Transformed and parameter-expanded Gibbs samplers for multilevel linear and generalized linear models

Andrew Gelman* Zaiying Huang† David A. van Dyk‡

W. John Boscardin§

October 2, 2003

Abstract

Hierarchical linear and generalized linear models can be computed using Gibbs samplers and Metropolis algorithms; however, such models often have many parameters and convergence of the most natural Gibbs and Metropolis algorithms can be slow.

We examine two reparametrizations for multilevel linear and generalized linear models. First, we consider a simple rotation of the regression coefficients based on an estimate of their posterior covariance matrix. This leads to a Gibbs algorithm based on updating the transformed parameters one at a time or a Metropolis algorithm with vector jumps; either of these algorithms can perform much better (in terms of total CPU time) than the two standard algorithms: one-at-a-time updating of untransformed parameters or vector updating using a linear regression at each step.

Second, we consider a parameter expansion using working parameters, as was done for the EM algorithm by Meng and van Dyk (1997) and Liu, Rubin, and Wu (1998), which, compared to the usual algorithms, is much less susceptible to getting stuck near zero values of the variance parameters. The parameter expansion also has an interesting interpretation as a true model expansion as well as a computational tool.

We present an innovative evaluation of the algorithms in terms of their speed in getting unstuck from remote areas of parameter space, along with some more standard evaluation of computation and convergence speeds. We illustrate our methods with examples from our applied work. Our goal are to develop a fast and reliable computer program so that one can fit a hierarchical linear model as easily as one can now fit a nonhierachical model, and to increase understanding of Gibbs samplers for hierarchical models in general.

Keywords: Bayesian computation, Markov chain Monte Carlo, multilevel modeling, mixed effects models, PX-EM algorithm, random effects regression, working parameters.

1 Introduction

1.1 Background

Gaussian hierarchical linear models (also called multilevel models, random effects regressions, and mixed effects models) are an important tool in many areas of statistics (see, for example, Robinson,

*Department of Statistics and Department of Political Science, Columbia University, New York, gelman@stat.columbia.edu, http://www.stat.columbia.edu/~gelman/

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

‡Department of Statistics, University of California, Irvine

§Department of Statistics, University of California, Los Angeles

1991, Longford, 1993, and Goldstein, 1995, for recent reviews). In recent years, there has been increasing interest in the Bayesian formulation (Lindley and Smith, 1972), which allows one to account for the uncertainty in the variance parameters, a feature that is particularly important when the hierarchical variances are hard to estimate or to distinguish from zero (see Carlin and Louis, 2000, and Gelman et al., 1995, for discussions and examples). Modern Bayesian inference generally entails computing simulation draws of the parameters from the posterior distribution. 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 Gelman et al., 1995, and Gilks, Richardson, and Spiegelhalter, 1996).

Unfortunately, the standard algorithms for computing posterior simulations from hierarchical models can have convergence problems, especially when hierarchical variance components are near zero. We consider improvements based on transformations and parameter expansion, evaluating the various simulation algorithms from the standpoint of computation time per iteration, number of iterations required for approximate convergence, and the autocorrelation in the resulting chains. *Auxiliary variables* (Besag and Green, 1993) and *data augmentation* (Tanner and Wong, 1987) can simplify and reduce dependence in the conditional specifications in a joint distribution. *Parameter expansion*, on the other hand, aims to introduce working parameters which add structure to the conditional distribution used in the steps of the EM algorithm or draws of the Gibbs sampler. The method was introduced by Liu, Rubin, and Wu (1998) for the EM algorithm and extended to the data augmentation algorithm by Meng and van Dyk (1999) and Liu and Wu (1999). Meng and van Dyk (1999) introduce the terms *conditional* and *marginal* augnentation to indicate whether we condition on or average over the working parameter in the iteration; conditional augmentation indexes a family of transformations by the working parameters. Here we compare and extend the new algorithms to hierarchical linear and generalized linear models (see also Meng and van Dyk, 1998, Liu, Wu and Rubin, 1998, van Dyk, 2000, and van Dyk and Meng, 2001).

This paper proceeds as follows. The rest of Section 1 reviews the basic hierarchical linear and generalized models along with standard Gibbs/Metropolis algorithms based on scalar or vector updating using linear regression computations. Section 2 and 3 present two improvements in computation: the scalar updating algorithm with rotation and the parameter-expanded hierarchical model. Section 4 presents theoretical arguments on the computation time per iteration and convergence times for several proposed Gibbs sampler implementations, including an innovative argument based on the speed of the algorithm in getting unstuck from remote areas of parameter space. Section 5 presents two examples, and Section 6 concludes with recommendations and ideas for further research.

1.2 Notation for hierarchical linear and generalized linear models

1.2.1 Hierarchical linear model

In setting up any mathematical notation, there is a tension between conciseness and generality. To keep the focus in this paper on the basic computational ideas, we work with a relatively simple form of the Gaussian hierarchical linear regression model with a vector of data y and a vector of coefficients β that are arranged in batches of random effects:

$$y \sim \mathrm{N}(X\beta, \Sigma_y) \tag{1}$$

$$\beta \sim \mathrm{N}(\beta_0, \Sigma_\beta). \tag{2}$$

Hierarchical linear models with more than two levels can be expressed in this caqnonical form by extending the vector β to include the higher-level linear parameters in the hierarchy and augmenting y and X appropriately (see Gelman et al., 1995, and Hodges, 1998). For simplicity, we assume the mean vector β_0 is known.

A key aspect of the model is the parameterization of the variance matrices, Σ_y and Σ_β. We shall assume that each Σ_z is diagonal with a vector of unknown elements $\sigma_z = (\sigma_{zk} : k = 1, \ldots, K_z)$, where we are using z as a floating subscript to stand for y or β:

$$\begin{aligned} \Sigma_y &= \mathrm{Diag}(W_y \sigma_y^2) \\ \Sigma_\beta &= \mathrm{Diag}(W_\beta \sigma_\beta^2), \end{aligned} \tag{3}$$

and W_y and W_β are indicator matrices of 0's and 1's, with exactly one 1 in each row. Thus, each component of σ_y and σ_β is the standard deviation assigned to a subset, or batch, of the components of y or β. The data are partitioned into K_y batches, the regression coefficients are partitioned into K_β batches, and the columns of W_y and W_β correspond to the batches.

