for sparse matrices as described in Rue (2001) and Knorr-Held and Rue (2002).

	Method	mean	sd	AC1	AC5	AC10	ESS	ES/s
ϕ_7	UMCMC	0.011	0.059	0.85	0.56	0.42	2558.52	2.06
ϕ_7	SMCMC	0.013	0.059	0.05	0.01	0	87324.52	6.3
ϕ_7	RUMCMC	0.014	0.059	0.53	0.33	0.22	7613.83	6.06
ϕ_7	RSMCMC	0.012	0.058	0.05	0	0.01	85412.01	6.45
ϕ_{15}	UMCMC	0.065	0.072	0.86	0.62	0.51	1770.87	1.42
ϕ_{15}	SMCMC	0.065	0.071	0.13	0.07	0.07	46110.47	3.33
ϕ_{15}	RUMCMC	0.066	0.071	0.63	0.44	0.32	4436.28	3.53
ϕ_{15}	RSMCMC	0.065	0.069	0.12	0.06	0.06	46418.03	3.51
θ_{56}	UMCMC	-0.048	0.067	0.29	0.06	0.04	34158.58	27.48
θ_{56}	SMCMC	-0.05	0.068	0.14	0.1	0.09	42524.81	3.07
θ_{56}	RUMCMC	-0.049	0.068	0.45	0.12	0.05	22295.57	17.74
θ_{56}	RSMCMC	-0.051	0.068	0.13	0.09	0.1	53139.88	4.01
τ_h	UMCMC	263.305	135.471	-0.02	0.01	0.01	1000	0.8
τ_h	SMCMC	263.213	143.781	0.39	0.02	0.05	464.63	0.03
τ_h	RUMCMC	257.536	127.679	0.01	-0.04	0.02	1000	0.8
τ_h	RSMCMC	253.287	134.534	0.37	-0.02	0.03	505.27	0.04
τ_c	UMCMC	109.862	61.903	0.15	-0.05	0	773.02	0.62
τ_c	SMCMC	109.883	62.774	0.45	0.04	-0.04	353.41	0.03
τ_c	RUMCMC	106.315	59.725	0.06	0.05	-0.02	885.35	0.7
τ_c	RSMCMC	110.55	60.444	0.42	-0.01	0.01	418.74	0.03

Table 2.1: Selected results for the Minnesota breast cancer detection data set. Each chain was run for 1 million iterations. Only every 10th sample for the θ_i and ϕ_i (and every 1000th sample for τ_h and τ_c, due to the oversampling in SMCMC and RSMCMC) was saved, so the above lag1, lag5 and lag10 ACs are for this thinned sample.

	Method	mean	sd	AC1	AC5	AC10	ESS	ES/s
ϕ_{75}	UMCMC	-0.011	0.06	0.84	0.54	0.4	3144.41	2.53
ϕ_{75}	SMCMC	-0.009	0.06	0	0	0	99207.1	7.16
ϕ_{75}	RUMCMC	-0.011	0.061	0.54	0.33	0.2	7918.18	6.3
ϕ_{75}	RSMCMC	-0.009	0.06	0	0	0	100000	7.55
θ_{69}	UMCMC	-0.017	0.051	0.3	0.16	0.11	16195.42	13.03
θ_{69}	SMCMC	-0.019	0.051	0.03	0.02	0.02	88135.5	6.36
θ_{69}	RUMCMC	-0.018	0.051	0.33	0.11	0.05	24610.39	19.58
θ_{69}	RSMCMC	-0.018	0.051	0.02	0.02	0.01	88732.19	6.7
θ_3	UMCMC	-0.05	0.062	0.25	0.05	0.02	41067.86	33.04
θ_3	SMCMC	-0.051	0.063	0.12	0.11	0.1	42327.36	3.06
θ_3	RUMCMC	-0.05	0.062	0.47	0.14	0.06	19442.6	15.47
θ_3	RSMCMC	-0.051	0.062	0.11	0.1	0.09	61166.16	4.62
τ_h	UMCMC	312.323	139.511	-0.04	0.03	0.01	1000.0	0.87
τ_h	SMCMC	306.829	146.865	0.32	0.02	0.07	502.76	0.04
τ_h	RUMCMC	308.54	138.365	-0.01	-0.08	-0.02	1000	0.8
τ_h	RSMCMC	304.573	140.563	0.27	-0.03	-0.02	595.29	0.04
τ_c	UMCMC	136.488	65.949	0.1	0.01	0.03	811.67	0.65
τ_c	SMCMC	130.292	60.755	0.35	0.04	-0.04	454.9	0.03
τ_c	RUMCMC	131.086	62.205	0.03	-0.02	0.01	940.56	0.75
τ_c	RSMCMC	131.894	62.717	0.36	0.01	0.01	473.75	0.04

Table 2.2: Selected results for the Minnesota colorectal cancer detection data set. Each chain was run for 1 million iterations. Only every 10th sample for the θ_i and ϕ_i (and every 1000th sample for τ_h and τ_c, due to the oversampling in SMCMC and RSMCMC) was saved, so the above lag1, lag5 and lag10 ACs are for this thinned sample.

	UMCMC ESS	SMCMC ESS	diff	ratio
θ_{71}	853.08	100000	99146.92	117.22
ϕ_{13}	940.32	77742.14	76801.82	82.68
θ_{27}	59.57	2558.03	2498.46	42.94

	UMCMC ES/s	SMCMC ES/s	diff	ratio
θ_{71}	0.69	7.22	6.53	10.52
ϕ_{13}	0.76	5.61	4.85	7.42
θ_{27}	0.05	0.18	0.14	3.85

Table 2.3: Selected results for Minnesota cancer mortality data set. Each chain was run for 1 million iterations.

	RUMCMC ESS	RSMCMC ESS	diff	ratio
ϕ_{15}	25955	75468.5	49513	2.91
θ_{17}	79213	94511.5	15298	1.19
θ_{82}	93285	82025	-11260	0.88

	RUMCMC ES/s	RSMCMC ES/s	diff	ratio
ϕ_{15}	20.65	5.7	-14.95	0.28
θ_{17}	63.02	7.14	-55.88	0.11
θ_{82}	74.21	6.19	-68.02	0.08

Table 2.4: Selected results for Minnesota cancer mortality data set. Each chain was run for 1 million iterations.

Chapter 3

Exact Simulation Algorithms

With the availability of faster computers, exact simulation algorithms that have been abandoned in favor of "off-the-shelf" MCMC methods, may well prove useful again. There is a large class of problems for which, given some preliminary work and thought, exact Monte Carlo methods such as rejection samplers, or even more complicated algorithms such as perfect samplers using Markov chains, may be practical. Thus difficult theoretical justifications and complicated issues such as convergence assessment (cf. Rosenthal (1995a, b), Roberts and Rosenthal (1999), Jones and Hobert (2003), Cowles and Carlin (1996), Cowles and Rosenthal (1999)) can be avoided. One of the main problems with MCMC approaches is that it is difficult to determine an appropriate length for running the Markov chain, which is often referred to as the problem of diagnosing convergence of the chain. There have been several approaches to diagnosing convergence, including using spectral analysis (Geweke, 1992; Geyer, 1992), large-sample normal theory with multiple chains (Gelman and Rubin, 1992) and graphical methods using cumulative

sum path plots (Yu and Mykland, 1998). As pointed out in Cowles and Carlin (1996), all of the proposed methods can fail to detect non-convergence problems, even for relatively simple problems. Theoretical work to determine bounding times for convergence is generally very difficult and often leads to bounds that are too conservative, and hence of not much practical use. Monte Carlo standard errors for samplers that produce i.i.d. draws from the distribution of interest can also be easily computed. These Monte Carlo (rather than MCMC) samplers may sometimes have the added benefit of being easy to program, as is the case with rejection samplers. Also, when it is relatively cheap to generate proposals, but the mixing is slow, perfect simulation can have advantages in terms of storage requirements as well, since proposals that are of little use (perhaps because of their high auto-correlation with previous samples), will not have to be stored when running an exact sampler.

This chapter begins with a general overview of exact Monte Carlo methods such as rejection sampling and importance sampling, and then describes the basic ideas behind perfect simulation, where Markov chains are used to produce i.i.d. samples.

3.1 Exact Monte Carlo Methods

This section describes two standard but powerful Monte Carlo algorithms that do not rely on Markov chains.

3.1.1 Rejection Sampling

Rejection sampling (or accept-reject sampling), first introduced by Von Neumann (1951), is a powerful and fairly general method for producing random variates. Let $P(x)$ be the unnormalized density of interest and $R(x)$ be the proposal density. For a rejection sampling algorithm to work, the necessary condition is

$$P(x)/R(x) \leq K, \text{ for some } K > 0, \quad K < \infty. \tag{3.1}$$

For the most efficient rejection sampler, K should be the *smallest* constant which satisfies $P(x)/R(x) \leq K$. An estimate of K, say $\hat{K}$, can be used since it may be prohibitively difficult to analytically obtain the smallest K that satisfies (3.1). R is usually called an enveloping density. The algorithm proceeds as follows:

1. Generate $X \sim R(\cdot)$ and $U \sim \text{Unif}(0, 1)$.
2. If $U \leq \frac{P(X)}{KR(X)}$, accept the sample X, otherwise reject the sample.
3. X is then a random variable with distribution $P(\cdot)$.
4. Repeat this procedure until the requisite number of samples have been accepted.

It is easy to see that it is not necessary for $P(\cdot)$ or $R(\cdot)$ to be normalized, since the normalizing constants cancel out. To get a heuristic sense for why this algorithm works, consider Figure 3.1. Say we draw a large number of samples from the envelope density R. Then, the corresponding histogram, scaled according to the constant K, would look roughly like the curve labeled

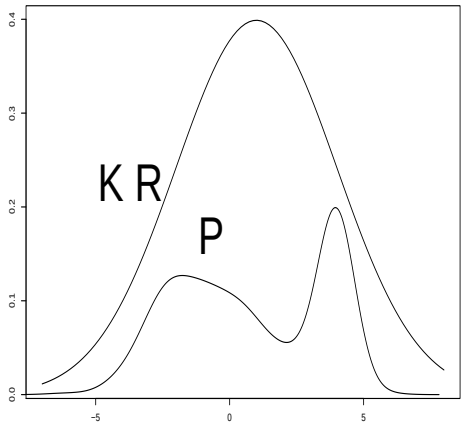


Figure 3.1: Heuristics for Rejection Sampling

"KR". Now, consider rejecting all the samples that lie above the "P" curve since this is what the rejection step effectively does. Then, the resulting histogram (with the samples above the "P" curve removed) would look like the "P" curve, the density of interest.

Since this is an algorithm used frequently in this thesis, a short proof of the rejection sampling algorithm is given here for completeness. Similar proofs for this algorithm can also be found in any of a number of standard references (see Devroye, 1986 or Ripley, 1987).

Theorem: The random variables generated from the above algorithm are i.i.d. with density, $P(\cdot)$.

Proof: Let R^* and P^* be the cumulative distribution functions of the distributions R and P respectively. If $X \sim R(\cdot)$, and assume (from above) that

$\frac{P(X)}{R(X)} \leq K$, and hence $\frac{dP^*(X)}{dR^*(X)K} \leq 1$ with probability 1. Then,

$$\begin{aligned}
P(X \leq x | X\text{accepted}) =& \frac{P(X \leq x, U \leq \frac{dP^*(X)}{KdR^*(X)})}{P(U \leq \frac{dP^*(X)}{KdR^*})} \\
=& \frac{E_R\left[P(X \leq x, U \leq \frac{dP^*}{KdR^*}|X)\right]}{E_R\left[P(U \leq \frac{dP^*}{KdR^*}|X)\right]} \\
=& \frac{E_R\left[I_{(X \leq x)} P(U \leq \frac{dP^*}{KdR^*}|X)\right]}{E_R\left[\frac{dP^*}{KdR^*}\right]} \\
=& \frac{E_R\left[I_{(X \leq x)} \frac{dP^*}{dR^*} \frac{1}{K}\right]}{\frac{1}{K}} \\
=& P^*(x).
\end{aligned}$$

Thus, $X \sim P(\cdot)$, as required. While it appears that x is implicitly assumed to be univariate, the above argument works just as well for the multivariate case, with the inequality above simply acting component-wise on the elements of x.

Now, the Monte Carlo standard error for $\hat{g}_N$ with accepted samples, $\{X_1, \cdots, X_N\}$ is

$$\text{SE}_{MC}(\hat{g}_N) = \sqrt{Var(g(X_1))}/N.$$

Thus computing Monte Carlo standard errors is trivial for a rejection sampler (or, for that matter, any sampler that produces i.i.d. draws from the distribution of interest).

Among useful variants of the rejection sampler are the empirical-sup rejection sampler (Caffo et al., 2002), the adaptive rejection sampler (Gilks and Wild, 1992) and the Metropolised rejection sampler (Tierney, 1994). A brief outline of only two of these methods is given here. For a nice overview of several related methods, the reader is referred to Gilks (1996).

E-sup Rejection Sampling

The empirical sup or E-sup rejection sampler (Caffo et al., 2002) works by replacing the exact supremum (the bound, K) with the maximum obtained from simulated candidates. This is similar to the approach adopted here, since the difficulty in calculating the supremum necessitates replacing it by an estimate. Caffo et al. (2002) use theoretical arguments and numerical work to show that a practically perfect sample may still be obtained in this manner. However, one significant difference between their approach and the approach described here is that even when a violation of the bound is detected, they keep the samples with the known incorrect bound. They provide diagnostics for failure of the method due to a bad choice of candidate distribution, and show how the bound converges to the right bound, by proving the existence of a central limit theorem for the samples thus obtained. In the work here, any time a sample is produced for which the bound does not hold, all samples that were produced up to that point are discarded; the bound is then updated, and the rejection sampling procedure is updated until a set of samples is obtained that does not violate the empirically determined bound. This avoids any dependence between the samples, and keeps the computation of Monte Carlo standard errors as simple as possible. Also, to the best of our knowledge, the correct supremum is used to obtain the final set of samples returned by the algorithm.

Metropolised Rejection Sampling

The metropolised rejection sampler (Tierney, 1994) deals with estimated suprema in a rejection algorithm by embedding the rejection sampler in a

Markov chain. To construct the algorithm, take the distribution R and a constant $\widehat{K}$ such that the set $C = \{x : P(x)/R(x) \leq \widehat{K}\}$ has reasonably high probability under R. Then, a rejection sampling algorithm can be implemented by generating independent variables $Y \sim R$ and $U \sim$Unif$(0,1)$, until (Y,U) is obtained s.t. $U \leq \frac{P(x)}{R(x)\widehat{K}}$. Then, the accepted pair has density $f(x) \propto \min\{P(x), \widehat{K}R(x)\}$. The Y thus obtained can be used as a candidate for a Metropolis-Hastings algorithm with acceptance probability

$$\alpha(x,y) = \begin{cases} 1 & \text{for } x \in C \\ 1/w(x) & \text{for } x \notin C, y \in C \\ \min\{w(y)/w(x), 1\} & \text{for } x \notin C, y \notin C. \end{cases}$$

where $w(\cdot) = P(\cdot)/\widehat{K}R(\cdot)$. If the envelope is adequate during the run, that is $\widehat{K}$ is large enough so every sample lies in C, then the chain produces i.i.d. samples from P, as desired. If the algorithm occasionally produces samples that lie outside C, then it will sometimes reject candidates, thereby introducing some dependence between samples, which compensates for the inadequacy of the envelope.

Regeneration

As an aside, the following subsection describes how the metropolised rejection scheme is a way to introduce regenerations into an MCMC sampler. A regenerative process is one where there is a sequence of random times at which the process starts over independently and identically. Regenerations in a Markov chain are the times at which the Markov chain starts over independently and identically. Identifying the regeneration times in a Markov chain can be very useful in analyzing an MCMC algorithm since the 'tours'

of the process between these times are i.i.d. and can therefore be used to construct reliable estimates of Monte Carlo standard error.

Let $\{X_t : t = 1, 2, \dots\}$ be a Markov chain on a state space $\mathcal{X}$. Let $P(x, A) = P(X_{t+1} \in A | X_t = x)$ be the transition probabilities, and denote the higher-order transition probabilities by $P^k(x, A) = P(X_{t+k} \in A | X_t = x)$. A subset $\mathcal{S} \subseteq \mathcal{X}$ is a *small set*, or (k_0, ϵ, ν)-small if there exists a probability measure ν on $\mathcal{X}$, a positive integer k_0, and $\epsilon > 0$ such that the *minorization condition* is satisfied

$$P^{k_0}(x, A) \geq \epsilon \, \nu(A), \quad x \in S, \quad A \subseteq \mathcal{X}.$$

If a set is (k_0, ϵ, ν)-small, then it is also (k_0, ϵ', ν)-small for any $\epsilon' < \epsilon$. Thus, an underestimate of ϵ may still be useful (though perhaps not optimal) if we want to apply small set ideas.

One interpretation of the above definition of a small set is that, roughly, once the chain is in the set $\mathcal{S}$, the chain "forgets" its current state with probability ϵ, and will simply jump to the distribution ν, ignoring its current state. When the chain makes such a jump, it "regenerates", since it is effectively starting over again from the distribution ν. Small sets are also useful for *coupling* chains, a notion that will be described in the context of the perfect simulation algorithm in Section 3.2.

In the metropolised rejection sampler, whenever the current state of the chain is in C, any candidate is accepted, i.e., the next state is generated as a draw from the proposal, irrespective of the particular state within C. Thus, visits to C, the small set, represent regeneration times since the chain 'starts over' at that point (it has no memory of the previous states). For further background on regeneration, see Nummelin (1984), Meyn and Tweedie (1993)

and Mykland, Tierney and Yu (1995); for an introductory level description see Tierney (1996).

3.1.2 Importance Sampling

Although this is not a major focus of this thesis, importance sampling is a very powerful Monte Carlo method and can easily be used in the context of our problem. It is therefore described here for completeness. Let $P(x)$ be an unnormalized density, and let $R(x)$ be a density that approximates $P(x)$. Say the goal is to compute the expectation for some integrable function $g(x)$ with respect to $P(x)$ (a normalized version), then we need to compute

$$E_P[g(X)] = \frac{\int g(x)P(x)dx}{\int P(x)dx} = \frac{\int g(x)\frac{P(x)}{R(x)}R(x)dx}{\int \frac{P(x)}{R(x)}R(x)dx} = \frac{\int g(x)w(x)R(x)dx}{\int w(x)R(x)dx},$$

where $w(x) = \frac{P(x)}{R(x)}$ are the importance sampling weights. It does not matter whether the $R(x)$ used in the ratio $w(x)$ is normalized or not, since the normalizing constant cancels out from the numerator and denominator. We can then estimate the expectation of g by generating a sample $X_1, \cdots, X_N \overset{i.i.d.}{\sim} R(\cdot)$ and computing $\hat{g}_N =$

$$\frac{\sum_{i=1}^N g(X_i)w(X_i)/N}{\sum_{i=1}^N w(X_i)/N} = \frac{\sum_{i=1}^N g(X_i)w(X_i)}{\sum_{i=1}^N w(X_i)} \to \frac{\int g(x)P(x)dx}{\int P(x)dx} \text{ as } N \to \infty.$$

To summarize, the importance sampling algorithm is as follows:

1. Generate $X_1, \cdots, X_N \overset{i.i.d.}{\sim} R(\cdot)$.

2. To estimate E(g(X)) for $X \sim P(\cdot)$, compute

$$\hat{g}_N = \frac{\sum_{i=1}^N g(X_i)w(X_i)}{\sum_{i=1}^N w(X_i)}, \text{ where } w(X_i) = \frac{P(X_i)}{R(X_i)}.$$

In theory, the same sample can be used to estimate moments of several functions, g(X) of *several* distributions of interest. The importance sampler also has the advantage that, unlike the rejection sampler, it does not require a distribution that envelopes the distribution of interest, though having the importance density be an envelope automatically implies that the associated C.L.T. holds and that there is a consistent estimate for the variance, provided the envelope has a finite second moment. If the envelope is a standard density, the existence of a finite second moment is typically already established. The importance sampler also does not require explicit knowledge of an upper bound on the ratio between $R(\cdot)$ and $P(\cdot)$. However, if the weights of a large number of samples are very small (corresponding to a rejection sampler where the acceptance rates are very low), the importance sampler necessitates 'carrying around' a very large number of samples that are not very useful. Computing Monte Carlo errors is also more difficult for an importance sampler. As an aside, note that if $g(X_i)w(X_i)$ is constant for each i then the Monte Carlo standard error is 0, so importance sampling can theoretically be used to *reduce* Monte Carlo standard errors. This is actually how "importance sampling" gets its name - from its use as a variance reduction method. Of course, dramatic reductions in Monte Carlo standard error are typically hard to obtain in realistic problems.

Thus, while there are benefits of chosing to use importance sampling rather than rejection sampling, there are several situations in which rejection sampling (or a closely related variant) may be much more practical. This is particularly true in situations where the method of generating samples from the enveloping density is reasonably cheap. In fact, the spatial

models considered here appear to fall into this category, i.e., a rejection sampler seems more practical, particularly because most of the samples that need to be saved for importance sampling computations can be of fairly high dimension, but may actually have very small importance weights.

3.2 Perfect Simulation

Since the perfect tempering algorithm described in this chapter is a "Coupling from the Past" (CFTP) algorithm, this section describes some of the basic ideas behind CFTP. For a gentle introduction to perfect simulation and CFTP algorithms, see Casella et al. (2001) and the references therein. This section also gives a general idea of approaches that have been proposed for generalizing CFTP algorithms for routine use in Bayesian inference.

3.2.1 Introduction to Coupling from the Past

Coupling Markov Chains

Recall from the description of regeneration in Section 3.1.1, that a Markov chain $\{X_t, t = 0, 1, \ldots, \}$ on state space $\mathcal{X}$ with transition probabilities $P(x, A)$ satisfies a minorization condition if for some $\mathcal{S} \subseteq \mathcal{X}$, and $\epsilon > 0$,

$$P^k(x, A) \geq \epsilon\nu(A), \quad x \in S, \quad A \subseteq \mathcal{X} \tag{3.2}$$

If we run multiple copies of the chain, which all happen to be in the small set $\mathcal{S}$ at the same time t, then we can construct the copies jointly so that with probability ϵ, at time $t + k$ they are all in the same exact state. In practice, Markov chains are coupled if they use an identical source of randomness,

typically the same sequence of uniform random variates. This can be more easily understood in the context of the 'splitting" construction of Nummelin (1984). We can define the "residual kernel"

$$\text{Res}(x, A) = (1 - \epsilon)^{-1}(P(x, A) - \epsilon\nu(A)),$$

then $\text{Res}(x, \cdot)$ is a probability measure on $\mathcal{X}$. Also,

$$P(x, A) = \epsilon\nu(A) + (1 - \epsilon)\text{Res}(x, A), \quad A \subseteq \mathcal{A}, \quad x \in \mathcal{S}$$

Hence, if the Markov chain enters the small set, i.e., $X_t = x \in \mathcal{S}$, we can determine the value of X_{t+1}, by simply flipping an ϵ-coin (a coin with an ϵ probability of heads). If the coin is heads, we draw $X_{t+1} \sim \nu$, if it is tails, we draw $X_{t+1} \sim \text{Res}(x, \cdot)$. In practice, it may be difficult to draw from $\text{Res}(x, \cdot)$, but Mykland et al. (1995) provide a technique, originally due to Nummelin (1984), that solves this problem.

The Propp-Wilson algorithm

In a seminal paper Propp and Wilson (1996) outline a clever technique by which they use Markov chains to produce samples from the stationary distribution of interest, rather than samples that are merely draws from the stationary distribution *asymptotically*. In essence, their idea is as follows: consider running an ergodic simulation of a Markov chain using a sequence of random numbers with a known distribution. If the simulation had been running infinitely long, then at time 0, a sample from the chain would have the exact stationary distribution of the chain. By figuring out the state at time 0 by looking at a finite number of the same sequence of random numbers

used in the recent past, we can obtain a sample that is from the stationary distribution of the chain. The Propp-Wilson algorithm determines on its own when to stop and produces samples from the exact stationary distribution of the chain.

The Propp-Wilson algorithm involves running coupled chains from a distant point in the past up until the present. Furthermore, the algorithm is designed so it is able to determine, during the running of the algorithm itself, the distance into the past that one needs to go. In their paper, they apply the algorithm succesfully to Gibbs distributions associated with various statistical mechanics models, including the Ising model, or choose uniformly at random from the elements of a finite distributive lattice. What follows is a brief overview of their algorithm.

The basic idea underlying the Propp-Wilson algorithm (sometimes called monotone coupling from the past) is to simulate the Markov chain by performing random moves until some predetermined amount of time has elasped, in the hope that all the chains started at all possible starting points coalesce or come together in that time. If this has not happened, then the algorithm restarts the chain further back in time, adding new random moves to the beginning of the chain (but keeping the old moves the same). If this process is repeated enough times, and we move back in time far enough, eventually the chains will coalesce, and the result will be an exact draw from the stationary distribution.

Of course, when the number of states is large, the algorithm outlined in the previous paragraph is not feasible. However, the algorithm may become feasible if it is possible to impose a partial ordering on the state-space, and

if there is a way to couple the Markov chain with itself so that it respects the partial ordering of the state-space under time-evolution. This coupling helps ascertain that coalescence has occurred by simply studying two Markov chains — one whose initial state was the maximal element and the other with the minimal element of the state space as its initial state. The following subsection describes some of the basic theory underlying the CFTP algorithm.

3.2.2 Coupling from the Past: General Theory

Define a single update of an ergodic (irreducible and aperiodic) Markov chain with n states, numbered 1 through n to be MCupdate$(\cdot)$. Given a current state, i, MCupdate returns a new state j. MCupdate$(\cdot)$ is a randomized subroutine usually specified by the Metropolis-Hastings algorithm and an associated sequence of uniform random variates. Let MCupdate construct a chain that has the distribution of interest, P, as its stationary distribution. A standard Markov chain Monte Carlo algorithm started at time $-M$ at initial state X^* would then proceed as follows:

Fixed Time Forward Simulation Algorithm

$X_{-T} \rightarrow X^*$
For $t = -M$ to -1
    $X_{t+1} \rightarrow$ MCupdate(X_t)
return X_0.

Propp and Wilson call the above algorithm a fixed time simulation since M is fixed a priori. The problem with this algorithm, of course, is that it is hard to determine how large M needs to be for X_0 to be "close enough" to being a

draw from the stationary distribution, P. "Close enough" is usually defined in terms of the total variational distance between P and the distribution from which X_0 is drawn. The total variational distance between two probability distributions, p_1, p_2 on a state space $\mathcal{X}$ with associated Borel σ-algebra, $\mathcal{B}$, is defined as

$$||p_1 - p_2|| = \sup_{A \subseteq \mathcal{B}} |p_1(A) - p_2(A)|$$

The Propp-Wilson algorithm gives us a way to circumvent this problem.

We first define a map from the state space to itself, $f_t(i) = \text{MCupdate}(i)$. For convenience, we can define RandomMap() to be a subroutine whose values are themselves functions from the state space to itself, and let $f_t() = \text{RandomMap}()$. In reality, an update of the Markov chain at state X_t will return the value X_{t+1}, by using a uniform random variate U_t and the Metropolis-Hastings algorithm. However, we leave the description of RandomUpdate abstract here since that will help fix general ideas about perfect simulation. Let the composition, $F_{t1}^{t2} = f_{t2} \circ f_{t2-1} \circ \cdots \circ f_{t1+1} \circ f_{t1}$. The output of the fixed time simulation algorithm above can then be written as $F_{-M}^{0}(X^*)$. Clearly, $F_t^0 = F_{t+1}^0 f_t$, so there is no need to keep track of each individual map. It is useful to think about the algorithms in terms of these maps because of the following observation: if a map $F_{t^*}^0$ becomes a constant map, i.e., if $F_{t^*}^0(i) = F_{t^*}^0(j)$ for all i, j in the state space, then this condition will hold for any $t \leq t^*$. When $F_{t^*}^0$ becomes a constant map, "coalescence" is said to have occurred from time t^* to time 0, and there is no need to go further back in time, since, for all $t \leq t^*$, F_t^0 is also a constant map. The algorithm is then:

Backward Simulation Algorithm

$t \rightarrow 0$

$F_t^0 \rightarrow$ the identity map

repeat

$t \rightarrow t - 1$

$f_t \rightarrow$ RandomUpdate()

$F_t^0 \rightarrow F_{t+1}^0 \circ f_t$

until F_t^0 is a constant map

return the unique value in the range of F_t^0.

Theorem 1 (Propp-Wilson): With probability 1, the coupling-from-the-past protocol returns a value, and this value is distributed according to the stationary distribution of the Markov chain.

Proof: There exists an L such that, for all states i and j, there is a positive chance of going from state i to state j in L steps (by the ergodicity of the Markov chain). Hence, for each t, F_{t-L}^t has a positive chance of being constant. Thus, each of the maps $F_{-L}^0, \ldots F_{-2L}^{-L}$ has some positive probability $\epsilon > 0$, of being constant. Since the events $F_{-L}^0, \ldots F_{-2L}^{-L}$ are independent (by construction, since they use independent uniform random variates), with probability 1, one of these maps must be constant. Hence, F_{-M}^0 is constant for sufficiently large M. When the algorithm reaches back M steps into the past, it will terminate and return a value, $F_{-\infty}^0$. Since $F_{-\infty}^0$ is obtained from $F_{-\infty}^{-1}$ by running the Markov chain one step further, $F_{-\infty}^0$ and $F_{-\infty}^{-1}$ have the same probability distribution. Thus, the output $F_{-\infty}^0$ is distributed according to the unique stationary distribution.

By determining a constant map, we have essentially ascertained what would have happened if we had run the Markov chain from the indefinite past (time $t = -\infty$) until the present (time $t = 0$). The distance into the past is determined dynamically by the particular sequence of U_ts used to generate the backward simulations. The unique value returned by the algorithm above is therefore a draw from the desired distribution, P.

It is essential in the CFTP algorithm that we simulate from the past up to the present. If we simply ran the chain forward from time 0 into the future, found the smallest M such that the value of F_0^M is independent of x and then output that value, the resulting samples would not follow the exact stationary distribution. For instance, if a Markov chain has some states with a unique predecessor, then such states can never occur at the exact instant when all n histories coalesce. Thus, such states would never be returned by the algorithm and the set of 'perfect' samples obtained would not be distributed according to P.

We can now look at how the above algorithm is implemented. We start by making explicit the fact that each constant map is determined for a *fixed* sequence of uniform random variates $\{\ldots, U_{-2}, U_{-1}\}$. Let $\psi(i, U_t)$ be the deterministic function that returns the next state of the Markov chain, i.e., $f_t(i) = \psi(i, U_t)$. Then, we have the following theorem:

Theorem 2 (Propp-Wilson): Let $\{\ldots, U_{-2}, U_{-1}\}$ be i.i.d. random variables. Assume that with probability 1, there exists t for which the map F_t^0 is constant, with a constant value that we may denote by $\psi(\ldots, U_{-2}, U_{-1})$. Then the random variable $\psi(\ldots, U_{-2}, U_{-1})$, which is defined with probability 1, has distribution governed by π.

Since, in practice, several different Markov update rules may be used to produce a single Markov chain, and one may cycle among them, Propp and Wilson prove a more general version of the above theorem.

Theorem 3 (Propp-Wilson): Let $\{\ldots, U_{-2}, U_{-1}\}$ be i.i.d. random variables, and let $\psi_t(\cdot,\cdot)(t<0)$ be a sequence of deterministic functions with the property that for all t and j,

$$\sum_i P(i) Prob(\psi_t(i, U_t) = j) = P(j),$$

so $\psi_t(\cdot,\cdot)$ preserves the stationary distribution, P. Define $f_t(i) = \psi_t(i, U_t)$ and $F_t^0 = f_{-1} \circ f_{-2} \circ \cdots \circ f_t$. Assume that with probability 1, there exists t for which the map F_t^0 is constant, with a constant value that we may denote by $\psi(\ldots, U_{-2}, U_{-1})$. Then the random variable $\psi(\ldots, U_{-2}, U_{-1})$, which is defined with probability 1, has distribution governed by π.

Proof: Let X be a random variable on the state-space of the Markov chain governed by the steady-state distribution P, and for all $t \leq 0$, let Y_t be the random variable $F_t^0(X)$. Each Y_t has distribution P, and the sequence $Y_{-1}, Y_{-2}, \ldots$ converges almost surely to some state $Y_{-\infty}$, which must also have distribution P. Put $\psi(\ldots, U_{-2}, U_{-1}) = Y_{-\infty}$.

