Sampling for Bayesian computation with large datasets*

Zaiying Huang† Andrew Gelman‡

April 27, 2005

Abstract

Multilevel models are extremely useful in handling large hierarchical datasets. However, computation can be a challenge, both in storage and CPU time per iteration of Gibbs sampler or other Markov chain Monte Carlo algorithms. We propose a computational strategy based on sampling the data, computing separate posterior distributions based on each sample, and then combining these to get a consensus posterior inference. With hierarchical data structures, we perform cluster sampling into subsets with the same structures as the original data. This reduces the number of parameters as well as sample size for each separate model fit. We illustrate with examples from climate modeling and newspaper marketing.

Keywords: Bayesian inference, cluster sampling, Gibbs sampler, hierarchical model, multilevel model, MCMC, Metropolis algorithm, particle filtering

1 Introduction

Multilevel models (also called hierarchical linear models, random effects regressions, and mixed effects models; see, for example, Robinson, 1991, Longford, 1993, and Goldstein, 1995) are extremely useful in handling hierarchical datasets. In applications with small to medium sized data sets, Bayesian methods have found great success in statistical practice. In particular, applied statistical work has seen a surge in the use of Bayesian hierarchical models for modeling multilevel or relational data (Carlin and Louis, 2000) in a variety of fields including health, education, environment and business (see, for example, Gelman et al., 1995, and Carlin and Louis, 2000). Modern Bayesian inference generally entails computing simulation draws of the parameters from the posterior distribution (see Besag et al., 1995, for an overview). For hierarchical linear models, this can be done fairly easily using the Gibbs sampler (see Gelfand and Smith, 1990). For hierarchical generalized linear models, essentially the same algorithm can be used by linearizing the likelihood and then applying a Metropolis-Hastings accept/reject rule at each step (see Gilks, Richardson, and Spiegelhalter, 1996). However, computation for fitting a Bayesian model with a massive dataset can be a big challenge, both in storage and CPU time per iteration of Gibbs sampler or other Markov chain Monte Carlo algorithms.

*We thank Upmanu Lall for helpful comments and the National Science Foundation for financial support.

†Department of Statistics, Columbia University, New York, huang@stat.columbia.edu

‡Department of Statistics, Columbia University, New York, gelman@stat.columbia.edu

In this paper, we propose a computational strategy of randomly partitioning the data into disjoint subsets, computing posterior distributions separately for each subset, and then combining to get a consensus posterior inference. With hierarchical data structures, we perform cluster sampling into subsets with the same structures as the original data. This reduces the number of parameters as well as the sample size for each separate model fit. Our method goes beyond previous approaches to computation using data subsetting (see Chopin, 2002, and Ridgeway and Madigan, 2002), in our use of cluster sampling and also in that we select subsets of the data using random sampling, which should lead the inferences from the separate subsets to be more representative of the target inference for the entire dataset. Section 2 of this paper lays out the basic idea on which our computation strategy is based. In Section 3, we illustrate with two practical examples from our work in statistical application with large datasets in newspaper marketing and climate modeling, and we conclude in Section 4 with suggestions for further research.

2 Sampling for posterior inference

2.1 Basic examples

2.1.1 Dividing a dataset into $K = 2$ parts

We first use a simple example to discuss our basic idea. Let $y_1, \dots, y_n$ be an independent, identically distributed sample from $\mathrm{N}(\theta, \sigma^2)$, where θ and σ^2 are parameters of interest which can estimated by the sample mean $\bar{y}$ and sample variance s_y^2, respectively. From the central limit theorem for sums of random variables, these two estimates have good statistical properties such as consistency and asymptotic normality (see Bickel and Doksum, 1976).

Now suppose we randomly divide $(y_1, \dots, y_n)$ into $K = 2$ parts, $(u_1, \dots, u_k)$ and $(v_1, \dots, v_l)$, with $k + l = n$. From these two random subsets, we can compute $\bar{u}$,s_u^2 and $\bar{v}$,s_v^2, which may also be used to estimate θ and σ^2. If k and l are sufficiently large, all these estimates are consistent and asymtotically normal. We can combine them using their estimated precisions (inverted variances) as weights to obtain more accurate estimates. If we denote the precisions for $\bar{u}$, $\bar{v}$, s_u^2, and s_v^2 by w_u, w_v, $w_{s_u^2}$, and w_{s_v}, then we have the following combined estimates:

$$\begin{aligned} \bar{y_c} &= \frac{w_u \bar{u} + w_v \bar{v}}{w_u + w_v} \\ s_c^2 &= \frac{w_{s_u} s_u^2 + w_{s_v} s_v^2}{w_{s_u} + w_{s_v}} \end{aligned} \tag{1}$$

which should be better than the estimates obtained from either subset. If the estimated precisions are proportional to the sample size k or l, then equation (1) reduces to the simple form:

$$\begin{aligned} \bar{y_c} &= \frac{k\bar{u} + l\bar{v}}{n} \\ s_c^2 &= \frac{k s_u^2 + l s_v^2}{n} \end{aligned} \tag{2}$$

which become simple means if the sample sizes k and l are equal. This computation strategy does not reduce computation for this example. We are presenting it just to introduce our ideas.

2.1.2 Classical linear models

Now consider the following regression model with fixed prior distribution:

$$y|\beta \sim \mathrm{N}(X\beta, \Sigma_y) \tag{3}$$

$$\beta \sim \mathrm{N}(\beta_0, \Sigma_\beta). \tag{4}$$

In this model y is the full dataset and X is the design matrix and we assume Σ_y, β_0, and Σ_β are known. We have posterior distribution $\beta|y \sim \mathrm{N}(\hat{\beta}, V_\beta)$, where

$$\begin{aligned} \hat{\beta} &= \left(X^t\Sigma_y^{-1}X + \Sigma_\beta^{-1}\right)^{-1}\left(X^t\Sigma_y^{-1}y + \Sigma_\beta^{-1}\beta_0\right), \\ V_\beta^{-1} &= X^t\Sigma_y^{-1}X + \Sigma_\beta^{-1}. \end{aligned}$$

Suppose we divide y into K disjoint subsets denoted by $y^{(1)}, \ldots, y^{(K)}$, and the corresponding partition of the design matrix X is $X^{(1)}, \ldots, X^{(K)}$. For $k = 1, \ldots, K$, we have $\beta|y^{(k)} \sim \mathrm{N}(\hat{\beta}^{(k)}, V_\beta^{(k)})$, where

$$\begin{aligned} \hat{\beta}^{(k)} &= \left(X^{(k)t}\Sigma_y^{-1}X^{(k)} + \Sigma_\beta^{-1}\beta_0\right)^{-1}\left(X^{(k)t}\Sigma_y^{-1}y^{(k)} + \Sigma_\beta^{-1}\beta_0\right), \\ (V_\beta^{(k)})^{-1} &= X^{(k)t}\Sigma_y^{-1}X^{(k)} + \Sigma_\beta^{-1}. \end{aligned}$$

We can express $\hat{\beta}$ and V_β^{-1} in terms of $\hat{\beta}^{(1)}, \ldots, \hat{\beta}^{(K)}$ and $V_\beta^{(1)}, \ldots, V_\beta^{(K)}$ as follows,

$$\begin{aligned} \hat{\beta} &= V_\beta^{-1}\sum_{k=1}^{K}\left((V_\beta^{(k)})^{-1}\hat{\beta}^{(k)} - \frac{K-1}{K}\Sigma_\beta^{-1}\beta_0\right), \\ V_\beta^{-1} &= \sum_{k=1}^{K}\left((V_\beta^{(k)})^{-1} - \frac{K-1}{K}\Sigma_\beta^{-1}\right). \end{aligned}$$