1.2.2 Prior distributions on the variance parameters

We assign independent inverse-χ^2 prior distributions to the variance parameters:

$$\begin{aligned} \sigma_{yk}^2 &\sim \text{Inv-}\chi^2(\nu_{yk}, \sigma_{0yk}^2), \quad \text{for} \quad k = 1, \ldots, K_y \\ \sigma_{\beta k}^2 &\sim \text{Inv-}\chi^2(\nu_{\beta k}, \sigma_{0\beta k}^2), \quad \text{for} \quad k = 1, \ldots, K_\beta. \end{aligned} \tag{4}$$

The prior degrees of freedom ν and scale parameters σ_0 can either be preset or assigned a non-informative prior distribution. First, $p(\sigma_y, \sigma_\beta) \propto \Pi_{k=1}^{K_y} \sigma_y^{-1}$ corresponds to $(\nu_{yk} = 0, \sigma_{0yk} = 0)$ for $k = 1, \ldots, K_y$, and $(\nu_{\beta k} = -1, \sigma_{0\beta k} = 0)$ for $k = 1, \ldots, K_\beta$. Second, variance components with known values correspond to $\nu_{zk} = \infty$ and are, of course, not altered in any of the Gibbs sampler algorithms. Third, components of β with noninformative flat prior distributions are associated with variance components $\sigma_{\beta k}$ that are fixed at ∞. Fourth, components of β with fixed or known values are associated with variance components $\sigma_{\beta k} = 0$.

1.2.3 Hierarchical generalized linear model

We also consider generalized linear models that are identical to the above, except that (1) is replaced by

$$y \sim \text{glm}(X\beta) \quad \text{or} \quad y \sim \text{glm}(X\beta, \Sigma_y), \tag{5}$$

with the simpler form being used for models such as the binomial and Poisson with no free variance parameters. We use the general "glm" notation to include whatever probability model and link function are being used (McCullagh and Nelder, 1989). We illustrate in Section 5 with a hierarchical logistic regression.

1.2.4 Extensions of the models

The model can be extended in various ways, including nondiagonal variances (Gelman et al., 1995), multivariate clustering of regression parameters (Barnard, McCulloch, and Meng, 1997; Daniels and Kass, 1999), informative prior distributions on the regression coefficients, unequal variances (Boscardin and Gelman, 1996), robust models, and many other variants in the statistics and econometrics literature, perhaps most notably models for variable selection and model averaging (Raftery, 1996). Most of these extensions can be fit into our framework by adding additional parameters and thus additional steps in any Gibbs/Metropolis algorithms. Rather than try to present an ideal algorithm to handle all contingencies, we try here to understand what works with an important and nontrivial basic family of models.

1.3 Standard Gibbs algorithms for the hierarchical normal model

There are two standard Gibbs sampler algorithms for the hierarchical linear model. In both algorithms, the variances parameters (σ_j, σ_β) are drawn from their joint distribution given β and y. The algorithms differ in their sampling of β: in one version the components of β are drawn one at a time, and in the other version the vector β is drawn all at once. When we speak of an *iteration* of either algorithm, we refer to a single update of all of the free parameters in the model.

The *all-at-once* algorithm draws the vector β from $p(\beta|\sigma_y^2, \sigma_\beta^2, y)$ by running the regression of y_* on X_* with variance matrix Σ_*, where

$$y_* = \begin{pmatrix} y \\ \beta_0 \end{pmatrix}, \qquad X_* = \begin{pmatrix} X \\ I \end{pmatrix}, \qquad \text{and} \qquad \Sigma_* = \begin{pmatrix} \Sigma_y & 0 \\ 0 & \Sigma_\beta \end{pmatrix} \tag{6}$$

combine the data and prior information (Gelman et al., 1995; see also Dempster, Rubin, and Tsutakawa, 1991, and Hodges and Carlin, 1999). The regression computation yields an estimate and a variance matrix, which we label $\hat{\beta}$ and V_β, with the understanding that both are functions of the data and the variance parameters. To simulate β in the Gibbs sampler we need not compute

V_β explicitly; rather, we can compute the QR decomposition $\Sigma_*^{-1/2} X_* = QR$ and then draw

$$\text{all-at-once Gibbs update:} \qquad \beta = \hat{\beta} + R^{-1} z, \tag{7}$$

where z is a vector of independent unit normals. (We can write $\hat{\beta}(\sigma)$ and $R(\sigma)$ to emphasize their dependence on the variance components.)

The *one-at-time* algorithm samples β component-wise, conditional on all the other parameters:

$$\text{one-at-a-time Gibbs update: for each } j\text{, sample} \quad \beta_j \sim p(\beta_j | \beta_{-j}, \Sigma_y, \sigma_\beta, y), \tag{8}$$

where β_{-j} is all of β except β_j. Expression (8) is a simple univariate normal distribution.

The one-at-a-time computations are slightly more difficult to set up, in the general case, than the vector Gibbs sampler, but they have the advantage of never requiring large matrix operations. If the updating step (8) is set up carefully, with the appropriate intermediate results held in storage, this algorithm can be very efficient in terms of computation time (Boscardin, 1996). There is also the potential for further speeding the scalar computations by taking advantage of zeros in the X matrix.