3.2.3 General-Purpose Perfect Simulation

The Propp-Wilson algorithm is useful for exact sampling with Markov chains on large state spaces. However, since monotone-CFTP relies on the Markov chain having special structure, the algorithm they describe does not apply universally, even though it is practical for a large number of Markov chains. Since the original Propp-Wilson paper, perfect simulation has be-

come an area of active research (see, for instance, David Wilson's website on perfect sampling at `http://dimacs.rutgers.edu/~dbwilson/exact`). In particular, there have been several papers that have worked towards more general purpose perfect sampling algorithms that would be useful in practical settings such as Bayesian inference. Murdoch and Green (1998) describe methods based on gamma-coupling and rejection sampling that are relatively straightforward to understand, but require a closed form for the transition kernel and entail cumbersome algebraic manipulation. Green and Murdoch (1999) describe a perfect simulation algorithm for continuous state spaces, including several coupling techniques aimed at routine application in the context of Bayesian inference using random walk Metropolis algorithms. Møller (1999) extends CFTP ideas to infinite or uncountable state spaces such as autogamma, auto-Poisson and autonegative binomial models, using Gibbs sampling in combination with sandwiching methods originally introduced for perfect simulation of point processes. However, as clearly stated in Møller (1999), the methods he describes are unable to handle conditional autoregressions (which arise in the spatial models considered here). The algorithm studied here is the perfect tempering algorithm described in Moller and Nicholls (1999), which attempts to extend the practicality of perfect simulation to somewhat more general Bayesian hierarchical modeling situations.

Although the work in this thesis does not show that perfect simulation is practical for general Bayesian hierarchical modeling contexts, the contributions here demonstrate that, given some structure in the model, it may be feasible to construct perfect samplers for fairly complex, high-dimensional, continuous distributions. The successful application of these methods to pop-

ular disease mapping models (Chapter 4), and to standard Bayesian variance components models (Chapter 5), demonstrates that perfect samplers may be available for some commonly used, realistic models.

3.3 Perfect Tempering

This section provides a short review of simulated tempering, along with an overview of the perfect tempering algorithm, borrowing notation from Møller and Nicholls (1999).

3.3.1 Simulated Tempering

Let P be the distribution of a random variable X. Suppose we wish to sample from the distribution P. The simulated tempering algorithm involves defining a Markov chain on a mixture of distributions where P is one of the distributions in the mixture. Each distribution in the mixture has an associated level (or "temperature" from the analogy to the metallurgical process which gives simulated tempering its name). Denote the tempering levels by $\mathcal{T} = \{0, 1, .., N^*\}$, with $N^* \geq 1$. In general, the "cold distribution" H_{N^*} is set to P, the distribution of interest. The H_N distributions 'interpolate' between H_0 and the target distribution, P, moving from distributions that are easy to sample from to distributions that are more difficult to sample from, but closer to the target distribution.

Let Ω_n be the state space corresponding to the tempering level, n. Define $\Omega = \bigcup_{n=0}^{N^*}(\Omega_n \times \{n\})$. We can use a Metropolis-Hastings algorithm to generate a Markov chain $\{Z_t\} = \{(X_t, N_t)\}$ on the augmented state space

Ω, with invariant distribution H. When the current state is (X_t, N_t), the temperature for the next state of the Markov chain, n', is proposed according to some distribution $l(n, n')$, and the parameters for the next state, x', are proposed according to some distribution $f_{n,n'}(x, x')$. In the algorithm used here, a single Metropolis-Hastings accept-reject ratio, $\alpha((x, x'), (n, n'))$, is used to update the the proposed temperature and parameters though it is also possible to update the temperature and parameters using separate Metropolis-Hastings accept-reject steps. Distribution H is defined over the space Ω and is given as the π-weighted mixture $H(x, n) = \pi_n H_n(x)$, where each $\pi_n > 0$, $\sum \pi_i = 1$, $n = 1, \ldots, N^*$, is a constant. If $Z = (X, N) \sim H$, then the π_is, or 'pseudo-priors', control the marginal distribution of Z in its tempering variable N. Therefore, if we have samples from H, we can estimate expectations of interest, since, for real functions f with $E_P|f| < \infty$,

$$E_P|f| = \frac{E_H\{\sum_{t=0}^{L} f(X_t) I_{N_t=N^*}\}}{E_H\{\sum_{t=0}^{L} I_{N_t=N^*}\}}. \tag{3.3}$$

In essence, if we run the Markov chain $\{Z_t\}$, and only utilize the states at the temperature N^*, we can estimate expectations with respect to the distribution P. For a more detailed description of simulated tempering, see Geyer and Thompson (1995) and Marinari and Parisi (1992).

Regenerations in Simulated Tempering

Suppose we choose the "hot distribution", H_0, such that independent sampling is possible from it — this is necessary for introducing i.i.d. draws in the sampler. Then, every time the chain is at temperature $n = 0$, the next value of the chain is drawn independently of previous values of the chain, and

hence the chain "regenerates". The simplest way to do this is to assume that H_0 is a distribution with point mass on an atom, $\mathbf{0}$, say. Regeneration is useful for coupling Markov chains, and is thus helpful for implementing perfect simulation algorithms. If we know for a given sequence of uniform random variates, that a Markov chain is in the hot distribution (where it regenerates), then we can simply draw the next sample *without* needing to know the previous state of the chain. Thus, it is easy to introduce regenerations in a simulated tempering algorithm. This avoids the need to run the chain from infinitely far in the past to obtain its current state (at a regeneration point), and therefore forms the basis of the perfect tempering algorithm described in the next section. For more on regeneration, atoms and coupling refer to Sections 3.1.1, 3.2.1 and 4.5.

3.3.2 Perfect Simulation via Simulated Tempering

Møller and Nicholls (1999) use simulated tempering ideas in a CFTP framework, similar in spirit to the Propp-Wilson algorithm, to develop a perfect simulation algorithm which they call perfect tempering. This algorithm can, in principle, be used for distributions where the state space is unbounded. The following outline of the algorithm closely follows their description.

If it were possible to initialize $Z_0 \sim H$ (from the stationary distribution), then we could simply run the chain forward and obtain 'exact' samples from the stationary distribution, G, using the method described by (3.3). The issue then becomes one of obtaining $Z_0 \sim H$. This is done via a perfect simulation algorithm that uses a random walk $\{D_t\}$ on the temperatures $\{0, ..N^*\}$ of the simulated tempering chain. The coupling of the chains occurs because

the *same* uniform random variates are used for updates of both chains. We construct $\{D_t\}$ such that it dominates $\{(X_t, N_t)\}$ in the following sense: if at time t, $D_t \geq N_t$, then at time $t+1$, $D_{t+1} \geq N_{t+1}$. Formally, this implies

$$Pr\{N_{t+1} \leq D_{t+1} | N_t = n, D_t = m\} = 1 \text{ whenever } n \leq m. \tag{3.4}$$

Denote a sequence of simulations of $\{D_t\}$ by $\{D_t\}^{(a)}$, where $\{D_t\}^{(a)}$ runs forward from a negative time $-T-a$, with T=1 (a much larger T can also be used if necessary). Denote a single update of the random walk chain $\{D_t\}$ by RWupdate$(m; u^1, u^2)$, and a single update of the simulated tempering chain by STupdate$(z; u^1, u^2)$, where u_1, u_2 are uniform random variates and m and z are the current state of the random walk and simulated tempering chain respectively. Details of both updates are given in Appendix B.3. The perfect tempering algorithm proceeds in two phases. **Phase 1** is as follows:

1. Fix a realization of uniform random variates, $\mathbf{u}$. For instance, we could have $\mathbf{u} = ((u^1_{-1}, u^2_{-1}), (u^1_{-2}, u^2_{-2}), \cdots)$. Initialize $a = 1$.

2. Simulate the random walk $\{D_t\}^{(a)}$, after initializing $D^{(a)}_{-T-a} = N^*$.

3. Consider three possible events: (A) where $\{D_t\}^{(a)} = 0$ for some $t < 0$, (B) where $\{D_t\}^{(a)} = \{D_t\}^{(a-1)}$ (it couples with the previous simulation) and (C) where $\{D_t\}^{(a)}$ reaches time $t = 0$ without (A) or (B) occurring. If (B) or (C) occurs, a new simulation is started one step further back in time, i.e., a is set to $a+1$, and we return to the previous step. If (A) occurs, terminate Phase 1 of the algorithm, and let a_0 be the index of this sequence. Let $-\tau^* = \inf\{-T - a_0 < t \leq 0; D^{a_0}_t\}$, so τ^* is the first hitting time for the $D^{a_0}_t$ simulation.

Note that every tempering path, simulated from $t \leq -\tau^*$ using the fixed random numbers $\mathbf{u}$, is in state $(\mathbf{0}, 0)$ at $t = -\tau^*$. We keep τ^* and $\mathbf{u}$ from Phase1, and start **Phase 2**:

1. Set $Z_{-\tau^*} = (\mathbf{0}, 0)$ and simulate forward from $t = -\tau^*$ up to $t = -1$ using $Z_{t+1} = \text{STupdate}(Z_t; u_t^1, u_t^2)$.

2. We know $Z_0 \sim H$. If $N_0 = N^*$, then we accept the sample X_0. If $N_0 \neq N^*$, we reject the sample and start over with Phase 1 again.

This algorithm terminates with probability 1, returns $\tau^* < \infty$ and $Z_0 \sim G$. A proof of this result is obtained by closely following the proof of Theorem 2 in Propp and Wilson (1996), as described in the previous section.

Figure 3.2 gives a flavour for how the algorithm works by following the trajectory of two coupled random walks $\{D_t\}^{(a)}, a = 1, 2$ on the temperatures $\{0, 1, 2\}$. Two scenarios are considered, Scenario A and Scenario B. In Scenario A, u_{-1} instructs the random walk to stay at the same level when it moves from time $t = -1$ to $t = 0$. The random walk that starts at time $t = -1$ is denoted by a thick line, and the random walk that starts at time $t = -2$ by a thin line. Since $D^{(1)}$ is started at level 2 at $t = -1$, it stays at level 2 at $t = 0$. Now, $D^{(2)}$ is also started at level 2 at $t = -2$. There are two possibilities when it moves to $t = -1$: one where it stays at level 2, and the other where it moves down to level 1. In the first case, it will automatically move where $D^{(1)}$ moved (since it uses the same value u_{-1}. In the second case, it can either move up one level (from level 1 to level 2) or stay at level 1. In Scenario B, u_{-1} 'instructs' the random walk $D^{(1)}$ to move down one level (to level 1) from time $t = -1$ to $t = 0$. If we follow the trajectory of $D^{(2)}$, in the

same way as we did for Scenario A, we find, again, that $D^{(2)}$ can never be at a higher level than $D^{(1)}$ at any time where they two chains overlap.

In general, if we were to follow the possible trajectories of all successive random walks, we would find that,

$$\text{For all } t \geq a_2, \quad D_t^{(a_1)} \leq D_t^{(a_2)} \quad \text{when } a_1 < a_2,$$

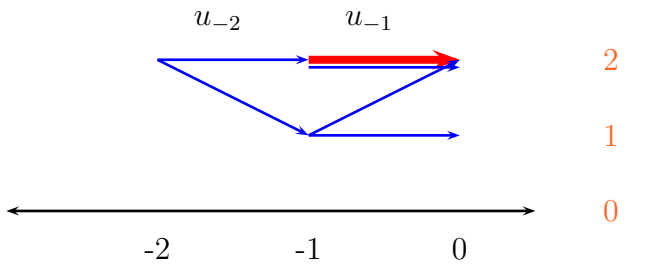
that is, whenever the two chains overlap, $D_t^{(a_1)}$ is never any higher than $D_t^{(a_2)}$ for the overlapping portions of the chain. Hence, we have

$$\text{If } D_{-\tau^*}^{(a^*)} = 0, \text{ for some } (a^*, -\tau^*), \text{ then } D_{-t}^{(a^*)} = 0 \text{ when } -t \leq -\tau^*.$$

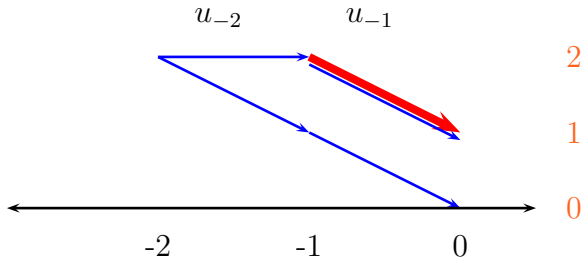
This is the situation illustrated in Figure 3.3. By the domination condition, we then automatically have $Z_{-\tau^*} = (\mathbf{0}, 0)$, for this particular sequence of uniform random variates, $((u_{-1}^1, u_{-1}^2), (u_{-2}^1, u_{-2}^2), \cdots)$. We could then simple run $\{Z_t\}$ chain forward from time $t = -\tau^*$ without having to know any of the previous states of the chain. It is therefore as if we had run the chain from time $-\infty$, without ever having to physically do so.

Let the Metropolis-Hastings acceptance probabilities for an update of the chain $\{D_t\}$ be $\tilde{\alpha}(n, n')$, where n is the current state and n' is the proposed future state. Similarly, let $\alpha((x, n), (x', n'))$ be the acceptance probability for the $\{Z_t\}$ chain, where (x, n) is the current state and (x', n') is the proposed future state. For the domination condition (3.4) to hold, we need $\tilde{\alpha}$ and α to satisfy the following conditions for all $x, x' \in \Omega$.

$$\begin{aligned} \tilde{\alpha}(n, n+1) &\geq \alpha((x, n), (x', n+1)) \\ \tilde{\alpha}(n, n-1) &\leq \alpha((x, n), (x', n-1)) \\ \tilde{\alpha}(n, n) &\geq \alpha((x, n), (x', n)). \end{aligned} \tag{3.5}$$



Scenario A



Scenario B

Figure 3.2: Trajectories of random walks on $\{0, 1, 2\}$ (thick line=random walk started at $t = -1$, thin line=random walk started at $t = -2$).

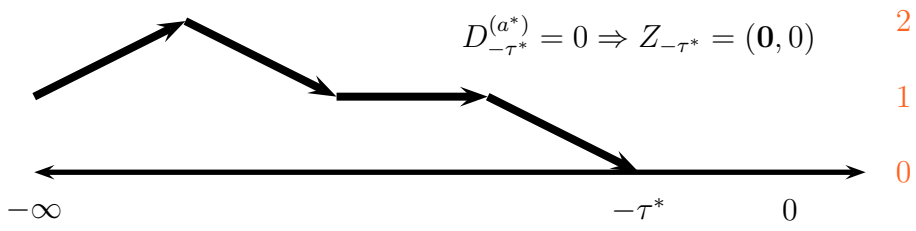


Figure 3.3: Identifying τ^* in a simulated tempering chain

The conditions in (3.5) can be satisfied if $\forall(x, n) \in \Omega$, with $n < N^*$, the Hastings ratio,

$$\begin{aligned} r((x,n),(x',n+1)) =& 1/r((x',n+1),(x,n)) > 0, \\ K_n \times \frac{\pi_{n+1}}{\pi_n} \geq& r((x,n),(x',n+1)). \end{aligned} \tag{3.6}$$

The first condition above requires symmetry and positivity of the Hastings ratio, while the second condition is a rejection sampling-like condition that requires that we have an envelope for the target distribution. Recall that an envelope for a distribution P on state space Ω is defined as any distribution R such that for some $B > 0$,

$$\sup_{X \in \Omega} P(X)/R(X) \leq B < \infty.$$

In fact, Møller and Nicholls (1999) show that if (3.6) is satisfied then the simulated tempering chain $\{Z_t\}$ is a uniformly ergodic chain.

Perfect tempering is closely related to rejection sampling, but can be more efficient than a rejection sampler, as seen in Section 4.8. In perfect tempering, the rejection envelope can be used as a proposal distribution in

Metropolis-Hastings updates for the target distribution, or also as a tempering distribution in the simulated tempering schedule. Two different variants of perfect tempering were tried. The first algorithm has a simulated tempering schedule of three distributions, where H_0 is a point mass on the atom $\mathbf{0}$, H_1 is the enveloping distribution R, and H_2 is the target distribution P. The second algorithm has only two tempering distributions with H_0 as before, and H_1 as the target distribution. For both algorithms, the envelope, R, is used as the proposal for sampling from the target distribution. The second version of the algorithm was found to be more efficient. We describe, in fairly general terms, two schemes for two-temperature perfect tempering here. They differ in terms of the distribution H_0 : while in Scheme A the state space for H_0 is a distribution on a point mass, in Scheme B, H_0 is the enveloping R distribution on the entire state space. In practice, Scheme A requires less time for the generation of proposals, since at temperature $t = 0$, the atom is generated with probability 1. The results in this chapter are therefore for Scheme A. Details of the algorithm are given in Appendix B.3.

Scheme A: Let $H_0 = \mathcal{A}$ with probability 1, let $H_1 = P(\cdot)$, and the 'pseudo-priors' on H_0, H_1 be π_0, π_1. $\mathcal{A}$ is an atom. Let p_{ij} be the probability of proposing level j when the current level is i. The proposal distributions are set as follows: $f_{00} = 1_{\mathcal{A}}\ p_{00}$, $f_{10} = 1_{\mathcal{A}}\ p_{10}$, and leave f_{01}, f_{11} as general proposals, where p_{ij} is the probability that level j is proposed when in state i.

The appropriate ratios (r's) for the Metropolis-Hastings algorithms are

$$\begin{aligned}
r((\mathcal{A},0),(x',1)) =& \frac{H_1(x')\pi_1 f_{10}((\mathcal{A},0))}{H_0(\mathcal{A})\pi_0 f_{01}((x',1))} = \frac{\pi_1 H_1(x') f_{10}((\mathcal{A},0))}{\pi_0 f_{01}((x',1))}. \\
r((\mathcal{A},0),(\mathcal{A},0)) =& \frac{H_0(\mathcal{A})\pi_0 f_{00}((\mathcal{A},0))}{H_0(\mathcal{A})\pi_0 f_{00}((\mathcal{A},0))} = 1. \\
r((x,1),(\mathcal{A},0)) =& \frac{H_0(\mathcal{A})\pi_0 f_{01}((x,1))}{H_1(x)\pi_1 f_{10}((\mathcal{A},0))} = \frac{\pi_0 f_{01}((x,1))}{H_1(x)\pi_1 f_{10}((\mathcal{A},0))}. \\
r((x,1),(x',1)) =& \frac{H_1(x')\pi_1 f_{11}((x,1))}{H_1(x)\pi_1 f_{11}((x',1))} = \frac{H_1(x') f_{11}((x,1))}{H_1(x) f_{11}((x',1))}.
\end{aligned}$$

The notation here implicitly assumes that the proposals do not depend on the current state. In the algorithm, the proposals are set as follows $f_{01} = R(\cdot)\, p_{01}$, $f_{11} = R(\cdot)\, p_{11}$, where $R(\cdot)$ is a rejection envelope.

Scheme B: Let H_0 be a distribution on the entire state space. H_0 will typically be $R(\cdot)$. Everything else stays the same as in Scheme A. The corresponding Metropolis-Hastings ratios are

$$\begin{aligned}
r((x,0),(x',1)) =& \frac{H_1(x')\pi_1 f_{10}((x,0))}{H_0(x)\pi_0 f_{01}((x',1))}. \\
r((x,0),(x',0)) =& \frac{H_0(x')\pi_0 f_{00}((x,0))}{H_0(x)\pi_0 f_{00}((x',0))} = \frac{H_0(x') f_{00}((x,0))}{H_0(x) f_{00}((x',0))}. \\
r((x,1),(x',0)) =& \frac{H_0(x')\pi_0 f_{01}((x,1))}{H_1(x)\pi_1 f_{10}((x',0))}. \\
r((x,1),(x',1)) =& \frac{H_1(x')\pi_1 f_{11}((x,1))}{H_1(x)\pi_1 f_{11}((x',1))} = \frac{H_1(x') f_{11}((x,1))}{H_1(x) f_{11}((x',1))}.
\end{aligned}$$

Chapter 4

Exact Methods for Bayesian Disease Mapping

This chapter describes how the exact sampling methods in Chapter 3 may be applied to the spatial models discussed in Chapter 2. The block sampling methods of Chapter 2 are fairly successful in helping the Markov chains explore the state space efficiently for these models, but they do not always perform optimally for all situations. More importantly, as briefly discussed in Chapter 1, it is generally hard to assess just how "well" a Markov chain Monte Carlo algorithm is performing. Exact Monte Carlo methods, circumvent all the usual diagnostics issues that arise with regular MCMC. The problem with exact simulation, of course, is that it is rarely practical for multidimensional continuous posterior distributions. Exact simulation is generally not even considered an *option* for the kind of models studied here. This chapter describes systematic algorithms to produce i.i.d. draws from posterior distributions for the spatial Poisson models, and discuss how these methods work

in practice.

The method of integrating out model parameters to sample from a lower dimensional marginal distribution was explored in work by Wolfinger and Kass (2000), Everson and Morris (2000) and Everson (2001) for variance component models and Gamerman et al. (2002) for space-varying regression models. However, a major difference in the approach of these authors is that they use Markov chain Monte Carlo techniques to sample from the marginal distribution of the variance components, and therefore do not obtain i.i.d. draws from the distribution of interest. Also, these authors deal with models where it is possible to analytically obtain the exact marginal distribution of the variance components. The remaining model parameters can then easily be sampled from their posterior distribution, conditional on the variance components. In contrast, for the spatial models considered here, exact formulas are neither available for the marginal distribution of the variance parameters nor for the conditional distributions for the remaining model parameters. The exact sampler described here therefore needs to sample the *entire* joint posterior distribution at the same time, since sampling from the marginal distribution of the variance components is not feasible. Even if the number of variance components is held constant, the sampling problem can become progressively harder with increase in the number of regions. The increase in the number of regions leads to an increase in the number of random effects in the spatial model (we have to deal with the "curse of dimensionality"). In contrast, the difficulty of sampling from the normal random effects variance components models does not increase in the same way as the number of random effects for such models grows, as long as the number of variance

components stays fixed. For instance, Chapter 5 describes a situation where an increase in the number of random effects does not complicate the marginal distribution of the variance components. Since the conditional distribution of the remaining parameters is a normal distribution, i.i.d. sampling from the full posterior distribution for such models is not affected as much by the increase in the number of random effects.

4.1 Spatial Modeling of Areal Data

This section describes two different spatial models for areal data (i.e., data arising as sums or averages over geographic regions). Let the number of regions be N and the number of adjacencies (number of pairs of regions that are adjacent to one another) be J. The number of disease events in region i is Y_i for all models.

4.1.1 Model 1

This model contains parameters that capture the spatial structure in the data. Let these spatial or clustering parameters be represented as $\boldsymbol{\phi}$. The model is then described as follows :

$$Y_i|\phi_i \sim \text{Poi}(E_i e^{\phi_i}) \tag{4.1}$$

where E_i is a fixed estimate of disease events in region i, which does not account for differences in disease risk from region to region. E_i is determined before obtaining the data. ϕ_i is the log-relative risk of disease for the ith region. The clustering parameters are described by a conditionally au-

toregressive (CAR) model. They are assumed to follow a Gaussian Markov random field (GMRF), and their distribution is specified as

$$\phi_i|\phi_{j\neq i} \sim N(\mu_{\phi_i}, \sigma^2_{\phi_i}), i = 1, ..., N \ ,$$
$$\text{where } \mu_{\phi_i} = \frac{\sum_{j\neq i} w_{ij}\phi_j}{\sum_{j\neq i} w_{ij}} \text{ and } \sigma^2_{\phi_i} = \frac{1}{\tau_c \sum_{j\neq i} w_{ij}}.$$
$$w_{ij} = 1 \text{ if } i\text{th and } j\text{th region are neighbors, else 0.}$$

The μ_{ϕ_i} for a region i is thus a weighted average of the clustering parameters in other regions. The prior on ϕ_i leaves the overall level of the GMRF unspecified; the prior is therefore improper due to translation invariance. The distribution can also be written as

$$\boldsymbol{\phi}|\tau_c \sim CAR(\tau_c) \propto \tau_c^{(N-1)/2} \exp\left(-\frac{\tau_c}{2}\boldsymbol{\phi}^T Q \boldsymbol{\phi}\right)$$

where

$$Q_{ij} = \begin{cases} n_i & \text{if } i = j \\ 0 & \text{if } i \text{ is not adjacent to } j \\ -1 & \text{if } i \text{ is adjacent to } j \end{cases}$$

To complete the Bayesian specification, a Gamma prior is placed on the precision parameter.

$$\tau_c \sim G(\alpha_c, \beta_c).$$

4.1.2 Model 2

This is the same model considered in Chapter 2. Besag, York and Mollie (1991) describe a spatial model using a Poisson likelihood for the areal data, with parameters that capture the clustering and heterogeneity in log-relative risks separately.

$$Y_i \sim \text{Poi}(E_i e^{\mu_i}) \tag{4.2}$$

where μ_i is the log-relative risk of disease for the ith region. μ_i is modeled linearly as

$$\mu_i = \theta_i + \phi_i \ , \ i = 1, ..., N,$$

where

$$\theta_i|\tau_h \sim N(0, 1/\tau_h), \ i = 1, ..., N,$$
$$\boldsymbol{\phi}|\tau_c \sim CAR(\tau_c)$$

The θ_i's are independent and identically distributed normal variables, while the ϕ_i's follow a CAR model (as in Model 1). In this way, each θ_i captures the ith region's extra-Poisson variability due to area-wide heterogeneity, while each ϕ_i captures the ith region's excess variability attributable to regional clustering. The priors on the precision parameters are independent Gamma distributions.

$$\tau_h \sim G(\alpha_h, \beta_h), \quad \tau_c \sim G(\alpha_c, \beta_c).$$

4.2 Proposal Distributions: Finding Envelopes

For each of the models considered, a Gaussian approximation to the likelihood is used. The model parameters ($\boldsymbol{\phi}$ and $(\boldsymbol{\phi}, \boldsymbol{\theta})$ for Models 1 and 2 respectively) can be integrated out from the corresponding approximate posterior distribution to obtain approximate marginal posterior distributions for the precision parameters (τ_c and (τ_h, τ_c) respectively). The approximate conditional distribution of the model parameters given the precision parameters can be derived analytically, and this distribution is used to guide the choice of a proposal distribution for the model parameters. The outline of the technique for generating a proposal is as follows:

1. Let the posterior distribution of interest be $P(\cdot|Y)$. Use a Gaussian approximation to the likelihood and a delta method approximation for the variance to obtain an approximate posterior distribution, S.

2. Analytically integrate out model parameters from S to get S_1, an approximate marginal posterior distribution for the precision parameters.

3. Based on S, find the approximate conditional distribution for the model parameters, S_2 (conditional on precision parameters).

4. Find a log-t distribution (or a bivariate log-t distribution for the two precision components), R_1, that is similar to S_1.

5. Find a multivariate-t distribution, R_2, that is similar to S_2.

6. Use numerical maximization or empirically determine the rejection sampling bound K (Caffo et al., 2002).

7. Generate a sample for the precision components from R_1, then generate model parameters from R_2, conditional on the precision components sampled. The sample of precision components and model parameters jointly constitute a proposal.

The following sections briefly describe how to obtain the approximate distributions (S_1, S_2) and the envelopes (R_1, R_2).

4.2.1 Approximate Marginal Distributions

This subsection provides an outline of the derivation of the approximate marginal distributions (S_1) of the precision parameters for each model. De-

tails are given in Appendix B.1.

Model 1

The Poisson likelihood (4.1) can be approximated as

$$Y_i \sim N(E_i e^{\phi_i}, E_i e^{\phi_i}) \tag{4.3}$$

Let $\hat{\mu}_i$ be $\log(Y_i/E_i)$. Using the delta method, we get

$$\hat{\mu}_i \approx N(\phi_i, 1/Y_i). \tag{4.4}$$

We can calculate the approximate full joint posterior distribution $S(\boldsymbol{\phi}, \tau_c)$, which we can then analytically integrate with respect to $\boldsymbol{\phi}$ to obtain $S_1(\tau_c)$. $S_1(\tau_c)$ serves as the approximate marginal posterior distribution for τ_c.

Model 2

Similarly, a normal approximation for (4.2) is

$$Y_i \sim N(E_i e^{\mu_i}, E_i e^{\mu_i}) \tag{4.5}$$

Let $\hat{\mu}_i$ be $\log(Y_i/E_i)$. Again, the delta method gives us

$$\hat{\mu}_i \sim N(\theta_i + \phi_i, 1/Y_i). \tag{4.6}$$

We can then analytically integrate the approximate joint posterior distribution $S(\boldsymbol{\Theta}, \tau_h, \tau_c)$ with respect to $\boldsymbol{\Theta}$ to obtain $S_1(\tau_h, \tau_c)$, the approximation to the marginal distribution of τ_h, τ_c.

4.2.2 Approximate Conditional Distributions

This section describes how we derive the approximate posterior distribution of the model parameters, conditional on the precision parameters.

Model 1

Using standard linear model results from Lindley and Smith (1972), we can obtain the approximate conditional distribution from $S(\boldsymbol{\phi}, \tau_c)$:

$$S_2(\boldsymbol{\phi}|\tau_c, \hat{\boldsymbol{\mu}}) \sim N\left(\left(V^{-1} + \tau_c Q\right)^{-1} V^{-1}\hat{\boldsymbol{\mu}}, \left(V^{-1} + \tau_c Q\right)^{-1}\right) \tag{4.7}$$

where V^{-1}=diag$(Y_1, ..Y_N)$, and $\hat{\boldsymbol{\mu}}, Q$ are as before.

Model 2

The hierarchical model can first be reexpressed in the "constraint case" formulation as described in Hodges (1998). Following Chapter 2, we can express Model 2 in the form :

$$\left[\begin{array}{c} \hat{\boldsymbol{\mu}} \\ \hline \mathbf{0_{(N+J)\times 1}} \end{array}\right] = \left[\begin{array}{c|c} I_{N\times N} & I_{N\times N} \\ \hline -I_{N\times N} & 0_{N\times N} \\ \hline 0_{J\times N} & A_{J\times N} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\theta} \\ \hline \boldsymbol{\phi} \end{array}\right] + E.$$

where $A_{J\times N}$ is a matrix with each row having a -1 and 1 in the kth and lth columns respectively, for each pair of adjacent regions (k, l). The rows of A represent the following pairwise difference form of the prior on the ϕ_i's

$$(\phi_i - \phi_j)|\tau_c \sim N(0, 1/\tau_c) \text{ for each } i, j \text{ that are adjacent regions.}$$

This is identical to (2.3) from Chapter 2. The above specification can be expressed as $Y = \mathrm{X}\boldsymbol{\Theta} + \mathrm{E}$, where X and Y are known, $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \boldsymbol{\phi})^T$ is unknown, and E is an error term with block diagonal covariance matrix

$$\mathrm{Cov}(E) \;=\; \Gamma \;=\; \left[\begin{array}{c|c|c} \Gamma_1 & 0 & 0 \\ \hline 0 & \Gamma_2 & 0 \\ \hline 0 & 0 & \Gamma_3 \end{array}\right], \tag{4.8}$$

where $\Gamma_1 = \text{Diag}(1/Y_1, .., 1/Y_N)$, $\Gamma_2 = \tau_h^{-1} I_{N\times N}$, $\Gamma_3 = \tau_c^{-1} I_{N\times N}$. It can be shown that the posterior density of $\boldsymbol{\Theta}$ is

$$S_2(\boldsymbol{\Theta}|\hat{\boldsymbol{\mu}}, \Gamma) \sim N((X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}\hat{\boldsymbol{\mu}}) \ , \ (X^T\Gamma^{-1}X)^{-1}) \ . \tag{4.9}$$

If we let $V^{-1} = \text{Diag}(Y_1, \ldots, Y_N)$, this is equivalent to

$$S_2(\boldsymbol{\Theta}|\hat{\boldsymbol{\mu}}, \Gamma) \sim N\left(C^{-1}(-\frac{1}{2}D^T) \ , \ C^{-1}\right) \ ,$$

where

$$C_{2N\times 2N} = \begin{bmatrix} \overbrace{V^{-1} + \tau_h I}^{\boldsymbol{\theta}} & \overbrace{+V^{-1}}^{\boldsymbol{\phi}} \\ +V^{-1} & V^{-1} + \tau_c Q \end{bmatrix},$$

and

$$D_{1\times 2N} = (-2\hat{\boldsymbol{\mu}}^T V^{-1}, -2\hat{\boldsymbol{\mu}}^T V^{-1}).$$

4.2.3 Envelopes from Approximate Distributions

A good envelope will have the property that it matches the distribution of interest well. It may seem reasonable therefore, as a first attempt, to use $S_1 S_2$ as an envelope for the posterior distribution P. However, $S_1 S_2$ turns out not be an envelope for the P distribution. The approach here is to use heavy-tailed distributions, R_1 and R_2, that have roughly the same shape and scale as S_1 and S_2 respectively. For Model 1, several different distributions for R_1 were tried. Among those considered were the gamma, Weibull and log-normal distributions, but a two parameter log-t distribution was found to be the most appropriate distribution due to its tail behavior and flexibility. For Model 2, the distribution used as $R_1(\tau_h, \tau_c)$ was a product of independent log-t distributions. For Model 1, a multivariate-t distribution was used as

$R_2(\boldsymbol{\phi};\tau_c)$, with the same mean and variance as $S_2(\boldsymbol{\phi}|\tau_c)$. Similarly, R_2 for Model 2 is obtained by using a multivariate-t analogue of $S_2(\boldsymbol{\theta},\boldsymbol{\phi}|\tau_h,\tau_c)$. It is easy to draw a sample from R_1R_2 for both models by sequential sampling. For instance, for Model 1, draw $\tau_c^* \sim R_1$, and, conditional on the value τ_c^*, draw $\boldsymbol{\theta}^*$ from $R_2(\boldsymbol{\theta}|\tau_c^*)$.