2.1.3 Simple linear hierarchical models

We consider a simple hierarchical model:

$$y_i|\beta, \alpha_i \sim \mathrm{N}(x_i^t\beta + \alpha_i, \sigma_i^2), \qquad \text{for } i = 1, \ldots, n \tag{5}$$

$$\beta|\beta_0, \Sigma_\beta \sim \mathrm{N}(\beta_0, \Sigma_\beta) \tag{6}$$

$$\alpha_i|\tau \sim \mathrm{N}(0, \tau^2). \tag{7}$$

For simplicity we assume σ's, β_0, Σ_β, and τ are all known. Let $y = (y_1, \ldots, y_n)^t$ (full dataset) and $X = (x_1, \ldots, x_n)^t$ (design matrix for β), then we have $\beta|y \sim \mathrm{N}(\hat{\beta}, V_\beta)$, where

$$\begin{aligned} \hat{\beta} &= V_\beta\left(\frac{X^ty}{\sigma^2 + \tau^2} + \Sigma_\beta^{-1}\beta_0\right), \\ V_\beta^{-1} &= \left(\frac{X^tX}{\sigma^2 + \tau^2} + \Sigma_\beta\right). \end{aligned}$$

Suppose we divide y into K disjoint subsets denoted by $y^{(1)}, \ldots, y^{(K)}$ and the corresponding partition of the design matrix X is $X^{(1)}, \ldots, X^{(K)}$. For $k = 1, \ldots, K$, we have $\beta | y^{(k)} \sim \mathrm{N}(\hat{\beta}^{(k)}, V_\beta^{(k)})$, where

$$\begin{aligned} \hat{\beta}^{(k)} &= \left(\frac{X^{(k)t} X^{(k)}}{\sigma^2 + \tau^2} + \Sigma_\beta^{-1} \beta_0 \right)^{-1} \left(\frac{X^{(k)t} y^{(k)}}{\sigma^2 + \tau^2} + \Sigma_\beta^{-1} \beta_0 \right), \\ (V_\beta^{(k)})^{-1} &= \frac{X^{(k)t} X^{(k)}}{\sigma^2 + \tau^2} + \Sigma_0^{-1}. \end{aligned}$$

We can express $\hat{\beta}$ and V_β^{-1} in terms of $\hat{\beta}^{(1)}, \ldots, \hat{\beta}^{(K)}$ and $V_\beta^{(1)}, \ldots, V_\beta^{(K)}$ as follows,

$$\begin{aligned} \hat{\beta} &= V_\beta^{-1} \sum_{k=1}^{K} \left(V_\beta^{(k)})^{-1} \hat{\beta}^{(k)} - \frac{K-1}{K} \Sigma_\beta^{-1} \beta_0 \right) \\ V_\beta^{-1} &= \sum_{k=1}^{K} \left((V_\beta^{(k)})^{-1} - \frac{K-1}{K} \Sigma_\beta^{-1} \right). \end{aligned}$$

Therefore, we have a closed form to combine the subset inferences for this kind of linear hierarchical model.

2.2 General approach for hierarchical models

Now we turn to consider two-level Bayesian hierarchical models, which we can write in a general notation as,

$$\begin{aligned} y_i | \theta, \varphi &\sim p(y_i | \varphi, \theta_{j(i)}) \quad \text{for } i = 1, \ldots, n \\ \theta_j | \phi &\sim p(\theta_j | \phi) \quad \text{for } j = 1, \ldots, J, \end{aligned} \tag{8}$$

where i the individual index, j is the batch index, and $j(i)$ is an indexing variable indicating the batch j within which observation i falls. In this model, we assume that the data from different batches are independent. The parameters φ and ϕ may be vectors: φ includes common parameters in the likelihood (e.g., fixed effect coefficients and standard deviations of error terms in a regression), and ϕ includes hyperparameters (e.g., the mean and standard deviation of random effect coefficients and group-level regression coefficients). In addition, we assume φ and ϕ have a noninformative prior distribution, $p(\varphi, \phi) \propto 1$. We denote the complete vector of parameters by $\xi = (\theta, \psi)$, where $\theta = (\theta_1, \ldots, \theta_J)$ and $\psi = (\varphi, \phi)$, and the posterior distribution is then $p(\xi | y)$, implying the following likelihood for the data in model (8):

$$p(y | \theta, \varphi) = \prod_{j=1}^{J} \prod_{i=1}^{n} p(y_i | \varphi, \theta_{j(i)}).$$

The posterior distribution for the model parameters is then,

$$p(\xi | y) \propto \prod_{j=1}^{J} \prod_{i=1}^{n} p(y_i | \varphi, \theta_{j(i)}) p(\theta_j | \phi).$$

In order to keep the same hierarchical structure as the original dataset, we perform cluster sampling to divide the original dataset into subsets. This is equivalent to simply sampling the index set for j into subsets. Next, we will compute the separate posterior distributions for each subset sampled, and then combine these to obtain a consensus posterior inference. This strategy reduces the number of parameters as well as the sample size for each separate fit and thus saves storage and CPU time for each iteration when implementing the Gibbs sampler or other Markov chain Monte Carlo algorithms.

As with sampling in general, we subset the data at random so that the samples will be representative of the population. In this case, we have K disjoint samples, and the quantity of interest is the log-likelihood. As in classical sampling, we are estimating a population total (the log-likelihood of all the data) from a sample total (log-likelihood of any subset K). We are using cluster sampling to reduce our computation time, which is analogous to the usual motivation of cluster sampling to reduce costs (see, e.g., Lohr, 1999).

2.3 Computation time per iteration and motivation for cluster sampling

We explore computational costs in the context of linear regression models:

$$\begin{aligned} y_i|\beta &\sim \mathrm{N}(y_i|(X\beta)_{j(i)}, \sigma^2) \qquad \text{for } i = 1, \dots, n \\ \beta_j|\mu_\beta &\sim \mathrm{N}(\beta_j|\mu_\beta, \tau^2) \qquad \text{for } j = 1, \dots, J \\ p(\beta_0, \sigma, \mu_\beta, \tau) &\propto 1, \end{aligned} \tag{9}$$

where $\beta = (\beta_0, \beta_1, \dots, \beta_J)$ are regression coefficients, among which β_0 are common parameters shared by each batch of observations, β_j are only related to the j^{th} batch of observations, σ is the standard deviation of the error terms, and μ_β and τ are hyperparameters. Let $\psi = (\beta_0, \sigma, \mu_\beta, \tau)$, the vector of common parameters. We assume that the length of ψ is negligible compared to m and n, where n and m are the length of $y = (y_1, \dots, y_J)$ and $\beta = (\beta_1, \dots, \beta_J)$, respectively, and thus we can ignore the computation time per iteration for updating ψ. For the vector updating algorithm in fitting the target model, the QR decomposition of the X matrix and backsolving the two upper-triangular systems takes $O(2nm^2)$ floating-point operations (flops); see Golub and van Loan (1983). If we divide the data into K parts of equal size by cluster sampling, i.e., simply sampling the batch index set $\{1, 2, \dots, J\}$. In fitting each separate subset model, the QR decomposition of the X matrix and backsolving the two upper-triangular systems takes $O(\frac{2n}{K}(\frac{m}{K})^2) = O(2nm^2/K^3)$ floating-point operations (flops). Thus sampling vector updating algorithms take $O(2nm^2/K)$ flops in total, compared to non-sampling vector updating algorithms $O(2nm^2)$. For the non-sampling scalar updating algorithm, updating all m components of β takes $O(10nm)$, and for the sampling scalar updating algorithm, updating all m components of β also takes $O(10nm)$. In many of the problems we work with, m is quite large—typically some fraction of n—and thus the vector and scalar updating algorithms require $O(n^3)$ and $O(n^2)$ flops per iteration, respectively, compared to

$O(n^3/K)$ and $O(n^2)$ for the sampling vector and scalar algorithms, respectively. Also if startup and loading time included, the sampling computation strategy should save computation time for vector updating and scalar updating algorithms.

2.4 Combining inferences from the subsets

Once we have the K separate inferences for the parameters shared by the K subsets, we need to combine these. Several methods are possible:

1. Using a normal approximation (motivated by the central limit theorem applied to our random sampling procedure). We can obtain the K separate inferences using normal approximation and combine the K normal distributions. We discuss this approach in Section 2.4.1.