For either algorithm, updating the variance components is simple; their conditional posterior distributions, given β, are independent,

$$\sigma_{zk}^2 | \beta \sim \text{Inv-}\chi^2(\nu_{zk} + n_{zk}, \frac{\nu_{zk}\sigma_{0zk} + SS_{zk}}{\nu_{zk} + n_{zk}}),$$

where n_{zk} is the number of components of z that correspond to the variance parameter σ_{zk} (that is, the sum of the elements of the kth column of W_z), and SS_{zk} is the kth component of the appropriate vector of residual sum of squares:

$$\begin{aligned} SS_y &= W_y \text{Diag}((y - X\beta)\Sigma_y^{-1}(y - X\beta)^t), \qquad \text{or} \\ SS_\beta &= W_\beta \text{Diag}((\beta - \beta_0)\Sigma_\beta^{-1}(\beta - \beta_0)^t). \end{aligned} \tag{9}$$

1.4 A Metropolis adaptation for the hierarchical generalized linear model

For the hierarchical generalized linear model, both algorithms above must be modified since the likelihood is not conjugate with the Gaussian prior distribution. The most straightforward adaptation is to perform the Gibbs sampler, accepting or rejecting at each step based on the Metropolis-Hastings rule. For the one-at-time algorithm, one approach for the one-dimensional jumps is the adaptive Metropolis algorithm of Gilks, Best, and Tan (1995). For the all-at-once algorithm, a natural approach to updating β is, by analogy to maximum likelihood computations, to run the regression based on a linearization of the likelihood at the current values of the variance components. This involves computing the conditional (on the variance components) posterior mode and second derivative matrix and using a multivariate Gaussian or t distribution to generate proposals.

2 Conditional augmentation in one-at-a-time algorithms

If the components of β are highly correlated in their posterior distribution, then the one-at-a-time Gibbs sampler (or, for the generalized linear model, the associated Metropolis-Hastings algorithms) can move slowly. The vector Gibbs sampler avoids this problem by sampling all the components of β at once, but this algorithm can be slow in computation time because it requires a full linear regression computation at each step. (We formalize this run-time computation in Section 4.1.)

Optimally, we would combine the computational speed of the one-at-a-time algorithm with the fast convergence of the all-at-once algorithm. Suppose we introduce a fixed working parameter A, via $\xi = A\beta$ and draw the variance components jointly given ξ and y and the components of ξ one at a time given the variance components and y. We would like to choose A to optimize, or at least improve, computational performance. One possibility is to set $A = [R(\sigma^{(t)})]^{-1}$, where $\sigma^{(t)}$ represents the current draw of the variance parameters and R comes from the QR decomposition of $\Sigma_*^{-1/2} X_*$ (see (7)) at each iteration, which orthogonalizes the elements of ξ and result in an algorithm that is equivalent to the all-at-once updating. Unfortunately, this strategy does not save time because it requires the computation of $\hat{\beta}$ and R, i.e., the full regression computation, at each iteration.

A less computationally burdensome method fixes A at some nearly optimal value that is constant across iterations and thus does not need to be recomputed. A natural approach is to set $A = [R(\hat{\sigma}_0)]^{-1}$, where $\hat{\sigma}_0$ is some estimate of the variance parameters, perhaps estimated by running a few steps of an EM-type algorithm (e.g., van Dyk, 2000). Given $\hat{\sigma}_0$, the idea is to run the regression just once, to compute A, and then run the Gibbs conditional of A, i.e., using ξ, As long as $R(\sigma)$ is reasonably stable, using a reasonable estimate of σ should keep the components of ξ close to independent in their conditional posterior distribution. A natural modification is to set A to average R over its posterior distribution, which can be computed for example by updating occasionally (for example, every 200 steps) online; e.g., set $A^{(0)} = R(\hat{\sigma}_0)^{-1}$ and then, for $t = 200, 400, \ldots$,

$$A^{(t)} = \frac{1}{t+200}\left[A^{(200l)}t + [R(\sigma^{(t)})]^{-1}\right].$$

Similarly, we can apply Metropolis-Hastings algorithms to the components of ξ one at a time, either as corrections to approximate Gibbs sampler jumps for generalized linear models (where the nonconjugate conditional posterior densities make exact Gibbs impractical), or simply using spherically-symmetric Metropolis jumps on ξ, starting with a unit normal kernel with scale $2.4/\sqrt{d}$ (where d is the dimension of β) and tuning to get an approximate acceptance rate of 1/4 (see Gelman, Roberts, and Gilks, 1996).

3 Marginal augmentation

In those algorithms we have conditioned on the working parameter in the spirit of Meng and van Dyk's (1999) conditional augmentation (β corresponds to their augmented data). That is, we

choose A according to some optimization strategy either once for the entire run, computed during the iteration, or recomputed every 200 (say) iterations. This equivalent to a reparametrization; once A is fixed, we can consider ξ to be a simple transformation of β. We now consider averaging over the working parameter using marginal augmentation.

3.1 Motivation and notation for the expanded model

As we discuss in detail in Section 4.3.2, any of the above Gibbs or Metropolis algorithms can be slow to converge when estimated hierarchical variance parameters are near zero. The problem is that, if the current draw of $\sigma_{\beta k}$ is near 0, then in the updating step of β, the derivation of the parameters β in batch k from their prior means in β_0 will themselves be estimated to be very close to 0 (because their prior distribution has scale $\sigma_{\beta k}$). Then, in turn, the variance parameter will be estimated to be close to 0 because it is updated based on the relevant β_j's (see (9)). Ultimately, the stochastic nature of the Gibbs sampler allows it to escape out of this trap but this may require many iterations.

Van Dyk and Meng (2000) showed how the method of marginal augmentation can substantially improve convergence in this setting; see also Meng and van Dyk (1998), Liu, Rubin, and Wu (1998), van Dyk (2000), and Liu (2002). Marginal augmentation indexes a class of models with a working parameter, α; the elements in the class are equivalent in that

$$\int p(y, \beta|\sigma, \alpha) d\beta = p(y|\sigma) \qquad \text{for all} \quad \alpha \in \mathcal{A}.$$

that is, $p(y|\sigma, \alpha)$ does not depend on α: this is the definition of a working parameter. Clearly, the working parameter will not affect inference based on $p(y|\sigma)$. In contrast to conditional augmentation which chooses a value of α to optimize (or improve) convergence, marginal augmentation marginalizes over a working prior density, $p(\alpha)$:

$$\int p(y, \beta|\sigma, \alpha) p(\alpha) d\beta d\alpha = p(y|\sigma),$$

where, as usual, we assume the working and model parameters are a priori independent.