To obtain an R_1 reasonably similar to S_1 for Model 1, $S_1(\tau_c)$ was plotted on the log scale, and a 'matching' t-distribution with parameters μ_c, σ_c was found by visually matching the mode and variance of the two distributions. The corresponding log-t distribution, $R_1(\tau_c;\mu_c,\sigma_c)$, was used as the proposal for τ_c. For Model 2, the profiles of $S_1(\tau_h,\tau_c)$, $S_{1a}(\tau_h)$ and $S_{1b}(\tau_c)$ along τ_h and τ_c respectively, were plotted. Envelopes for the precision parameters were again found on the log-scale by matching t-distributions to the approximate log-scale profiles. The corresponding log-t distributions $R_{1a}(\tau_h)$ and $R_{1b}(\tau_c)$ were used jointly as the proposal, $R_1(\tau_h,\tau_c)$, for (τ_h,τ_c).

Proofs for Envelopes

Finally, to be sure that the proposal distribution, does in fact envelope the posterior distribution for each of the two models, some analytical work is required. Note that it is only necessary to show that R_1R_2 envelopes P, it is not necessary to prove that R_1 envelopes S_1 or that R_2 envelopes S_2. Appendix B.2 has proofs to that show that, for some constants K_1, K_2 to be found numerically, $K_1R_1(\tau_c)R_2(\boldsymbol{\phi};\tau_c)$ is an envelope for the Model 1 posterior distribution, $P(\boldsymbol{\phi},\tau_c)$, and $K_2R_1(\tau_h,\tau_c)R_2(\boldsymbol{\theta},\boldsymbol{\phi};\tau_h,\tau_c)$ is an envelope for the Model 2 posterior distribution, $P(\boldsymbol{\theta},\boldsymbol{\phi},\tau_h,\tau_c)$.

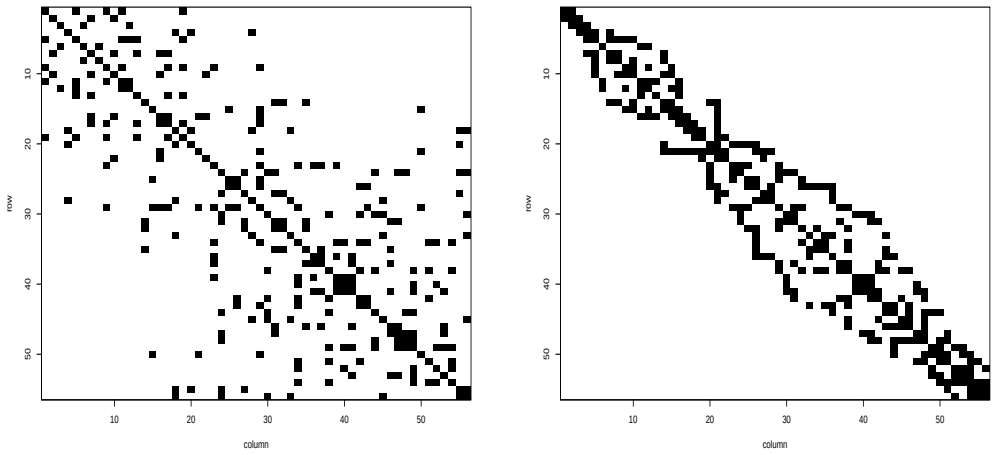
4.3 Using Sparse Matrix Algorithms

It is easy to see that the steps that take up most of the computational time involve operations on large matrices. In particular, computing the proposal distribution for the random effects parameters involves several matrix operations for each accept-reject step of the algorithm. We can exploit the structure of the matrices to dramatically speed up computation time. Sparse matrix algorithms can be very helpful in this context.

A sparse matrix is defined, somewhat vaguely, as a matrix which has very few nonzero elements (Saad, 1996). Special techniques can be utilized to take advantage of the large number of zero elements and their locations. This section describes how to exploit the sparsity of the matrices that naturally arise in the spatial models considered. The techniques used to accelerate computation are similar to those described in Rue (2001).

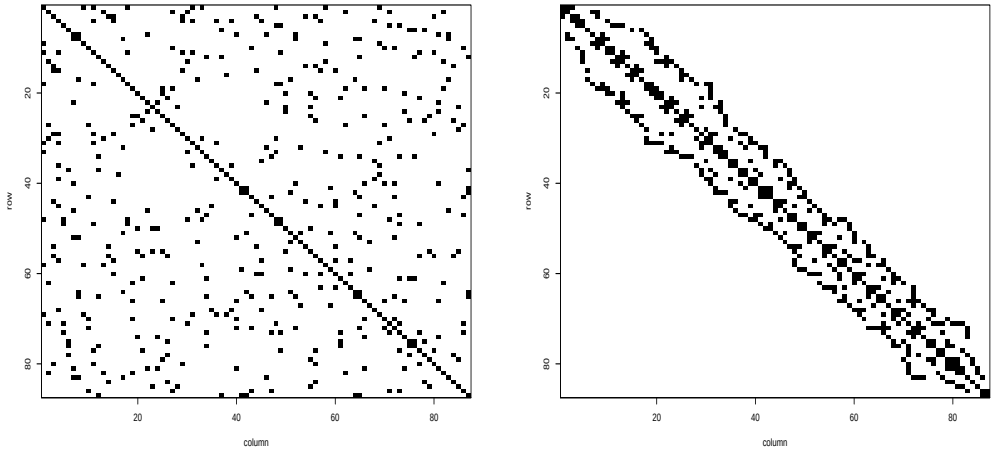
As a first step, we need to minimize the bandwidth of the sparse matrices that arise in the algorithm. This is done by relabeling the regions, (i.e. reordering the nodes of the graph that describes the region) so that the corresponding precision matrix (Q) has minimal bandwidth. Figures 4.1 and 4.2 show the results of this operation for two example adjacency structures — Scotland (used for the Scottish lip cancer data set, Clayton and Kaldor, 1987), and Minnesota. Once the bandwidth of the matrix has been minimized, we can use sparse matrix techniques for computing Cholesky decompositions or inverses much more efficiently.

These sparse matrix algorithms are useful for computing $R_2(\boldsymbol{\phi}|\tau_c)$ for Model 1 and $R_2(\boldsymbol{\Theta}|\tau_h, \tau_c)$ for Model 2. This step involves computing the respective covariance matrices, Σ_T, and the means, μ_N. For Model 1, the



alphabetical ordering (bw=38) after reordering (bw=11)

Figure 4.1: Minimizing bandwidth for Q matrix for Scottish lip cancer



alphabetical ordering (bw=80) after reordering (bw=13)

Figure 4.2: Minimizing bandwidth for Q matrix for Minnesota cancer data

covariance matrix is obtained by inverting the matrix $V^{-1} + \tau_c Q$ which is a sparse matrix. For Model 2, we need to compute

$$C^{-1}_{2N\times 2N} = \begin{bmatrix} V^{-1} + \tau_h I & V^{-1} \\ V^{-1} & V^{-1} + \tau_c Q \end{bmatrix}^{-1} = \begin{bmatrix} C^{11} & C^{12} \\ C^{21} & C^{22} \end{bmatrix},$$

where

$$\begin{aligned} C^{22} &= (V^{-1} + \tau_c Q - V^{-1}(V^{-1} + \tau_h I)^{-1} V^{-1})^{-1} \\ &= (V^{-1} + \tau_c Q - \text{Diag}\left(\frac{Y_i^2}{Y_i + \tau_h}\right))^{-1} \\ C^{12} &= -(V^{-1} + \tau_h I)^{-1} V^{-1} C^{22} \\ &= \text{Diag}\left(-\frac{Y_i}{Y_i + \tau_h}\right) C^{22} \\ C^{11} &= (V^{-1} + \tau_h I)^{-1} - C^{12} V^{-1} (V^{-1} + \tau_h I)^{-1} \\ &= \text{Diag}\left(\frac{1}{Y_i + \tau_h}\right) - C^{12}\,\text{Diag}\left(\frac{Y_i}{Y_i + \tau_h}\right) \\ C^{21} &= \text{Transpose}(C^{12}). \end{aligned}$$

As can be seen from above, the only time when a matrix inversion is necessary is for computing C^{22}. It is easy to see that the matrix to be inverted is a sparse matrix and we can therefore use sparse matrix algorithms in our computations.

As with matrix inversion, we can exploit the block structure of the matrix to make Choleski decomposition much faster. Let the Choleski decomposition of matrix C be $C = U^T U$, and

$$C_{2N\times 2N} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} U_{11}^T & \mathbf{0} \\ U_{21} & U_{22}^T \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ \mathbf{0} & U_{22} \end{bmatrix}$$

so that

$$U_{11}U_{11}^T = C_{11} \Rightarrow U_{11} = \text{Diag}(\sqrt{Y_i + \tau_h})$$
$$U_{12} = (U_{11}^T)^{-1}C_{12} \Rightarrow U_{12} = \text{Diag}\left(\frac{Y_i}{\sqrt{Y_i + \tau_h}}\right)$$
$$U_{22}^T U_{22} = C_{22} - U_{21}U_{12} = V^{-1} + \tau_c Q - \text{Diag}\left(\frac{Y_i^2}{Y_i + \tau_h}\right)$$

Since $C_{22} - U_{21}U_{12}$ is a sparse matrix, it is clear that we need only perform a Choleski decomposition for this sparse matrix of dimension $N \times N$ rather than a Choleski decomposition for a $2N \times 2N$ matrix. This will result in a dramatic reduction in computational time, especially for large N.

Avoiding Matrix Inversion

Matrix inversion turns out to be completely unnecessary in the algorithm. The following description explains how to avoid matrix inversion using standard results from matrix algebra (cf. Golub and Van Loan, 1995).

(1) For generating a proposal from a multivariate-t density, $MT(A^{-1}b, A^{-1}; \nu)$, we can do the following:

1. Find the permutation, $\mathcal{P}$, that minimizes the bandwidth of A. $\mathcal{P}$ is obtained by permuting the rows of an identity matrix according to the relabeling of nodes (rows and columns of the matrix) that induces the minimum bandwidth. This step is done only once when we run the exact simulation algorithms. Let $B = \mathcal{P}A\mathcal{P}^T$ be the matrix with the minimized bandwidth, and $d = \mathcal{P}b$, be the permuted mean vector.

2. Compute (band) Choleski decomposition $B = U^T U$.

3. Simulate $z \sim N(\mathbf{0}, I)$, and $s \sim \chi^2_\nu$.

4. Solve the systems of equations: $U^T v = d, Um = v$ and $Uy = z$.

5. We have thus efficiently computed, $m = B^{-1}d$, and $y \sim N(\mathbf{0}, B^{-1})$.

6. Let $w = y/(\sqrt{s/\nu})$.

7. Set $r = m + y$. We then have $r \sim MT(B^{-1}d, B^{-1}; \nu)$.

8. Set $x = \mathcal{P}^T r$, then $x \sim MT(A^{-1}b, A^{-1}; \nu)$, as required.

(2) For evaluation of the density $MT(A^{-1}b, A^{-1}; \nu)$ at the sampled point, x (above), we can equivalently evaluate the density $MT(B^{-1}d, B^{-1}; \nu)$ at the corresponding point, r. Also, note that for numerical stability, we need to work with the sum of logs when evaluating such densities.

$$= \frac{|B|^{0.5}\Gamma\left(\frac{\nu+N}{2}\right)}{(\nu\pi)^{N/2}\Gamma(\nu/2)} \times \left(1 + \frac{1}{\nu}(r - B^{-1}d)^T B(r - B^{-1}d)\right)^{-(\nu+N)/2}$$

It is easy to see why the above statement is true since

$$(r - B^{-1}d)^T B(r - B^{-1}d) = (\mathcal{P}x - (\mathcal{P}A\mathcal{P}^T)^{-1}\mathcal{P}b)^T \mathcal{P}A\mathcal{P}^T (r - B^{-1}d)$$

$$=(\mathcal{P}x - (\mathcal{P}A\mathcal{P}^T)^{-1}\mathcal{P}b)^T \mathcal{P}A\mathcal{P}^T (\mathcal{P}x - (\mathcal{P}A\mathcal{P}^T)^{-1}\mathcal{P}b)$$

$$=(\mathcal{P}(x - A^{-1}b))^T \mathcal{P}A\mathcal{P}^T (\mathcal{P}(x - A^{-1}b))$$

$$=(x - A^{-1}b)^T A(x - A^{-1}b) \text{ since } \mathcal{P}^T = \mathcal{P}^{-1}$$

Also, $\det(B) = \det(\mathcal{P}A\mathcal{P}^T) = \det(\mathcal{P}\mathcal{P}^T)\det(A) = \det(A)$.

We can do the above computation entirely in terms of the precision matrix, and we already have $B^{-1}d = m$, so not much extra work needs to be done. The only possibly time consuming operation here is computing $\det(B)^{0.5}$. However, we have already found a U in step (1) above such that $U^T U = B$, so $\det(B)^{0.5} = \det(U^T U)^{0.5} = \det(U)$, where the determinant of U will just be a product of its diagonal elements since it is upper triangular.

Computing time can also be saved by noting that the vector b (and hence the vector d) above remains fixed throughout. For Model 1: $b = V^{-1}\hat{\boldsymbol{\mu}}$, and for Model 2: $b = (\hat{\boldsymbol{\mu}}^T V^{-1}, \hat{\boldsymbol{\mu}}^T V^{-1})$, where the quantity $V^{-1}\hat{\boldsymbol{\mu}}$ needs to be computed only once.

The band-Choleski factorization, U, of the band matrix M of dimension $n \times n$ and bandwidth b_w, is computed using only nb_w^2 floating point operations (flops), and the matrix U requires $nb_w S$ bytes of storage (S is the number of bytes needed to store a floating point number). Solving an upper (or lower) triangular banded system of equations $U^T v = d$ takes $2nb_w$ flops by using band back-substitution. In contrast, a regular choleski factorization takes $n^3/3$ flops, and solving a general upper (or lower) triangular system of equations takes n^2 flops. For details, see Golub and Van Loan (1995).

High quality routines for computing the band Choleski factorization are available in the public domain Lapack library written in `Fortran` which can be downloaded from `http://www.netlib.org`. The appropriate routine is `dpbtrf`. The Gibbs-Poole-Stockmeyer-King algorithm (Lewis, 1982) for minimizing the bandwidth of a given positive definite matrix algorithm can be downloaded from `http://www.netlib.org/toms/582`. When the dimensions of the matrices involved are even larger than those considered in this chapter, parallel computing algorithms for Choleski decompositions and other matrix manipulations may be used. For an attempt at using parallel matrix algorithms for similar Bayesian models see Whiley and Wilson (2002). For a more general reference on parallel computation for sparse matrices, see Saad (1996).

4.4 Laplace Approximation

The Laplace approximation (Tierney and Kadane, 1986; Tierney, Kass and Kadane, 1989) provides an alternative way to compute approximate distributions. This approach is briefly described here.

First order normal approximations to posterior distributions typically involve using the posterior mode and the negative inverse Hessian of the log posterior at the mode, as the mean and covariance matrix respectively of the normal distribution. There are also other ways to obtain such first order approximations, but if the true posterior is far from being normal, the first order approximations will perform quite poorly. Tierney and Kadane (1986) provide a much more accurate *second* order approximation, which they refer to as Laplace's method.

Take f, a smooth positive function of x, and h, a smooth function of x with $-h$ having unique maximum at $\hat{x}$. Assume x has dimension m. The Laplace method gives the first order approximation

$$\int f(x)e^{-nh(x)dx} \approx f(\hat{x}) \left(\frac{2\pi}{n}\right)^{m/2} |\tilde{\Sigma}|^{1/2} \exp(-nh(\hat{x})),$$

where $\tilde{\Sigma} = D^2h(\hat{x})$, the hessian evaluated at $\hat{x}$. As before, the goal is to calculate μ_g, the expectation of $g(x)$ w.r.t. distribution $P(.)$. Let $-nh(x)$ be the unnormalized log posterior density, so $-nh(x) \propto \log(P(x))$, where n is typically sample size. Then,

$$\mu_g = E(g(x)) = \int g(x)P(x)dx = \frac{\int g(x)e^{-nh(x)}dx}{\int e^{-nh(x)}dx}.$$

If we applied Laplace's method in a naive fashion, to the numerator and denominator separately ($f = g$ in the numerator, $f = 1$ in the denominator), we would obtain the same estimator as the first order approximation

described in the previous paragraph. However, if we use a clever idea due to Tierney and Kadane (1986), we can obtain a much more accurate second order approximation in the following way. First assume that $g > 0$. Notice that the numerator of the expression above can also be written as $\int e^{\log(g(x))-nh(x)}dx = \int e^{-nh^*(x)}dx$. Laplace's method can be applied with $f = 1$ to both the numerator and denominator. The approximation is then:

$$\mu_g \approx \frac{|\Sigma^*|^{1/2}\exp(-nh^*(x^*))}{|\tilde{\Sigma}|^{1/2}\exp(-nh(\hat{x}))} \tag{4.10}$$

where $\hat{x}, \tilde{\Sigma}$ is as before, and x^* is the maximizer of $-h^*$ and $\Sigma^* = D^2h^*(x^*)$. This method produces a second order approximation with accuracy of $\mathcal{O}(\frac{1}{n^2})$ (Tierney and Kadane, 1986), whereas the accuracy of the first order approximation was $\mathcal{O}(\frac{1}{n})$.

Now, the Laplace method can also be used to calculate approximations for marginal densities. To obtain the Laplace approximation, $S_1^{(L)}(\tau_c)$ for the marginal distribution of the precision parameter for Model 1, we can do the following:

$$P(\tau_c) \propto \int \exp(\log(P(\boldsymbol{\phi}, \tau_c)))d\boldsymbol{\phi}.$$

If we set $f = 1$ and $h = n^{-1}\log(P(\boldsymbol{\phi}, \tau_c))$, we can use (4.10) to get the approximation

$$S_1^{(L)}(\tau_c) \propto |\tilde{\Sigma}(\tau_c)|^{1/2}\exp(\log(P(\hat{\boldsymbol{\phi}}(\tau_c), \tau_c))) = |\tilde{\Sigma}(\tau_c)|^{1/2}P(\hat{\boldsymbol{\phi}}(\tau_c), \tau_c),$$

where $\hat{\boldsymbol{\phi}}(\tau_c)$ is the maximum of $\log(P(\cdot, \tau_c))$, and $\tilde{\Sigma}(\tau_c)$ is the inverse of the Hessian of $-\log(P(\cdot, \tau_c))$ at $\hat{\boldsymbol{\phi}}(\tau_c)$. Notice that to use this approximation in the disease mapping context, we would need to resort to numerical maximization and numerical derivatives to compute $\hat{\boldsymbol{\phi}}(\tau_c)$ and $\tilde{\Sigma}(\tau_c)$, which would

increase in difficulty as the dimensions of the problem increases. However, as described in Section 4.2, the approximations used here can be obtained analytically, and these approximations do not become harder to obtain even as the dimensions of the problem grows.

4.5 Related Algorithms

This section briefly discusses a few other perfect simulation algorithms and touches upon reasons why they may or may not be practical for sampling from the spatial models considered here.

4.5.1 Perfect Slice Sampling

Let $\pi(x), x \in \mathcal{X}$ be the distribution of interest. The simple slice sampler (Mira and Tierney, 2001; Roberts and Rosenthal, 1999; Neal, 2003) is an MCMC algorithm where an auxiliary variable, $u \in \mathcal{U}$ is introduced, and a joint distribution for u and x is constructed by taking the marginal distribution for x unchanged and defining the conditional distribution u given x in a convenient way. In particular, the conditional distribution of $u|x$ is specified as a uniform over the interval $(0, \pi(x))$.

A Markov chain is then constructed over the enlarged state space, $\mathcal{X} \times \mathcal{U}$, with stationary distribution $\pi(x, u) \propto I_{\{u<\pi(x)\}}(x, u)$, as its unique stationary distribution, where $I_A(x)$ is the indicator function of the set A. The usual way to implement this Markov chain is by iteratively performing Gibbs updates on x and u:

1. Vertical update: $u|x$ sampled uniformly over $(0, \pi(x))$.

2. Horizontal update: $x|u$ is typically sampled uniformly over $A(u) = \{x : \pi(x) > u\}$. For a more general description of the horizontal update, refer to Mira et al. (2001).

Consider the case where the distribution of interest is $\pi(x) \propto L(x)p(x)$, where in a Bayesian context, $L(x)$ would be the likelihood and $p(x)$ would be the prior. The convenient choice for the conditional distribution of u is then Unif$(0, L(x))$.

Now, if $\pi(x) = p(x)\prod_{i=1}^{n} L_i(x)$, then a collection of auxiliary variable $u = (u_1, \ldots, u_n)$ can be used, where we take $p(u_i|x)$ as a Unif$(0, L_i(x))$ for all i. Now the joint density is given by:

$$\pi(x, u_1, \ldots, u_n) \propto p(x) \prod_{i=1}^{n} I_{\{0 \leq u_i \leq L_i(x)\}}(x, u_i),$$

so the marginal density of x is $\pi(x)$. The full conditionals for the u_i are Unif$(0, L_i(x))$ while the full conditional for x is simply $p(x)$ restricted to $\{x : L_i(x) \geq u_i, i = 1, \ldots n\}$. This version of the slice sampler is called the product slice sampler (Besag and Green, 1993; Damien, Wakefield and Walker 1999; Edwards and Sokal, 1988; Mira and Tierney, 2001; Roberts and Rosenthal, 1999; Swendsen and Wang, 1987). It can be used for numerous hierarchical and nonconjugate Bayesian settings if the L_is are readily invertible, as shown in Damien et al. (1999).

The perfect slice sampler (Mira, Møller and Roberts, 2001) uses the inherent monotonicity properties of the the slice sampler to construct an algorithm that produces *exact* draws from a stationary distribution of interest, $P(\cdot)$, when $P(\cdot)$ is bounded everywhere. The analogous perfect product slice sampler has the potential for being of more practical use in multidimensional

settings, but is less straightforward to construct.

Mira et al. (2001) describe the application of their methods for sampling a one dimensional distribution, where they show that as the enveloping distribution becomes a poorer match to $P(\cdot)$, their perfect sampler appears to be less negatively affected by the non-optimatility of the envelope than the rejection sampler. Thus the simple perfect slice sampler becomes increasingly efficient relative to the rejection sampler, as the envelope becomes a poorer fit to the $P(\cdot)$. This is similar to the results observed empirically in Section 4.7 when comparing the perfect tempering algorithm with a rejection sampler.

The perfect slice sampler, like the perfect tempering algorithm, uses an auxiliary variables method to produce a perfect sampler. The algorithm, also requires a "bounding process" (like the enveloping distribution for perfect tempering). However, unlike the perfect tempering algorithm, it does not require knowledge of the supremum of the ratio of $P(\cdot)$ to this bounding process. The major drawback to the perfect slice sampler is that, while fairly general in theory, it suffers from the same practical implementation problems that the slice sampler faces. In particular, the horizontal slice is often very difficult to perform, since it necessitates identifying level sets for complicated density functions. This is generally very difficult to do in complicated Bayesian hierarchical modeling settings, but there is some recent work by Agarwal and Gelfand (2001) that suggests that it may be practical in the context of the spatial models considered in this thesis. It would be of interest to see if it is possible to extend the construction of the perfect slice sampler in Mira et al. (2001) to such models.

4.5.2 Perfect Forward Tempering

Brooks, Fan and Rosenthal (2003) describe a perfect simulation algorithm that uses simulated tempering to implement a single-sweep forward simulation without the need for recursively searching through negative times. Assume that a simulated tempering chain can be constructed such that it satisfies the domination conditions described in Subsection 3.3.2. As before, suppose the time when the dominating random walk chain ($\{D_t\}$) lands at temperature 0 is $-\tau^*$. Brooks et al. (2003) observe that τ^* follows a Geometric distribution with some parameter ϵ, say. The perfect *forward* tempering version of the C.F.T.P. perfect tempering algorithm proceeds as follows:

1. Draw $T \sim \text{Geo}(\epsilon)$.
2. Draw $X_1 \sim H_0$, and set $Z_1 = (X_1, 0)$.
3. Run the residual chain forward for $T - 1$ iterations to produce the chain $\{Z_1, \ldots, Z_T\}$
4. Z_T is an exact draw from the stationary distribution (Brooks et al.,2003).

The idea behind the algorithm can be understood by looking at a C.F.T.P. algorithm (running from time $-T$ to 0) that corresponds to this forward simulation algorithm. Suppose we can establish a k_0-step minorization condition for the chain. The random variable T is then conditionally independent of $\{X_{-T\,k_0}, \ldots, X_0\}$, conditional on X_{-Tk_0}, and on there being no further regenerations from time $-Tk_0 + 1$ to 0. Since the residual kernel produces dependent samples, it ensures that no regenerations occur between time $-Tk_0 + 1$

to 0. For the simulated tempering chain here, we obtain a single step minorization by embedding two steps in each update of the chain (so $k_0 = 1$) (see Appendix B.4). The important issue then is finding a way to obtain ϵ. This ϵ can be found from the minorization condition (3.2), which is difficult to establish in general. One way to get a handle on ϵ is from the following proposition (Brooks et al., 2003)

Proposition : Suppose that the simulated tempering chain defined on $(\mathcal{X} \times \mathcal{T})$ has a temperature t_{R} such that

$$\mathcal{S}^* = \{(x, t_{\mathrm{R}}) : x \in \mathcal{X}\} \text{ is } (k_2, \epsilon_2, \nu) - \text{small.} \tag{4.11}$$

Suppose further that, from any state in any temperature, there is probability of at least ϵ_1 of reaching $\mathcal{S}^*$ (i.e., jumping to t^* after k_1 steps), so that

$$P^{k_1}((x,t), \mathcal{S}^*) \geq \epsilon_1 \text{ for all } (x,t) \in (\mathcal{X} \times \mathcal{T}). \tag{4.12}$$

Then the entire state space $(\mathcal{X} \times \mathcal{T})$ is $(k_1 + k_2, \epsilon_1\epsilon_2, \nu)$-small, so

$$P^{k_1+k_2}((x,t), A) \geq \epsilon_1 \, \epsilon_2 \, \nu(A) \;\; \forall (x,t) \in (\mathcal{X} \times \mathcal{T}), \; \forall A \in \mathcal{B}$$

If it were possible to compute ϵ_1 and ϵ_2 as above, it would in theory be possible to run the perfect forward tempering algorithm; the algorithm would be practical only if $\epsilon_1\epsilon_2$ is not too small. It is easy to construct the simulated tempering chain in such a way that it becomes obvious how to compute ϵ_2. For instance, we can take the set $\mathcal{S}^*$ to be an atom $\mathcal{A}$ (as is the case in Section 3.3). Then $\mathcal{S}^*$ is (k_2, ϵ_2, ν)-small with $k_2 = 1, \epsilon_2 = 1$, and $\nu = 1_{\mathcal{A}}$. We can have $\mathcal{S}^*$ be a small set in a simulated tempering chain as long as the value of the chain upon entering $\mathcal{S}^*$ is drawn independently of the previous value, which is easy to incorporate into the algorithm if the

probability distribution associated with temperature t_{R} is one from which independent samples can be drawn. It is therefore not necessary for $\mathcal{S}^*$ to be an atom to satisfy (4.11). The more difficult issue is finding ϵ_1 to satisfy (4.12), since this requires getting a handle on the transition probabilities of the Markov chain, which is typically a hard problem. However, if the simulated tempering chain satisfies the domination condition (3.5), there is an obvious lower bound for the probability that the chain moves from any state to the state with temperature t_{R}, so ϵ_1 is easy to estimate as well. Like the CFTP version of perfect tempering, this algorithm also relies on an estimate of the supremum (K) of the rejection sampling ratio, since this is implicitly used in computing ϵ_1 (see Appendix B.4 for details).

Brooks et al. (2003) also show that, whenever we have a uniformly ergodic Markov chain, we can generate a perfect draw from the stationary distribution simply by running the residual chain for a Geometric(ϵ) time, where ϵ is the minorization parameter. Therefore, an alternative is to consider the possibility of obtaining ϵ for an independence chain with an enveloping distribution as the proposal distribution. Now, let $\pi(x), x \in \mathcal{X}$ be the distribution of interest. Suppose we have an enveloping distribution, $p(x)$ such that

$$\frac{\pi(x)}{p(x)} \leq \kappa, \ \ \forall x \in \mathcal{X},$$

then Mengerson and Tweedie (1996) show that

$$P(x, A) \geq \epsilon Q(A) \ \ \forall x \in \mathcal{X}, \ A \in \mathcal{B}$$

with $\epsilon = \kappa^{-1}$, and $\mathcal{B}$ is the Borel σ-algebra associated with $\mathcal{X}$. At first glance, it appears that we have already computed a good estimate of κ for the purpose of implementing the perfect tempering and rejection sampling

algorithms. This seems to directly give us an estimate of ϵ, which would allow us to run the perfect forward tempering algorithm without having to do any more setup work. However, on closer inspection, it is clear that we do not actually have an estimate of ϵ. For the perfect tempering algorithm, we have an *unnormalized* distribution, $P \propto \pi$, and an *unnormalized* enveloping distribution, $R \propto p$, and we have proved that there exists a $B > 0$ such that

$$\frac{P(x)}{R(x)} \leq B, \ \forall x.$$

Without explicitly knowing the normalizing constants, knowing B does not tell us anything about κ, so we do not have an estimate of ϵ. This makes it impossible to implement the perfect forward tempering algorithm without doing some additional work. It may be possible to estimate the ratio of the normalizing constants using methods described in Geyer (1994), but this adds a whole new level of difficulty to the problem of implementing perfect forward sampling.

In terms of efficiency, the forward tempering algorithm may not offer any noticeable advantage over the C.F.T.P. version of the algorithm since tracing the random walks back and forth adds an insignificant amount of computing work; once the coalescence time is determined in the C.F.T.P. perfect tempering algorithm, the rest of the computation involves simply running the simulated tempering chain forward, which is identical to the work involved in the perfect forward tempering algorithm. Perfect forward tempering avoids the need to keep sequences of uniform random variates, though this is likely to be insignificant relative to the overall storage and computing requirements of the algorithm, and is therefore not a significant advantage. The most attractive feature of the forward simulation algorithm is its slightly reduced

programming complexity relative to the C.F.T.P. versions of the algorithm. For this reason, if it is possible to construct a simulated tempering algorithm, and determine the corresponding minorization parameter, ϵ, perfect forward tempering should be considered a viable option.

4.6 Related Models

This section briefly discusses exact computation for two models that are closely related to the disease mapping models studied in this chapter.

4.6.1 Proper CAR Prior Models

The impropriety in the prior on the $\boldsymbol{\phi}$ parameters for these models arises out of the singularity of Q, the precision matrix for the joint distribution on the $\boldsymbol{\phi}$s. There are proper versions of these priors that are also often used; the following description shows how the exact sampling methods accomodate these priors. Recall that the CAR prior is

$$\boldsymbol{\phi}|\tau_c \propto \exp\left(-\frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}\right).$$

$$Q_{ij} = \begin{cases} n_i & \text{if } i = j \\ 0 & \text{if } i \text{ is not adjacent to } j \\ -1 & \text{if } i \text{ is not adjacent to } j \end{cases}$$

This matrix is often written in the form $Q = M^{-1}(I - \alpha_p C)$, where we set the 'propriety' parameter $\alpha_p = 1$, I is the identity matrix, $M = \text{Diag}\{1/n_i\}$, and the matrix $C = \{c_{ij}\}$ with $c_{ij} = 1/n_i$ if i is a neighbor of j, and $c_{ij} = 0$ otherwise. There are other versions of this prior that 'fix' the impropriety

in the Q matrix by allowing $\alpha_p \in (0,1)$, thus causing the new matrix, Q^* say, to be positive-definite. In fact, some authors (Cressie, personal communication), refer to *this* prior as the CAR prior, to distinguish it from the improper prior, which they call an IAR (intrinsic autoregression) prior. However, this proper prior does not deliver enough spatial similarity unless α_p is fairly close to 1, that is, the prior does not induce enough spatial smoothing unless α_p is nearly 1. On the other hand, as α_p gets really close to 1, the prior tends to impropriety (at least numerically). One solution prescribed by some authors is to use an informative prior on α_p that insists on larger values of α_p. Carlin and Banerjee (2003) briefly describe these issues in the context of multivariate CAR models.