2. Updating from prior to posterior sequentially. Starting with one subset, we can fit the model to this subset and obtain the posterior distribution by normal approximation or by some nonparametric approach, and then fit the model to next subset using the posterior distribution from last step as a prior and continue this procedure step by step until to the last one subset. The inference from the last is conditional on all the data.

3. Starting with one subset, then adding other data using importance sampling (see Chopin, 2002, and Ridgeway and Madigan, 2002). We must generalize the method of Chopin (2002) and Ridgeway and Madigan (2002) to apply to hierarchical models where not all parameters are present in each subset. Consider model (8) and suppose $K = 2$. We sample L (say L=10,000) points from the posterior $p(\phi, \varphi, \theta_1|y_1)$. For each l, we can augment $(\phi^l, \varphi^l, \theta_1^l)$ to $(\phi^l, \varphi^l, \theta_1^l, \theta_2^l)$ by sampling θ_2^l from the prior $p(\theta_2|\phi^l)$. This means that $(\phi^l, \varphi^l, \theta_1^l, \theta_2^l)$ is sampled from the distribution

$$\begin{aligned} g &= p(\phi, \varphi, \theta_1|y_1)p(\theta_2|\phi) \\ &= p(\phi, \varphi)p(\theta_1|\phi)p(y_1|\varphi, \theta_1)p(\theta_2|\phi). \end{aligned}$$

   The goal is to sample from the full posterior distribution,

$$\begin{aligned} p &= p(\phi, \varphi, \theta_1, \theta_2|y) \\ &= p(\phi, \varphi)p(\theta_1|\phi)p(\theta_2|\phi)p(y_1|\varphi, \theta_1)p(y_2|\varphi, \theta_2). \end{aligned}$$

   For each l, we can compute the importance ratio $p/g = p(y_2|\theta_2^l)$.

4. Generalizing method 2 above, we can use a combination of importance resampling and birth-death process (Gilks and Berzuini, 2001). This approach starts with one subset, fitting the model to this subset and obtaining a sample set from the posterior distribution for this subset, and then generates a new sample set for the model with the next subset by combining an importance resampling step and a birth-death step. The processs continues until including all subsets.

2.4.1 Using the normal approximation

Here we discuss the first approach above by considering the normal model (8). We randomly divide the data set y into K parts $y^{(1)}, \dots, y^{(K)}$ using cluster sampling to keep the same hierarchical structure. If the sample size for each parts is very large, the marginal posterior distributions $p(\psi|y^{(1)})$, ... , $p(\psi|y^{(K)})$ are each approximately normal and so is $p(\psi|y)$ (see Le Cam and Yang, 1990, Gelman et al., 1995, and Shen and Wasserman, 2000). For $k = 1, \dots, K$, we compute $p(\psi, \theta_k|y^{(k)})$ and obtain posterior simulations, and then use these simulations to obtain a normal approximation $p(\psi|y^{(k)}) \approx \mathrm{N}(\hat{\psi}_k, V_{\psi_k})$. Combining these approximation gives $p(\psi|y) \approx \mathrm{N}(\hat{\psi}, V_\psi)$, where

$$\hat{\psi} = \left(\sum_{k=1}^{K} V_{\psi_k}^{-1}\right)^{-1} \left(\sum_{k=1}^{K} V_{\psi_k}^{-1}\hat{\psi}_k\right) \quad \text{and} \quad V_\psi = \left(\sum_{k=1}^{K} V_{\psi_k}^{-1}\right)^{-1}.$$

Then, if desired, one can use $\mathrm{N}(\hat{\psi}, V_\psi)$ as a prior distribution and obtain posterior inferences for the θ_k's. In fact, this whole procedure could be used as a starting point for the Gibbs sampler. We use this approximate approach to combine the separate simulations from the separate fit in the examples in the next Section 3.

2.4.2 Importance sampling

It is possible to improve the above combined inference using importance sampling. We consider two possible approaches. For each subset k, we have inferences from (ψ, θ_k). In the first approach, we work with the marginal distribution of ψ. We may do this by sampling L (say 10,000) points from $\mathrm{N}(\hat{\psi}, V_\psi)$ and then computing the importance ratio $p(\psi|y)/\mathrm{N}(\psi|\hat{\psi}, V_\psi)$. But we cannot in general do that because we cannot in general compute $p(\psi|y)$. In the second approach, we augment each draw with draws θ_1, ... ,θ_K constructed from the K fitted models and work with the joint distribution of $(\psi, \theta_1, \dots, \theta_K)$. As before we will sample L points from $p(\psi|y)$ and then for each of these simulations ψ^l, we sample θ_1, ..., θ_K from $\mathrm{N}(A_1 + B_1\psi^l, V_1)$, ..., $\mathrm{N}(A_K + B_K\psi^l, V_K)$, respectively.

Now the problem is how to determine all these A's, B's, and V's. For each k, we can use simulations of (ψ, θ_k) to obtain a normal approximation,

$$\mathrm{N}\left(\left(\begin{array}{c} \hat{\psi} \\ \hat{\theta}_k \end{array}\right), \left(\begin{array}{cc} V_\psi & V_{\psi\theta_k} \\ V_{\psi\theta_k}^T & V_{\theta_k} \end{array}\right)\right),$$

and then use the multivariate normal distribution (e.g., formula (A.1) from Gelman et al., 1995) to get

$$\theta_k|\psi, y^{(k)} \approx \mathrm{N}(A_k + B_k\psi, V_k).$$

This is a reexpression of the normal distribution of θ_k which gives us all the A's, B's, and V's. For each simulation draw l, we can compute the joint posterior density $p(\psi^l, \theta_1^l, \dots, \theta_K^l|y)$ and the approximation

$$g(\psi^l, \theta_1^l, \dots, \theta_K^l) \approx \mathrm{N}(\psi^l|\hat{\psi}, V_\psi) \prod_{k=1}^{K} \mathrm{N}(\theta_k^l|A_k + B_k\psi^l, V_k),$$

and then compute the importance ratio p/g.

2.4.3 Numerical examples

We use two simple examples to show how importance sampling can improve inferences. Our first example is a random effects model for effects of an educational testing experiment in 8 schools, described by Rubin (1981) and Gelman et al. (1995). This is a small example but the nonnormality of its posterior distribution makes it interesting. For each school $j = 1, ..., 8$, an experiment was performed to estimate the treatment effect α_j in that school; a regression analysis applied to the data from that school alone yielded an estimate and stand error, which we label y_j and σ_j. The sample size within each school is large enough that it was reasonable to model $y_j|\alpha_j \sim \mathrm{N}(\alpha_j, \sigma_j^2)$, with σ_j known. The parameters from the 8 schools are modeled as normally distributed: $\alpha_j \sim$ iid $\mathrm{N}(\mu, \tau^2)$, with a noninformative uniform hyperprior density, $p(\mu, \tau) \propto 1$. Interest lies in the individual school parameters α_j and also in the hyperparameters (μ, τ). Suppose we divide the dataset into $K = 2$ parts. In this example, the posterior distribution of τ is far from normal. Figure 1 shows the comparison of the inferences from the full dataset and each subsets, the combined inferences by normal approximation and their improvement by importance sampling.

Our second example is similar to the first one, but with more groups. In this example, we randomly select 200 retail dealers of a newspaper company. For each dealer $j = 1, \dots, 200$, we have enough historical data to estimate their daily sales levels (copies per day) and standard errors, which we label y_j and σ_j. This is a simpler version of the full model for this problem described in Section 3.1. We set up the same random effects model for this example but in this example, the normal approximation fits well,and so importance sampling improves the combined inference only slightly. Figure 2 displays the separate and combined inferences.