The choice of working prior density affects the convergence behavior of the algorithm; generally more diffuse densities improve mixing (at least when the working parameter transforms the data augmentation independent of the model parameter; see Meng and van Dyk, 1999). Unfortunately, improper densities lead to non-positive recurrent Markov chains since $p(\alpha|y)$. With judicious choice of the improper density, however, we can ensure that subchain corresponding to the model parameters is recurrent with stationary distribution $p(\sigma|y)$; see Meng and van Dyk (1999) and Liu and Wu (1999).

We follow the parameterization chosen by Liu, Rubin, and Wu (1998), which in our notation is

$$A^{-1} = \text{Diag}(W_\beta \alpha), \tag{10}$$

where W_β is the matrix of 0's and 1's from (3), including the batching of the β_j's in their prior distribution. we thus have introduced one parameter α_k for each of the hierarchical variance parameters $\sigma_{\beta k}$. we shall see that, with this new parameter, the Gibbs sampler is less prone to move slowly in regions of parameter space with $\sigma_{\beta k}$ near zero.

Reformulating the model in terms of $\xi = A\theta$ yields

$$y \sim \mathrm{N}(XA^{-1}\xi, \sigma_y) \tag{11}$$

$$\alpha \sim \mathrm{N}(\alpha_0, \Sigma_\alpha) \tag{12}$$

$$\xi \sim \mathrm{N}(\xi_0, \Sigma_\xi), \tag{13}$$

where $\xi_0 = A\beta_0$ and $\Sigma_\xi = A\Sigma_\beta A)$. This prior variance matrix is diagonal since both A and Σ_β are diagonal: $\Sigma_\xi = \mathrm{Diag}(W_\beta(\alpha * \sigma_\beta)^2)$, where * represents componentwise multiplication.

Adding the new parameter α potentially generalizes the hierarchical model in an interesting direction—allowing interactions in the parameters (which is different than interactions among the regression predictions). Our primary purpose in this paper, however, is to use the expanded model to improve computational efficiency.

3.2 Equivalence between the parameter-expanded model and the usual model

Model (11)–(13) reduces to model (1)–(2) if each element of α is either fixed at a constant or has a uniform prior distribution. That is, the two models are equivalent if Σ_α is diagonal, with each of its diagonal elements equal to either zero or infinity. Under equivalence, the parameters change their interpretation slightly:

Model (1)–(2)	Model (11)–(13)
the vector β	the vector $(W_\beta) * \xi$
a signle β_j	$\alpha_{k(j)}\xi_j$
the vector σ_β	the vector $\lvert\alpha\rvert * \sigma_\xi$
a single $\sigma_{\beta k}$	$\lvert\alpha_k\rvert\sigma_{\xi k}$

Thus, if we would like to work with model (1)–(2), but we use model (11)–(13) for computational efficiency, then it is the summaries on the right side of the above table that we will want to report. For each batch k of regression coefficients, the coefficients β_j and the variance parameters $\sigma_{\beta k}$ estimated from model (11)–(13) must be multiplied by the estimated scale parameter α_k so that they can be given their original interpretations as β_j's and σ_{jk} in the original model (1)–(2).

In fact, with these special prior distributions the parameters α_k and $\sigma_{\beta k}$ have no separate meaning and only affect the final inferences from the model through their product. Thus, if a noninformative prior distribution is assigned to $\sigma_{\beta k}$, then we must constrain the model so that α_k is identified. We do so by setting each such α_k to 1 at the end of each iteration.

3.3 Gibbs sampler computation for the extended model

There are many possible implementations of the Gibbs sampler in this model; the simplest to describe alternately updates the vectors α, ξ, and σ. To update α, we must reexpress equation (11) as

$$y \sim \mathrm{N}(X\mathrm{Diag}(\xi)W_\beta\alpha, \Sigma_y). \tag{14}$$

Conditional on all the other parameters of the model, this is just a normal linear regression of y on α with known design matrix, $X\mathrm{Diag}(\xi)W_\beta$, and known variance matrix, Σ_y, along with a normal prior distribution with known mean α_0 and known variance Σ_α. A conditional posterior simulation of α cab be obtained by running a standard regression program (augmenting the data vector, design matrix and the variance matrix to include the prior distribution) to obtain an estimate $\hat{\alpha}$ and covariate matrix, then drawing from the multivariate normal distribution centered at that estimate and with that covariance matrix (see, e.g., Gelman et al., 1995).

Similarly, to update ξ, we reexpress (11) as

$$y \sim \mathrm{N}(X\mathrm{Diag}(W_\beta\alpha)\xi, \Sigma_y) \tag{15}$$

and work with a regression in which $X\mathrm{Diag}(W_\beta\alpha)$ is the design matrix.

We can update the parameters in Σ_y and Σ_ξ as before.

3.4 Parameter expansion for generalized linear models

For the generalized linear model, the same general ideas hold if we simply replace (11) by

$$y \sim \mathrm{glm}(X((W_\xi) * \xi), \Sigma_y) \qquad \text{or} \qquad y \sim \mathrm{glm}(X((W_\xi) * \xi)),$$

with the equivalences to the original model as described in Section 3.2. As in Section 3.3, the Gibbs sampler for the new model must include a step to update α. Since the model is nonconjugate, this can must be performed either a linerization of the model $y \sim \mathrm{glm}(X((W_\xi) * \xi), \Sigma_y)$—considered as a likelihood for α—followed by a draw from the corresponding approximate Gauusian conditional prior distribution for α and a Metropolis-Hastings accept/reject step. As with the normal model computation in Section 3.3, α is typically a relatively short vector, and so we are not bothered by simply updating it in vector fashion using linear regression computations. Computation for ξ and the variance parameters are as before.

3.5 Setting up the computation

We perform all computations in terms of the regression model (11)–(13), without assuming any special structure in the design matrix X.