If we consider the prior with fixed $\alpha_p \in (0,1)$, we find that the only change to the joint posterior distribution involves replacing Q with $Q^* = M^{-1}(I - \alpha_p C)$. The new Q^* matrix is just a diagonally dominant version of the Q matrix from before:

$$Q^*_{ij} = \begin{cases} n_i & \text{if } i = j \\ 0 & \text{if } i \text{ is not adjacent to } j \\ -\alpha_p & \text{if } i \text{ is not adjacent to } j \end{cases}$$

We can derive a proposal distribution, $R(\boldsymbol{\theta}, \boldsymbol{\phi}, \tau_h, \tau_c)$ in exactly the same way as for the proper prior model, simply by replacing Q by Q^*. The proof in Appendix B.2 can also be used almost verbatim to obtain a proof that R is, in fact, an envelope for the posterior distribution for this model. The methods should be as successful in practice for the proper CAR prior as it is for the improper prior since the overall structures of the matrices involved are identical to those that result from the improper prior. It is possible that the

propriety of the prior may even lead to some additional numerical stability. Thus, the exact sampling methods developed here generalize to this case fairly easily.

4.6.2 Models with Covariates

When there is additional information that is relevant to the relative risk for disease events in each region, the disease mapping models can incorporate this information as covariates. If the influence on log-relative risk is assumed to be linear in the covariates, the resulting model is typically described in the following way:

$$Y_i \sim \text{Poi}(E_i e^{\mu_i})$$

with μ_i now modeled as

$$\mu_i = X_i^T \beta + \theta_i + \phi_i \ , \ i = 1, ..., N,$$

with $\boldsymbol{\theta}, \boldsymbol{\phi}$ modeled as before, and X_i^T are explanatory, region-level, spatial covariates, having parameter coefficients (vector) $\boldsymbol{\beta}$. Thus, the $\boldsymbol{\theta}, \boldsymbol{\phi}$ parameters capture any variability, spatial or otherwise, not accounted for by the covariates. For now, assume the prior on $\boldsymbol{\beta}$ is $h(\boldsymbol{\beta})$.

Let $P^*(\boldsymbol{\Theta}, (\tau_h, \tau_c), \boldsymbol{\beta})$ be the posterior distribution, and $\hat{P}^*(\boldsymbol{\Theta}, (\tau_h, \tau_c), \boldsymbol{\beta})$ be the approximate posterior distribution, where the approximations follow in the same fashion as for Model 2. We can also, in the same way as before, obtain the approximate marginal posterior distribution $S_1^*(\tau_h, \tau_c, \beta)$, and the approximate conditional distribution, $S_2^*(\boldsymbol{\theta}, \boldsymbol{\phi} | \tau_h, \tau_c, \boldsymbol{\beta})$. Details are given in Appendix B.2.3. If we have a probability density function that is easy to simulate from, $R_{1\beta}(\boldsymbol{\beta})$, such that $S_{1\beta}(\boldsymbol{\beta}) / R_{1\beta}(\boldsymbol{\beta}) \leq K_\beta$ for all $\boldsymbol{\beta}$, for some

$K_\beta < \infty$ then, following Appendix B.2.3, we can obtain appropriate proposal distributions that envelope the posterior distribution. We can therefore do rejection sampling or perfect tempering for this model. However, there are still several major issues that make this a difficult problem:

1. Finding an appropriate $R_{1\beta}$ that envelopes $h(\boldsymbol{\beta})$: this may be quite easy if $h(\boldsymbol{\beta})$ is a standard distribution. If $h(\boldsymbol{\beta})$ is improper, this is impossible to do, and hence, we will not be able to directly use the proposal distributions for the remaining parameters (envelopes) without doing some more analytical work, and perhaps having to derive modified envelopes.

2. Even if we determine a general distribution, $R_{1\beta}$, that envelopes $h(\boldsymbol{\beta})$, tuning the parameters of the distribution to obtain a good 'match' to $h(\boldsymbol{\beta})$ for each new data set may be difficult since $\boldsymbol{\beta}$ is likely to be of greater than two or three dimensions.

3. If the number of covariates gets large, even if we do have a provable enveloping distribution for the joint posterior distribution, there is no guarantee that the envelope will be practical. That is, the rejection sampling algorithms and perfect simulation algorithms may take too long to produce i.i.d. draws to be practical.

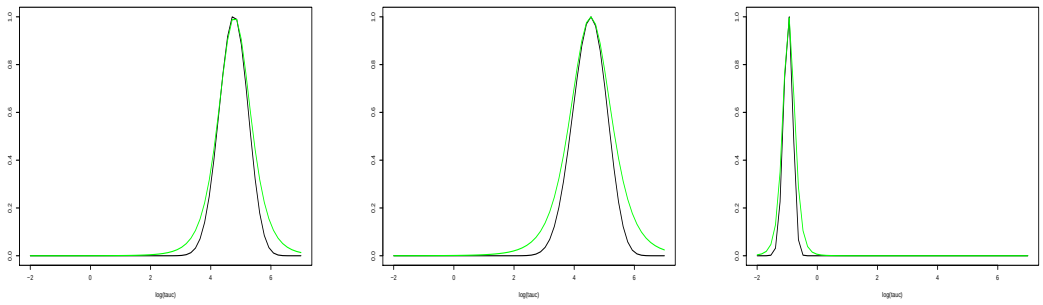
From the above discussion, it is clear that there is reason to believe that exact sampling methods may work for models with covariates. However, it is not clear that the exact sampling methods would readily work for a variety of prior distributions on the covariates, or with a large number of covariates. It

may also be necessary to do some analytical work before it becomes possibles to actually implement the exact sampling algorithms in these settings.

4.7 Examples: Application to Cancer Data

Perfect tempering and rejection sampling methods were applied to the three Minnesota cancer data sets studied in Chapter 2. The posterior distribution for Model 1 has 88 dimensions, while it has 176 dimensions for Model 2. All the computer code was implemented in `R` (Ihaka and Gentleman, 1996).

The results described here are for the second variant of the perfect tempering algorithm (as described in 3.3.2) since that was found to be more efficient. The perfect sampler was also compared to a rejection sampling algorithm that uses the same enveloping distributions. The algorithms were compared in two ways — the amount of time taken for every sample produced, and the acceptance rates. Since acceptance rates are an easy way to measure the efficiency of a rejection sampler, we defined a corresponding notion of acceptance rates for the perfect sampler as the number of samples accepted divided by the total number of draws from the enveloping distribution. This measure of acceptance rates for the perfect sampler does not account for the time taken by the algorithm to compute Metropolis-Hastings ratios, but it does give a sense of its efficiency on a scale that is roughly similar to the measuring of acceptance rates for the rejection sampler.



colorectal breast cancer cancer mortality

Figure 4.3: Envelope for marginal posterior for Minnesota cancer data: $S_1(\tau_c)$ (black), envelope $R_1(\tau_c)$ (green): both log-scale.

4.7.1 Model 1

Envelopes can be found for the approximate marginal distribution, $S_1(\tau_c)$ for each of the three data sets by simply plotting the function (as shown in Figure 4.3). The bounding constant K is obtained by numerically maximizing the ratio of $S_1(\tau_c)$ to $R_1(\tau_c)$, then checking the bounds for randomly generated samples. The bounding constant, along with the appropriate parameters for $R_1(\tau_c)$ are all the necessary ingredients for running the perfect sampling algorithm. The rejection sampler and the perfect sampler were each run for $10,000$ iterations with the same envelopes and bounding constant for each of the three data sets. The results are summarized in Table 4.1.

4.7.2 Model 2

The approximate bivariate marginal distribution, $S_1(\tau_h, \tau_c)$ can be plotted as shown for the breast cancer data in Figure 4.4. However, to find the appropriate 'matching' envelopes, it is useful to look at profiles of the approximate marginal posterior distributions, $S_{1a}(\tau_h)$ and $S_{1b}(\tau_c)$ (on log-scale) and find t-distribution that match them reasonably well. Figure 4.5 shows plots of the profiles and the respective envelopes for the breast cancer data. The bounding constant K can be obtained in the same way as for Model 1. Again, the perfect sampling algorithm and rejection sampler were run for $10,000$ iterations each. The results are summarized in Table 4.2.

Of course, another option that could be explored is one where the perfect sampler produces groups of samples that are correlated within groups, but independent between groups. This method may be a nice compromise between independent and dependent sampling and is, in fact, the method used by Møller and Nicholls (1999).

4.8 Perfect versus Rejection Sampling

This section compares perfect tempering and rejection sampling, while also discussing the results from the application of both methods to some data sets. While the domination condition required for the perfect tempering algorithm (3.4) is in theory much less restrictive than the rejection sampling condition (3.1), in practice, it seems that the rejection sampling condition is necessary to satisfy the domination condition. Therefore, it seems as if perfect tempering can only be applied when rejection sampling is also possible. There

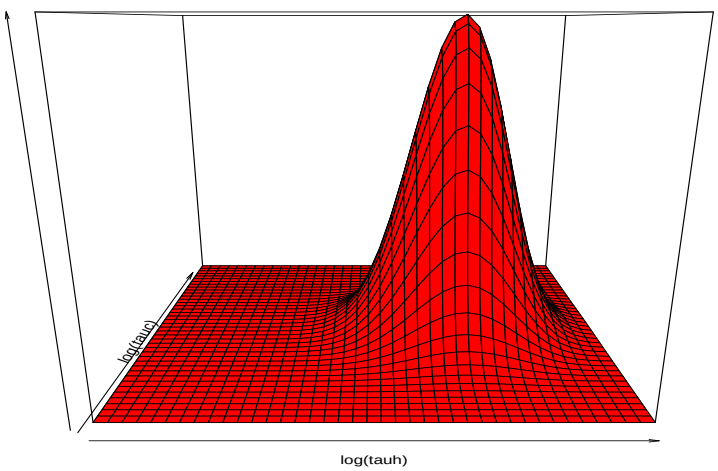


Figure 4.4: Bivariate marginal posterior for Minnesota breast cancer data

are several advantages to using rejection sampling. Since rejection sampling is much more intuitive than any perfect simulation algorithm that relies on Markov chains, it is much easier to explain and understand the algorithm. Also, it is much easier to program a rejection sampling algorithm than it is to write a program that implements perfect simulation. This greatly reduces the chances of programming errors. Since the rejection sampler is simpler, and has been studied for a much longer period of time, there are also ways to "correct" for the incorrect specification of the supremum K of the rejection sampling ratio via techniques such as Metropolised rejection (Tierney, 1994) and E-sup rejection (Caffo et al., 2002). Issues arising from incorrect specification of K are not as well understood in the perfect tempering algorithm.

While the above discussion makes it seem like rejection sampling is always preferable to perfect tempering, there is still enormous potential for perfect

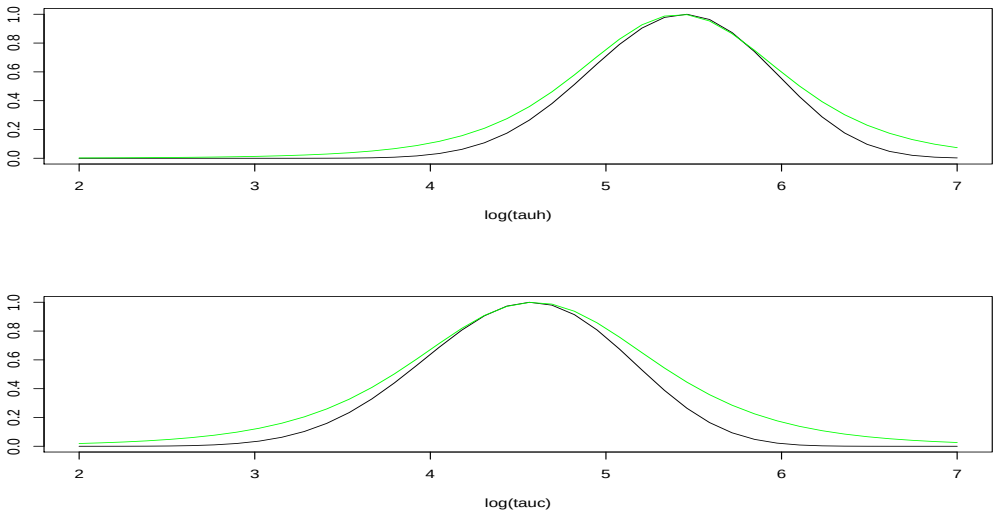


Figure 4.5: Marginal posterior profiles for Minnesota breast cancer data

tempering to be useful in practice. This is due to the fact that it can be more efficient than a rejection sampler — this becomes increasingly valuable as the dimensionality and difficulty of the distribution increases, and rejection sampling becomes less practical. Tables 4.1 and 4.2 show cases where there can be significant gains in efficiency, sometimes as high as five times the number of exact samples per second. While this is not a dramatic speed-

data set	samples/sec		acceptance rates	
	perfect	rejection	perfect	rejection
breast cancer	6.83	2.89	0.11	0.028
colo-rectal cancer	18.57	7.04	0.31	0.07

Table 4.1: Perfect versus Rejection: Model 1

data set	samples/sec		acceptance rates	
	perfect	rejection	perfect	rejection
breast cancer	4.0	1.26	0.147	0.041
colo-rectal cancer	5.25	1.95	0.211	0.059

Table 4.2: Perfect versus Rejection: Model 2

up, with increase in dimensions, perfect sampling is likely to be even faster relative to rejection sampling (due to the difference in acceptance rates). The perfect sampler appears to be more effective relative to the rejection sampler when the envelope is poor, a phenomenon also noticed in the context of the perfect slice sampler (Mira, Møller and Roberts, 2001).

Also, there are many ways to fine-tune a perfect tempering algorithm in terms of changing or adding tempering distributions, or varying the pseudo-priors (π_i's in Section 3.3). Hence, it is reasonable to believe that with a better understanding of the perfect tempering algorithm, and the simulated tempering algorithm underlying it, there may be even greater gains in efficiency of the perfect tempering algorithm when compared to the rejection sampler. The results here encourage future research in perfect tempering.

4.9 Discussion

The results in this chapter demonstrate that it is feasible to do i.i.d. sampling for Bayesian hierarchical models for disease mapping models as described in Section 4.1. The application of this method to real data sets shows that exact sampling, both via rejection sampling and perfect tempering, can work

for realistic problems that have been known to cause problems for regular Markov chain samplers. Strategies for exact simulation are not much more computationally expensive than block sampling MCMC methods. Also, the preliminary work required by the user to obtain appropriate proposal distributions is less than the standard diagnostic work that is involved when using an MCMC sampler. It is very easy to program this entirely in R (Ihaka and Gentleman, 1996). By using perfect sampling rather than the usual Markov chain Monte Carlo methods, issues such as assessing convergence and determining if the sampler is mixing well, can be completely avoided. Obtaining Monte Carlo errors is very easy, while it is often a harder problem when using standard MCMC algorithms (cf. Geyer, 1992 and Jones and Hobert, 2001). Also, the sampling techniques used here certainly provide an independent check of, or a useful diagnostic for, any MCMC techniques that people use routinely for such models.

From this study of proposal distributions for producing envelopes for this model, it is clear that there is a lot to be gained by using heavy-tailed distributions as proposals for both the precision components and the conditional distribution of the model parameters. This suggests that for similar models, and for cases for this model where it is more difficult to do perfect sampling (for instance, due to numerical difficulties caused by very high dimensions), it may be worth using heavy tailed distributions as proposals for block sampling Markov chain Monte Carlo algorithms. We can run an independence chain where the proposal distribution is an envelope, which results in a uniformly ergodic Markov chain (Mengerson and Tweedie, 1996). It is important to note that rejection sampling is still often a practical alternative to MCMC

methods in some cases; whenever possible, it should at least be used as a benchmark when studying perfect sampling algorithms.

Since the supremum of the ratio of the target and candidate densities (K) is determined empirically or via maximization routines, only an approximate value of K is actually used in the perfect sampling algorithm. However, after adjustments to the estimate of K over a few iterations, it should converge fairly quickly to the right value (see Caffo et al., 2002). This is consistent with the empirical evidence from implementations studied here as well. The perfect sampler here can be thought of as a sampler that takes a rejection sampler and makes it more efficient; therefore, the arguments that justify using the estimated K for rejection sampling should apply for the perfect sampler. A proof that the proposal distributions in this thesis truly envelope the distribution of interest involved some analytical work, but the proposals themselves are standard distributions from which it is easy to draw samples.

The success with enveloping a hierarchical model that involves a Poisson likelihood, improper CAR priors and some non-identifiable parameters in the prior, is encouraging. Similar or even better results may be obtained for models that have Gaussian likelihoods, and priors where all parameters are well identified. Perfect sampling may be possible for realistic problems for which even standard Gibbs or MCMC techniques do not perform well.

Chapter 5

Exact Simulation for Variance Component Models

This chapter outlines a systematic, easy to use method for sampling from posterior distributions for a class of popular Bayesian hierarchical models with two variance components. The low dimensionality of the variance components in these models can be exploited to obtain a rejection sampling envelope to produce i.i.d. samples from their marginal distributions. Unlike the spatial models studied in Chapter 4, the problem here ultimately translates to sampling from low dimensional distributions since it is possible to have access to the actual marginal posterior distributions of the variance components (recall that for the spatial models, only *approximate* marginal posterior distributions were available). Also, conditional on the sampled variance components, the random effects have a multivariate normal distribution (again, for the spatial models only approximate conditional distributions are available).

Rejection sampling for multivariate full conditional distributions using multivariate normal and multivariate split-t distribution envelopes is discussed in Zeger and Karim (1991) and Carlin and Gelfand (1991), but these rejection samplers are embedded in Gibbs sampling schemes. The use of the 'integrated out' marginal posterior distribution of the variance components has been studied by Wilkinson and Yeung (2001,2002) in the context of using sparse matrix algorithms for MCMC computation for large Bayesian linear models. Wolfinger and Kass (1996) in unpublished work discuss an attempt at rejection sampling precision parameters using gamma densities as envelopes, but do not prove that they actually have an envelope. In Wolfinger and Kass (2000), they use the same distributions in an independence Metropolis-Hastings algorithm. In this work, log-t distributions are used as enveloping distributions, and a proof is given to show that it is an envelope. Thus, once a rejection sampler is implemented for the precision parameters, it is possible to use sequential sampling to obtain samples from the random effects. Everson and Morris (2000) provide a rejection sampling algorithm for a 2-level Normal hierarchical model having possibly multivariate outcome vectors, but with known level-1 (data level) variances. Everson (2001) describes a method for rejection sampling for a model with univariate outcomes, but his priors are imposed on the variances rather differently, as combinations of truncated distributions imposed on transformed versions of the variances. This, in turn, leads to marginal posterior distributions for the variances that can be directly enveloped by a combination of truncated and inverse Gamma distributions. The random effects model for which exact sampling algorithms are developed here are closely related to the model

described in Everson (2001), but the priors on the variance components are directly placed on the variances themselves, rather than on the transformed form of the variances.

5.1 The Bayesian Linear Hierarchical Model

This section illustrates how the exact sampling approach can be applied to the Bayesian analogue of the standard, normal theory one-way random effects model with conjugate priors. This is a model studied in the seminal Gibbs sampling paper by Gelfand and Smith (1990). Convergence rates for a fully blocked Gibbs sampler for this model has also been analysed in a paper by Hobert and Jones (2003). This model has three levels. First, conditional on $\boldsymbol{\theta} = \{\theta_1, \ldots, \theta_K\}^T$, and precision λ_e, the data values, Y_{ij} are independent

$$Y_{ij}|\theta_i, \lambda_e \sim N(\theta_i, \lambda_e^{-1}),\ i = 1, \cdots K, j = 1, \cdots m$$

At the second stage, conditional on μ and λ_θ, $\boldsymbol{\theta}$ and λ_e are independent with

$$\boldsymbol{\theta}|\mu, \lambda_\theta \sim N(\mu\mathbf{1}, \lambda_\theta^{-1} I) \quad \text{and } \lambda_e \sim G(a_2, b_2)$$

where $\mathbf{1}$ is a $K \times 1$ column vector of ones, I is a $K \times K$ identity matrix, and a_2, b_2 are known positive constants. The parameterization used (as before), is $x \sim G(\alpha, \beta)$ if its density is proportional to $x^{\alpha-1}e^{-x/\beta}I(x > 0)$. The third stage is

$$\mu \sim N(\mu_0, \lambda_0^{-1}) \quad \text{and } \lambda_\theta \sim G(a_1, b_1)$$

Therefore, the random effects in this model are $\boldsymbol{\Theta} = (\boldsymbol{\theta}, \mu)$, the variance components are $\Lambda = (\lambda_\theta, \lambda_e)$, and the fixed constants are $a_1, b_1, a_2, b_2, \mu_0, \lambda_0^{-1}$.

5.2 Marginal and Conditional Distributions

The joint posterior distribution, $P(\mathbf{\Theta}, \Lambda)$ is obtained in the usual way by taking the product of the likelihood and the prior distributions. The distribution of the random effects, conditional on the variance parameters, $P(\mathbf{\Theta}|\Lambda)$, is easily obtained from the joint distribution.

$$P(\mathbf{\Theta}|\Lambda) \sim N(C^{-1}(-\frac{1}{2}D^T)\ ,\ C^{-1})\ . \tag{5.1}$$

where

$$C_{(K+1)\times(K+1)} = \begin{bmatrix} \overbrace{(m\lambda_e + \lambda_\theta)I}^{\boldsymbol{\theta}} & \overbrace{-\lambda_\theta \mathbf{1}}^{\mu} \\ -\lambda_\theta \mathbf{1}^T & K\lambda_\theta + \lambda_0 \end{bmatrix},$$

and

$$D_{1\times(K+1)} = (-2\lambda_e \mathbf{Y}^T W, -2\lambda_0 \mu_0).$$

As shown in Appendix C.1, it is not necessary to compute the matrix inverse of C (C^{-1} can be directly computed from Λ). The marginal distribution of the variance components, $S(\Lambda)$, can be obtained by integrating $P(\mathbf{\Theta}, \Lambda)$ with respect to the random effects.

$$\begin{aligned} S(\lambda_\theta, \lambda_e) \propto & \det(C)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}} + D\hat{\mathbf{\Theta}})\right) \\ & \times \lambda_e^{Km/2+a_2-1} \exp\left(-\lambda_e(1/b_2 + \sum_{i,j} Y_{ij}^2/2)\right) \lambda_\theta^{K/2+a_1-1} \exp(-\lambda_\theta/b_1). \end{aligned} \tag{5.2}$$

Details of the derivation of these distributions are given in Appendix C.1.

To use the rejection sampling algorithm, an appropriate enveloping distribution for S is derived. Following the work in Chapter 4, the enveloping

distribution is taken to be $R(\lambda_e, \lambda_\theta; \mu_e, \sigma_e, \mu_\theta, \sigma_\theta)$, a product of independent log-t distributions, so

$$R(\lambda_e, \lambda_\theta) \propto \frac{1}{\lambda_e}\left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_e) - \mu_e}{\sigma_e}\right)^2\right]^{-(\nu+1)/2} \times \frac{1}{\lambda_\theta}\left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_\theta) - \mu_\theta}{\sigma_\theta}\right)^2\right]^{-(\nu+1)/2}. \tag{5.3}$$

For some K to be found numerically, Appendix C.2 has a proof that shows $R(\lambda_e, \lambda_\theta)$ is an envelope for $S(\lambda_e, \lambda_\theta)$. To sample from this model, one would proceed as follows

1. Plot the log transformed version of $S(\lambda_e, \lambda_\theta)$, say $S_{\log}(x, y)$, where $x = \log(\lambda_e), y = \log(\lambda_\theta)$.

2. Plot the profiles, $S^{(2)}_{\log}(x, \cdot)$, $S^{(1)}_{\log}(\cdot, y)$, and match two t distributions, $R^{(1)}_{\log}(x; \mu_{\lambda_e}, \sigma_{\lambda_e})$ and $R^{(2)}_{\log}(y; \mu_{\lambda_\theta}, \sigma_{\lambda_\theta})$, to these profiles respectively. A simple way to match a distribution to its profile is by setting the mean of the t distribution to the mode, and changing the variance of the distribution to match the variance of the profile by trial and error. This method appears to be quite easy and effective in practice.

3. The enveloping distribution is now a product of log-t distributions that use the parameters of the t-distributions from above, $R(\lambda_e, \lambda_\theta; \mu_{\lambda_e}, \sigma_{\lambda_e}, \mu_{\lambda_\theta}, \sigma_{\lambda_\theta})$.

4. We need to compute the bound, K. This can be done by generating a large number of samples from $R(\lambda_e, \lambda_\theta)$ and using the empirical maximum of the ratio $S(\lambda_e, \lambda_\theta)/R(\lambda_e, \lambda_\theta)$, say $\widehat{K}$. This empirical maximum

can be used as a starting value in a procedure that numerically maximizes the ratio, to find $\tilde{K}$. The envelope thus obtained is $\tilde{K}R(\lambda_e, \lambda_\theta)$.

5. Use $\tilde{K}R(\lambda_e, \lambda_\theta)$ in a rejection sampling algorithm. If a sample is obtained that violates the estimated upper bound $\tilde{K}$, change the bound accordingly and restart the rejection sampling algorithm.

6. Using (5.1), we can sample the random effects $\boldsymbol{\Theta}$, conditional on the sampled values of the precision parameters from the previous step.

5.3 Results

This section describes the application of the exact sampling algorithm on the styrene data analyzed by Lyles, Kupper and Rappaport (1997). While Lyles et al. use a frequentist random effects model, a Bayesian analysis of this data is considered here. This data set was also studied by Jones and Hobert (2001) in the context of computing convergence rates for a blocked Gibbs sampler. Jones and Hobert (2003) show that the simple, fully blocked Gibbs sampler is geometrically ergodic for this model. To remain consistent with the notation from previous chapters, the Gamma distribution is parameterized here as $x \sim G(\alpha, \beta)$ if $p(x) \propto x^{\alpha-1}e^{-x/\beta}$. This is a different parameterization from the one Jones and Hobert use: the a_1, b_1 here are identical to their corresponding parameters, but the a_2, b_2 parameters here are reciprocals of their a_2, b_2. The hyperparameter settings considered here is setting #2 in Jones and Hobert, with $a_1 = 601.76$, $b_1 = 1/77.573$, $a_2 = 31.273$, $b_2 = 1/17.674$, $\mu_0 = 4.809$, and $\lambda_0 = 0.1$.

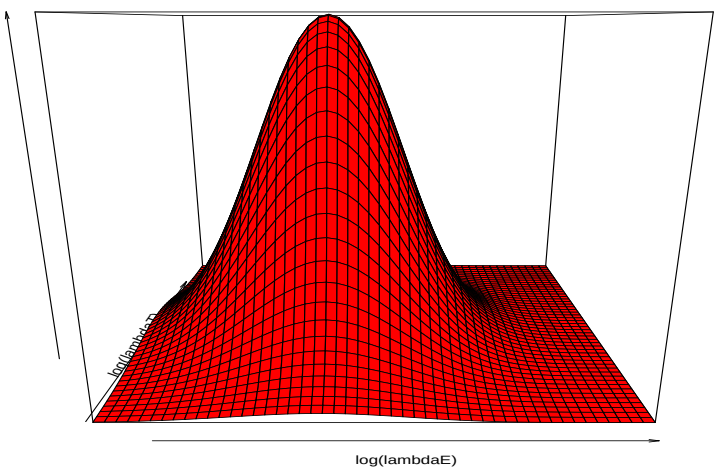


Figure 5.1: Bivariate marginal posterior distribution for styrene data

Figure 5.1 is a plot of the marginal posterior distribution of Λ, and Figure 5.2 shows how the profiles are enveloped. The acceptance rates for sampling the variance components using the rejection sampler is very high, around 80% with the appropriate envelopes. Since each sample (two log-t variates) is very cheap to produce, the algorithm is extremely efficient. Since the rejection sampler here only deals with a two dimensional distribution, and computing the accept-reject ratio is very easy, it is possible to be very conservative and pick a much larger $\tilde{K}$ than the value estimated. If no sampled value ever violates the bound on the envelope, all the samples obtained can be considered as i.i.d. draws from the distribution of interest. Conditional on the sampled precision parameters, the random effects can be drawn easily since they are simply multivariate random variates. Also, the fact that the covariance matrix for these multivariate normals are very easy to compute

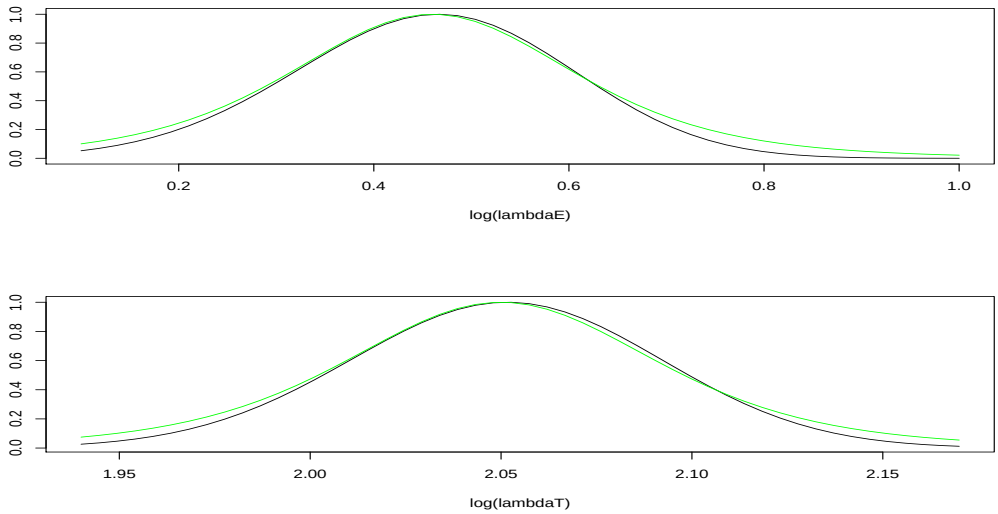


Figure 5.2: Marginal posterior profiles for styrene data

(see Appendix C.1), makes each draw very cheap as well.

The exact sampling procedure used here makes it possible to sample from the posterior of this important hierarchical model, *without* having to resort to MCMC methods and any associated ad hoc diagnostic techniques. Excellent proposal distributions for the marginal precision components are very easy to obtain via simple graphical methods. The rejection sampling algorithm is also very easy to implement correctly, and the procedure produces reliable, easy to compute Monte Carlo estimates. The rejection sampler has the added benefit, in this case, of not being affected by the curse of dimensionality when the number of random effects increases. This is because the rejection sampler continues to operate on the marginal posterior distribution of the precision parameters, which is two-dimensional, and the corresponding random effects can be quickly sampled from easy to compute conditional distributions.

Chapter 6

Conclusions and Future Work

This thesis has successfully demonstrated the application of efficient Monte Carlo and MCMC methods to some important Bayesian hierarchical models. In Chapter 2, a number of methods were described for the systematic construction of efficient MCMC block sampling algorithms for Bayesian spatial models. The chapter ended with a summary of experiences with implementing several of these algorithms, along with a recommendation of the algorithms found to be most efficient. In Chapter 3, a description was given of exact Monte Carlo methods, i.e., methods that produce independent and identically distributed samples, rather than the dependent samples returned by MCMC methods. Since it is usually hard to implement these algorithms for hard models, Chapter 4 described methods by which it becomes possible to implement exact sampling algorithms for the kind of spatial models described in Chapter 2. In particular, Chapter 4 discusses ways in which good analytical approximations to marginal posterior distribution of the variance components, and conditional distributions of the random effects can be ob-

tained. By using simple graphical methods, heavy-tailed variants of these approximate distributions can be easily computed. These can then be used to derive proposal distributions for a rejection sampling algorithm, as well as for a perfect simulation algorithm called perfect tempering. Chapter 4 ends with a comparison of the rejection sampling and perfect tempering methods. Chapter 5 describes how to use rejection sampling to draw i.i.d. samples from the posterior distribution of a Bayesian hierarchical model with two variance components, and ends by demonstrating the practicality and ease of use of such methods for real data sets.

For the Bayesian variance components model of Chapter 5, the rejection sampling algorithm is very efficient, easy to use, and completely avoids MCMC diagnostics issues. Exact sampling would therefore be an excellent practical option for such models. For the disease mapping models, if the rejection sampler or perfect tempering algorithm produces samples efficiently enough to be practical, the i.i.d. samples obtained from these algorithms should be used for inference. Of course, it is best to use rejection sampling before trying the perfect tempering algorithm, since the rejection sampler is easier to implement and debug. If neither the rejection sampler nor the perfect tempering algorithm returns i.i.d. samples quickly enough for inference, perhaps the optimal strategy to use would be to run *both* an exact sampler and an efficient MCMC algorithm. The exact sampler can be used to provide several i.i.d. samples; each i.i.d. sample can be used to independently 'restart' a fast MCMC algorithm such as the SMCMC algorithm of Chapter 2 (though any MCMC algorithm may be used in theory). This may be a practical compromise between fast, dependent MCMC sampling, and slower

i.i.d. sampling. Another option would be to run an independence Metropolis chain using the rejection proposal (envelope distribution), which would result in a uniformly ergodic chain. Some general work in the area of combining i.i.d. and dependent samples has been described by Murdoch and Rosenthal (1999), and Bandyopadhyay and Aldous (2002), but practitioners who want to take advantage of perfect simulation methods would greatly benefit from a thorough and detailed analysis of strategies for combining MCMC and exact Monte Carlo methods.