3 Applied examples

3.1 Newspaper marketing

Data and model

In this example, we consider a simple two-level model with no covariates. A newspaper company has a dispersed retail network to support its retail business and is interested in the demand distributions for each retailer so that it can plan its future business. More specifically, the company is interested in simultaneously addressing two objectives: (1) maximization of its profits and (2) maximization of the sales volume. The demand distribution for each retailer plays critical role in addressing the two objectives. We work with historical daily sales data for several months for each retailer. For simplification, we only consider sales on Tuesdays. We introduce the following notation:

Y_{jt} copies sold for retailer j at time t

D_{jt} copies allocated to retailer j at time t

Z_{jt} copies demanded for retailer j at time t (unobservable if $Z_{jt} > D_{jt}$, that is, sellout).

For any j and t, we have

$$Z_{jt} = \min(Y_{jt}, D_{jt})$$

In addition, we let $y_{jt} = \log(Y_{jt})$, $d_{jt} = \log(D_{jt})$ and $z_{jt} = \log(Z_{jt})$. Thus, we still have $z_{jt} = \min(y_{jt}, d_{jt})$. We set up a Bayesian hierarchical model for the logarithm of demand, z_{jt}. We consider the model,

$$z_{jt} = \theta_j + \epsilon_{jt} \qquad \text{for } j = 1, \ldots, J \text{ and } t = 1, \ldots, T \tag{10}$$

In our general notation, $i = (j, t)$ and $n = JT$.

The parameter in the model are defined as follows:

- θ_j is the normal sales of retailer j; we assign it a normal population distribution with mean θ and standard deviation τ.
- ϵ_{jt} are are independent error terms assumed normally distributed with mean 0 and standard deviation σ.

In this model, log sales are right-censored. In order to handle the censored observations, we introduce censoring indicators I_{jt}:

$$I_{jt} = \begin{cases} 1 & \text{if } z_{jt} > y_{jt} \\ 0 & \text{otherwise.} \end{cases}$$

Thus, in this model, the data information we have are pairs, (y_{jt}, I_{jt}).

Equation (10) implies the following likelihood for the sales data:

$$p(y, I|\theta, \mu, \sigma) = \prod_{j,t:I_{jt}=0} \mathrm{N}(y_{jt}|\theta_j, \sigma^2) \prod_{j,t:I_{jt}=1} (1 - \Phi(\frac{y_{jt} - \theta_j}{\sigma})),$$

where Φ is the normal cumulative distribution function.

The prior distribution for the retailer effects is given by,

$$p(\theta|\mu) = \prod_{j=1}^{J} \mathrm{N}(\theta_j|\mu, \tau^2),$$

and we assume μ, τ and σ have a noninformative prior distribution,

$$p(\mu, \tau, \sigma) \propto 1.$$

Finally, the joint posterior density is proportional to the product of all the above pieces:

$$p(\theta, \mu, \tau, \sigma|y, I) \propto \prod_{j,t:I_{jt}=0} \mathrm{N}(y_{jt}|\theta_j, \sigma^2) \prod_{j,t:I_{jt}=1} (1 - \Phi(\frac{y_{jt} - \theta_j}{\sigma})) \prod_{j=1}^{J} \mathrm{N}(\theta_j|\mu, \tau^2). \tag{11}$$

Implementation

In order to simulate density (11), we consider z_{jt} (for those j and t such that $I_{jt} = 1$) as augmented parameters (see van Dyk and Meng, 2001). We indicate these augmented parameters by θ. In this way, it is easy to set up a Gibbs sampler in two steps. First, given z_{jt}, (10) is a normal linear multilevel model, and all its parameters can be updated using a Gibbs sampler. The linear parameters θ, μ have a joint normal conditional distribution, and the variance parameters τ^2 and σ^2 have independent inverse-χ^2 distributions. The Gibbs sampler thus alternates between these two blocks of parameters. Second, given all the model parameters, for any j and t, $z_{jt} = y_{jt}$ if $I_{jt} = 0$, otherwise, z_{jt} has independent normal conditional distribution with mean θ_j and standard deviation σ and truncated at y_{jt}, from which we can draw a sample until it is greater than y_{jt}. We alternate these two steps which can be improved including a covariance adjustment step (see, Liu, 2003). Two chains reached approximate convergence in the sense that $\hat{R}$ is less than 1.2 (see Gelman and Rubin, 1992) after 400 iterations. All computations were performed in statistical language S-Plus, using the `apply, colSums, rowSums,` etc., functions so that there was no internal looping in the Gibbs sampler updating.

In performing our computation strategy, we randomly divide the $J = 4787$ retailers into 2, 4, 6, and 8 subsets of equal size in this example, respectively.

Simulation results

Figure 3 shows the estimates of overall mean μ, the standard deviation of retailer's effects τ on Tuesdays for the newspaper company, and the standard deviation of residuals σ in the model of single copy sales, respectively. F represents estimates from fitting the whole dataset model, S_k from fitting the k^{th} subset model and C from combining the estimates from the subset models. The dots, thick lines, and thin lines display the median, 50%, and 95% posterior intervals. The first row shows the posterior inferences from 2 random subsets of equal size, the second from 4 random subsets, the third row from 6 random subsets and the last row from 8 random subsets. Because of sampling error, the estimates from each subset model vary across the subsets. But the estimates from the full data set model and the combined estimates are very close for all the partitions. Therefore we can obtain an effective consensus posterior inference in this example using our computational strategy.

Figure 4 compares the combined simulation results from the posterior distributions of the hyperparameters μ, τ and σ in the model of single copy sales on Tuesdays for the newspaper company, respectively. C_1 represents the estimates from fitting model (10) with the whole dataset and C_k represents the combined estimates from fitting model (10) with k random subsets.

Saving memory requirements and computation time

If the dataset is too large, the traditional computational methods might not be implemented in small computers because small computers can not allocate the dynamic memory requested by the computation program. However, in our strategy, we just fit all the subset models once at a time, and each subset model is much smaller than the whole model in the sense that the numbers of parameters and observations in each subset model are only a small proportion of those in the whole model. Thus we will never use up the computer memory if we randomly divide the whole data into enough subsets. Another advantage of our computation strategy is the potential saving of computation time. Figure 5 shows the total computation times per iteration of fitting the full dataset model and all subset models with dividing the full dataset into different number of subsets. If the number of subsets is not greater than 16 in this example, our computation strategy saves computation time. Dividing the data set into 8 subsets has minimum computation time.

3.2 Climate modeling

Data and model

In the second example, we consider the forecast skill of global circulation models (GCM) based on Africa precipitation data in the fall (October, November, December) season. The models divide the globe into a grid, on which Africa covers 527 boxes, and we have 41 observed precipitation values (between 1950 and 1990) for each box in a given season. Here we consider three GCM models, each of which gives us 10 predicted precipitation values for each box in a given season (Mason et al., 1999 and Rajagopalan et al., 2000). In this example, we use y_{jt} and $x_{jt}^{m,k}$, represent observed and k^{th} predicted precipitation anomaly, respectively, for m^{th} GCM ensemble in box j and at time t. We use as predictors the average values,

$$\bar{x}_{jt}^{m} = \frac{1}{10}\sum_{k=1}^{10} y_{jt}^{m,k}.$$

In this example, we set up the following multilevel model using y_{jt} as response variable and $\bar{x}_{jt}^{m}$ as predictors:

$$y_{jt} = \alpha_j + \delta_t + \sum_{m=1}^{M} \beta_j^m \bar{x}_{jt}^{m} + \gamma_j n_t + \xi_j \eta_t + \epsilon_{jt}, \tag{12}$$

The parameters in the model are defined as follows:

- δ_t is an offset for time t; we assign it a normal population distribution with mean 0 and standard deviation σ_δ.

- α_j is an offset for location j; we assign it a normal population distribution with mean μ_α and standard deviation σ_α.

- β_j^m is the coefficient of ensemble forecast m in location j; we assign it a normal population distribution with mean μ_m and standard deviation σ_m.

- γ_j is the effect of Nino3 (a predictor for the effect of El nino) in location j; we assign it a normal population distribution with mean μ_γ and standard deviation σ_γ.