The algorithms must begin with estimates of the variance parameters σ_{zk} and also, for the one-at-a-time algorithms, an estimate of ξ. In order to reliably monitor the convergence of the Gibbs

sampler, it is desirable to start the Gibbs sampler runs at a set of points that are "overdispersed": that is, from a distribution that includes and is more variable than the starting distribution. More precisely, it is desired that the simulation draws from the mixture of all the simulated sequences should decrease in variability as the simulation proceed, with the variability within each sequence increasing, so that approximate convergence can be identified with the empirical mixing of the simulated sequences (see Gelman and Rubin, 1992). An important implication of overdispersion is that we have to worry about the speed of the algorithms when they are started far from the posterior mode.

4 Theoretical arguments

4.1 Computation time per iteration

The QR decomposition of the W matrix and backsolving the two upper-triangular systems takes $O(2nm^2)$ floating-point operations (flops), where n and m are the length of y and ξ, respectively; see Golub and van Loan (1983). For the scalar updating algorithm, updating all m components of ξ and tranforming back to ξ takes only $O(10nm)$ flops. The computation for updating α require another $O(nm)$ flops. (We assume that the length of α and the number of variance parameters is neglible compared to m and n.) In many of the problems we work with, m is quite large—typically some fraction of n—and thus the scalar updating algorithms require $O(n^2)$ flops per iteration, compared to $O(n^3)$ for the vector algorithms.

4.2 Covergence rate of vector vs. scalar updating of regression parameters

An interesting tradeoff occurs as the number of predictors become large. As the batch sizes $n_{\beta k}$ increase, the dimensions of the X_* matrix in (6) increase also, making the least-squares regression computation slower. However, having more replication increases the precision of the estimates of the variance parameters, which means that once they have been reasonably estimated, a one-time transformation as described in Section 2 will do a good job of making the components of ξ close to independent. Thus, *increasing the complexity of the model can make the Gibbs sampler computations more stable on the transformed space.*

We shall try to understand the efficiency of the vector and scalar algorithms by considering different amounts of knowledge about the variance parameters σ in the model (see (3)).

First suppose all the variance parameters are known. Then the vector updating of ξ converges in a single step, meaning that Gibbs sampler draws are equivalent to independent draws from the posterior distribution of ξ. In contrast, the speed of the untransformed scalar updating alogorithm is roughly determined by the second eigenvalue of the posterior variance matrix V_ξ. In many hierarchical models, it is possible to keep the correlations in V_ξ low by centering and scaling the parameter ξ (see Gelfand, Sahu and Carlin, 1995, and Gilks and Roberts, 1996). These techniques

will not necessarily to be applied easily in all cases, however. Thus, the computational savings from the scalar updating algorithm can possibly be lost if those algorithms require much more time to converge. Finally, the transformed scalar updating algorithm will converge in one step, and be just as efficient as the vector algorithm, if the variance parameter are known.

Second, consider a setting where the variance parameters are known, but are accurately determined by the posterior distribution. This should be the case in large data sets with large values of n_{yk} and $n_{\beta k}$. In this case, the transformed scalar updating should be nearly as efficient as the vector algorithm, since the only reason the orthogonalizing transformation varies is because of uncertainty in Σ_y and Σ_ξ.

Third, suppose that the variance parameters are poorly estimated, as could occur if some of the variance components $\sigma_{\xi k}$ had few associated ξ_j's, along with a weak or noninformative prior distribution. In this case, the transformed scalar updating, with any particular transformation, might not work very well, and it would make sense to occasionally update the transformation based on current values of Σ_y and Σ_ξ in the simulation.

4.3 Effect of parameter expansion on convergence rate

The parameter expansion algorithm affects only the updating of α and σ_ξ, leaving the operations for ξ and σ_y unchanged. Because the parameter expansion operates independently for each hierarchical variance component k, we consider its effect on the convergence of these $n_{\beta k}$ components separately. We shall use the notation $\tau_k = \alpha_k \sigma_\xi$. The hope is that the new parameterization can be outperform the old in three settings:

- τ_k near the posterior mode (relevant in problems for which τ_k is well estimated in the posterior distribution),
- τ_k near 0 (relevant because of possible extreme starting points and also for problems in which τ_k has a high posterior probability of being near 0),
- τ_k near ∞ (relevant because of possible extreme starting points and also for problems in which τ_k has a high posterior probability of being large).

For each of these case, we consider first the speed of the corresponding EM algorithms (a simpler problem because deterministic) and then consider the Gibbs sampler. The EM algorithms we shall consider are those that treat the components of ξ as missing data and thus converge to the posterior mode of $(\alpha, \sigma_y, \sigma_\xi)$, averaging over ξ.

We work through the details in the context of a simple model with one variance component $\tau = \alpha\sigma_\xi)$ in order to illustrate the principles that we conjecture to hold for hierarchical models in

general. Consider the following model, with variance expressed in parameter-expanded form:

$$\begin{aligned} \text{For } j = 1, ..., m: \quad y_j &\sim \mathrm{N}(\alpha\xi, 1) \\ \xi_j &\sim \mathrm{N}(0, \sigma_\xi^2) \\ p(\alpha, \sigma_j) &\propto 1. \end{aligned}$$

The Gibbs sampler for this model (in this case, the vector and the componentwise algorithms are identical) is:

$$\begin{aligned} \beta_j &\leftarrow \mathrm{N}\left(\frac{1}{\alpha}\frac{y_j}{1+\frac{1}{\tau^2}}, \frac{1}{\alpha^2}\frac{1}{1+\frac{1}{\tau^2}}\right), \qquad \text{for } j = 1, ..., m \\ \alpha &\leftarrow \mathrm{N}\left(\frac{\sum_{j=1}^m \beta_j y_j}{\sum_{j=1}^m \beta_j^2}, \frac{1}{\sum_{j=1}^m \beta_j^2}\right) \\ \sigma_\beta^2 &\leftarrow \sum_{j=1}^m \beta_j^2 / \chi_{m-1}^2. \end{aligned} \tag{16}$$

The EM updating is simply,

$$\begin{aligned} \alpha &\leftarrow \alpha \frac{s_y^2}{\frac{1}{1+\frac{1}{\tau^2}} s_y^2 + 1} \\ \sigma_\beta^2 &\leftarrow \frac{1}{\alpha^2}\left(\frac{1}{1+\frac{1}{\tau^2}}\left(\frac{1}{1+\frac{1}{\tau^2}} s_y^2 + 1\right)\right), \end{aligned} \tag{17}$$

where

$$s_y^2 = \frac{1}{m}\sum_{j=1}^m y_j^2.$$

at each step. τ is set to $|\alpha|\sigma_\beta$. The EM algorithm converges approximately to $\hat{\tau} = \sqrt{s_y^2 - 1}$, or 0 if $s_y \leq 1$.