While exact simulation has been succesfully applied for some important Bayesian models in this thesis, the methods presented here are still quite specialized. An interesting area of future research would involve finding ways to apply exact methods much more routinely. With some additional effort, it is quite likely that these techniques can be extended to several other important models which have a small number of precision components. There are some difficulties to overcome for this work to be generalized: to carry out perfect tempering or rejection sampling, there is a need for establishing finite upper bounds on the ratio between the distribution of interest and the proposal distribution. Not only is it generally difficult to come up with proposals that satisfy this property, but there is a fair amount of analytical work involved in establishing these bounds. Each model or problem therefore requires a unique methodology. While it may be prohibitively difficult to establish a general method for exact sampling from a very large class of models, with some ingenuity, it may be possible to develop fairly effective methods for a class of models which have a small number of variance components. This would cover a large number of useful modeling situations, and therefore would

be of practical use to the statistics community. As shown in this work, and also by several other authors, exploiting the sparsity of matrices that arise naturally can result in substantial computational speed-ups, thereby making exact sampling methods practical in a lot of situations where it would otherwise take too long to generate good proposals.

The data sets studied in this thesis provide a sense for how well the exact simulation algorithms may work in practice. However, as the dimensions of the problem increases further, methods such as rejection sampling are known to become less practical. On the other hand, the efficiency of MCMC methods is also adversely affected by an increase in dimensions. Some preliminary work with a spatial data set resulting in a 1000-dimensional posterior distribution suggests that while it takes very long for each i.i.d. draw for such large dimensional distributions, the slow-mixing MCMC algorithms for sampling from such distributions may benefit from independent 'restarts' provided by an exact sampler. A useful area of future research would involve a thorough study of the utility of exact sampling algorithms as the dimensions of the problem gets large.

The programs for exact simulation have all been written in `R`, and are therefore relatively easy to read, use and implement. Another focus of future work would be to make the software used for these i.i.d. samplers much more "user-friendly". Ideally, it should be very easy to specify the model, input the data, and have the program produce i.i.d. draws, without any additional user input. The more automated the procedure gets, the more practical and useful it will be to practitioners.

While perfect tempering requires less stringent conditions on the proposal

distribution than the rejection sampling algorithm, in practice, it appears that the requirements are identical. It would be of great interest to see if there is a way to actually produce proposal distributions, and satisfy the domination conditions necessary for the perfect tempering algorithm without having to satisfy the rejection sampling conditions. This would make the perfect tempering algorithm more general and flexible. There is also a lot that can be done with the perfect tempering algorithm in terms of tuning the "parameters" that control how the algorithm runs. A deeper understanding of how it works, and learning how to fine-tune the algorithm would perhaps make it even more efficient, and therefore even more useful for sampling from complicated, multivariate distributions.

Another interesting area of research would involve exploring the use of what are commonly referred to as "variational techniques" as described in Jordan (1998). While variational techniques are methods for situations in which even Monte Carlo methods are impractical (Markov chain Monte Carlo methods are totally infeasible), these methods of providing upper (or lower) bounds for probabilities may be useful when we want to find bounding processes for certain distributions. Jaakolla and Jordan (1997) show how to use variational techniques in a Bayesian modeling situation by applying it to a Bayesian logistic regression model. An advantage of the methods they use is that they are extremely efficient and often produce estimates that are very accurate. A major drawback, however, is that it is hard to quantify the accuracy of the estimates. The difficulty with using variational methods to directly construct exact simulation algorithms is because the upper and lower bounds are complicated functions themselves. Typically, E-M (Expectation

Maximization) algorithms are used to maximize or minimize such functions to find fairly tight upper or lower bounds for the probabilities of interest. Hence, it may be difficult to readily obtain samples from such bounding processes. However, with some ingenuity, it may be possible to use these bounds to construct reasonable approximate distributions for the purpose of exact sampling algorithms. A successful combination of perfect simulation ideas with variational techniques would make it possible to obtain estimates whose accuracy we would be able to ascertain.

There may also be other useful generalizations or variants of the heavy-tailed distributions used here; it may be possible to use them to provide enveloping distributions even for fairly poorly behaved posterior distributions. For instance, there is work by Dickey (1966) and Dreze (1977) where they introduce poly-t distributions for obtaining approximations to posterior distributions. Using poly-t distributions, the results described here may be generalized for a larger set of priors. Sampling from the enveloping distribution for fairly complex models may also be greatly simplified by using these poly-t distributions.

An interruptible perfect simulation algorithm is one where an impatient user can abort any runs that appear to be taking too long, without biasing the samples that are finally obtained. The perfect simulation algorithms used in this thesis are non-interruptible since the Møller-Nicholls algorithm is based on the general ideas of the Propp-Wilson algorithm, which is known to be non-interruptible. Hence it would be of interest to see if there is an interruptible analogue of the Møller and Nicholls algorithm that is along the lines of the interruptible algorithm proposed by Fill (1998).

Bibliography

[1] Agarwal, D.K. and Gelfand, A.E. (2001) "Slice Gibbs sampling for simulation based fitting of spatial models," Technical report, University of Connecticut.

[2] Bandyopadhyay, A. and Aldous, D. (2002) "How to combine fast heuristic Markov chain Monte Carlo with slow exact sampling," *Electronic Commun. Probab*, **6**, 79-89.

[3] Bendat, J. and Sherman, S. (1955) "Monotone and convex operator functions," *Transactions Amer.Math.Soc.*, **79**, 58-71.

[4] Bernardinelli, L., Clayton, D. and Montomoli, C. (1995) "Bayesian estimates of disease maps : How important are priors?" *Statistics in Medicine*, **14**, 2411-2431.

[5] Besag, Julian and Green, P.J. (1993), "Spatial statistics and Bayesian computation (Disc: p53-102)", *Journal of the Royal Statistical Society, Series B*, **55**, 25-37.

[6] Besag, J., Green, P., Higdon, D. and Mengerson, K. (1995) "Bayesian computation and stochastic systems," *Statistical Science*, **10**, 3–41.

[7] Besag, J. and Kooperberg, C. (1995) "On conditional and intrinsic autoregressions," *Biometrika*, **82**, 733–746.

[8] Besag, J., York, J. and Mollie, A. (1991) "Bayesian image restoration, with applications in spatial statistics (with discussion)," *Annals of the Institute of Statistical Mathematics*, **43**, 1–59.

[9] Best, N.G., Waller, L.A., Thomas, A., Conlon, E.M. and Arnold, R.A. (1999) "Bayesian models for spatially correlated diseases and exposure data," in *Bayesian Statistics 6*, J.M.Bernardo et al. , eds., Oxford: Oxford University Press, pp.131-156.

[10] Brooks, S.P., Fan Y., and Rosenthal, J.S. "Perfect Forward Simulation via Simulated Tempering," Technical report, 2002-05, Statistical Laboratory, Cambridge.

[11] Caffo, B.S., Booth, J.G. and Davison, A.C. (2002) "Empirical sup rejection sampling," to appear in *Biometrika*.

[12] Cancer Surveillance and Control Program (1997) "Case completeness and data quality audit: Minnesota Cancer Surveillance System 1994–1995," Technical report, Minnesota Department of Health.

[13] Carlin, B.P. and Banerjee, S. (2003), "Hierarchical multivariate CAR models for spatio-temporally correlated survival data," in *Bayesian Statistics 7*, J.M.Bernardo et al. , eds., Oxford: Oxford University Press.

[14] Carlin, B.P. and Gelfand, A.E. (1991), "An iterative Monte Carlo method for nonconjugate Bayesian analysis", *Statistics and Computing*, **1**, 119–128.

[15] Casella, G., Lavine, M. and Robert, C.P. (2001), "Explaining the perfect sampler", *The American Statistician*, **55** (4), 299–305.

[16] Clayton, D.G. and Kaldor, J. (1987) "Empirical Bayes estimates of age-standardized relative risks for use in disease mapping," *Biometrics*, **43**, 671-681.

[17] Cowles, M.K. (2003) "Efficient model-fitting and model-comparison for high-dimensional Bayesian geostatistical models," *Journal of Statistical Planning and Inference*, **112 (1-2)**, 221-239.

[18] Cowles, M.K. (2002) "MCMC sampler convergence rates for hierarchical normal linear models : a simulation approach," to appear in *Statistics and Computing*.

[19] Cowles, M.K. and Carlin, B.P. (1996) "Markov chain Monte Carlo convergence diagnostics: A comparative review," *Journal of the American Statistical Association*.

[20] Cowles, M.K. and Rosenthal, J.S. (1998) "A simulation approach to convergence rates for Markov chain Monte Carlo algorithms," *Statistics and Computing*, **8**, 115-124.

[21] Damien,P., Wakefield, J. and Walker, S. (1999) "Gibbs sampling for Bayesian non-conjugate and hierarchical models by using auxiliary variables," *Journal of the Royal Statistical Society, Series B*, **61**, 331-344.

[22] Devroye, L. (1986) "Non-Uniform Random Variate Generation," New York: Springer-Verlag.

[23] Dickey, J.M. (1968) "Three multidimensional-integral identities with Bayesian applications", *The Annals of Mathematical Statistics*, **39**, 1615-1628.

[24] Dreze, J.H. (1977), "Bayesian regression analysis using poly-t densities", *Journal of Econometrics*, **6**, 329-354.

[25] Eberly, L.E. and Carlin, B.P. (2000) "Identifiability and convergence issues for Markov chain Monte Carlo fitting of spatial models," *Statistics in Medicine*, **19**, 2279-2294.

[26] Edwards, R. and Sokal, A. (1988) "Generalization of the Fortium-Kasteleyn-Swendsen-Wang representation and Monte Carlo algorithm," *Phys.Rev. D*, **38**, 2009-2012.

[27] Everson, Philip J. (2001), "Exact Bayesian inference for normal hierarchical models", *Journal of Statistical Computation and Simulation*, **68**, 223-241.

[28] Everson, Philip J., and Morris, Carl N.(2000), "Inference for multivariate normal hierarchal models", *Journal of the Royal Statistical Society, Ser. B*, **62**, 399-412.

[29] Gamerman, D., Moreira, A.R.B and Rue, H. (2002) "Space-varying regression models: specifications and simulation," to appear in *Computational Statistics and Data Analysis*.

[30] Gelman, A. and Rubin, D.B. (1992) "Inference from iterative simulation using multiple sequences," *Statistical Science*, **7**, 457-511.

[31] Gelfand, A. and Smith, A.F.M. (1990) "Sampling-Based Approaches to Calculating Marginal Densities," *Journal of the American Statistical Association*, **85**, 398-409.

[32] Gelfand, A., Sahu, S. and Carlin, B.P. (1995) "Efficient parameterizations for normal linear mixed models," *Biometrika*, **82**, 479-488.

[33] Geman, S. and Geman, D. (1992) "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," IEEE Trans. Pattn. Anal. Mach. Intel., **6**, 721-741.

[34] Geweke, J. (1992) "Evaluating the Accuracy of Sampling-based Approaches to the Calculation of Posterior Moments (with discussion)," in *Bayesian Statistics 4*, Bernardo, J.M., Berger, J., Dawid, A.P. and Smith, A.F.M. eds., Oxford: Oxford University Press, pp.169-188.

[35] Geyer, C. J. (1991) "Estimating Normalizing Constants and Reweighting Mixtures in Markov Chain Monte Carlo," Technical Report No. 568. School of Statistics, University of Minnesota.

[36] Geyer, C.J. (1992) "Practical Markov Chain Monte Carlo," *Statistical Science*, **7**, 473-483.

[37] Geyer, C.J. (1996) "Estimation and optimization of function," in *Markov Chain Monte Carlo in Practice*, Gilks et al., eds.

[38] Geyer, C.J. and Thompson, E.A. (1995), "Annealing Markov chain Monte Carlo with applications to ancestral inference", *Journal of the American Statistical Association*, **90**, 909–920.

[39] Gilks, W. (1996), "Full conditional distributions" in *Markov Chain Monte Carlo in Practice*, Gilks et al., eds.

[40] Golub, G.H. and Van Loan, C.F. (1995) *Matrix Computations, 3rd ed.*, Baltimore, MD: Johns Hopkins University Press,

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

[42] Green, P.J. and Murdoch, D.J. (1999) "Exact sampling for Bayesian inference: towards general purpose algorithms (with discussion)," in *Bayesian Statistics 6*, J.M.Bernardo et al. , eds., Oxford: Oxford University Press, pp.301-321.

[43] Haran, M., Hodges, J.S. and Carlin, B.P. (2003) "Structured Markov Chain Monte Carlo for Disease Mapping", to appear in *Journal of Computational and Graphical Statistics*.

[44] Hastings, W. K. (1970), "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, **57**, 97-109.

[45] Hodges, J.S. (1998) "Some Algebra and Geometry for Hierarchical Models, Applied to Diagnostics (with discussion)," *Journal of the Royal Statistical Society, Ser. B*, **60**, 497-536.

[46] Hodges, J.S. and Carlin, B.P. (2001) "A note on the precision of the conditionally autoregressive prior in spatial models," Research Report 2001-024, Division of Biostatistics, University of Minnesota.

[47] Horn, R.A. and Johnson, C.R. (1985) "Matrix Analysis", Cambridge University Press.

[48] Ihaka, R. and Gentleman, R. (1996) "R: A Language for Data Analysis and Graphics," *Journal of Computational and Graphical Statistics*, **5**, 299–314.

[49] Jones, G.L. and Hobert, J.P. (2001) "Honest explorations of intractable probability distributions via Markov chain Monte Carlo," *Statistical Science*, **16**, 312-334.

[50] Jones, G.L. and Hobert, J.P. (2003) "Sufficient Burn-in for Gibbs Samplers for a Hierarchical Random Effects Model," To appear in *The Annals of Statistics*.

[51] Jordan, M.I., Ghahramani, Z., Jaakkola, T.S., and Saul, L.K. (1998), "An introduction to variational methods for graphical models," in *Learning in Graphical Models*, Jordan, M.I. ed., Kluwer Academic Publishers Group (Dordrecht; Norwell, MA), 105-161.

[52] Kass, R.E., Carlin, B.P., Gelman, A. and Neal, R. (1998) "Markov chain Monte Carlo in practice: a roundtable discussion," *American Statistician*, **52**, 93-100.

[53] Knorr-Held, L. and Rue, H. (2002) "On block updating in Markov random field models for disease mapping," *Scandinavian Journal of Statistics*, **29**, 597–614.

[54] J.G. (1982) "Algorithm 582: The Gibbs-Poole-Stockmeyer and Gibbs-King Algorithms for Reordering Sparse Matrices," ACM-TRANS. MATH. SOFTWARE, **8**, 190.

[55] Lindley, D.V. and Smith, A.F.M. (1972) "Bayes estimates for the linear model (with discussion)," *Journal of the Royal Statistical Society, Ser. B*, **34**, 1-41.

[56] Liu, J.S. (1994) "The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem," *Journal of the American Statistical Association*, **89**, 958-966.

[57] Liu, J.S. and Sabatti, C. (1999) "Simulated sintering: Markov chain Monte Carlo with spaces of varying dimensions," in *Bayesian Statistics 6*, J.M.Bernardo et al. , eds., Oxford: Oxford University Press, pp.389-413.

[58] Liu, J.S., Wong, W.H. and Kong, A. (1994) "Covariance structure of the Gibbs sampler with applications to the comparisons of estimators and augmentation schemes," *Journal of the American Statistical Association*, **89**, 958-966.

[59] Löwner, K. (1934) "Über monotone Matrixfunktionen," *M*ath. Zeitschrift, 38:177-216.

[60] Lyles, R.H., Kupper, L.L. and Rappaport, S.M. (1997) "Assessing regulatory compliance of occupational exposures via the balanced one-way random effects ANOVA model," *Journal of Agricultural, Biological Environmental Statistics*, **2**, 64-86.

[61] Marinari, E. and Parisi, G. (1992), "Simulated Tempering: A New Monte Carlo Scheme," *Europhysics Letters*, **19**, 451-458.

[62] Mengerson, K. and Tweedie, R.L.(1996), "Rates of convergence of the Hastings and Metropolis algorithms, " *Annals of Statistics*, **24**, 101-121.

[63] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H. and Teller, E. (1953) "Equations of state calculations by fast computing machine," *J.Chem.Phys.*, **21**, 1087-1091.

[64] Mira, A., Møller, J., and Roberts, G. O. (2001), "Perfect slice samplers", Journal of the Royal Statistical Society, Series B, **63**, 593-606.

[65] Mira, A. and Tierney, L. (2001) "Efficiency and convergence properties of slice samplers," *Scandinavian Journal of Statistics*, **29**, 1-12.

[66] Møller, J. (1999) "Perfect simulation of conditionally specified models," *Journal of the Royal Statistical Society, Series B*, **61**, 251-264.

[67] Møller, J. and Nicholls, G. (1999) "Perfect simulation for sample-based inference," to appear in *Statistics and Computing*.

[68] Murdoch, D.J. and Rosenthal, J.S. (1999) "Efficient use of exact samples," *Statistics and Computing* **10**, 237-243.

[69] Murdoch, D.J. and Green, P.J. (1998) "Exact sampling from a continuous state space," in *Scandinavian Journal of Statistics*, **25**, 483-502.

[70] Meyn, S.P. and Tweedie, R.L. (1994) "Markov Chains and Stochastic Stability," Springer-Verlag.

[71] Mykland, P., Tierney, L., and Yu, B. (1995), "Regeneration in Markov chain samplers", Journal of the American Statistical Association, 90 , 233-241

[72] Neal, R.M. (1996) "Sampling from multimodal distributions using tempered transitions," *Statistics and Computing*, **6**, 353–366.

[73] Neal, R.M. (2003) "Slice sampling," *Annals of Statistics* (to appear).

[74] Nummelin, E. (1984) "General Irreducible Markov Chains and Non-Negative Operators," *Cambridge University Press*.

[75] Pinto, R. L. and Neal, R. M. (2001) "Improving Markov chain Monte Carlo estimators by coupling to an approximating chain", Technical Report No. 0101, Dept. of Statistics, University of Toronto.

[76] Propp, J.G. and Wilson, D.B. (1996) "Exact sampling with coupled Markov chains and applications to statistical mechanics," *Random Structures and Algorithms*, **9**, 223-252.

[77] Ripley, B.D. (1987) "Stochastic Simulation," New York: Wiley.

[78] Roberts G.O. and Rosenthal, J.S (1999) "Convergence of slice sampler Markov chains," *J*ournal of the Royal Statistical Society, Series B, **61**, 643-660.

[79] Roberts G.O. and Sahu, S. (1997) "Updating schemes, correlation structure, blocking and parameterization for the Gibbs Sampler," *Journal of the Royal Statistical Society, Series B*, **59**, 291-317.

[80] Rosenthal, J.S. (1995a) "Rates of convergence for Gibbs sampling for variance component models," *Annals of Statistics*, **23**, 740-761.

[81] Rosenthal, J.S. (1995b) "Minorization conditions and convergence rates for Markov chain Monte Carlo," *Journal of the American Statistical Association*, **23**, 740-761.

[82] Rue, H. (2001) "Fast sampling of Gaussian Markov Random Fields," *Journal of the Royal Statistical Society, Series B*, **63**, 325-338.

[83] Saad, Y. (1996) *Iterative Methods for Sparse Linear Systems*, Boston:PWS.

[84] Sargent, D.J., Hodges, J.S. and Carlin, B.P. (2000) "Structured Markov Chain Monte Carlo," *Journal of Computational and Graphical Statistics*, **9**, 217-234.

[85] Sinharay, S. and Stern, H. S. (2000) "Bayes Factors for Variance Component Testing in Generalized Linear Mixed Models," *International Society of Bayesian Analysis*, **6**, 507-516.

[86] Swendsen, R. and Wang, J. (1987) "Non-universal critical dynamics in Monte Carlo simulations," *Phys. Rev. Letters*, **58**, 86-88.

[87] Tanner, M. and Wong, W.H. "The calculation of posterior distributions by data augmentation (with discussion)," Journal of the American Statistical Association, **82**, 528-540.

[88] Tierney, L. (1994) "Markov chains for exploring posterior distributions (with discussion),", *Annals of Statistics*, **22**, 1701-1728.

[89] Tierney, L. (1996) "Introduction to general state space Markov chain theory," in *Markov Chain Monte Carlo in Practice*, Gilks et al., eds., 59-74.

[90] Tierney, L. and Kadane, J.B. (1986) "Accurate approximations for posterior moments and marginal densities," *Journal of the American Statistical Association*, **81**, 82-86.

[91] Tierney, L., Kass, R.E. and Kadane, J.B. (1989) "Fully exponential Laplace approximations to expectations and variances of non-positive functions," *Journal of the American Statistical Association*, **84**, 710-716.

[92] Von Neumann, J. (1951) "Various techniques used in connection with random digits," *National Bureau of Standards Applied Mathematics*, **12**, 36-38.

[93] Waller, L.A., Carlin, B.P., Xia, H., and Gelfand, A.E. (1997) "Hierarchical spatio-temporal mapping of disease rates," *Journal of the American Statistical Association*, **92**, 607–617.

[94] Whiley, M. and Wilson, S. (2002) "Parallel Algorithms for Markov chain Monte Carlo methods in Latent Spatial Gaussian Models,".

[95] Wilkinson, D.J. and Yeung, S.K.H (2002) "A sparse matrix approach to Bayesian computation in large linear models," *Computational Statistics and Data Analysis*, (to appear).

[96] Wilkinson, D.J. and Yeung, S.K.H (2002) "Conditional simulation from highly structured Gaussian systems, with application to blocking-

MCMC for the Bayesian analysis of very large linear models," *Statistics and Computing*, **12**, 287-300.

[97] Wilson, D.B. (2000) "How to couple from the past using a read-once source of randomness," *R*andom Structures and Algorithms, **16**, 85-113.

[98] Wolfinger, R.D. and Kass, R.E. (2000), "Nonconjugate Bayesian analysis of variance component models", *Biometrics*, **56**, 768-774.

[99] Yu, B. and Mykland, P.A. (1998) "Looking at Markov samplers through CUSUM path plots: A simple diagnostic idea," *Statistics and Computing*, **8**, 275-286.

[100] Zeger, S.L. and Karim, M.R. (1991), "Generalized linear models with random effects: A Gibbs sampling approach", *Journal of the American Statistical Association*, **86**, 79–86.

Appendix A

Block Sampling

A.1 Alternate Constraint Case Formulation

As alluded to at the end of Section 2.1.3, there is an alternate way to set up the design matrix X and the corresponding variance-covariance matrix Γ using a different specification of the joint conditional distribution of the ϕ_is (the clustering parameters). For completeness, and to outline a technique for running SMCMC marginally faster, we briefly describe the derivation of this alternate formulation. First, recall that :

$$p(\phi_1, \phi_2, ..., \phi_N|\tau_c) \propto \exp\left(-\frac{\tau_c}{2}\boldsymbol{\phi}^T Q \boldsymbol{\phi}\right).$$

where

$$Q_{ij} = \begin{cases} c & \text{if i=j where c is the number of neighbors of region i} \\ 0 & \text{if i is not adjacent to j} \\ -1 & \text{if i is adjacent to j} \end{cases}$$

Now, the spectral decomposition of $Q = UDU^T$ where $U^TU = I_N$ and D is the diagonal matrix containing eigenvalues of Q. Let D_1 be the section

of the matrix D which contains the non-zero eigenvalues. D_1 will typically have dimension $(N-1)\times(N-1)$, and we will henceforth assume it does. Let U_1 consist of the $(N-1)$ columns of the U matrix corresponding to the non-zero eigen-values in D, i.e., ${U_1}{D_1}{U_1}^T = UDU^T$. Let U_2 correspond to the columns of U excluded from U_1. Thus,

$$\begin{aligned} p({U_1}^T\boldsymbol{\phi}, {U_2}^T\boldsymbol{\phi}|\tau_c) \propto p(\phi_1,\phi_2,...,\phi_N|\tau_c) &\propto \exp\left(-\frac{\tau_c}{2}\boldsymbol{\phi}^T Q\boldsymbol{\phi}\right) \\ &= \exp\left(-\frac{\tau_c}{2}\boldsymbol{\phi}^T U \left[\begin{array}{c|c} D_1 & 0 \\ \hline 0 & 0 \end{array}\right] U^T\boldsymbol{\phi}\right) \\ &= \exp\left(-\frac{\tau_c}{2}\left[\boldsymbol{\phi}^T U_1 D_1 U_1^T\boldsymbol{\phi}\right]\right). \end{aligned}$$

We therefore have

$$p({U_1}^T\boldsymbol{\phi}|\tau_c) \sim N\left(0, {D_1}^{-1}\frac{1}{\tau_c}\right),$$

which is a proper distribution. The new constraint case specification for the clustering parameters, $\boldsymbol{\phi}$, in the design matrix must now incorporate $U_1^T\boldsymbol{\phi}$ instead of the pairwise differences used in (2.6). The corresponding design matrix, X^* of dimension $(3N-1)\times 2N$ and Γ^* matrix are as follow :

$$X^* = \left[\begin{array}{c|c} I_{N\times N} & I_{N\times N} \\ \hline -I_{N\times N} & 0_{N\times(N-1)} \\ \hline 0_{(N-1)\times N} & U_1^T \end{array}\right],$$

$$\Gamma^* = \left[\begin{array}{c|c|c} \mathrm{Diag}(1/Y_1, 1/Y_2, \ldots, 1/Y_N) & 0_{N\times N} & 0_{N\times N} \\ \hline 0_{N\times N} & \frac{1}{\tau_h} I_{N\times N} & 0_{N\times N} \\ \hline 0_{(N-1)\times N} & 0_{(N-1)\times N} & {D_1}^{-1}\times\frac{1}{\tau_c} \end{array}\right].$$

All the SMCMC schemes described in section (2.2) would still work by simply replacing X by X^* and Γ by Γ^*. However, it is very easy to show that this alternative formulation of the joint conditional distribution of the ϕ_is results in the same SMCMC candidate mean and covariance matrix as the one described in (2.2), i.e.,

$$(X^{*T}\Gamma^{*-1}X^*)^{-1}(X^{*T}\Gamma^{*-1}Y) = (X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}Y)$$
$$(X^{*T}\Gamma^{*-1}X^*)^{-1} = (X^T\Gamma^{-1}X)^{-1}$$

We omit details. The only real benefit from this reformulation is in terms of marginally reducing the speed of the algorithm since the matrix computations here would involve smaller dimensions. For instance, for the Minnesota cancer data sets we analyzed, the design matrix, X was of dimension 383×174 and Γ was of dimension 383×383 while X^* was of dimension 260×174 and Γ^* was of dimension 260×260. Since the SMCMC procedures typically involve only occasional matrix computations (once every 100 samples for the SMCMC scheme discussed here) and the reduction in dimension is not dramatic, the alternative representation does not represent a significant reduction in the execution time of the sampler.

A.2 Algorithm Details

The following algorithm corresponds to the SMCMC scheme described in Section 2.2:

1. Initialize the parameter values $\mathbf{\Theta}$ to any starting values. We set $\theta_i = 0$ and $\phi_i = \log(Y_i/E_i)$, for i=1 to N. This starts each $\mu_i = \theta_i + \phi_i$ at $\log(Y_i/E_i)$, its MLE value.

2. At every 100th iteration, sample τ_h and τ_c from their posterior distributions. For instance, with our conjugate Gamma priors $\tau_h|\boldsymbol{\Theta} \sim \text{Gamma}(\alpha_h + N/2, 1/(\sum_{i=1}^{N} \theta_i^2/2 + 1/\beta_h))$ and, $\tau_c|\boldsymbol{\Theta} \sim \text{Gamma}(\alpha_c + N/2, 1/(\sum_{i\sim j}^{N} (\phi_i - \phi_j)^2/2 + 1/\beta_h))$. Use these values of τ_h and τ_c in Γ. Recall that Γ is a diagonal matrix containing τ_h and τ_c as described in (2.7).

3. Every time a new sample for the hyperparameters (τ_h, τ_c) is generated, compute candidate mean, $M_{cand}^{(t)} = (X^T(\Gamma)^{-1}X)^{-1}(X^T(\Gamma)^{-1}\hat{\boldsymbol{\mu}})$ and candidate variance-covariance matrix, $\Sigma_{cand} = (X^T\Gamma^{-1}X)^{-1}$. $\hat{\boldsymbol{\mu}}$ is a vector with $\hat{\boldsymbol{\mu}}_{\boldsymbol{i}} = \log(Y_i/E_i)$, the MLE for μ_i. This plays the role of data that is linear in the model parameters. To avoid having to compute inverses for the candidate mean and candidate covariance, and for greater numerical stability, we use some standard linear algebra tricks (see Golub and Van Loan (1996)):

   (a) Cholesky decompose $(X^T(\Gamma)^{-1}X) = LL^T$. Since Γ is diagonal, it is very easy to invert.

   (b) Solve $LL^T\beta = X^T(\Gamma)^{-1}\hat{\boldsymbol{\mu}}$ for $L^T\beta = \eta$, say. This is a simple lower triangular system of equations.

   (c) Solve $L^T\beta = \eta$ for β using the solution η from above. This is now a simple upper triangular system of equations. Thus $\beta = (X^T(\Gamma)^{-1}X)^{-1}(X^T(\Gamma)^{-1}\hat{\boldsymbol{\mu}})$ as required.

   (d) To get Σ_{cand} error structure, multiply the vector of independent normals by the inverse of L^T

4. Generate $\boldsymbol{\Theta}^* \sim N(M_{cand}, \Sigma_{cand})$ using the results of the previous step.

5. Accept $\mathbf{\Theta^*}$, i.e., set $\mathbf{\Theta} = \mathbf{\Theta^*}$ with probability α, the usual Metropolis-Hastings ratio. Note that the Hastings ratio can be computed efficiently using L and L^{-1} already computed above so there is still no need to really evaluate the covariance matrix.

$$\alpha = \min\left(1, \frac{p(\Theta^*)q(\Theta)}{q(\Theta^*)p(\Theta)}\right)$$

where it can be shown that

$$\log(p(\Theta)) \propto -\sum_{i=1}^{N} E_i e^{\mu_i} + \sum_{i=1}^{N} \mu_i Y_i - \frac{\tau_h}{2}\sum_{i=1}^{N}\theta_i^2 - \frac{\tau_c}{2}\sum_{i\sim j}(\phi_i - \phi_j)^2,$$
$$\log(q(\Theta)) \propto -\frac{1}{2}(\Theta - M_{cand})^T \Sigma_{cand}(\Theta - M_{cand}).$$

6. Return to step 2. Note that we "oversample" $\mathbf{\Theta}$, i.e., for each (τ_h, τ_c) sampled, 100 samples of $\mathbf{\Theta}$ are generated from an independence chain using the candidate mean and covariance computed in step 3.

Distributions for Algorithms Used in SMCMC

Univariate algorithm (UMCMC). The full conditionals for the precision parameters below are used in almost all the algorithms

$$\tau_h|... \sim G\left(N/2 + \alpha_h, \left(\sum_{i=1}^{N}\theta_i^2/2 + 1/\beta_h\right)^{-1}\right)$$
$$\tau_c|... \sim G\left(M/2 + \alpha_c, \left(\sum_{i\sim j}(\phi_i - \phi_j)^2/2 + 1/\beta_c\right)^{-1}\right)$$

We use random walk Metropolis-Hastings to sample from these distributions, with different standard deviations for the ϕ_is and θ_is random walk proposals.

The standard deviations were tuned to have acceptance rates between 30% to 60%. The full conditionals for the θ_is, ϕ_is are given below.

$$\theta_i|... \propto \exp(\theta_i Y_i - E_i e^{\theta_i+\phi_i}) \times \exp(-\tau_h \theta_i^2/2)$$
$$\phi_i|... \propto \exp(\phi_i Y_i - E_i e^{\theta_i+\phi_i}) \times \exp(-\tau_c \sum_{i\sim j}(\phi_i - \phi_j)^2/2)$$

Reparameterized univariate algorithm (RUMCMC). Set $\mu_i = \theta_i + \phi_i$, then sample (μ_i, ϕ_i)s instead of (θ_i, ϕ_i)s. The **full joint distribution** is now

$$\begin{aligned} P(\cdot) = [\boldsymbol{\Theta}, (\tau_h, \tau_c)|Y] \propto & \exp\left(\sum_{i=1}^{N}(\mu_i Y_i - E_i e^{\mu_i})\right) \\ & \times \tau_h^{N/2} \exp\left(-\frac{1}{2}(\boldsymbol{\mu}-\boldsymbol{\phi})^T(\tau_h I)(\boldsymbol{\mu}-\boldsymbol{\phi})\right) \\ & \times \tau_c^{M/2} \exp\left(-\frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}\right) \\ & \times \tau_h^{\alpha_h - 1} \exp\left(-\tau_h/\beta_h\right) \times \tau_c^{\alpha_c - 1} \exp\left(-\tau_c/\beta_c\right). \end{aligned}$$

And the corresponding full conditionals for the precision parameters are:

$$\tau_h|... \sim G\left(N/2 + \alpha_h, \left(\sum_{i=1}^{N}(\mu_i - \phi_i)^2/2 + 1/\beta_h\right)^{-1}\right)$$
$$\tau_c|... \sim G\left(M/2 + \alpha_c, \left(\sum_{i\sim j}(\phi_i - \phi_j)^2/2 + 1/\beta_c\right)^{-1}\right)$$

The full conditionals for the μ_is, ϕ_is are given below. As in UMCMC, we use random walk Metropolis-Hastings to sample from these distributions:

$$\mu_i|... \propto \exp(\mu_i Y_i - E_i e_i^{\mu}) \times \exp(-\tau_h(\mu_i - \phi_i)^2/2)$$
$$\phi_i|... \propto \exp(-\tau_h(\mu_i - \phi_i)^2/2 - \tau_c \sum_{i\sim j}(\phi_i - \phi_j)^2/2)$$

We also tried a variation on the above RUMCMC algorithm by noticing that the full conditional distribution of ϕ_i is proportional to a Normal density,

and thereby Gibbs sampling these parameters.

$$\phi_i|... \sim N\left(\frac{\tau_h\mu_i + \tau_c\sum_{i\sim j}\phi_j}{\tau_h + \tau_c N_i}, \frac{1}{\tau_h + \tau_c N_i}\right)$$

The difference in performance was negligible, so we did not report our results.