- ξ_j is the effect of time indicator η (1 for the first 20 observations, 0 for the last 21 observations) in location j; we assign it a normal population distribution with mean μ_ξ and standard deviation σ_ξ.
- ϵ_{it}'s are independent error terms assumed normally distributed with mean 0 and standard deviation σ. We can model the error terms ϵ_{jt}'s as independent because any dependence that would have occurred between them is captured by the variables α_j, y_{jt}^m, n_t, δ_t and η_t.

Likelihood

Model (12) has the following likelihood:

$$p(y|\theta) = \prod_{j=1}^{527} \prod_{t=1}^{41} \mathrm{N}(y_{jt}|\gamma_j + \delta_t + \sum_{m=1}^{3} \beta_j^m \bar{x}_{jt}^m + \gamma_j n_t + \xi_j \eta_t, \sigma^2)$$

where θ indicates all the parameters in the model.

Prior distribution

The prior distributions for the location offsets, the time offsets, the forecast model effects, nino3 effects, and first-twenty-year effects are given by,

$$\begin{aligned}
p(\alpha|\mu_\alpha, \sigma_\alpha) &= \prod_{j=1}^{527} \mathrm{N}(\alpha_j|\mu_\alpha, \sigma_\alpha^2) \\
p(\delta|\sigma_\delta) &= \prod_{t=1}^{41} \mathrm{N}(\delta_t|0, \sigma_\delta^2) \\
p(\beta|\vec{\mu}_\beta, \vec{\sigma}_\beta) &= \prod_{j=1}^{527} \prod_{m=1}^{3} \mathrm{N}(\beta_j^m|\mu_{\beta_j}, \sigma_{\beta_j}^2) \\
p(\gamma|\mu_\gamma, \sigma_\gamma) &= \prod_{j=1}^{527} \mathrm{N}(\gamma_j|\mu_\gamma, \sigma_\gamma^2) \\
p(\xi|\mu_\xi, \sigma_\xi) &= \prod_{j=1}^{527} \mathrm{N}(\xi_j|\mu_\xi, \sigma_\xi^2)
\end{aligned}$$

where $\vec{\mu_\beta} = (\mu_{\beta_1}, \mu_{\beta_2}, \mu_{\beta_3})$ and $\vec{\sigma_\beta} = (\sigma_{\beta_1}, \sigma_{\beta_2}, \sigma_{\beta_3})$. We next assign noninformative prior distributions to the remaining parameters in the model:

$$p(\mu_\alpha, \vec{\mu}_\beta, \mu_\gamma, \mu_\xi, \sigma_\alpha, \sigma_\delta, \vec{\sigma}_\beta, \sigma_\gamma, \sigma_\xi, \sigma) \propto 1.$$

Posterior distribution

The joint posterior distribution for the model parameters is:

$$
\begin{aligned}
p(\theta|y) \quad \propto \quad & \prod_{j=1}^{527}\prod_{t=1}^{41} \mathrm{N}(y_{jt}|\alpha_j + \delta_t + \sum_{m=1}^{3} \beta_j^m \bar{x}_{jt}^m + \gamma_j n_t + \xi_j \eta_t, \sigma^2) \prod_{j=1}^{527} \mathrm{N}(\alpha_j|\mu_\alpha, \sigma_\alpha^2) \\
& \times \prod_{t=1}^{41} \mathrm{N}(\delta_t|0, \sigma_\delta^2) \prod_{j=1}^{527}\prod_{m=1}^{3} \mathrm{N}(\beta_j^m|\beta_0^m, \sigma_m^2) \\
& \times \prod_{j=1}^{527} \mathrm{N}(\gamma_j|\mu_\gamma, \sigma_\gamma^2) \prod_{j=1}^{527} \mathrm{N}(\xi_j|\mu_\xi, \sigma_\xi^2).
\end{aligned} \tag{13}
$$

Implementation

In this example, we randomly divide the $J = 527$ locations into 2, 3, 4, and 5 subsets of equal size, respectively, and then obtain simulations for each separate model. The density (13) is a normal linear mutilevel model and so it is easy to update all its parameters using the Gibbs sampler. In order to make full use of zeros in the design matrix to increase computation speed, we use a scalar updating algorithm rotating the vector of linear parameters to achieve approximate independence.

Simulation results

In this example, we have 55 hyperparameters to estimate: 8 variance components, 6 overall means, and 41 δ's. Figure 6 gives a summary of posterior inference for some of the hyperparameters from model (12). F represents the results, estimated based on the whole dataset, S_k the results, estimated based on the k^{th} ($k = 1, 2, 3$) subset data (here we divide the whole dataset into three subsets.) and C the combined inference results. Again, in this example the estimates from each subset model vary across the subsets because of sampling error. But the estimates from the full data set model and the combined estimates are very close. Thus our computation strategy can give same posterior inference as the traditional computation methods.

Figure 7 gives a summary of posterior inference for some of the hyperparameters from model (12). C_1 represents the results, estimated based on the whole dataset, and C_k the combined results, estimated based on k ($k = 2, 3, 4, 5$) random subsets. Again, the estimates from the full dataset model and the combined estimates are very close for all the different partisans and thus our computation strategy can give same posterior inference as the traditional computation methods for this example. However, the differences between the estimates with the full dataset and the combined estimates increase with the number of subsets.

Figure 8 shows the posterior inferences for the δ's. Again, the estimates from the full dataset model and the combined estimates are very close for all the different partitions and thus our computation strategy can give the almost same posterior inference as the traditional computation methods for this example. However, the differences between the estimates with the full dataset and the combined estimates increase with the number of subsets.

4 Discussion

Problems involving computation time and storage space often arise for large models with large datasets. We do not want to be forced to use unrealistically simple models or to restrict our data, just because of computational problems. However, there is good news: with lots of data, there is lots of replication, which implies that inferences from a random subset of the data should be representative of the whole. Our solution is to divide data into subsets using random sampling (cluster sampling) rather than simply taking the data sequentially. Our approach has potential for parallel implementation (although we have not implemented it this way). A challenge is then to combine the separate inferences. We have success with a normal approximation, with the potential for correction using importance sampling or the Metropolis algorithm. Open problems include: (1) rigorous theoretical justification (based on the Central Limit Theorem applied to the log-likelihood), (2) choice of the number of subsets K, and (3) the best method for combining the subset inferences. In addition, for huge problems, perhaps it would be effective to take a few subsets and stop if inferences are good enough.

References

Besag, J., P., Green, P., Higdon, D., and Mengersen, K. (1995) Bayesian computation and stochastic systems (with discussion). *Statistical Science* **10**, 3–41.

Bickel, P., and Doksum, K. (1976) *Mathematical Statistics: Basic Ideas and Selected Topics.* San Francisco: Holden-Day.

Carlin, B. P., and Louis, T. A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis.* London: Chapman and Hall, 2nd edition.

Chopin, N. (2002). A sequential particle filter method for static models. *Biometrika* **89**, 539–552.

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

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

Gelman, A., and Rubin, D. B. (1992). Inference from iterative simulation using multiple sequences (with discussion). *Statistical Science* **7**, 457–511.

Gilks, W. R., Richardson, S., and Spiegelhalter, D., eds. (1996). *Practical Markov Chain Monte Carlo.* London: Chapman and Hall.

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

Goldstein, H. (1995). *Multilevel Statistical Models.* London: Edward Arnold.

Golub, G. H., and van Loan, C. F. (1983). *Matrix Computations.* Baltimore, Maryland: Johns

Hopkins University Press.

Le Cam, L., and Yang, G. L. (1990). *Asymptotics in Statistics: Some Basic Concepts.* New York: Springer-Verlag.

Lindley, D. V., and Smith, A. F. M. (1972). Bayes estimates for the linear model. *Journal of the Royal Statistical Society B* **34**, 1–41.

Liu, C. (2003). Alternating subspace-spanning resampling to accelerate Markov chain Monte Carlo simulation. *Journal of the America Statistical Association.*

Longford, N. (1993). *Random Coefficient Models.* Oxford: Clarendon Press.

Lohr, S. (1999). *Sampling: Design and Analysis.* Pacific Grove, Calif.: Duxbury.