4.3.1 Speed of the algorithms near the posterior mode

For τ_l near the posterior mode (assuming that mode is not 0), the speed of the Gibbs sampler is roughly the same as the corresponding EM algorithm. Liu, Rubin, and Wu (1997) consider the EM algorithm for the hierarchical normal model and find that parameter expansion is uniformly better than the usual algorithm (which, in the parameter expansion context, corresponds to holding the parameters α_l fixed at 1).

4.3.2 Speed of the algorithms for τ near 0

For τ_l near 0, the usual algorithm, in both the EM and Gibbs contexts, is notoriously slow, but we shall see that the parameter expanded versions move much faster.

We first consider the EM algorithm. Under the usual parametrization, α is fixed at 1, and only σ_β is updated. For τ near 0, we can expand (17) as

$$\begin{aligned} \tau &\leftarrow \frac{\tau}{1+\tau^2}\sqrt{1+\tau^2+\tau^2 s_y^2} \\ &= \tau(1+\frac{1}{2}(s_y^2-1)\tau^2+\ldots). \end{aligned}$$

Thus, for τ near 0, even the *relative* change in τ at each step approaches 0. If $s_y^2 \leq 1$, this means that τ will approach 0 at an infinitely slow rate; if $s_y^2 > 1$ and τ happens to be started near 0, then it will move away from 0 hopelessly slowly.

In contrast, the parameter-expanded algorithm updates both α and σ_β, yielding, for τ near 0,

$$\begin{aligned} \tau &\leftarrow \tau\frac{s_y^2}{\sqrt{1+(s_y^2+1)\tau^2}} \\ &= \tau(s_y^2+\ldots), \end{aligned}$$

which, for $s_y^2 \neq 1$ is geometric convergence, which should be acceptable in practice as long as s_y^2 is not too close to 1 and τ is not started at an extremely small value.

We now consider the Gibbs sampler, which, under the conventional parameterization, adds two sorts of variation to the EM algorithm: (1) $\mathrm{E}(\sum_{j=1}^m \beta_j^2)$ is replaced by the random variable $\sum_{j=1}^m \beta_j^2$, and (2) division by a χ_m^2 random variable. (Under the conventional parameterization, the updating of α in (17) does not occur since α is fixed at 1.) After working out the algebra, we find that when the current value of τ in the Gibbs sampler is near 0, each of these steps adds a coefficient of variation $1/(\sqrt{2}m)$ to the updating step for τ^2; so, in combination, they give the random updating of τ a coefficient of variation of $1/m$. Within the context of the usual parametrization, with τ near 0, this variation acts like a random walk, and it implies that the Gibbs sampler will require on the order of $m(\log\tau_0)^2$ iterations to "escape" from a starting τ_0 near 0. (See Rosenthal, 1995, for more formal versions of this argument.)

For the parameter-expanded algorithm, the Gibbs sampler adds another random component in the updating of α. We can express the updated τ as

$$\tau \leftarrow \left| \frac{\sum_{j=1}^m \gamma_j y_j}{\sqrt{\sum_{j=1}^m \gamma_j^2}} + z \right| / x, \tag{18}$$

where the components of $\gamma = \beta/\alpha$ can be written as

$$\gamma_j = \frac{1}{1+\frac{1}{\tau^2}} + \frac{1}{\sqrt{1+\frac{1}{\tau^2}}} z_j,$$

and the following random variables are independent:

$$z \sim \mathrm{N}(0,1), \qquad x \sim \chi_{m-1}^2, \qquad z_j \sim \mathrm{N}(0,1), \qquad \text{for } j = 1, ..., m. \tag{19}$$

The random variables in (18) induce a distribution for τ on the left of (18) that depends only on m, $s_y^2 = \frac{1}{m}\sum_{j=1}^m y_j^2$, and the previous value of τ. For τ near 0, this distribution has a standard deviation on the order of $1/\sqrt{m}$ *on the absolute scale.* Therefore, it is impossible for τ to get stuck at values less than $1/\sqrt{m}$, no matter how low its value was on the previous iteration. This is a stunning improvement over the Gibbs sampler for the conventional parametrization.

This jumpy behavior of the parameter-expanded algorithm near $\tau = 0$ is reminiscent of the nonzero lowest energy state in quantum mechanics, which is related to the uncertainty principle—if τ is precisely localized near 0, then the "momentum" of the system will be high (in that τ is likely to jump far away). When m is large, the minimal "energy" of $1/\sqrt{m}$ becomes closer to 0, which corresponds in a physical system to the classical limit with many particles. In a statistical sense, this behavior is reasonable because the precision of the marginal posterior distribution for τ is determined by the number of coefficients m in the batch.

4.3.3 Speed of the algorithms for τ near ∞

If τ is started at a very high value, both the usual parameterization and the parameter-expanded algorithm perform reasonably well. From $\tau = \infty$, the usual EM algorithm moves in one step to $\tau = \sqrt{s_y^2 + 1}$ and the parameter-expanded EM algorithm moves in one step to $\tau = s_y^2/\sqrt{s_y^2 + 1}$. Of these two, the parameter-expanded version is preferable (since it is closer to the convergence points, $\sqrt{s_y^2 + 1}$), but they are both reasonable. Similarly, τ will not get stuck near ∞ under either Gibbs sampler implementation.

5 Examples

We consider two examples that have arisen from applied research. For each example and each algorithm, we compute computation time per iteration, number of iterations untill convergence (as determined by the condition that the variance ratio $\hat{R}$ (Gelman and Rubin, 1992) is less than 1.2 for all model parameters), and simulation efficiency (measured by the autocorrelations).