Blocking-by-geographical-proximity:

1. Sample τ_h and τ_c from their gamma full conditionals at every iteration.

2. Instead of sampling all of $\boldsymbol{\Theta}$ at once, sample each smaller block one at a time. For instance, if a block b of size k contains regions $b_1, b_2, b_3, ..b_k$, then the corresponding block of parameters sampled would be $\{\theta_{b_1}, \theta_{b_2}, ..., \theta_{b_k}, \phi_{b_1}, \phi_{b_2}, ..., \phi_{b_k}\}$. This is done for each block as follows :

   (a) Take the SMCMC proposal mean vector and covariance matrix and break it into sections corresponding to the mean and covariance for each block of parameters respectively. These are the proposal mean and covariance matrix for each block of parameters. Thus, we are using an approximation to the conditional distribution of the parameters as a proposal for the marginal distribution of the parameters.

   (b) Accept-reject each block separately using a Metropolis-Hastings ratio.

All-at-once Structured MCMC/blocking with precision components:

1. Sample τ_h and τ_c from some proposal distribution. For instance, we use a proposal (suggested by Knorr-Held and Rue (2000)) which generates a τ_h candidate by multiplying the current value of τ_h by a Uniform on

$(1/f, f)$ where f is a tuning constant. We could also use a Gamma proposal but it appears to be difficult to tune the parameters to obtain good acceptance rates for the large block.

2. Using the generated value of τ_h and τ_c, compute the SMCMC proposal mean and covariance matrix.

3. Sample $\boldsymbol{\Theta}$ using the SMCMC proposal.

4. Accept-reject (τ_h, τ_c, $\boldsymbol{\Theta}$) as one large block using a Metropolis-Hastings ratio.

Reparameterized Structured MCMC (RSMCMC):
This algorithm is the SMCMC analogue of the reparameterizated univariate algorithm (RUMCMC). The algorithm follows exactly the same steps as the SMCMC algorithm, with the only difference being that $\boldsymbol{\Theta}$ is now $(\boldsymbol{\mu}, \boldsymbol{\phi})$ instead of $(\boldsymbol{\theta}, \boldsymbol{\phi})$, and the proposal distribution is adjusted according to the new parameterization. It is obtained in the same manner as before, as described in Section 2.1.3.

Consider a data set of N regions with C pairs of adjacent neighbors. Thus, there are $2N + 2$ model parameters: $\{\mu_i : \ i = 1, \ldots N\}$, $\{\phi_i : \ i = 1, \ldots N\}$, τ_h and τ_c. The RSMCMC algorithm requires that we transform the Y_i data points to $\hat{\mu}_i = \log(Y_i/E_i)$, which can be conveniently thought of as the response since they should be roughly linear in the model parameters (the θ_i's and ϕ_i's). For the constraint case formulation, the different levels of the model are written down case by case. The data cases are $\hat{\mu}_i$, $i = 1, \ \ldots, N$. The constraint cases for the μ_i's are $\mu_i|\phi_i \sim N(\phi_i, 1/\tau_h)$, $i = 1, \ \ldots, N$. For the constraint cases involving the ϕ_i's, the differences between

the neighboring ϕ_i's can be used to get an unconditional distribution for the ϕ_i's using pairwise differences as in (2.3).

We proceed in the same fashion as in Subsection 2.1.3. The SMCMC candidate generating distribution is thus of the form (2.2), with the Y_i's replaced by $\hat{\boldsymbol{\mu}}$:

$$\Theta|\hat{\boldsymbol{\mu}}, \Gamma \sim N((X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}\hat{\boldsymbol{\mu}})\ ,\ (X^T\Gamma^{-1}X)^{-1}),$$

where $\boldsymbol{\Theta}=\{\mu_1, \cdots \mu_N, \phi_1, \cdots \phi_N\}^T$. The conditional distribution of the μ_i's is given by

$$p(\mu_1, \mu_2, ..., \mu_N|\phi_1, \phi_2, ..., \phi_N, \tau_h) \propto \tau_h^{N/2} \exp\left\{-\frac{\tau_h}{2}\sum_{i=1}^{N}(\mu_i - \phi_i)^2\right\}.$$

The response vector is $\hat{\boldsymbol{\mu}}^T = \{\log(Y_1/E_1), \ldots, \log(Y_N/E_N)\}$, as before. The $(2N + C) \times 2N$ design matrix for the spatial model is now defined as:

$$X = \left[\begin{array}{c|c} I_{N\times N} & 0_{N\times N} \\ \hline -I_{N\times N} & I_{N\times N} \\ \hline 0_{C\times N} & A_{C\times N} \end{array}\right]. \tag{A.1}$$

The design matrix is divided into two halves, the left half corresponding to the N μ_i's and the right half referring to the N ϕ_i's. The top section of this design matrix is a $N \times 2N$ matrix relating $\hat{\mu}_i$ to the model parameters μ_i and ϕ_i. In the ith row, a 1 appears in the ith and $(N + i)$th columns while 0s appear elsewhere. Thus the ith row corresponds to $\mu_i = \theta_i + \phi_i$. The middle section of the design matrix is an $N \times 2N$ matrix which imposes a stochastic constraint on each μ_i separately (θ_i's are i.i.d normal with mean ϕ_i, variance $1/\tau_h$). The bottom section of the design matrix is a $C \times 2N$ matrix with each row having a -1 and 1 in the $(N + k)$th and $(N + l)$th columns respectively,

corresponding to a stochastic constraint being imposed on $\phi_l - \phi_k$ (using the pairwise difference form of the prior on the ϕ_i's as described in (2.3) with regions l and k being neighbors). The variance-covariance matrix Γ is a diagonal matrix with the top left section corresponding to the variances of the data cases, i.e., the $\hat{\mu}_i$'s. Using the variance approximations described above, the $(2N + C) \times (2N + C)$ block diagonal variance-covariance matrix is as before the reparameterization

$$\Gamma = \left[\begin{array}{c|c|c} \text{Diag}(1/Y_1, 1/Y_2, \ldots, 1/Y_N) & 0_{N\times N} & 0_{N\times C} \\ \hline 0_{N\times N} & \frac{1}{\tau_h} I_{N\times N} & 0_{N\times C} \\ \hline 0_{C\times N} & 0_{C\times N} & \frac{1}{\tau_c} I_{C\times C} \end{array}\right]. \qquad \text{(A.2)}$$

Once we have altered the set up to account for the reparameterization, we can implement the RSMCMC algorithm in similar fashion to the SMCMC algorithm.

Appendix B

Exact Sampling for Disease Mapping

B.1 Deriving Proposals

B.1.1 The Multivariate-t Distribution

We generate a multivariate-t variate of dimension N, with mean $\boldsymbol{\mu}$, variance Σ and ν degrees of freedom in the following manner:

(1) Generate $Z \sim N_N(\mathbf{0}^T, I)$

(2) Generate $s \sim \chi^2_\nu$

(3) Set $T = Z/\sqrt{(s/\nu)}$

(4) Set $\Sigma_T = \Sigma(\frac{\nu-2}{\nu})$.

(5) Find U=choleski decomposition of Σ_T, s.t., $U^T U = \Sigma_T$.

(6) Set $M = U^T T + \boldsymbol{\mu}$.

Now,

$$\begin{aligned} E(M) =& E(E(M|s)) = E(\mathbf{0} + \boldsymbol{\mu}) = \boldsymbol{\mu}, \text{ and} \\ Var(M) =& E(Var(M|s)) + Var(E(M|s)) = E(\frac{\nu}{s}\Sigma_T) + Var(\boldsymbol{\mu}) \\ =& E(\frac{\nu}{s}\Sigma\frac{\nu-2}{\nu}) + 0 = \Sigma(\nu-2)E(\frac{1}{s}) = \Sigma(\nu-2)\frac{1}{\nu-2} = \Sigma. \end{aligned}$$

where E(1/s) is computed as the expectation of an inverse gamma with parameters $(\nu/2, 2)$. The density of the multivariate-t distribution of N dimensions with mean, μ_t, variance, Σ_t and ν degrees of freedom is:

$$x \sim \frac{|\Sigma_t|^{-0.5}\Gamma\left(\frac{\nu+N}{2}\right)}{(\nu\pi)^{N/2}\Gamma(\nu/2)} \times \left(1 + \frac{1}{\nu}(x-\mu_t)^T\Sigma_t^{-1}(x-\mu_t)\right)^{-(\nu+N)/2}.$$

B.1.2 Deriving Marginal and Conditional Distributions

We first note that if x is n-dimensional and

$$f(x) = \exp\left\{-\frac{1}{2}\left(x^TCx + Dx + k\right)\right\}$$

then

$$\int f(x)dx = (2\pi)^{n/2}\det(C)^{-1/2}\exp\left\{-\frac{1}{2}\left(\hat{x}^TC\hat{x} + D\hat{x} + k\right)\right\}$$

where $\hat{x}$ solves

$$2C\hat{x} + D^T = 0$$

In the following subsections we show how we derive the marginal posterior distributions of interest for the various models we consider.

Model 1

The joint distribution for Model 1 is

$$P(\boldsymbol{\phi}, \tau_c) \propto \exp\left(\sum_{i=1}^{N}(\phi_iY_i - E_ie^{\phi_i}) - \frac{1}{2}\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi} - \tau_c/\beta_c\right)\tau_c^{M/2+\alpha_c-1}.$$

The approximate joint distribution for Model 1 based on (4.3) and (4.4) is

$$\hat{P}(\boldsymbol{\phi}, \tau_c) \propto \exp\left(-\frac{1}{2}(\hat{\boldsymbol{\mu}} - \boldsymbol{\phi})^T V^{-1}(\hat{\boldsymbol{\mu}} - \boldsymbol{\phi}) - \frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi} - \tau_c/\beta_c\right) \tau_c^{M/2+\alpha_c-1}$$

where V^{-1}=diag$(1/Y_1, ..1/Y_N)$. To obtain the approximate marginal posterior distribution $S_1(\tau_c)$ upto a constant, we integrate out $\boldsymbol{\phi}$ from above.

$$S_1(\tau_c) = \tau_c^{M/2+\alpha_c-1} \exp(-\tau_c/\beta_c) \det(C)^{-1/2} \exp\left\{-\frac{1}{2}\left(\hat{\boldsymbol{\phi}}^T C\hat{\boldsymbol{\phi}} + D\hat{\boldsymbol{\phi}}\right)\right\}. \tag{B.1}$$

with

$$\begin{aligned} &C = V^{-1} + \tau_c Q, \quad D = -2\hat{\boldsymbol{\mu}}^T V^{-1}, \text{ and } \quad k = \hat{\boldsymbol{\mu}}^T V^{-1}\hat{\boldsymbol{\mu}} \\ &\hat{\boldsymbol{\phi}} = \left(V^{-1} + \tau_c Q\right)^{-1} V^{-1}\hat{\boldsymbol{\mu}}. \end{aligned} \tag{B.2}$$

The proposal for τ_c is a log-t distribution,

$$R_1(\tau_c; \mu_c, \sigma_c, \nu) \propto \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_c) - \mu_c}{\sigma_c}\right)^2\right]^{-(\nu+1)/2}. \tag{B.3}$$

Our proposal for the model parameters is a multivariate-t version of (4.7)

$$\begin{aligned} R_2(\boldsymbol{\phi}|\tau_c) &\sim MT(\mu_N, \Sigma_T, \nu) \\ &\propto \frac{|\Sigma_T|^{-0.5}\Gamma\left(\frac{\nu+N}{2}\right)}{(\nu\pi)^{N/2}\Gamma(\nu/2)} \times \left(1 + \frac{1}{\nu}(\boldsymbol{\phi} - \mu_N)^T\Sigma_T^{-1}(\boldsymbol{\phi} - \mu_N)\right)^{-(\nu+N)/2}, \end{aligned}$$

where $\mu_n = (V^{-1} + \tau_c Q)^{-1}V^{-1}\hat{\boldsymbol{\mu}}$, $\Sigma_T = \frac{\nu}{\nu-2}(V^{-1} + \tau_c Q)^{-1}$, $\nu = 50$. The enveloping distribution for $P(\boldsymbol{\phi}, \tau_c)$ is

$$R(\boldsymbol{\phi}, \tau_c) = R_1(\tau_c; \mu_c, \sigma_c, \nu) R_2(\boldsymbol{\phi}|\tau_c).$$

Model 2

The full posterior distribution for Model 2 is

$$P(\boldsymbol{\Theta},(\tau_h,\tau_c)) \propto \exp\left(\sum_{i=1}^{N}((\theta_i+\phi_i)Y_i - E_i e^{\theta_i+\phi_i})\right)\exp\left(-\frac{1}{2}\boldsymbol{\theta}^T(\tau_h I)\boldsymbol{\theta}\right)$$
$$\times\tau_h^{N/2+\alpha_h-1}\tau_c^{M/2+\alpha_c-1}\exp\left(-\frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}\right)\exp\left(-\frac{\tau_h}{\beta_h}-\frac{\tau_c}{\beta_c}\right). \tag{B.4}$$

The approximate joint posterior distribution based on (4.5), (4.6) is:

$$\hat{P}(\boldsymbol{\Theta},(\tau_h,\tau_c)) \propto \exp\left(-\frac{1}{2}(\hat{\boldsymbol{\mu}}-(\boldsymbol{\theta}+\boldsymbol{\phi}))^T V^{-1}(\hat{\boldsymbol{\mu}}-(\boldsymbol{\theta}+\boldsymbol{\phi}))\right)$$
$$\times\exp\left(-\frac{1}{2}\boldsymbol{\theta}^T(\tau_h I)\boldsymbol{\theta}-\frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}\right)$$
$$\times\tau_h^{N/2+\alpha_h-1}\tau_c^{M/2+\alpha_c-1}\exp\left(-\tau_h/\beta_h-\tau_c/\beta_c\right)$$

To obtain the marginal posterior distribution $S_1(\tau_h,\tau_c)$ upto a constant, we integrate out $\boldsymbol{\Theta}$ from the above function.

$$S_1(\tau_h,\tau_c) = \tau_h^{N/2+\alpha_h-1}\tau_c^{M/2+\alpha_c-1}\exp\left(-\tau_h/\beta_h-\tau_c/\beta_c\right).$$
$$\times\left(\det(\tau_h V^{-1}+\tau_c V^{-1}Q+\tau_h\tau_c Q)\right)^{-1/2}$$
$$\times\exp\left\{-\frac{1}{2}\left(\hat{\boldsymbol{\Theta}}^T C\hat{\boldsymbol{\Theta}}+D\hat{\boldsymbol{\Theta}}+k\right)\right\}.$$

with

$$C_{2N\times 2N}=\begin{bmatrix}\overbrace{V^{-1}+\tau_h I}^{\boldsymbol{\theta}} & \overbrace{+V^{-1}}^{\boldsymbol{\phi}}\\ +V^{-1} & V^{-1}+\tau_c Q\end{bmatrix},$$

and

$$D_{1\times 2N}=(-2\hat{\boldsymbol{\mu}}^T V^{-1},-2\hat{\boldsymbol{\mu}}^T V^{-1}),$$
$$\hat{\boldsymbol{\Theta}}^T=(\hat{\boldsymbol{\theta}}^T,\hat{\boldsymbol{\phi}}^T),$$

where $\hat{\boldsymbol{\theta}} = \frac{\tau_c}{\tau_h} Q(I + \frac{\tau_c}{\tau_h} Q + \tau_c VQ)^{-1}\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\phi}} = (I + \frac{\tau_c}{\tau_h} Q + \tau_c VQ)^{-1}\hat{\boldsymbol{\mu}}$. The proposal for (τ_h, τ_c) is a product of log-t distributions,

$$R_1(\tau_h, \tau_c) \propto \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_c) - \mu_c}{\sigma_c}\right)^2\right]^{-(\nu+1)/2} \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_h) - \mu_h}{\sigma_h}\right)^2\right]^{-(\nu+1)/2}.$$

Our proposal for the model parameters in this case, based on a multivariate-t version of (4.9) is

$$R_2(\boldsymbol{\Theta}; \tau_h, \tau_c) = \frac{|\Sigma_T|^{-0.5}\Gamma\left(\frac{\nu+N}{2}\right)}{(\nu\pi)^{N/2}\Gamma(\nu/2)} \left(1 + \frac{1}{\nu}(\boldsymbol{\Theta} - \mu_N)^T \Sigma_T^{-1}(\boldsymbol{\Theta} - \mu_N)\right)^{-(\nu+N)/2}.$$

where $\mu_N = (X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}\hat{\boldsymbol{\mu}})$, $\Sigma_T = (X^T\Gamma^{-1}X)^{-1}$, and $\nu = 50$. X, Γ are as described in Section 4.2.2. The envelope for $P(\boldsymbol{\Theta}, \tau_h, \tau_c)$ is

$$R(\boldsymbol{\Theta}, \tau_h, \tau_c) = R_1(\tau_h, \tau_c; \mu_h, \mu_c, \mu_c, \sigma_c, \nu) R_2(\boldsymbol{\Theta}|\tau_h, \tau_c).$$

B.2 Proofs for Envelopes

B.2.1 Model 1

We give proofs that the envelope, $R_1(\tau_c)R_2(\boldsymbol{\phi}; \tau_c)$, used for sampling from Model 1, $P(\boldsymbol{\phi}, \tau_c)$, truly is an envelope. We prove this by showing that there exist $b_1, b_2 > 0$, such that $\forall \boldsymbol{\phi} \in R^N, \tau_c > 0$, (i) and (ii) below are true

$$\left.\begin{aligned} &(i) f_1(\tau_c) = \frac{S_1(\tau_c)}{R_1(\tau_c)} \leq b_1, \\ &(ii) f_2(\boldsymbol{\phi}, \tau_c) = \frac{P(\boldsymbol{\phi}, \tau_c)}{S_1(\tau_c)R_2(\boldsymbol{\phi}; \tau_c)} \leq b_2 \end{aligned}\right\} \quad \Rightarrow \frac{P(\boldsymbol{\phi}, \tau_c)}{R_1(\tau_c)R_2(\boldsymbol{\phi}; \tau_c)} \leq b_1 b_2. \tag{B.5}$$

We begin by proving assertion (i). By rearranging terms and rewriting C and $\hat{\boldsymbol{\phi}}$ in terms of $V, \boldsymbol{\phi}, \tau_c, Q, \hat{\boldsymbol{\mu}}$ we get

$$f_1(\tau_c) \propto \tau_c^{M/2+\alpha_c} \exp(-\tau_c/\beta_c) \times \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_c) - \mu}{\sigma}\right)^2\right]^{(\nu+1)/2} \times \det(V^{-1} + \tau_c Q)^{-1/2} \exp\left\{\frac{1}{2}(V^{-1}\hat{\boldsymbol{\mu}})^T(V^{-1} + \tau_c Q)^{-1}(V^{-1}\hat{\boldsymbol{\mu}})\right\}.$$

Since Q is positive semi-definite (Besag and Kooperberg, 1995), and $V > 0$, we can use the Löwner partial ordering (Löwner, 1934; Bendat and Sherman, 1955; also see Horn and Johnson, 1985) to obtain:

$$\begin{aligned} &\det(V^{-1} + \tau_c Q)^{-1/2} \leq \det(V^{-1})^{-1/2} = \det(V)^{1/2} \quad \text{and} \\ &(V^{-1}\hat{\boldsymbol{\mu}})^T(V^{-1} + \tau_c Q)^{-1}(V^{-1}\hat{\boldsymbol{\mu}}) \leq \hat{\boldsymbol{\mu}}^T V^{-1}\hat{\boldsymbol{\mu}}. \end{aligned} \tag{B.6}$$

and therefore,

$$\begin{aligned} f_1(\tau_c) \leq &\tau_c^{M/2+\alpha_c} \exp(-\tau_c/\beta_c) \times \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_c) - \mu}{\sigma}\right)^2\right]^{(\nu+1)/2} \\ &\times \det(V)^{1/2} \exp\left\{\frac{1}{2}\hat{\boldsymbol{\mu}}^T V^{-1}\hat{\boldsymbol{\mu}}\right\}, \end{aligned} \tag{B.7}$$

which is clearly bounded for $\tau_c > 0$ since $\det(V)^{1/2} \exp\left\{\frac{1}{2}\hat{\boldsymbol{\mu}}^T V^{-1}\hat{\boldsymbol{\mu}}\right\}$ is constant and what remains is the ratio of a gamma to a log-t distribution.

We now prove the second part of (B.5). Canceling common terms, we get $f_2(\boldsymbol{\phi}, \tau_c)$ as

$$\frac{\exp\left(\sum_{i=1}^N (\phi_i Y_i) - \sum_{i=1}^N E_i e^{\phi_i} - \frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi} + \frac{1}{2}\left(\hat{\boldsymbol{\phi}}^T C\hat{\boldsymbol{\phi}} + D\hat{\boldsymbol{\phi}}\right)\right)}{\frac{|\Sigma_T|^{-0.5}\Gamma\left(\frac{\nu+N}{2}\right)}{(\nu\pi)^{N/2}\Gamma(\nu/2)} \times \left(1 + \frac{1}{\nu}(\boldsymbol{\phi} - \mu_N)^T\Sigma_T^{-1}(\boldsymbol{\phi} - \mu_N)\right)^{-(\nu+N)/2} \det(C)^{-0.5}}$$

Clearly, $|\Sigma_T|^{-0.5}$ cancels out with $\det(C)^{-0.5}$ (upto a constant), and

$$\hat{\boldsymbol{\phi}}^T C\hat{\boldsymbol{\phi}} + D\hat{\boldsymbol{\phi}} = -(V^{-1}\hat{\boldsymbol{\mu}})^T(V^{-1} + \tau_c Q)^{-1}(V^{-1}\hat{\boldsymbol{\mu}}).$$

Furthermore, we can show that $(\boldsymbol{\phi}-\mu_N)^T\Sigma_T^{-1}(\boldsymbol{\phi}-\mu_N)$ is bounded above by

$$(\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}+\boldsymbol{\phi}^T(V^{-1})\boldsymbol{\phi}-2(V^{-1}\hat{\boldsymbol{\mu}})^T\boldsymbol{\phi})(\frac{\nu-2}{\nu})+(\hat{\boldsymbol{\mu}}^TV^{-1}\hat{\boldsymbol{\mu}})(\frac{\nu-2}{\nu}).$$

Thus, we get $f_2(\boldsymbol{\phi},\tau_c)\leq$

$$\begin{aligned}
&\exp\left(\sum_{i=1}^N(\phi_iY_i)-\sum_{i=1}^N E_ie^{\phi_i}-\frac{1}{2}\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}-\frac{1}{2}(V^{-1}\hat{\boldsymbol{\mu}})^T(V^{-1}+\tau_cQ)^{-1}(V^{-1}\hat{\boldsymbol{\mu}})\right)\\
&\times\left(1+k_2\left(\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}-2(V^{-1}\hat{\boldsymbol{\mu}})^T\boldsymbol{\phi}+\hat{\boldsymbol{\mu}}^TV^{-1}\hat{\boldsymbol{\mu}}+\boldsymbol{\phi}^TV^{-1}\boldsymbol{\phi}\right)\right)^{(\nu+N)/2}\\
&=\exp\left(\sum_{i=1}^N(\phi_iY_i)-\sum_{i=1}^N E_ie^{\phi_i}-\frac{1}{2}\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}-\frac{1}{2}(V^{-1}\hat{\boldsymbol{\mu}})^T(V^{-1}+\tau_cQ)^{-1}(V^{-1}\hat{\boldsymbol{\mu}})\right)\\
&\times\left(1+k_2\left(\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}+(\boldsymbol{\phi}-\hat{\boldsymbol{\mu}})^TV^{-1}(\boldsymbol{\phi}-\hat{\boldsymbol{\mu}})\right)\right)^{(\nu+N)/2},
\end{aligned}$$

where $k_2=\frac{(\nu-2)}{2\nu^2}>0$. But, $(V^{-1}\hat{\boldsymbol{\mu}})^T(V^{-1}+\tau_cQ)^{-1}(V^{-1}\hat{\boldsymbol{\mu}})>0\ \forall\tau_c$ since $(V^{-1}+\tau_cQ)^{-1}$ is positive definite. Hence

$$\begin{aligned}
f_2(\boldsymbol{\phi},\tau_c)\leq&\exp\left(\sum_{i=1}^N(\phi_iY_i)-\sum_{i=1}^N E_ie^{\phi_i}-\frac{1}{2}\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}\right)\\
&\times\left(1+k_2\left(\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}+(\boldsymbol{\phi}-\hat{\boldsymbol{\mu}})^TV^{-1}(\boldsymbol{\phi}-\hat{\boldsymbol{\mu}})\right)\right)^{(\nu+N)/2}.
\end{aligned}$$

Using the fact that $(1+x+y)^p\leq(1+x)^p(1+y)^p$ if $x,y\geq0$ and $p>1$,

$$\begin{aligned}
f_2(\boldsymbol{\phi},\tau_c)\leq&\exp\left(\sum_{i=1}^N(\phi_iY_i)-\sum_{i=1}^N E_ie^{\phi_i}-\frac{1}{2}\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}\right)\\
&\times\left(1+k_2\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}\right)^{(\nu+N)/2}\left(1+k_2\sum_{i=1}^N(\phi_i-\hat{\mu}_i)^2Y_i\right)^{(\nu+N)/2}
\end{aligned}\tag{B.8}$$

Our proof has now been reduced to showing that the R.H.S. of (B.8) is bounded. We consider two cases:
(a) if $\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}\neq0$, then $\exp(-\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi})$ dominates $\left(1+k_2\boldsymbol{\phi}^T(\tau_cQ)\boldsymbol{\phi}\right)^{(\nu+N)/2}$, and $\exp\left(\sum_{i=1}^N(\phi_iY_i)-\sum_{i=1}^N E_ie^{\phi_i}\right)$ dominates $\left(1+k_2\sum_{i=1}^N(\phi_i-\hat{\mu}_i)^2Y_i\right)^{(\nu+N)/2}$.

Since $-\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi} \leq 0$ and $\exp\left(\sum_{i=1}^N(\phi_i Y_i) - \sum_{i=1}^N E_i e^{\phi_i}\right)$ is bounded, the R.H.S. of (B.8) is automatically bounded.

(b) If $\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi} = 0$ (either because $\boldsymbol{\phi}$ lies in the null space of Q or because $\tau_c \to 0$), then we need to prove that

$$\exp\left(\sum_{i=1}^N(\phi_i Y_i) - \sum_{i=1}^N E_i e^{\phi_i}\right)\left(1 + k_2\sum_{i=1}^N(\phi_i - \hat{\mu}_i)^2 Y_i\right)^{(\nu+N)/2} \tag{B.9}$$

is bounded. For N=1, we have

$$\exp\left(\phi_1 Y_1 - E_1 e^{\phi_1}\right)\left(1 + k_2(\phi_1 - \hat{\mu}_1)^2 Y_1\right)^{(\nu+1)/2}. \tag{B.10}$$

If $Y_1 = 0$, R.H.S. of (B.10) is $\exp(-E_1 e^{\phi_1})$, clearly bounded. Since we have $Y_1 > 0$, we know $\exp\left(\phi_1 Y_1 - E_1 e^{\phi_1}\right)\left(1 + k_2(\phi_1 - \hat{\mu}_1)^2 Y_1\right)^{(\nu+1)/2} \to 0$ both as $\phi_1 \to \infty$ and as $\phi_1 \to -\infty$. Since we know (B.10) is a continuous function of ϕ_1, by standard results from analysis, we have proved that (B.10) is bounded $\forall \phi_1$ by some constant B_1, say. For N dimensions,

$$\begin{aligned}
&\exp\left(\sum_{i=1}^N(\phi_i Y_i) - \sum_{i=1}^N E_i e^{\phi_i}\right)\left(1 + k_2\sum_{i=1}^N(\phi_i - \hat{\mu}_i)^2 Y_i\right)^{(\nu+N)/2} \\
=&\exp\left(\sum_{i=1}^{N-1}(\phi_i Y_i) - \sum_{i=1}^{N-1} E_i e^{\phi_i}\right)\exp\left(\phi_N Y_N - E_N e^{\phi_N}\right) \\
&\times\left(1 + k_2\sum_{i=1}^N(\phi_i - \hat{\mu}_i)^2 Y_i\right)^{(\nu+N)/2} \\
\leq&\exp\left(\sum_{i=1}^{N-1}(\phi_i Y_i) - \sum_{i=1}^{N-1} E_i e^{\phi_i}\right)\exp\left(\phi_N Y_N - E_N e^{\phi_N}\right) \\
&\times\left(1 + k_2\sum_{i=1}^{N-1}(\phi_i - \hat{\mu}_i)^2 Y_i\right)^{(\nu+N)/2}\left(1 + k_2(\phi_N - \hat{\mu}_N)^2 Y_N\right)^{(\nu+N)/2}
\end{aligned} \tag{B.11}$$

since we know that $(1+x+y)^p \leq (1+x)^p(1+y)^p$ if $x, y \geq 0$ and $p > 1$. The

R.H.S. of (B.9) is therefore

$$\leq \exp\left(\sum_{i=1}^{N-1}(\phi_i Y_i) - \sum_{i=1}^{N-1} E_i e^{\phi_i}\right)\left(1 + k_2 \sum_{i=1}^{N-1}(\phi_i - \hat{\mu}_i)^2 Y_i\right)^{(\nu+N)/2} \times B_1. \tag{B.12}$$

We can continue in the same manner to show that (B.9) is $\leq B_1^N$. We have thus proved that (B.9) is a bounded function, thereby proving (ii), as required. This completes the proof that we have an envelope for Model 1.