McCullagh, P., and Nelder, J. A. (1989). *Generalized Linear Models*, second edition. London: Chapman and Hall.

Meng, X. L., and van Dyk, D. (1997). The EM algorithm—an old folk-song sung to a fast new tune (with discussion). *Journal of the Royal Statistical Society B* **59**, 511–567.

Ridgeway, G., and Madigan, D. (2002). A sequential Monte Carlo method for Bayesian analysis of massive datasets. *Journal of Data Mining and Knowledge Discovery.*

Robinson, G. K. (1991). That BLUP is a good thing: the estimation of random effects (with discussion). *Statistical Science* **6**, 15–51.

Rubin, D. B. (1981). Estimation in parallel randomized experiments. *Journal of Educational Statistics* **6**, 377–401.

Shen, X. and Wasserman, L. (2000). Rates of convergence of posterior distributions. *Annals of Statistics* **29**, 687–714.

Van Dyk, D. A., and Meng, X. L. (2001). The art of data augmentation (with discussion). *Journal of Computational and Graphical Statistics* **10**, 1–111.

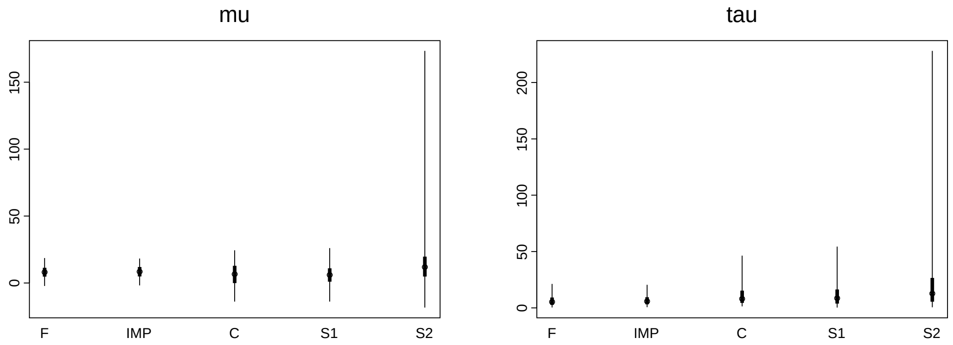

Figure 1: Estimates of μ and τ (posterior medians, 50% intervals, and 95% intervals) in the eight schools model. F represents estimates from fitting the model to the full dataset model, S_k estimates from fitting the model to the k^{th} subset ($k = 1, 2$), C estimates from combining the estimates from the subset models by normal approximation, "IMP" from importance sampling.

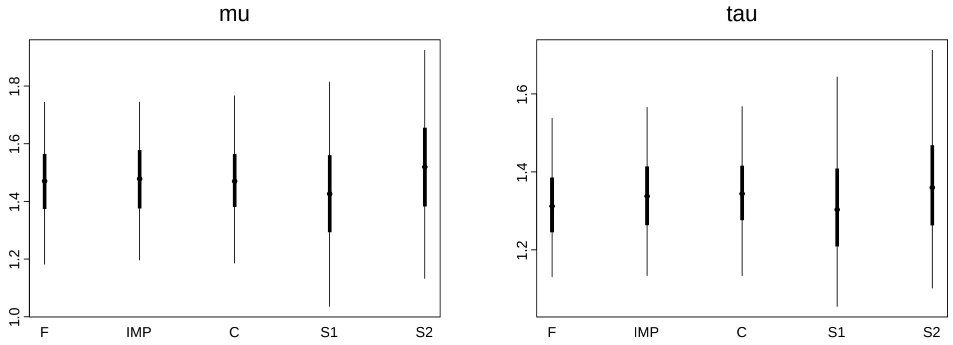

Figure 2: Estimates of μ and τ (posterior medians, 50% intervals, and 95% intervals) in the newspaper company sales model. F represents estimates from fitting the model to the full dataset model, S_k estimates from fitting the model to the k^{th} subset ($k = 1, 2$), C estimates from combining the estimates from the subset models by normal approximation, "IMP" from importance sampling.

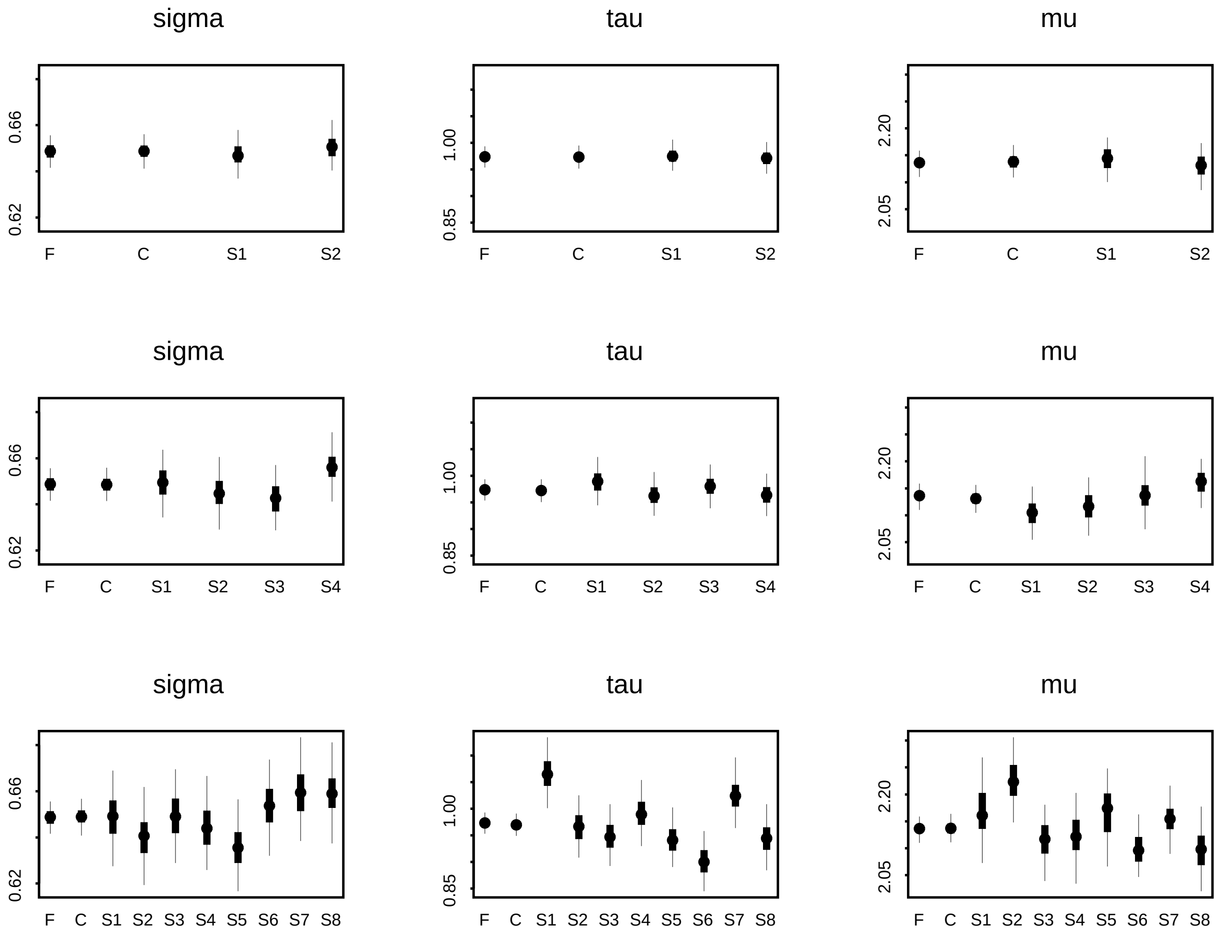

Figure 3: Estimates of overall mean μ, standard deviation of retailers' effects τ, and standard deviation of residuals σ in the model of single copy sales on Tuesdays for the newspaper company, respectively. F represents estimates from fitting the model to the full dataset, S_k from fitting to the k^{th} subset, and C from combining the estimates from the subset models. The dots and segments show medians, 50% intervals, and 95% intervals. The first row are the posterior inferences estimated from the data partitioned into 2 random subsets of equal size, the second row from 4 random subsets, the third row from 6 random subsets and the last row from 8 random subsets.