5.1 The examples

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). For each school $j = 1, ..., 8$, an experiment was performed to estimate the treatment effect θ_j in that school; a regression analysis applied to the data from that school alone yielded an estimate and standard error, which we label y_j and σ_j. The sample size within each school was large enough that it is reasonable to model $y_j|\mu, \theta_j \sim \mathrm{N}(\mu + \theta_j, \sigma_j^2)$, with σ_j known. The parameters from the 8 schools are modeled as normally distributed: $\theta_j \sim$ iid $\mathrm{N}(0, \tau^2)$, with a noninformative uniform hyperior density, $p(\mu, \tau) \propto 1$. Interest lies in the individual school parameters θ_j and also in the hyperparameters (μ, τ). This

particular example is quite small and should be easy to compute; in fact, since there is only one hierarchical variance component, it is in fact possible to draw posterior simulations noniteratively by first computing the marginal posterior density for τ at a grid of points and then simulating the parameters in the order τ, μ, θ (Rubin, 1981). The only possible difficulty for the Gibbs sampler is that the maximum likelihood estimate of the hierarchical scalar parameter, τ, is zero; there is also a substantial portion of the posterior density near $\tau = 0$, and there is the possibility that the algorithm may move slowly in this region. In this example, we use model $y_j|\mu, \alpha, \theta_j \sim \mathrm{N}(\mu+\alpha\theta_j, \sigma_j^2)$, with $p(\alpha) \propto 1$ as our working model for parameter expansion, where α is the working parameter that will play important role in simultation to avoid τ trapping in the 0 region.

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 the kth predicted precipitation anomaly for m^{th} GCM ensemble in box j and at time t. We use as predictors the average values,

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

We set up the following multilevel model using y_{jt} as the response and $\bar{x}_{jt}^{m}$ as predictors in Bayesian framework:

$$y_{jt} = \theta_j + \delta_t + \sum_{m=1}^{M} \beta_j^m \bar{x}_{jt}^m + \epsilon_{jt}, \quad \text{for } j = 1, \ldots, 527 \text{ and } t = 1, \ldots, 41. \tag{20}$$

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.
- γ_j is an offset for location j; we assign it a normal population distribution with mean μ_θ and standard deviation σ_θ.
- β_j^m is the coefficient of ensemble forecast m in location j; we assign it a normal population distribution with mean μ_{β_m} and standard deviation σ_{β_m}.
- ϵ_{it}'s are independent error terms assumed normally distributed with mean 0 and standard deviation σ_y.

- We assign noninformative hyperior distributions to the remaining parameters in the model:

$$p(\mu_\theta, \mu_{\beta_1}, \mu_{\beta_2}, \mu_{\beta_3}, \sigma_\theta, \sigma_\delta, \sigma_{\beta_1}, \sigma_{\beta_2}, \sigma_{\beta_3}, \sigma_y) \propto 1.$$

In this example, our parameter expansion model is

$$y_{jt} = \alpha_\theta \theta_j + \alpha_\delta \delta_t + \sum_{m=1}^{M} \alpha_{\beta^m} \beta_j^m \bar{x}_{jt}^m + \epsilon_{jt}, \quad \text{for } j = 1, \ldots, 527 \text{ and } t = 1, \ldots, 41, \tag{21}$$

with $p(\alpha_\theta, \alpha_\delta, \alpha_{\beta^1}, ..., \alpha_{\beta^m}) \propto 1$, where the α's are the working parameters.

5.2 Results

We present the results of the standard and PX-Gibbs algorithms. For each algorithm, starting points were obtained by running the EM algorithm (with initial guesses of 1 for all the variance components) and then drawing from a t_4 distribution. When the EM estimate of a variance component was zero (as in the study of the eight schools), we used 1 as a starting point.

Figure 1 compares the computation time of the standard and PX-Gibbs algorithms for the example of the eight schools. Each time we run 10 chains until approximate convergence ($\hat{R} < 1.2$ for all parameters). In Figure 1, V represents vector updating without parameter expansion, S scalar updating without parameter expansion, V+PX vector updating with parameter expansion, and S+PX scalar updating with parameter expansion. The dots display the combination of the average computation time per iteration and the average iterations required for convergence for each algorithm. The lines represent indifferent curves for total computation time. These are stright lines because the graph is on the logarithm scale. A point on a lower line indicates a more efficient algorithm in total computation time. Therefore the most efficient algorithm is scalar updating with parameter expansion, which only takes an average of 0.39 seconds in total computation time pr Gibbs sampler chain; the second is the vector updating with parameter expansion, which takes 0.69 seconds; the third is scalar updating, which takes 4.2 seconds; and the slowest is vector updating, which takes 8.7 seconds. In this example parameter expansion is about 22 times faster than traditional algorithms.

Figure 2 shows the simulation efficiency (measured by the autocorrelations and the sum of their absolute values) for the variance component τ. The vector and scalar updating with parameter expansion are much more efficient than the other two algorithms.

Because the design matrix for the second example (climate modeling) is too big, it is difficult to implement vector updating in R or S-Plus, and and thus we only consider the scalar updating algorithm. The computation speed (iterations per second) is 0.82 for the scalar updating with parameter expansion, compared to 0.78 for the scalar updating without parameter expansion. And the scalar updating with parameter expansion only need 400 iterations until convergence, compared