B.2.2 Model 2

We now prove that the envelope $R_1(\tau_h, \tau_c)R_2(\boldsymbol{\Theta}; \tau_h, \tau_c)$ is an envelope for $P(\boldsymbol{\Theta}, \tau_h, \tau_c)$ by showing that there exists $B > 0$, such that

$$f(\boldsymbol{\Theta}, \tau_h, \tau_c) = \frac{P(\boldsymbol{\Theta}, \tau_h, \tau_c)}{R_1(\tau_h, \tau_c)R_2(\boldsymbol{\Theta}; \tau_h, \tau_c)} \leq B, \quad \text{for } \boldsymbol{\Theta} \in R^{2N}, \tau_h, \tau_c > 0 \tag{B.13}$$

Using the same notation for Model 2 as in Appendix A.1,

$$\begin{aligned} f(\boldsymbol{\Theta}, \tau_h, \tau_c) \propto & \exp\left(\sum_{i=1}^{N}((\theta_i + \phi_i)Y_i - E_i e^{\theta_i + \phi_i})\right) \exp\left(-\frac{1}{2}\boldsymbol{\theta}^T(\tau_h I)\boldsymbol{\theta}\right) \\ & \times \tau_h^{N/2+\alpha_h-1} \tau_c^{M/2+\alpha_c-1} \exp\left(-\frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}\right) \exp\left(-\frac{\tau_h}{\beta_h} - \frac{\tau_c}{\beta_c}\right) \\ & \times \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_c) - \mu_c}{\sigma_c}\right)^2\right]^{(\nu+1)/2} \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_h) - \mu_h}{\sigma_h}\right)^2\right]^{(\nu+1)/2} \\ & \times \det(C)^{-0.5}\left(1 + k_3(\boldsymbol{\Theta} - \mu_N)^T C(\boldsymbol{\Theta} - \mu_N)\right)^{(\nu+N)/2}, \end{aligned} \tag{B.14}$$

with $k_3 = \frac{\nu-2}{\nu^2}$. Notice that $V^{-1} - (V + \tau_h V^2)^{-1}$ is positive definite (easily verified) and $\tau_c Q$ is non-negative definite, and $\det(A+B) \geq \det(A)$ for A, B

non-negative definite. So we have,

$$\begin{aligned} &\det(C) = \det(V^{-1} + \tau_h I)\det(V^{-1} + \tau_c Q - (V + \tau_h V^2)^{-1}) \\ &\Rightarrow \det(C)^{-1/2} \leq \tau_h^{-N/2} \Pi_{i=1}^{N} Y_i^{-1/2}. \end{aligned} \tag{B.15}$$

Now, $(\boldsymbol{\Theta} - \mu_N)^T C(\boldsymbol{\Theta} - \mu_N) = \boldsymbol{\Theta}^T C \boldsymbol{\Theta} + \mu_N^T C \mu_N - 2\boldsymbol{\Theta}^T C \mu_N$.
We know

$$\boldsymbol{\Theta}^T C \boldsymbol{\Theta} = (\boldsymbol{\theta} + \boldsymbol{\phi})^T V^{-1}(\boldsymbol{\theta} + \boldsymbol{\phi}) + \tau_h \boldsymbol{\theta}^T \boldsymbol{\theta} + \tau_c \boldsymbol{\phi}^T Q \boldsymbol{\phi}, \tag{B.16}$$

and,

$$\begin{aligned} \mu_N^T C \mu_N =& (X^T \Gamma^{-1} \hat{\boldsymbol{\mu}})^T (X^T \Gamma^{-1} X)^{-1} (X^T \Gamma^{-1} X)(X^T \Gamma^{-1} X)^{-1} (X^T \Gamma^{-1} \hat{\boldsymbol{\mu}}) \\ =& \hat{\boldsymbol{\mu}}^T (\Gamma^{-1} X (X^T \Gamma^{-1} X)^{-1} X^T \Gamma^{-1}) \hat{\boldsymbol{\mu}} \end{aligned}$$

Let $B = \Gamma^{-1} X (X^T \Gamma^{-1} X)^{-1} X^T \Gamma^{-1}$, $X^* = \Gamma^{-\frac{1}{2}} X$ and the S.V. decomposition of $X^* = UDV^T$, where $U^T U = I, V^T V = I$ (since X^* is full rank). So,

$$\begin{aligned} B =& \Gamma^{-\frac{1}{2}} X^* (X^{*T} X^*)^{-1} (\Gamma^{-\frac{1}{2}} X^*)^T \\ =& \Gamma^{-\frac{1}{2}} UDV^T (UDV^{T^T} UDV^T)^{-1} (\Gamma^{-\frac{1}{2}} UDV^T)^T \\ =& \Gamma^{-\frac{1}{2}} UDD^{-2} DU^T \Gamma^{-\frac{1}{2}} \\ =& \Gamma^{-\frac{1}{2}} I \Gamma^{-\frac{1}{2}} = \Gamma^{-1}. \end{aligned}$$

So,

$$\mu_N^T C \mu_N = \hat{\boldsymbol{\mu}}^T \Gamma^{-1} \hat{\boldsymbol{\mu}} = \hat{\boldsymbol{\mu}}^T V^{-1} \hat{\boldsymbol{\mu}} + \tau_h \hat{\boldsymbol{\mu}}^T \hat{\boldsymbol{\mu}}, \tag{B.17}$$

which can be seen by writing Γ^{-1} in its block matrix form. The last piece,

$$-2\boldsymbol{\Theta}^T C \mu_N = -2\boldsymbol{\Theta}^T X^T \Gamma^{-1} \hat{\boldsymbol{\mu}} = -2(\boldsymbol{\theta}^T (V^{-1} - \tau_h I) \hat{\boldsymbol{\mu}}) - 2(\boldsymbol{\phi}^T V^{-1} \hat{\boldsymbol{\mu}}). \tag{B.18}$$

From (B.16), (B.17) and (B.18), we get $\boldsymbol{\Theta}^T C\boldsymbol{\Theta} + \mu_N^T C\mu_N - 2\boldsymbol{\Theta}^T C\mu_N$

$$
\begin{aligned}
&=(\boldsymbol{\theta}+\boldsymbol{\phi})^T V^{-1}(\boldsymbol{\theta}+\boldsymbol{\phi}) + \tau_h\boldsymbol{\theta}^T\boldsymbol{\theta} + \tau_c\boldsymbol{\phi}^T Q\boldsymbol{\phi} + \hat{\boldsymbol{\mu}}^T V^{-1}\hat{\boldsymbol{\mu}} + \tau_h\hat{\boldsymbol{\mu}}^T\hat{\boldsymbol{\mu}} \\
&+ -2(\boldsymbol{\theta}^T(V^{-1}-\tau_h I)\hat{\boldsymbol{\mu}}) - 2(\boldsymbol{\phi}^T V^{-1}\hat{\boldsymbol{\mu}}) \\
&= \sum_{i=1}^N \mu_i Y_i + \tau_h\sum_{i=1}^N(\theta_i+\hat{\mu}_i)^2 + \tau_c\boldsymbol{\phi}^T Q\boldsymbol{\phi} - 2(\boldsymbol{\theta}+\boldsymbol{\phi})^T V^{-1}\hat{\boldsymbol{\mu}} + \hat{\boldsymbol{\mu}}^T V^{-1}\hat{\boldsymbol{\mu}}. \\
&= \sum_{i=1}^N \mu_i Y_i + \tau_h\sum_{i=1}^N(\theta_i+\hat{\mu}_i)^2 + \tau_c\boldsymbol{\phi}^T Q\boldsymbol{\phi} - 2\sum_{i=1}^N \mu_i(Y_i\hat{\mu}_i) + \sum_{i=1}^N \hat{\mu}_i^2 Y_i \\
&= \sum_{i=1}^N \mu_i Y_i(1-2\hat{\mu}_i) + \tau_h\sum_{i=1}^N(\theta_i+\hat{\mu}_i)^2 + \tau_c\boldsymbol{\phi}^T Q\boldsymbol{\phi} + \sum_{i=1}^N \hat{\mu}_i^2 Y_i.
\end{aligned}
$$

The above simplification shows that $(\boldsymbol{\Theta}-\mu_N)^T C(\boldsymbol{\Theta}-\mu_N)$

$$
= \sum_{i=1}^N \mu_i Y_i(1-2\hat{\mu}_i) + \tau_h\sum_{i=1}^N(\theta_i+\hat{\mu}_i)^2 + \tau_c\boldsymbol{\phi}^T Q\boldsymbol{\phi} + \sum_{i=1}^N \hat{\mu}_i^2 Y_i \tag{B.19}
$$

where $\mu_i = \theta_i + \phi_i$, and the constant, $\hat{\mu}_i = \log(Y_i/E_i)$. We use the fact that $|\sum_{i=1}^N a_i b_i|^p \leq N^{p-1}\sum_{i=1}^N |a_i|^p|b_i|^p$ when $p>0, a_i, b_i \in R$, along with (B.15) and (B.19), to show that (upto a constant),

$$
\begin{aligned}
f(.) \leq & \exp\left(\sum_{i=1}^N(\mu_i Y_i - E_i e^{\mu_i})\right) \\
& \times\tau_h^{\alpha_h}\exp\left(-\tau_h(\beta_h^{-1} + \sum_{i=1}^N \theta_i^2/2)\right)\left[1+\frac{1}{\nu}\left(\frac{\log(\tau_h)-\mu_h}{\sigma_h}\right)^2\right]^{(\nu+1)/2} \\
& \times\tau_c^{M/2+\alpha_c}\exp(-\tau_c(\beta_c^{-1}+\boldsymbol{\phi}^T Q\boldsymbol{\phi}))\left[1+\frac{1}{\nu}\left(\frac{\log(\tau_c)-\mu_c}{\sigma_c}\right)^2\right]^{(\nu+1)/2} \\
& \times\left(k_4^{(\nu+N)/2} + \left|k_3\sum_{i=1}^N \mu_i Y_i(1-2\hat{\mu}_i)\right|^{(\nu+N)/2} + \right. \\
& \left. + \tau_h^{(\nu+N)/2}\left(k_3\sum_{i=1}^N(\theta_i+\hat{\mu}_i)^2\right)^{(\nu+N)/2} + \tau_c^{(\nu+N)/2}(k_3\boldsymbol{\phi}^T Q\boldsymbol{\phi})^{(\nu+N)/2}\right),
\end{aligned}
$$

where $k_4 = 1 + k_3 \sum_{i=1}^{N} \hat{\mu}_i^2 Y_i$.

Thus, $f(\cdot) \leq f_1 f_2 f_3 (f_{4a} + f_{4b} + f_{4c} + f_{4d})$, where

$$f_1 = \exp\left(\sum_{i=1}^{N} (\mu_i Y_i - E_i e^{\mu_i})\right)$$

$$f_2 = \tau_h^{\alpha_h} \exp\left(-\tau_h(\beta_h^{-1} + \sum_{i=1}^{N} \theta_i^2/2)\right) \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_h) - \mu_h}{\sigma_h}\right)^2\right]^{(\nu+1)/2}$$

$$f_3 = \tau_c^{M/2+\alpha_c} \exp(-\tau_c(\beta_c^{-1} + \boldsymbol{\phi}^T Q \boldsymbol{\phi})) \left[1 + \frac{1}{\nu}\left(\frac{\log(\tau_c) - \mu_c}{\sigma_c}\right)^2\right]^{(\nu+1)/2}$$

$$f_{4a} = k_4^{(\nu+N)/2}$$

$$f_{4b} = \left| k_3 \sum_{i=1}^{N} \mu_i Y_i (1 - 2\hat{\mu}_i) \right|^{(\nu+n)/2}$$

$$f_{4c} = \tau_h^{(\nu+N)/2} \left(k_3 \sum_{i=1}^{N} (\theta_i + \hat{\mu}_i)^2 \right)^{(\nu+N)/2}$$

$$f_{4d} = \tau_c^{(\nu+N)/2} (k_3 \boldsymbol{\phi}^T Q \boldsymbol{\phi})^{(\nu+N)/2}.$$

We can show that f_1 is bounded in the same way as we did in our proof for Model 1. f_2 is just the ratio of a gamma density in τ_h, divided by a log-t density in τ_h. It is easy to see that this ratio is bounded for all $\tau_h > 0$ since the exponent of τ_h is $\alpha_h > 0$. Similarly, f_3 is bounded for all $\tau_c > 0$ since the exponent of τ_c is $M/2 + \alpha_c > 0$. Thus,

$$f_1 f_2 f_3 \leq K_0, \text{ for some } K_0 < \infty. \tag{B.20}$$

Clearly, f_{4a} is a constant, and

$$f_1 f_{4b} \propto \exp\left(\sum_{i=1}^{N} (\mu_i Y_i - E_i e^{\mu_i})\right) \left| \sum_{i=1}^{N} \mu_i Y_i (1 - 2\hat{\mu}_i) \right|^{(\nu+N)/2}$$

$$\leq \exp\left(\sum_{i=1}^{N} (\mu_i Y_i - E_i e^{\mu_i})\right) N^{(\nu+N)/2-1} \sum_{i=1}^{N} |Y_i(1 - 2\hat{\mu}_i)|^{(\nu+N)/2} |\mu_i|^{(\nu+N)/2}$$

where we again use the fact that $|\sum_{i=1}^{N} b_i\mu_i|^p \leq N^{p-1}\sum_{i=1}^{N}|b_i|^p|\mu_i|^p$. To prove that the R.H.S. above is bounded, it suffices to show that for each j, $|\mu_j|^{(\nu+N)/2}\exp\left(\sum_{i=1}^{N}(\mu_i Y_i - E_i e^{\mu_i})\right)$, is bounded, which is obvious. So,

$$f_1 f_{4b} \leq K_1, \text{ for some } K_1 < \infty. \tag{B.21}$$

Also,

$$f_2 f_{4c} = \tau_h^{\alpha_h+(\nu+N)/2}\exp(-\tau_h(\beta_h^{-1}+\sum_{i=1}^{N}\theta_i^2))\left(k_3\sum_{i=1}^{N}(\theta_i+\hat{\mu}_i)^2\right)^{(\nu+N)/2}$$

We simply observe that if $\tau_h \to \infty$, then for any value of $\boldsymbol{\theta}$ ($\theta_i \to \infty$ or $-\infty$), the product goes to zero. Similary, for $\theta_i \to \infty$ or $-\infty$, the product goes to zero. For $\tau_h \to 0$ and $\boldsymbol{\theta} \to \infty$ or $-\infty$, if τ_h dominates, then the expression reduces to $0.\exp(0)$, and if τ_h is dominated by $\boldsymbol{\theta}$, we have an expression of the form $\exp(-\infty).\infty = 0$. Hence,

$$f_2 f_{4c} \leq K_2, \text{ for some } K_2 < \infty. \tag{B.22}$$

Now,

$$\begin{aligned} f_3 f_{4d} = {} & \tau_c^{M/2+\alpha_c+(\nu+N)/2}(k_3\boldsymbol{\phi}^T Q\boldsymbol{\phi})^{(\nu+N)/2} \\ & \times \exp(-\tau_c(\beta_c^{-1}+\boldsymbol{\phi}^T Q\boldsymbol{\phi}))\left[1+\frac{1}{\nu}\left(\frac{\log(\tau_c)-\mu_c}{\sigma_c}\right)^2\right]^{(\nu+1)/2}. \end{aligned}$$

If $\boldsymbol{\phi}^T Q\boldsymbol{\phi} \to 0$, all that remains is the ratio of a gamma to a log-distribution, which is bounded. If $\boldsymbol{\phi}^T Q\boldsymbol{\phi} \to \infty$, $\exp(-\tau_c(\beta_c^{-1}+\boldsymbol{\phi}^T Q\boldsymbol{\phi}))$ dominates all other terms, and $f_3 f_{4d} \to 0$. Similarly, it is easy to see that $f_3 f_{4d}$ is bounded whenever $\tau_c \to \infty$ or 0, so by continuity,

$$f_3 f_{4d} \leq K_3, \text{ for some } K_3 < \infty. \tag{B.23}$$

From (B.20), (B.21), (B.22) and (B.23), it follows that

$$f(\boldsymbol{\Theta}, \tau_h, \tau_c) \leq f_1 f_2 f_3 (f_{4a} + f_{4b} + f_{4c} + f_{4d}) \leq B \quad \text{for some } B > 0. \quad \text{(B.24)}$$

We have thus proved that our envelope for Model 2 is truly an envelope.

B.2.3 Model 2 with Covariates

The full posterior distribution for Model 2 with covariates is

$$\begin{aligned} P^*(\boldsymbol{\Theta}, (\tau_h, \tau_c), \boldsymbol{\beta}) \propto & \exp\left(\sum_{i=1}^{N}((X_i^T\boldsymbol{\beta} + \theta_i + \phi_i)Y_i - E_i e^{X_i^T\boldsymbol{\beta}+\theta_i+\phi_i})\right) \\ & \times \exp\left(-\frac{1}{2}\boldsymbol{\theta}^T(\tau_h I)\boldsymbol{\theta}\right) h(\boldsymbol{\beta}) \\ & \times \tau_h^{N/2+\alpha_h-1}\tau_c^{M/2+\alpha_c-1} \exp\left(-\frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi} - \frac{\tau_h}{\beta_h} - \frac{\tau_c}{\beta_c}\right) \\ & = P(\boldsymbol{\Theta}, (\tau_h, \tau_c))h(\boldsymbol{\beta}), \end{aligned}$$

where $P(\boldsymbol{\Theta}, (\tau_h, \tau_c))$ is the posterior distribution for Model 2 without covariates. The approximate joint posterior distribution, letting $\mathbf{X} = \{X_1, \ldots, X_k\}$, is:

$$\begin{aligned} \hat{P}^*(\boldsymbol{\Theta}, (\tau_h, \tau_c), \boldsymbol{\beta}) \propto & \exp\left(-\frac{1}{2}(\hat{\boldsymbol{\mu}} - \mathbf{X}^T\boldsymbol{\beta} - (\boldsymbol{\theta} + \boldsymbol{\phi}))^T V^{-1}(\hat{\boldsymbol{\mu}} - \mathbf{X}^T\boldsymbol{\beta} - (\boldsymbol{\theta} + \boldsymbol{\phi}))\right) \\ & \times \exp\left(-\frac{1}{2}\boldsymbol{\theta}^T(\tau_h I)\boldsymbol{\theta} - \frac{1}{2}\boldsymbol{\phi}^T(\tau_c Q)\boldsymbol{\phi}\right) h(\boldsymbol{\beta}) \\ & \times \tau_h^{N/2+\alpha_h-1}\tau_c^{M/2+\alpha_c-1} \exp\left(-\tau_h/\beta_h - \tau_c/\beta_c\right) \end{aligned}$$

To obtain the marginal posterior distribution $S_1(\tau_h, \tau_c, \beta)$ upto a constant, we integrate out $\boldsymbol{\Theta}$ from the above function.

$$\begin{aligned} S_1^*(\tau_h, \tau_c, \boldsymbol{\beta}) =& \tau_h^{N/2+\alpha_h-1} \tau_c^{M/2+\alpha_c-1} \exp\left(-\tau_h/\beta_h - \tau_c/\beta_c\right). \\ & \times (\det(\tau_h V^{-1} + \tau_c V^{-1} Q + \tau_h \tau_c Q))^{-1/2} \\ & \times \exp\left\{-\frac{1}{2}\left(\hat{\boldsymbol{\Theta}}^T C \hat{\boldsymbol{\Theta}} + D\hat{\boldsymbol{\Theta}} + k\right)\right\} \\ & \times (\hat{\boldsymbol{\mu}} - \mathbf{X}^T\boldsymbol{\beta})^T V^{-1} (\hat{\boldsymbol{\mu}} - \mathbf{X}^T\boldsymbol{\beta})\ h(\boldsymbol{\beta}). \end{aligned}$$

with

$$C_{2N\times 2N} = \begin{bmatrix} \overbrace{V^{-1} + \tau_h I}^{\boldsymbol{\theta}} & \overbrace{+V^{-1}}^{\boldsymbol{\phi}} \\ +V^{-1} & V^{-1} + \tau_c Q \end{bmatrix},$$

and

$$D_{1\times 2N} = (-2(\hat{\boldsymbol{\mu}} - \mathbf{X}^T\boldsymbol{\beta})^T V^{-1}, -2(\hat{\boldsymbol{\mu}} - \mathbf{X}^T\boldsymbol{\beta})^T V^{-1})$$

$$\hat{\boldsymbol{\Theta}}^T = (\hat{\boldsymbol{\theta}}^T, \hat{\boldsymbol{\phi}}^T),$$

The conditional distribution of the model parameters is of the same form as in (4.9), with the only differences arising from the differences in the $\hat{\mu}^*$ vector (which now contains $\boldsymbol{\beta}$ terms).

$$S_2^*(\boldsymbol{\theta}, \boldsymbol{\phi} | \tau_h, \tau_c, \boldsymbol{\beta}) \sim N((X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}\hat{\boldsymbol{\mu}}^*)\ ,\ (X^T\Gamma^{-1}X)^{-1})\ .$$

Note that $S_1^*(\tau_h, \tau_c, \boldsymbol{\beta})$ can be decomposed as

$$S_1^*(\tau_h, \tau_c, \boldsymbol{\beta}) = S_1(\tau_h, \tau_c)\ S_{1\beta}(\boldsymbol{\beta}),$$

where $S_1(\tau_h, \tau_c)$ is identical to the function for the approximate marginal posterior distribution of the precision parameters for Model 2 with no covariates. Suppose we were to find a probability density function that is easy

to simulate from, $R_{1\beta}(\boldsymbol{\beta})$, such that $S_{1\beta}(\boldsymbol{\beta})/R_{1\beta}(\boldsymbol{\beta}) \leq K_\beta$ for all $\boldsymbol{\beta}$, for some $K_\beta < \infty$. Also, let $R_1(\tau_h, \tau_c)$ be as before (for Model 2 with no covariates). Let the multivariate-t version of S_2^*, be $R_2^*(\boldsymbol{\theta}, \boldsymbol{\phi}|\tau_h, \tau_c, \boldsymbol{\beta})$, much in the same way as described for the other models, and let

$$R^* = R_2^*(\boldsymbol{\theta}, \boldsymbol{\phi}|\tau_h, \tau_c, \boldsymbol{\beta}) R_{1\beta}(\boldsymbol{\beta}) R_1(\tau_h, \tau_c),$$

Therefore, following our proof from the previous subsection, we note that the only term that changes is the term we denote by f_1. We can then arrive at a similar result as in (B.21), and therefore have

$$\begin{aligned} P^*/R^* =& P(\boldsymbol{\theta}, \boldsymbol{\phi}) h(\boldsymbol{\beta}) / (R_2^*(\boldsymbol{\theta}, \boldsymbol{\phi}|\tau_h, \tau_c, \boldsymbol{\beta}) R_{1\beta}(\boldsymbol{\beta}) R_1(\tau_h, \tau_c)) \\ =& \frac{P(\boldsymbol{\theta}, \boldsymbol{\phi})}{R_2^*(\boldsymbol{\theta}, \boldsymbol{\phi}|\tau_h, \tau_c, \boldsymbol{\beta}) R_1(\tau_h, \tau_c)} \frac{h(\boldsymbol{\beta})}{R_{1\beta}(\boldsymbol{\beta})} \\ \leq& \frac{P(\boldsymbol{\theta}, \boldsymbol{\phi})}{R_2^*(\boldsymbol{\theta}, \boldsymbol{\phi}|\tau_h, \tau_c, \boldsymbol{\beta}) R_1(\tau_h, \tau_c)} K_\beta \leq K < \infty, \end{aligned}$$

thus giving us an envelope for the posterior distribution of Model 2 with covariates.

B.3 Perfect Tempering Algorithm Details

We give details of the second variant of our perfect tempering scheme as described in Section 3.3.2. This scheme has two tempering distributions, so $N^* = 2$. A single update of the simulated tempering chain is done by proposing a move 'up' from level n to $n+1$ with probability p and proposing a move 'down' from level n to $n-1$ with probability q, and proposing the corresponding value for a particular tempering level using the proposal distribution $f_{n,n'}(x, x')$. A move 'up' one level when $n = N^*$ and a move 'down'

one level when $n = 0$ are interpreted as staying at level n. Thus, a move from level n to n' is proposed with probability $l(n, n')$

$$l(n, n') = \begin{cases} 1-p & \text{for } n = 0, n' = 0 \\ p & \text{for } n = 0, n' = 1 \\ q & \text{for } n = 1, n' = 0 \\ 1-q & \text{for } n = 1, n' = 1 \end{cases}$$

with $p \in (0, 1)$, $q \in (0, 1-p]$. our algorithms.

$$f_{n,n'}(x, x') = \begin{cases} h_0(x') & \text{if } n' = 0 \\ h_1(x') & \text{if } n' = 1 \end{cases}$$

where h_0 is the distribution with point mass on the atom $\mathcal{A}$ and h_1 is the enveloping distribution ($R(\boldsymbol{\phi}, \tau_c)$ for Model 1, $R(\boldsymbol{\Theta}, \tau_h, \tau_c)$ for Model 2). Let K be the bound used for a rejection sampler, i.e., $K > 0$ is the smallest number s.t. $P(\cdot)/KQ(.) \leq 1$ everywhere. Since we set $\pi_0/\pi_1 = K$, the Metropolis-Hastings acceptance ratios of the proposed value are then:

$$\begin{aligned}
\alpha((x,0),(x',0)) &= \frac{P(x')\pi_0}{P(x)\pi_0}\frac{R(x)(1-p)}{R(x')(1-p)} = 1 \\
\alpha((x,0),(x',1)) &= \frac{P(x')\pi_1}{P(x)\pi_0}\frac{R(x)q}{R(x')p} = \frac{1}{K}\frac{P(x')}{R(x')}\frac{q}{p} \\
\alpha((x,1),(x',1)) &= \min\left(\frac{P(x')\pi_1}{P(x)\pi_1}\frac{R(x)(1-q)}{R(x')(1-q)}, 1\right) = \min\left(\frac{P(x')}{P(x)}\frac{R(x)}{R(x')}, 1\right) \\
\alpha((x,1),(x',0)) &= \min\left(\frac{P(x')\pi_0}{P(x)\pi_1}\frac{R(x)p}{R(x')q}, 1\right) = \min\left(K\frac{R(x)}{P(x)}\frac{p}{q}, 1\right)
\end{aligned} \tag{B.25}$$

Increasing the ratio of p to q has the effect of increasing the probability of a move from temperature 0 to 1, and decreasing the probability of a move from temperature 1 to 0. This is therefore desirable from the point of view

of having the simulated tempering chain stay in the state where the samples are from the distribution of interest. Increasing the ratio of p to q also has the effect of reducing the amount of time the chain spends in temperature 0, thereby reducing the chance of the dominating random walk chain hitting 0. This increases the time taken for coalescence of the chains (τ^*'s), which also has the effect of reducing the efficiency of the algorithm. Therefore, there is a trade-off involved in increasing this ratio, from the point of view of the efficiency of the overall perfect simulation algorithm.

Now, the Metropolis-Hastings ratios for the proposed values for the random-walk chain are

$$\begin{aligned} \tilde{\alpha}(n, n+1) &= \min\left(1, K_n \frac{\pi_{n+1}}{\pi_n}\right) \\ \tilde{\alpha}(n, n-1) &= \min\left(1, \frac{1}{K_{n-1}} \frac{\pi_{n-1}}{\pi_n}\right) \\ \tilde{\alpha}(n, n) &= 1 \end{aligned} \tag{B.26}$$

which, in our two temperature chain, translates simply to

$$\begin{aligned} &\tilde{\alpha}(0,0) = \tilde{\alpha}(1,1) = 1 \\ &\tilde{\alpha}(0,1) = \min\left(1, K \frac{\pi_1}{\pi_0} \frac{q}{p}\right) = \min\left(1, \frac{q}{p}\right) \\ &\tilde{\alpha}(1,0) = \min\left(1, \frac{1}{K} \frac{\pi_0}{\pi_1} \frac{p}{q}\right) = \min\left(1, \frac{p}{q}\right). \end{aligned} \tag{B.27}$$

It is easy to see that the $\{D_t\}$ process produced by RWupdate, using the above Metropolis-Hastings acceptance probabilities, dominates the $\{Z_t\}$ process in the sense of (3.5), and hence satisfies the requisite condition (3.4). We now describe each of the modules of the perfect tempering algorithm.

STupdate

A single update of the simulated tempering chain:

z =STupdate$((x, n); u^1, u^2)$

$z \leftarrow (x, n)$

if $(u^1 < p)$

    $n' \leftarrow n + 1$

else if $(u^1 > 1 - q)$

    $n' \leftarrow n - 1$

else

    $n' \leftarrow n.$

if $(0 \leq n' \leq N^*)$

    draw $x^{'} \sim f_{n,n'}(x, x')$

    if $u^2 \leq \alpha((x, n), (x^{'}, n'))$

        $z \leftarrow (x^{'}, n')$

return z.

Note that when evaluating $\alpha((x, 1), (x', 1))$, we can reuse the $P(x), Q(x)$ from the previous iteration of the algorithm, thereby saving computation if the evaluations of $P(x)$ and $Q(x)$ are expensive.

RWupdate

A single update of the random walk $\{D_t\}$:

$d =$RWupdate$(m; u^1, u^2)$

$d \leftarrow m$

if $(u^1 < p)$

    $m' \leftarrow m + 1$

else if $(u^1 > 1 - q)$

    $m' \leftarrow m - 1$

else

    $m' \leftarrow m$

if $(0 \leq m' \leq N^*)$ and $(u^2 < \tilde{\alpha}(m, m'))$

    $d \leftarrow m'$

return d.

PWperfect

Return a single value from the stationary distribution:

$t <- -T$

$D \leftarrow N^*$

$D^{'} \leftarrow N^*$

repeat

$D \leftarrow \text{RWupdate}(D; u_t^1, u_t^2)$

$t \leftarrow t + 1$

if $D = D^{'}$ or $(t = 0$ and $D \neq 0)$

$D^{'} \leftarrow \text{RWupdate}(D^{'}; u_t^1, u_t^2)$

until D=0 $\tau^* \leftarrow -t$

$Z^{equi} \leftarrow (\mathbf{0}, 0)$

for $t = -\tau^*$ to -1

$Z^{equi} \leftarrow \text{STupdate}(Z^{equi}; u_t^1, u_t^2)$

return(Z^{equi}, τ^*).

B.4 Perfect Forward Tempering

For the two-temperature simulated tempering algorithm, it is easy to compute the lower bound on the probability of moving from any point in the state space to the level at which the Markov chain regenerates (level 0). From before,

$$\begin{aligned}
&\alpha((x,0),(x',0)) = 1 \\
&\alpha((x,1),(x',0)) = \min\left(K\frac{R(x)}{P(x)}\frac{p}{q},1\right) \geq \min\left(\frac{p}{q},1\right) \\
\Rightarrow\ &\alpha((x,t),(x',0)) \geq \min\left(\frac{p}{q},1\right)\ \ \forall t.
\end{aligned}$$

If $p/q \geq 1$, then $\alpha((x,1),(x',0)) = 1$, all proposed moves from level 1 to level 0 would be accepted, and hence the chain would not spend enough time in H_1. However, $q/p \leq 1$ implies $\alpha((x,1),(x',0))$ is at best equal to the rejection sampling acceptance ratio; as q/p gets smaller, the probability of the chain moving from level 0 to 1 decreases. Thus, if $p/q \geq 1$ the chain will not spend much time at level 1, therefore reducing the number of useful samples.

Since we set $p/q < 1$, the probability of moving from any level to level 0 is bounded below by p/q. We can therefore set $\epsilon_1 = p/q$ (where ϵ_1 is as in Subsection 4.5.2). Since ϵ_2 can be set to 1, the minorization parameter, $\epsilon = \epsilon_1\epsilon_2 = p/q$, hence from Brooks et al. (2003) we can produce an i.i.d. draw from the stationary distribution by simply running the appropriate residual chain forward for a random length of $T \sim \text{Geo}(\epsilon)$ iterations.

While the approach adopted in Section 3.3.2 involves the use of two uniform random variates (u^1, u^2) at every step of the algorithm, the approach in Brooks et al. (2003) implicitly uses three uniform random variates at each

step, v^1, v^2, and v^3, say. Recall that the random variate u^1 determines the proposed value for the tempering level of the next update of the simulated tempering chain, and hence produces an identical proposal for the next update of the coupled random walk chain. The proposal for the parameters is obtained independently of u^1 and u^2. u^2 determines whether the joint proposal (new temperature and corresponding new parameters) is accepted using the Metropolis-Hastings ratio. Under this construction, the simulated tempering chain has a two-step minorization condition (see the Proposition in Section 4.5.2) since there is a lower bound for the probability of moving from *any* state to the hot distribution (step 1), and the chain regenerates when it reaches the hot distribution (step 2). The algorithm described in Brooks et al. (2003) uses v^1 to generate the proposal for the tempering level, and hence the proposal for the corresponding random walk. However, this random variate is not explicitly described in their algorithm since the distribution of the time that the random walk chain takes to reach the "hot" tempering level (level 0) is determined ahead of time to be a geometric random variable with success probability ϵ. The random variate v^2 determines whether the proposed temperature is accepted via the Metropolis-Hastings ratio. v^3 similarly controls the acceptance of the proposed parameter update, where the proposal for the parameters is drawn independently of v^1, v^2 and v^3. Since v^1, v^2, and v^3 are used at each step of the algorithm, a single update of the Brooks et al. (2003) algorithm corresponds to two updates of the simulated tempering chain. The chain then has a one-step minorization condition, so the residual chain is easily constructed.

Let α^* be the lower bound on the probability of a transition from any

state to a state at temperature 0. From before, $\alpha^* = \min(p/q, 1)$. Suppose the current state of the chain is (x_t, n_t). Following Brooks et al. (2003), we can describe a single update of the residual chain (as used in Section 4.5).

Simulated tempering residual chain

1. Draw n' from $l(n, n')$ using $u^1 \sim \text{Unif}(0, 1)$.

2. If $n' = 0$ draw $u^2 \sim \text{Unif}(\alpha^*, 1)$, else draw $u^2 \sim \text{Unif}(0, 1)$.

3. Draw $x' \sim f_{n,n'}(x, x')$. If $u^2 \leq \alpha\left((x_t, n_t), (x', n')\right)$ set $(x_{t+1}, n_{t+1}) = (x', n')$, else set $(x_{t+1}, n_{t+1}) = (x_t, n_t)$.

4. Draw $x'' \sim f_{n,n'}(x', x'')$ and $u^3 \sim \text{Unif}(0, 1)$.
   If $u^3 \leq \alpha\left((x_{t+1}, n_{t+1}), (x'', n_{t+1})\right)$ set $(x_{t+1}, n_{t+1}) = (x'', n_{t+1})$, else (x_{t+1}, n_{t+1}) stays the same as before.

If at time t, $n' = 0$ and $u^2 \leq \alpha^*$, all coupled simulated tempering chains will coalesce at time $t + 1$. On the other hand if every time $n' = 0$, we have $u^2 > \alpha^*$, it is no longer true that *all* chains will move to the hot distribution if the proposal $n' = 0$ is accepted, so the chain will not regenerate at $t + 1$. The residual chain is therefore constructed in two steps. The first step is a regular simulated tempering joint update of the tempering level and parameter vector (x, n). The only difference is that if the proposal n' is 0, the uniform random variate, u^2 is drawn from $\text{Unif}(\alpha^*, 1)$ instead of from $\text{Unif}(0, 1)$. Since $u^2 > \alpha^*$, it is no longer Thus, even if the chain moves to the hot distribution, it is no longer regenerating at that time. The second step is then a Metropolis-Hastings update of the parameter vector alone, keeping the tempering level constant from the first step.