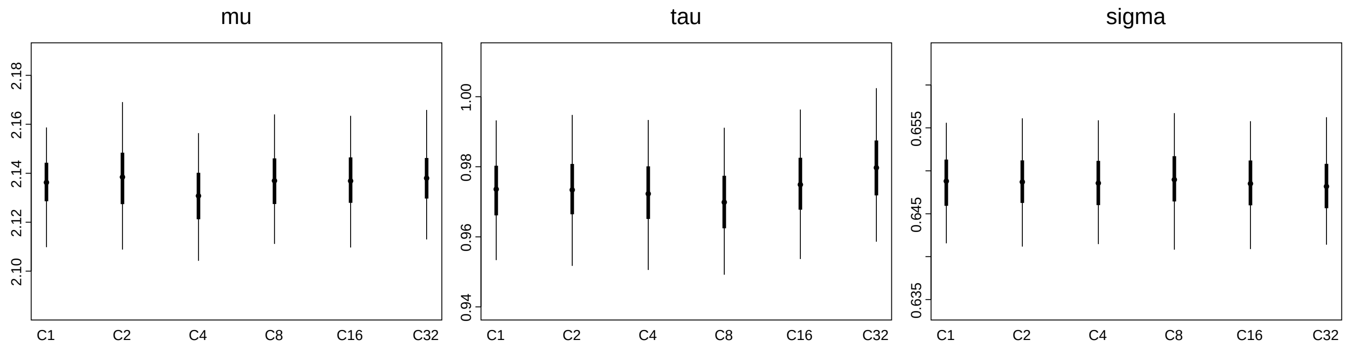

Figure 4: Estimates of μ, τ, and σ (posterior medians, 50% intervals, and 95% intervals) in the model of single copy sales on Tuesdays for the newspaper company. C_1 represents the estimates from fitting model (10) with the full data set and C_k represents the combined estimates from fitting model (10) with k random subsets ($k = 2, 4, 8, 16, 32$).

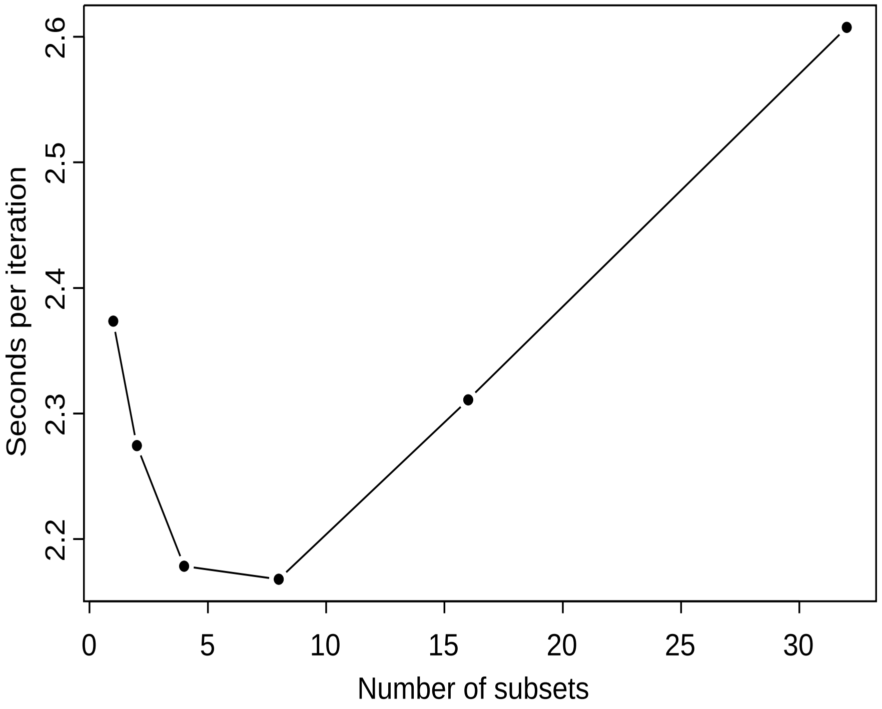

Figure 5: Total computation time per iteration of fitting the model to the full dataset and all the subset models with dividing the full dataset into different number of subsets. The curve drops at first because of the efficiency in working with smaller datasets and then increases because of fixed costs in the computation.

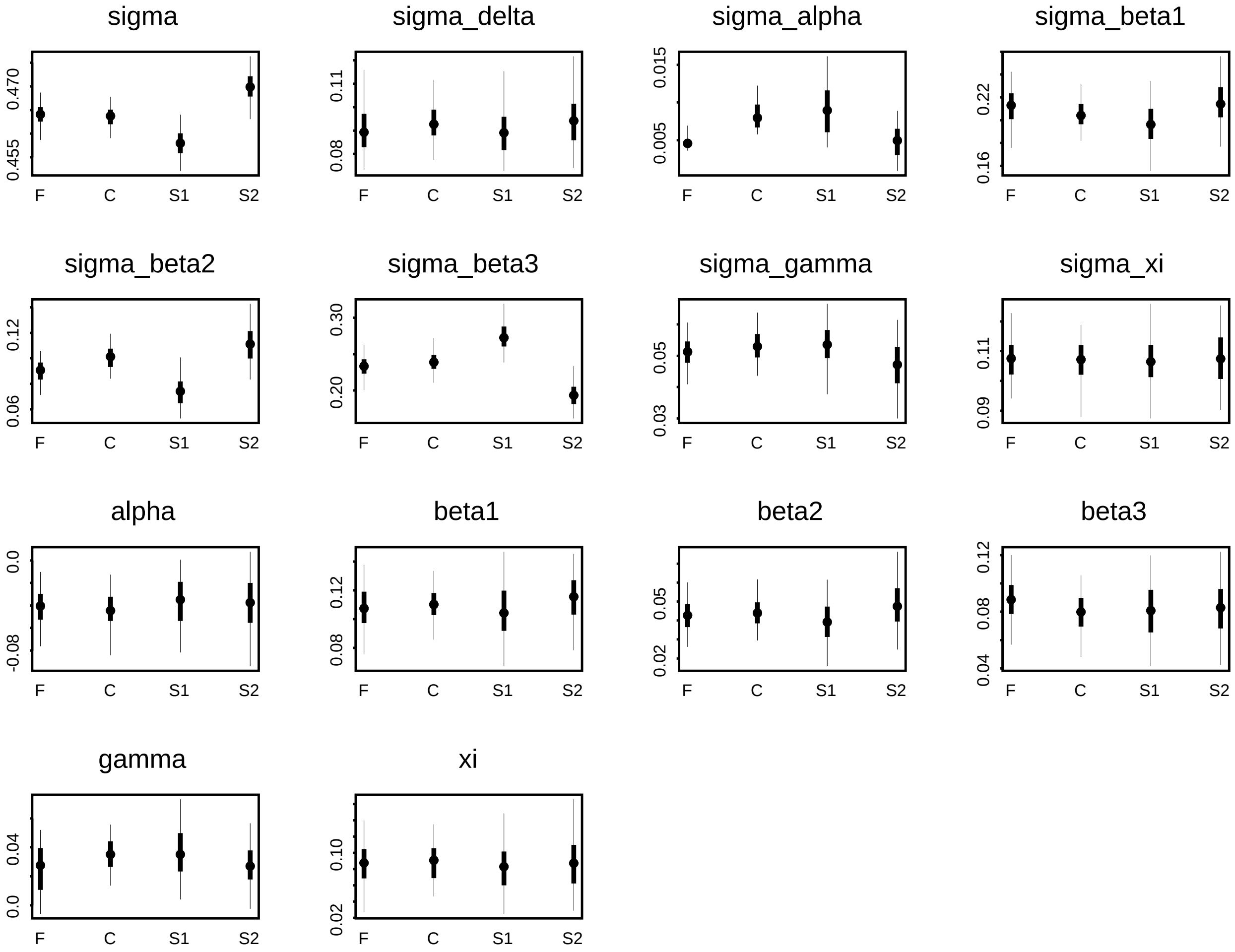

Figure 6: Estimates (posterior medians, 50% intervals, and 95% intervals) of the hyperparameters in the Africa climate model. F represents estimates from fitting the model to the full dataset model, S_k estimates from fitting the model to the k^{th} subset ($k = 1, 2, 3$) and C estimates from combining the estimates from the subset models.

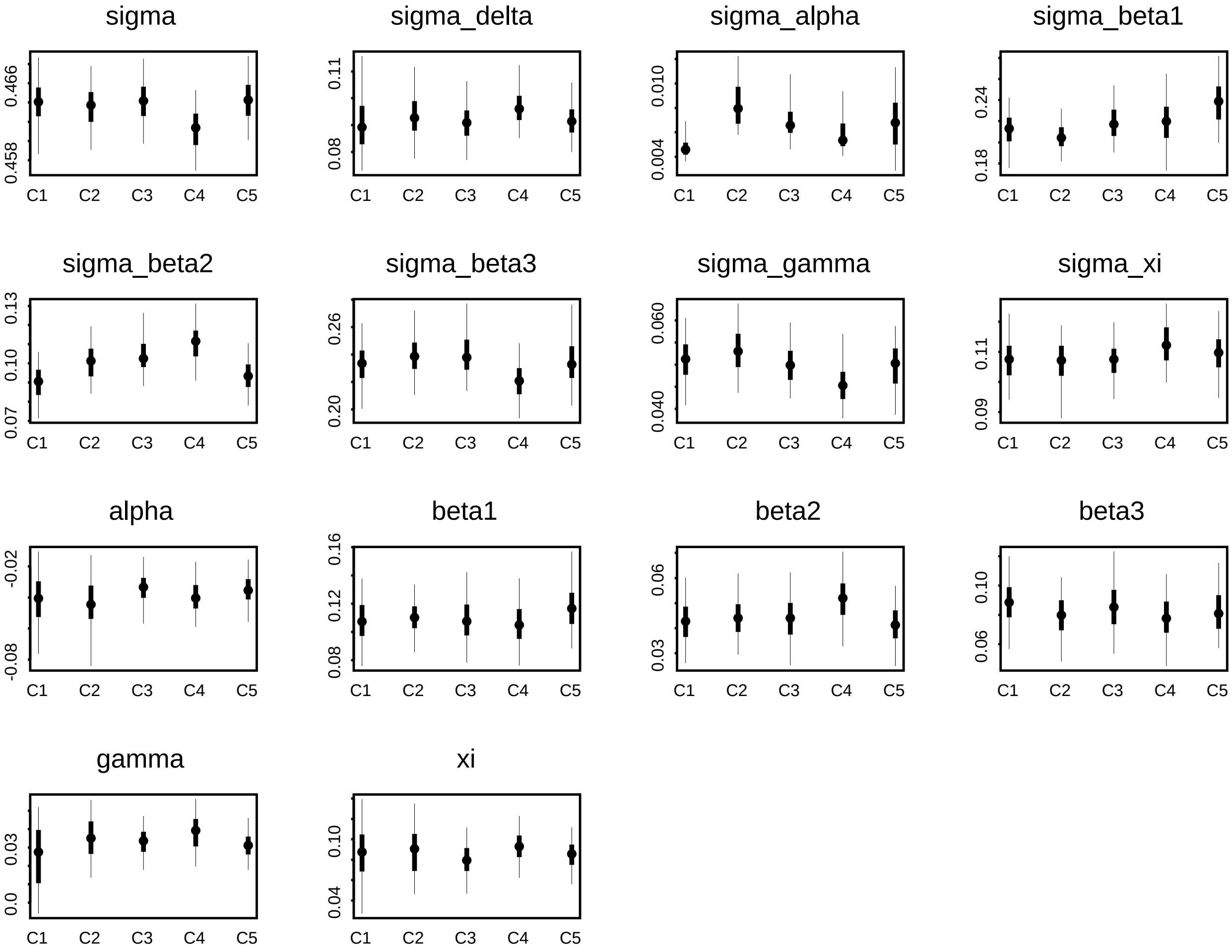

Figure 7: Estimates (posterior medians, 50% intervals, and 95% intervals) of the hyperparameters in the Africa climate model. C_1 represents the estimates from fitting model (12) with the full data set and C_k represents the combined estimates from fitting model (12) with k random subsets $(k = 2, 3, 4, 5)$.

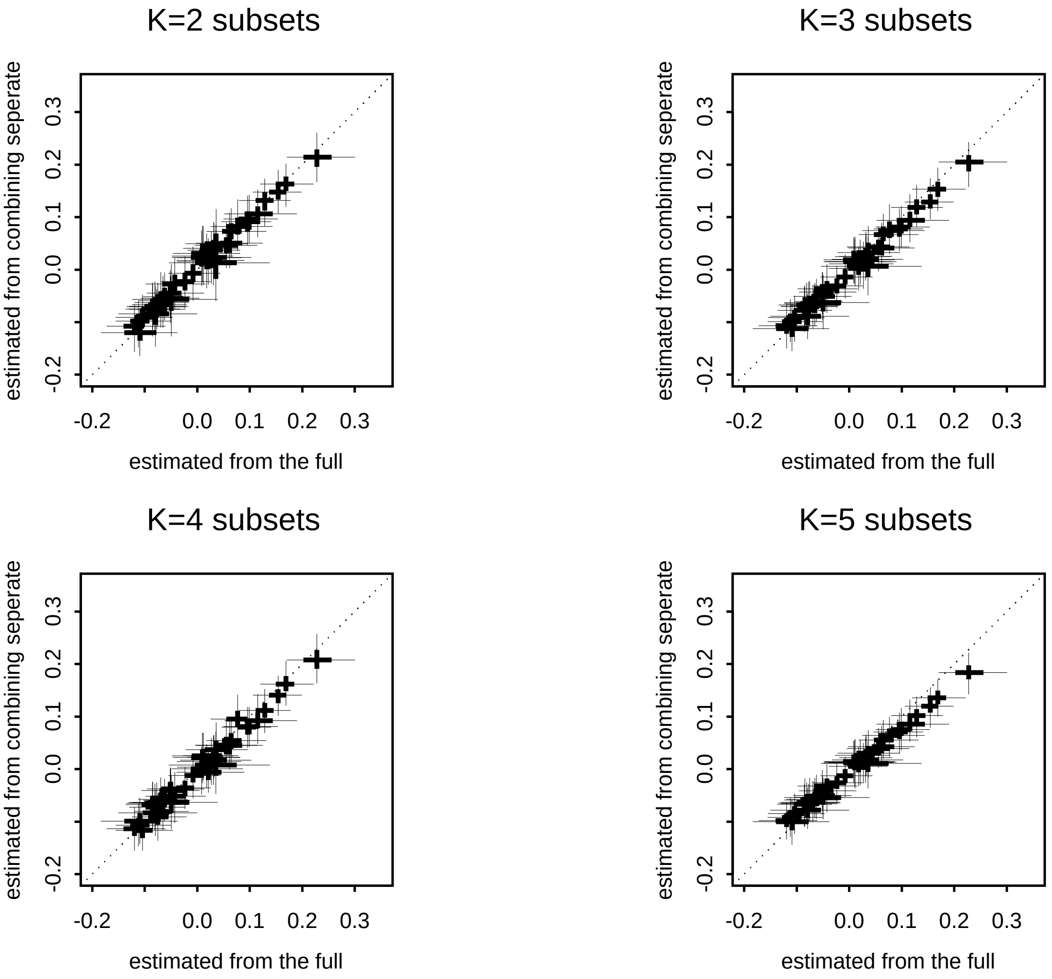

Figure 8: Estimates (posterior medians, 50% intervals, and 95% intervals) of δ's in the Africa climate model.