Appendix C

Exact Simulation for Linear Hierarchical Models

C.1 Marginal and Conditional Distributions

Let the data vector, $\mathbf{Y} = (Y_{11}, \ldots, Y_{1m}, Y_{21}, \ldots, Y_{2m}, \ldots, Y_{K1}, \ldots, Y_{Km})^T$, and let $W_{Km\times K} = (W_1, \ldots, W_K)^T$ where W_i is an $m \times K$ matrix with ones in its ith column, and zeros everywhere else (so $W^T W = mI_{K\times K}$). The full posterior distribution of $(\boldsymbol{\Theta}, \Lambda)$ is then

$$\begin{aligned} P(\boldsymbol{\Theta}, \Lambda) \propto & \det(\lambda_e I_{Km\times Km})^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\mathbf{Y} - W\boldsymbol{\theta})^T(\lambda_e I_{Km\times Km})(\mathbf{Y} - W\boldsymbol{\theta})\right) \\ & \times \det(\lambda_\theta I)^{\frac{1}{2}} \exp\left(-\frac{1}{2}(\boldsymbol{\theta} - \mu\mathbf{1})^T(\lambda_\theta I)(\boldsymbol{\theta} - \mu\mathbf{1})\right) \\ & \times \lambda_0^{\frac{1}{2}} \exp\left(-\frac{1}{2}\lambda_0(\mu - \mu_0)^2\right) \lambda_e^{a_2-1} \exp(-\lambda_e/b_2) \lambda_\theta^{a_1-1} \exp(-\lambda_\theta/b_1) \end{aligned}$$

$$
\begin{aligned}
&\propto \exp\left(-\frac{\lambda_e}{2}\sum_{i,j}(\mathbf{Y}-W\boldsymbol{\theta})^2\right)\\
&\times \exp\left(-\frac{1}{2}(\boldsymbol{\theta}-\mu\mathbf{1})^T(\lambda_\theta I)(\boldsymbol{\theta}-\mu\mathbf{1})\right)\exp\left(-\frac{1}{2}\lambda_0(\mu-\mu_0)^2\right)\\
&\times\lambda_e^{Km/2+a_2-1}\exp(-\lambda_e/b_2)\lambda_\theta^{K/2+a_1-1}\exp(-\lambda_\theta/b_1).\\
&\propto \exp\left(-\frac{\lambda_e}{2}\sum_{i,j}(Y_{ij}-\theta_i)^2-\frac{\lambda_\theta}{2}\sum_i(\theta_i-\mu)^2-\frac{\lambda_0}{2}(\mu-\mu_0)^2\right)\\
&\times\lambda_e^{Km/2+a_2-1}\exp(-\lambda_e/b_2)\lambda_\theta^{K/2+a_1-1}\exp(-\lambda_\theta/b_1).
\end{aligned}
$$

Let $P(\boldsymbol{\Theta},\Lambda)=f(\boldsymbol{\Theta})g(\Lambda)$, so f is

$$
\begin{aligned}
&\exp\left\{-\frac{1}{2}\left[(\mathbf{Y}-W\boldsymbol{\theta})^T(\lambda_e I_{Km\times Km})(\mathbf{Y}-W\boldsymbol{\theta})\right]\right\}\\
&\times\exp\left\{-\frac{1}{2}\left[(\boldsymbol{\theta}-\mu\mathbf{1})^T(\lambda_\theta I_{K\times K})(\boldsymbol{\theta}-\mu\mathbf{1})+\lambda_0(\mu-\mu_0)^2\right]\right\}\\
&=\exp\left\{-\frac{1}{2}\left[\boldsymbol{\theta}^T(m\lambda_e+\lambda_\theta)I_{K\times K}\boldsymbol{\theta}\right]\right\}\\
&\times\exp\left\{-\frac{1}{2}\left[\mu(K\lambda_\theta+\lambda_o)\mu+\boldsymbol{\theta}^T(-2\lambda_\theta\mathbf{1})\mu+(-2\lambda_e Y^TW)\boldsymbol{\theta}+(-2\lambda_0\mu_0)\mu\right]\right\}\\
&\times\exp\left\{-\frac{1}{2}\left[\lambda_e Y^TY+\lambda_0\mu_0^2\right]\right\}.
\end{aligned}
\tag{C.1}
$$

So, $f(\boldsymbol{\Theta})=\exp(-\frac{1}{2}(\boldsymbol{\Theta}^TC\boldsymbol{\Theta}+D\boldsymbol{\Theta}+\lambda_e\mathbf{Y}^T\mathbf{Y}+\lambda_0\mu_0^2))$ with

$$
C_{(K+1)\times(K+1)}=\begin{bmatrix}\overbrace{(m\lambda_e+\lambda_\theta)I}^{\boldsymbol{\theta}} & \overbrace{-\lambda_\theta\mathbf{1}}^{\mu}\\ -\lambda_\theta\mathbf{1}^T & K\lambda_\theta+\lambda_0\end{bmatrix},
$$

and

$$
D_{1\times(K+1)}=(-2\lambda_e\mathbf{Y}^TW,-2\lambda_0\mu_0).
$$

To compute the conditional distribution of $\mathbf{\Theta}$, given Λ, we have $f(\mathbf{\Theta})$

$$
\begin{aligned}
&\propto \exp(-\frac{1}{2}(\mathbf{\Theta}^T C\mathbf{\Theta} + D\mathbf{\Theta})) \\
&= \exp\left(-\frac{1}{2}\left(\mathbf{\Theta}^T C\mathbf{\Theta} - 2(-\frac{1}{2}DC^{-1})C\mathbf{\Theta} + (-\frac{1}{2}DC^{-1})C(-\frac{1}{2}DC^{-1})^T\right)\right) \\
&\times \exp\left(-\frac{1}{2}\left((\frac{1}{2}DC^{-1})C(-\frac{1}{2}DC^{-1})^T\right)\right) \\
&= \exp\left(-\frac{1}{2}((\mathbf{\Theta} - (-\frac{1}{2}C^{-1}D^T))^T C(\mathbf{\Theta} - (-\frac{1}{2}C^{-1}D^T)))\right) \times \exp(+\frac{1}{8}DC^{-1}D^T) \\
&\propto \exp\left(-\frac{1}{2}((\mathbf{\Theta} - (-\frac{1}{2}C^{-1}D^T))^T C(\mathbf{\Theta} - (-\frac{1}{2}C^{-1}D^T)))\right),
\end{aligned}
$$

which is a normal kernel, and hence C is the precision matrix, and $-\frac{1}{2}C^{-1}D^T$ is the mean of $\mathbf{\Theta}|\Gamma$. Note that from (4.9) that $C = X^T\Gamma^{-1}X$ and $-\frac{1}{2}D^T = X^T\Gamma^{-1}\mathbf{y}$. Therefore, the conditional posterior density of $\mathbf{\Theta}$ is

$$
\mathbf{\Theta}|\Lambda \sim N(C^{-1}(-\frac{1}{2}D^T)\ ,\ C^{-1})\ ,
$$

which appears as equation (5.1). We can avoid computing the inverse above by using similar methods to those described in Section 4.3. But since

$$
C^{-1}_{(K+1)\times(K+1)} = \begin{bmatrix} (m\lambda_e + \lambda_\theta)I & -\lambda_\theta\mathbf{1} \\ -\lambda_\theta\mathbf{1}^T & K\lambda_\theta + \lambda_0 \end{bmatrix}^{-1} = \begin{bmatrix} C^{11} & C^{12} \\ C^{21} & C^{22} \end{bmatrix},
$$

where

$$
\begin{aligned}
C^{22} =& (K\lambda_\theta + \lambda_0 - \frac{(-\lambda_\theta\mathbf{1}^T)(-\lambda_\theta\mathbf{1})}{m\lambda_e + \lambda_\theta})^{-1} = \frac{m\lambda_e + \lambda_\theta}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta} \\
C^{12} =& -\frac{1}{m\lambda_e + \lambda_\theta} - \lambda_\theta\mathbf{1}C^{22} = \frac{\lambda_\theta}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_\theta\lambda_0}\mathbf{1} \\
C^{11} =& \frac{1}{m\lambda_e + \lambda_\theta}I + \frac{\lambda_\theta^2}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_\theta\lambda_0}\frac{1}{m\lambda_e + \lambda_\theta}J_{K\times K} \\
C^{21} =& \text{Transpose}(C^{12}),
\end{aligned}
\tag{C.2}
$$

where J is a matrix of 1s, it is clear that we never really have to compute any matrix inverses to obtain C^{-1}. The computation will not become much harder with increasing dimensions.

We could also derive the conditional distribution in the framework of Hodges (1998), by rewriting the hierarchical model in the following way.

$$\begin{aligned} Y_{ij} =& \theta_i + \epsilon_{ij} \Rightarrow \mathbf{Y}_{Km\times 1} = \boldsymbol{\theta} + \boldsymbol{\epsilon} \\ \theta_i =& \mu + \delta_i \Rightarrow \mathbf{0}_{K\times 1} = -\boldsymbol{\theta} + \mu\mathbf{1} + \boldsymbol{\delta} \\ \mu =& \mu_0 + \xi \Rightarrow \mu_0 = \mu + \xi, \end{aligned}$$

or

$$\mathbf{y}|\boldsymbol{\Theta}, \Lambda = X\boldsymbol{\Theta} + \mathbf{E}$$

where $\mathbf{y}^T = (\mathbf{Y}^T, \mathbf{0}^T, \mu_0)$. So, we write

$$\mathbf{y} = \left[\begin{array}{c} \mathbf{Y}_{Km\times 1} \\ \hline \mathbf{0}_{K\times 1} \\ \hline \mu_0 \end{array}\right] = \left[\begin{array}{c|c} W_{Km\times K} & 0_{Km\times 1} \\ \hline -I_{K\times K} & \mathbf{1}_{K\times 1} \\ \hline \mathbf{0}_{1\times K} & -1_{1\times 1} \end{array}\right] \left[\begin{array}{c} \boldsymbol{\theta} \\ \hline \mu \end{array}\right] + E.$$

$$\mathrm{Cov}(E) \;=\; \Gamma \;=\; \left[\begin{array}{c|c|c} \Gamma_1 & 0 & 0 \\ \hline 0 & \Gamma_2 & 0 \\ \hline 0 & 0 & \Gamma_3 \end{array}\right],$$

where $\Gamma_1 = \lambda_e^{-1} I_{Km\times Km}$, $\Gamma_2 = \lambda_\theta^{-1} I_{K\times K}$, $\Gamma_3 = \lambda_0^{-1}$, so

$$\Gamma^{-1} = \mathrm{Diag}((\lambda_e, \cdots \lambda_e), (\lambda_\theta, \cdots \lambda_\theta), \lambda_0).$$

It can be shown (Hodges, 1998) that the conditional posterior density is

$$\boldsymbol{\Theta}|\Lambda \sim N((X^T\Gamma^{-1}X)^{-1}(X^T\Gamma^{-1}\mathbf{y})\ ,\ (X^T\Gamma^{-1}X)^{-1})\ .$$

Marginal Posterior Distribution

To compute the marginal posterior distribution, $S(\lambda_e, \lambda_\theta)$, we follow the method outlined in Appendix B.1. Setting $2C\boldsymbol{\Theta} + D^T = 0$ we get

$$(m\lambda_e + \lambda_\theta)\boldsymbol{\theta} - \lambda_\theta \mathbf{1}\mu = \lambda_e W^T \mathbf{Y}$$
$$-\lambda_\theta \mathbf{1}^T \boldsymbol{\theta} + (K\lambda_\theta + \lambda_0)\mu = \lambda_0 \mu_0.$$

Let the solutions be $\hat{\boldsymbol{\Theta}}^T = (\hat{\boldsymbol{\theta}}^T, \hat{\mu})$. We get

$$\hat{\mu} = \frac{\lambda_e \lambda_\theta \sum_{i=1}^K V_i + \lambda_0 \mu_0 (m\lambda_e + \lambda_\theta)}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta}$$
$$\hat{\boldsymbol{\theta}} = \frac{\lambda_e V}{m\lambda_e + \lambda_\theta} + \frac{\lambda_e \lambda_\theta^2 \sum_{i=1}^K V_i + \lambda_0 \mu_0 \lambda_\theta (m\lambda_e + \lambda_\theta)}{(m\lambda_e + \lambda_\theta)(Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta)} \mathbf{1}.$$

The marginal posterior distribution is then

$$\begin{aligned} S(\lambda_\theta, \lambda_e) \propto & \det(C)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\hat{\boldsymbol{\Theta}}^T C \hat{\boldsymbol{\Theta}} + D\hat{\boldsymbol{\Theta}})\right) \\ & \times \lambda_e^{Km/2 + a_2 - 1} \exp\left(-\lambda_e (1/b_2 + \sum_{i,j} Y_{ij}^2/2)\right) \\ & \times \lambda_\theta^{K/2 + a_1 - 1} \exp(-\lambda_\theta / b_1), \end{aligned}$$

which appears as equation (5.2).

C.2 Proof for Envelope

We show that $R(\lambda_e, \lambda_\theta)$ is an envelope for $S(\lambda_e, \lambda_\theta)$, i.e., for some $B < \infty$

$$f(\lambda_e, \lambda_\theta) = S(\lambda_e, \lambda_\theta)/R(\lambda_e, \lambda_\theta) \leq B \text{ for all } \lambda_e, \lambda_\theta > 0,$$

where

$$
\begin{aligned}
f(\lambda_\theta, \lambda_e) \propto & \det(C)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}} + D\hat{\mathbf{\Theta}})\right) \\
& \times \lambda_e^{Km/2+a_2} \exp\left(-\lambda_e(1/b_2 + \sum_{i,j} Y_{ij}^2/2)\right) \lambda_\theta^{K/2+a_1} \exp(-\lambda_\theta/b_1) \\
& \times \left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_e) - \mu_e}{\sigma_e}\right)^2\right]^{(\nu+1)/2} \left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_\theta) - \mu_\theta}{\sigma_\theta}\right)^2\right]^{(\nu+1)/2}.
\end{aligned}
$$

Since we can write

$$
C_{(K+1)\times(K+1)} = \begin{bmatrix} \lambda_\theta I & -\lambda_\theta \mathbf{1} \\ -\lambda_\theta \mathbf{1}^T & K\lambda_\theta + \lambda_0 \end{bmatrix} + \begin{bmatrix} m\lambda_e I & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} = C_1 + C_2.
$$

where C_1, C_2 above can be shown to be non-negative definite. So

$$
\det(C) \geq \det(C_1) = \lambda_0 \lambda_\theta^K, \Rightarrow \det(C)^{-\frac{1}{2}} \leq \lambda_0^{-\frac{1}{2}} \lambda_\theta^{-K/2}
$$

since $\det(C_1 + C_2) \geq \det(C_1)$ if $C_1, C_2 \geq 0$. Therefore (ignoring constants),

$$
\begin{aligned}
f(\lambda_\theta, \lambda_e) \leq & \exp\left(-\frac{1}{2}(\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}})\right) \exp\left(-\frac{1}{2} D\hat{\mathbf{\Theta}}\right) \\
& \times \lambda_e^{Km/2+a_2} \exp\left(-\lambda_e(1/b_2 + \sum_{i,j} Y_{ij}^2/2)\right) \lambda_\theta^{a_1} \exp(-\lambda_\theta/b_1) \\
& \times \left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_e) - \mu_e}{\sigma_e}\right)^2\right]^{(\nu+1)/2} \left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_\theta) - \mu_\theta}{\sigma_\theta}\right)^2\right]^{(\nu+1)/2}.
\end{aligned} \tag{C.3}
$$

By rearranging terms we have,

$$
\begin{aligned}
f(\lambda_\theta, \lambda_e) \leq & \exp\left(-\frac{1}{2}(\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}})\right) \exp\left(-\frac{1}{2} D\hat{\mathbf{\Theta}} - \lambda_e \sum_{i,j} Y_{ij}^2/2\right) \\
& \times \lambda_e^{Km/2+a_2} \exp\left(-\lambda_e/b_2\right) \left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_e) - \mu_e}{\sigma_e}\right)^2\right]^{(\nu+1)/2} \\
& \times \lambda_\theta^{a_1} \exp(-\lambda_\theta/b_1) \left[1 + \frac{1}{\nu}\left(\frac{\log(\lambda_\theta) - \mu_\theta}{\sigma_\theta}\right)^2\right]^{(\nu+1)/2}.
\end{aligned}
$$

It is easy to see that

$$\lambda_e^{Km/2+a_2}\exp\left(-\lambda_e/b_2\right)\left[1+\frac{1}{\nu}\left(\frac{\log(\lambda_e)-\mu_e}{\sigma_e}\right)^2\right]^{(\nu+1)/2}\leq B_1,$$

and

$$\lambda_\theta^{a_1}\exp(-\lambda_\theta/b_1)\left[1+\frac{1}{\nu}\left(\frac{\log(\lambda_\theta)-\mu_\theta}{\sigma_\theta}\right)^2\right]^{(\nu+1)/2}\leq B_2,$$

for some $B_1, B_2 < \infty$ since they are simply ratios of gamma kernels to log-t kernels. Therefore,

$$f(\lambda_\theta,\lambda_e)\leq\exp\left(-\frac{1}{2}(\hat{\mathbf{\Theta}}^T C\hat{\mathbf{\Theta}})\right)\exp\left(-\frac{1}{2}D\hat{\mathbf{\Theta}}-\lambda_e\sum_{i,j}Y_{ij}^2/2\right)\times B_1B_2. \tag{C.4}$$

Let,

$$g(\lambda_e,\lambda_\theta)=\exp\left(-\frac{1}{2}D\hat{\mathbf{\Theta}}\right)=\exp\left(\lambda_e V^T\hat{\theta}+\lambda_0\mu_0\hat{\mu}\right),$$

Since

$$\begin{aligned}\lambda_e V^T\hat{\theta}=&\lambda_e V^T\left(\frac{\lambda_e V}{m\lambda_e+\lambda_\theta}+\frac{\lambda_e\lambda_\theta^2\sum_{i=1}^K V_i+\lambda_0\mu_0\lambda_\theta(m\lambda_e+\lambda_\theta)}{(m\lambda_e+\lambda_\theta)(Km\lambda_e\lambda_\theta+m\lambda_e\lambda_0+\lambda_0\lambda_\theta)}\mathbf{1}\right)\\=&\frac{\lambda_e^2\sum_{i=1}^K V_i^2}{m\lambda_e+\lambda_\theta}+\frac{\lambda_e^2\lambda_\theta^2(\sum_{i=1}^K V_i)^2}{(m\lambda_e+\lambda_\theta)(Km\lambda_e\lambda_\theta+m\lambda_e\lambda_0+\lambda_0\lambda_\theta)}\\&+\frac{\lambda_0\mu_0\lambda_\theta\lambda_e\sum_{i=1}^K V_i}{Km\lambda_e\lambda_\theta+m\lambda_e\lambda_0+\lambda_0\lambda_\theta},\end{aligned}$$

we have

$$\begin{aligned}\log(g(\lambda_e,\lambda_\theta))=&\frac{\lambda_e^2\sum_{i=1}^K V_i^2}{m\lambda_e+\lambda_\theta}+\frac{\lambda_e^2\lambda_\theta^2(\sum_{i=1}^K V_i)^2}{(m\lambda_e+\lambda_\theta)(Km\lambda_e\lambda_\theta+m\lambda_e\lambda_0+\lambda_0\lambda_\theta)}\\&+\frac{\lambda_0\mu_0\lambda_\theta\lambda_e\sum_{i=1}^K V_i}{Km\lambda_e\lambda_\theta+m\lambda_e\lambda_0+\lambda_0\lambda_\theta}+\lambda_0\mu_0\hat{\mu}\\=&T_1+T_2+T_3+T_4,\text{ say.}\end{aligned} \tag{C.5}$$

We consider the last term first,

$$\begin{aligned} T_4 = \lambda_0\mu_0\hat{\mu} =& \frac{\lambda_0\mu_0\lambda_e\lambda_\theta \sum_{i=1}^K V_i}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta} + \frac{(\lambda_0\mu_0)^2(m\lambda_e + \lambda_\theta)}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta} \\ =& \frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km + m\lambda_0/\lambda_\theta + \lambda_0/\lambda_e} + \frac{(\lambda_0\mu_0)^2}{Km\lambda_e\lambda_\theta/(m\lambda_e + \lambda_\theta) + \lambda_0} \\ \leq& \frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km + m\lambda_0/\lambda_\theta + \lambda_0/\lambda_e} + \frac{(\lambda_0\mu_0)^2}{\lambda_0} \end{aligned}$$

since $Km\lambda_e\lambda_\theta/(m\lambda_e + \lambda_\theta) > 0$ and all other terms are positive. Now, if $\mu_0 \sum_{i=1}^K V_i \geq 0$, by a similar argument,

$$\frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km + m\lambda_0/\lambda_\theta + \lambda_0/\lambda_e} \leq \frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km}$$

and, if $\mu_0 \sum_{i=1}^K V_i < 0$,

$$\frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km + m\lambda_0/\lambda_\theta + \lambda_0/\lambda_e} < 0.$$

Thus, $\frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km+m\lambda_0/\lambda_\theta+\lambda_0/\lambda_e}$ is bounded above by $B_3 = \max(0, \frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km})$, so

$$\lambda_0\mu_0\hat{\mu} \leq B_3 + \frac{(\lambda_0\mu_0)^2}{\lambda_0} = B_4. \tag{C.6}$$

Now, consider the third term. If $\mu_0 \leq 0$, T_3 is bounded above by zero. We only need to consider the case where $\mu_0 > 0$

$$\begin{aligned} T_3 =& \frac{\lambda_0\mu_0\lambda_\theta\lambda_e \sum_{i=1}^K V_i}{Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta} = \frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km + (m\lambda_e\lambda_0 + \lambda_0\lambda_\theta)/\lambda_e\lambda_\theta} \\ \text{So, } T_3 \leq& \frac{\lambda_0\mu_0 \sum_{i=1}^K V_i}{Km} = B_5 < \infty. \end{aligned} \tag{C.7}$$

Hence, from (C.7), (C.6), (C.5), and (C.4) we have, by setting $B_6 = \exp(B_4 + B_5) \times B_1 B_2$,

$$f(\lambda_\theta, \lambda_e) \leq \exp\left(-\frac{1}{2}(\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}})\right) \exp\left(T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2\right) \times B_6$$
$$\log(f(\lambda_\theta, \lambda_e)) \leq -\frac{1}{2}\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}} + T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2 + \log(B_6). \tag{C.8}$$

Also, $T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2$ is

$$= \frac{\lambda_e^2 \sum_{i=1}^K V_i^2}{m\lambda_e + \lambda_\theta} + \frac{\lambda_e^2 \lambda_\theta^2 (\sum_{i=1}^K V_i)^2}{(m\lambda_e + \lambda_\theta)(Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta)} - \lambda_e \sum_{i,j} Y_{ij}^2/2$$
$$= \frac{\lambda_e^2 \sum_{i=1}^K V_i^2 (Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta) + \lambda_e^2\lambda_\theta^2 (\sum_{i=1}^K V_i)^2}{(m\lambda_e + \lambda_\theta)(Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta)} - \lambda_e \sum_{i,j} Y_{ij}^2/2.$$

Let $\text{den} = (m\lambda_e + \lambda_\theta)(Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta) = Km^2\lambda_e^2\lambda_\theta + m^2\lambda_e^2\lambda_0 + 2m\lambda_e\lambda_\theta\lambda_0 + Km\lambda_e\lambda_\theta^2 + \lambda_\theta^2\lambda_0$, and $h = \sum_{i,j} Y_{ij}^2/2$. Then $T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2$

$$= \frac{\lambda_e^2 \left\{\sum_{i=1}^K V_i^2 Km\lambda_e\lambda_\theta + \sum_{i=1}^K V_i^2 m\lambda_e\lambda_0 + \sum_{i=1}^K V_i^2 \lambda_0\lambda_\theta + \lambda_\theta^2 (\sum_{i=1}^K V_i)^2\right\}}{(m\lambda_e + \lambda_\theta)(Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta)}$$
$$+ \frac{\lambda_e^2 \{-\lambda_e hKm^2\lambda_\theta - \lambda_e hm^2\lambda_0 - \lambda_e^2 h2m\lambda_\theta\lambda_0 - \lambda_e^2 hKm\lambda_\theta^2\} - h\lambda_e\lambda_\theta^2\lambda_0}{(m\lambda_e + \lambda_\theta)(Km\lambda_e\lambda_\theta + m\lambda_e\lambda_0 + \lambda_0\lambda_\theta)}$$
$$= \frac{\lambda_e^2}{\text{den}} \left\{\lambda_e(Km\lambda_\theta + m\lambda_0)(\sum_{i=1}^K V_i^2 - hm) + \lambda_\theta^2((\sum_{i=1}^K V_i)^2 - hKm)\right\}$$
$$+ \frac{\lambda_e^2}{\text{den}} \left\{\lambda_0\lambda_\theta(\sum_{i=1}^K V_i^2 - 2hm)\right\} - \frac{h\lambda_e\lambda_\theta^2\lambda_0}{\text{den}}.$$

Since the denominator is positive, and $\sum_{i=1}^K V_i^2 - 2hm = \sum_{i=1}^K V_i^2 - m\sum_{i,j} Y_{ij} - 1/b_2 < 0$, along with the fact that $h\lambda_e\lambda_\theta^2\lambda_0 > 0$, the second line is bounded

above by $B_7 < \infty$, say. Hence, we have $T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2$

$$
\begin{aligned}
\leq & \frac{\lambda_e^2}{\text{den}} \left\{ \lambda_e(Km\lambda_\theta + m\lambda_0)(\sum_{i=1}^K V_i^2 - hm) + \lambda_\theta^2((\sum_{i=1}^K V_i)^2 - hKm) \right\} + B_7 \\
= & \frac{\lambda_e^2}{\text{den}} \left\{ \lambda_e(Km\lambda_\theta + m\lambda_0)(\sum_{i=1}^K V_i^2/2 - hm) + \lambda_\theta^2((\sum_{i=1}^K V_i)^2/2 - hKm) \right\} \\
& + \frac{\lambda_e^2}{\text{den}} \left\{ \lambda_e(Km\lambda_\theta + m\lambda_0)(\sum_{i=1}^K V_i^2/2) + \lambda_\theta^2((\sum_{i=1}^K V_i)^2/2) \right\} + B_7 \\
\leq & \frac{\lambda_e^2}{\text{den}} \left\{ \lambda_e(Km\lambda_\theta + m\lambda_0)(\sum_{i=1}^K V_i^2/2) + \lambda_\theta^2((\sum_{i=1}^K V_i)^2/2) \right\} \\
= & \frac{1}{\text{den}} \left\{ \lambda_e^3\lambda_\theta Km \sum_{i=1}^K V_i^2/2 + \lambda_e^3 m\lambda_0 \sum_{i=1}^K V_i^2/2 + \lambda_e^2\lambda_\theta^2((\sum_{i=1}^K V_i)^2/2) \right\} + B_7
\end{aligned}
\tag{C.9}
$$

where, for the last inequality, we use the fact that $\sum_{i=1}^K V_i^2/2 - hm < 0$, and $(\sum_{i=1}^K V_i)^2/2 - hKm < 0$. We now need to use some terms from $-\frac{1}{2}\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}}$ to help bound the remaining terms above. Some algebra shows that

$$
\begin{aligned}
& -\frac{1}{2}\hat{\mathbf{\Theta}}^T C \hat{\mathbf{\Theta}} = \frac{-\frac{1}{2}\lambda_e^2 \sum_{i=1}^K V_i^2 - \frac{1}{2}\hat{\mu}^2(Km\lambda_\theta + m\lambda_0 + \lambda_0\lambda_\theta)}{m\lambda_e + \lambda_\theta} \\
= & \frac{-\frac{1}{2}\lambda_e^2 \sum_{i=1}^K V_i^2(Km\lambda_e\lambda_\theta + m\lambda_0 + \lambda_0\lambda_\theta) - \frac{1}{2}\hat{\mu}^2(Km\lambda_e\lambda_\theta + m\lambda_0 + \lambda_0\lambda_\theta)^2}{\text{den}} \\
= & \frac{-\frac{1}{2}\lambda_e^3\lambda_\theta \sum_{i=1}^K V_i^2 Km - \frac{1}{2}\lambda_e^3\lambda_0 \sum_{i=1}^K V_i^2 m - \frac{1}{2}\lambda_e^2\lambda_\theta \sum_{i=1}^K V_i^2\lambda_0}{\text{den}} \\
& + \frac{-\frac{1}{2}\lambda_e^2\lambda_\theta^2(\sum_{i=1}^K V_i)^2 - \frac{1}{2}(\lambda_0\mu_0)^2 m^2\lambda_e^2 - \frac{1}{2}(\lambda_0\mu_0)^2\lambda_\theta^2}{\text{den}} \\
& + \frac{-(\lambda_0\mu_0)^2 m\lambda_e\lambda_\theta - \lambda_e^2\lambda_\theta \sum_{i=1}^K V_i\lambda_0\mu_0 m - \lambda_e\lambda_\theta^2 \sum_{i=1}^K V_i\lambda_0\mu_0}{\text{den}}.
\end{aligned}
\tag{C.10}
$$

From (C.11) and (C.9), we have

$$
\begin{aligned}
&-\frac{1}{2}\hat{\mathbf{\Theta}}^T C\hat{\mathbf{\Theta}} + T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2 \\
\leq &\frac{1}{\text{den}}\left\{-\frac{1}{2}\lambda_e^2\lambda_\theta \sum_{i=1}^K V_i^2\lambda_0 - \frac{1}{2}(\lambda_0\mu_0)^2 m^2\lambda_e^2 - \frac{1}{2}(\lambda_0\mu_0)^2\lambda_\theta^2 - (\lambda_0\mu_0)^2 m\lambda_e\lambda_\theta\right\} \\
+&\frac{1}{\text{den}}\left\{-\lambda_e^2\lambda_\theta \sum_{i=1}^K V_i\lambda_0\mu_0 m - \lambda_e\lambda_\theta^2 \sum_{i=1}^K V_i\lambda_0\mu_0\right\} \\
&\leq \frac{1}{\text{den}}\left\{-\lambda_e^2\lambda_\theta \sum_{i=1}^K V_i\lambda_0\mu_0 m - \lambda_e\lambda_\theta^2 \sum_{i=1}^K V_i\lambda_0\mu_0\right\} \\
&= \frac{1}{\text{den}}\left\{(-\sum_{i=1}^K V_i\mu_0)\left(\lambda_e^2\lambda_\theta\lambda_0 m + \lambda_e\lambda_\theta^2\lambda_0\right)\right\}.
\end{aligned}
\tag{C.11}
$$

Now, if $(-\sum_{i=1}^K V_i\mu_0) \leq 0$ the above terms are bounded above by zero and we are done. If $(-\sum_{i=1}^K V_i\mu_0) > 0$, we have

$$
\begin{aligned}
&\frac{1}{\text{den}}\left\{(-\sum_{i=1}^K V_i\mu_0)\left(\lambda_e^2\lambda_\theta\lambda_0 m + \lambda_e\lambda_\theta^2\lambda_0\right)\right\} + B_7 \\
\leq &\frac{(-\sum_{i=1}^K V_i\mu_0)\left(\lambda_e^2\lambda_\theta\lambda_0 m + \lambda_e\lambda_\theta^2\lambda_0\right)}{Km^2\lambda_e^2\lambda_\theta + m^2\lambda_e^2\lambda_0 + 2m\lambda_e\lambda_\theta\lambda_0 + Km\lambda_e\lambda_\theta^2 + \lambda_\theta^2\lambda_0} + B_7 \\
\leq &\frac{(-\sum_{i=1}^K V_i\mu_0)\left(\lambda_e^2\lambda_\theta\lambda_0 m\right)}{Km^2\lambda_e^2\lambda_\theta} + \frac{(-\sum_{i=1}^K V_i\mu_0)\lambda_e\lambda_\theta^2\lambda_0}{Km\lambda_e\lambda_\theta^2} \\
= &\frac{(-\sum_{i=1}^K V_i\mu_0)2\lambda_0}{Km} + B_7 = B_8,
\end{aligned}
$$

where $B_8 < \infty$. We therefore have

$$
-\frac{1}{2}\hat{\mathbf{\Theta}}^T C\hat{\mathbf{\Theta}} + T_1 + T_2 - \lambda_e \sum_{i,j} Y_{ij}^2/2 \leq B_8,
$$

and combining this with (C.8),

$$
\log(f(\lambda_\theta, \lambda_e)) \leq B_8 + \log(B_6), \text{for } \lambda_e, \lambda_\theta > 0.
$$

We have thus proved that $f(\lambda_\theta, \lambda_e) \leq B$ for $B = \exp(B_8)B_6 < \infty$, so $R(\lambda_\theta, \lambda_e)$ is an envelope for $S(\lambda_\theta, \lambda_e)$ as required.