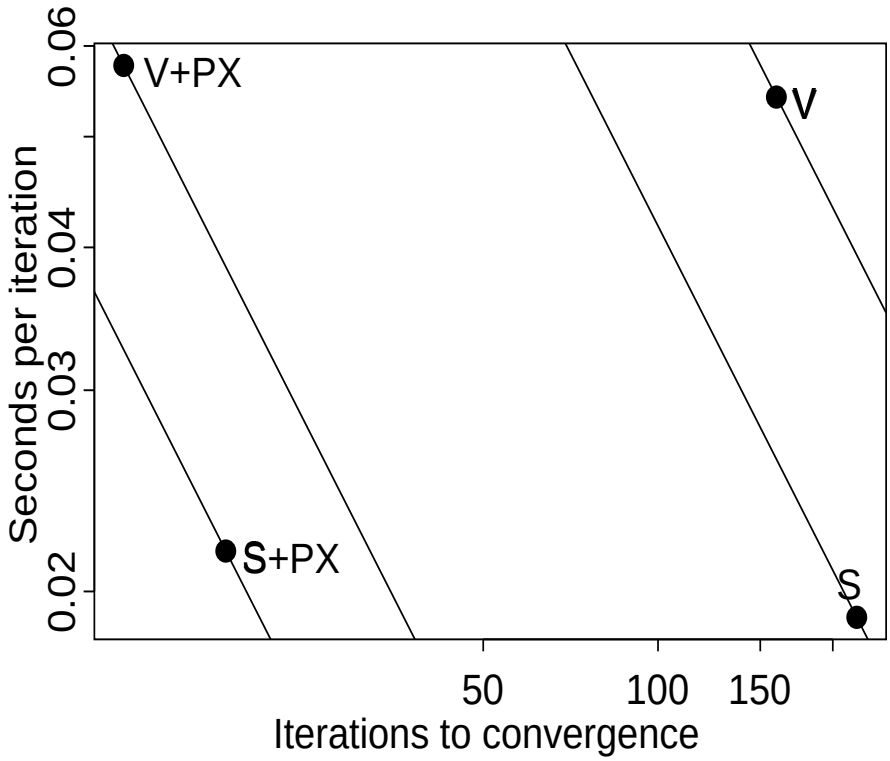

Figure 1: Computation time of the four algorithms for the eight schools example (plotted on a logarithm scale). V: Vector updating, S: Scalar updating, V+PX: Vector updating with paramater expansion, and S+PX: Scalar updating with paramater expansion. The dots display the combination of the computation time per iteration and the iterations required for convergence for each algorithm. The lines are indifferent curves in total computation time.

to 10,000 for the scalar updating without parameter expansion. Figure 3 compares the computation time of the two algorithms for the second example (climate modeling). Therefore the most efficeient algorithm is scalar updating with parameter expansion, which takes an average of 490 seconds per chain computation time compared to 13,350 seconds per chain for scalar updating.

Figure 4 and 5 show the simulation efficiency (measured by the autocorrelations) of the first 16 parameters for the scalar updating with parameter expansion and without parameter expansion, respectively. Parameter expansion dramatically improves the efficiency of the scalar updating algorithm.

6 Discussion

We see this paper as having three main contributions. First, we combine some ideas from the recent statistical literature to construct a family of improved algorithms for posterior simulations from hierarchical models. We suspect that the general ideas of approximate rotation for correlated

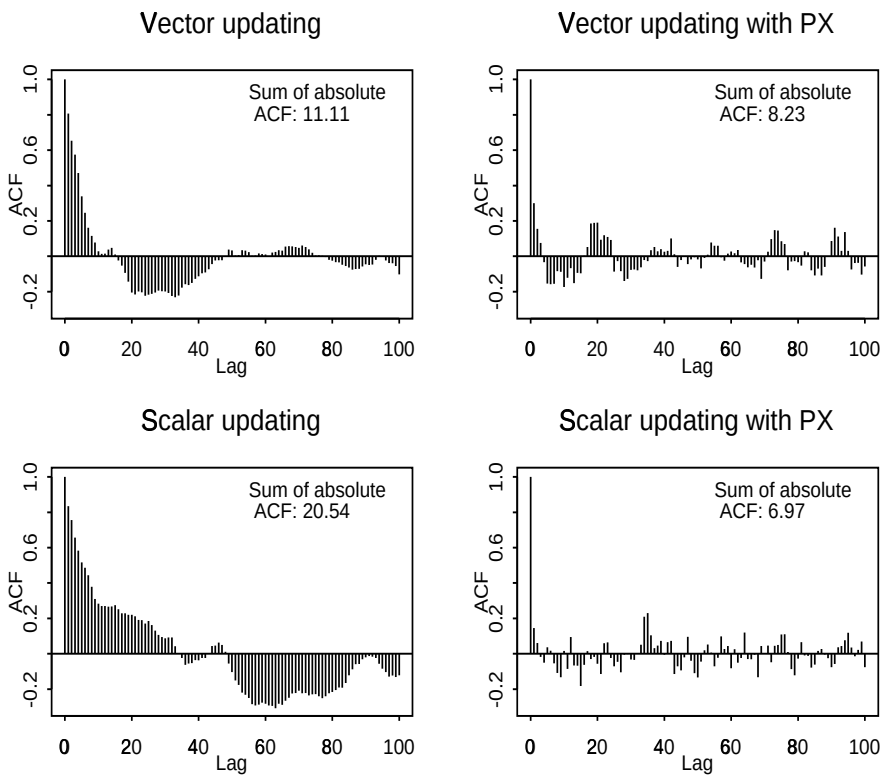

Figure 2: Simulation efficiency (measured by the autocorrelations) of variance component τ for the eight schools example.

parameters and parameter expansion for variance components will be useful in more elaborate settings such as multivariate and nonlinear models as well.

Second, the computations are set up in terms of an expanded model, following the work of Liu, Rubin, and Wu (1997) for the EM algorithm. Once this model has been set up, the next natural step is to see if its additional parameters can be given their own statistical meaning, as discussed in Section 3.3. There is a history in statistics of duality between computational tools and new models (for example, Besag (1974) motivated conditional autoregression by way of the Hammersley-Clifford theorem for joint probability distributions, and Green (1995) introduced a reversible-jump Markov chain algorithm that has enabled and motivated the use of mixture posterior distributions of varying dimension). Multiplicative parameter expansion for hierarchical variance components may turn out to be another useful model generalization that was originally motivated for computational reasons.

Third, we connect computational efficiency to the speed at which the various iterative algorithms can move away from corners of parameter space, in particular, near-zero estimates of variance components. When the number of linear parameters m in a batch is high, the corresponding variance component can be accurately estimated from data, which means that a one-time rotation can bring the linear parameters to approximate independence, leading to rapid convergence with one-at-a-

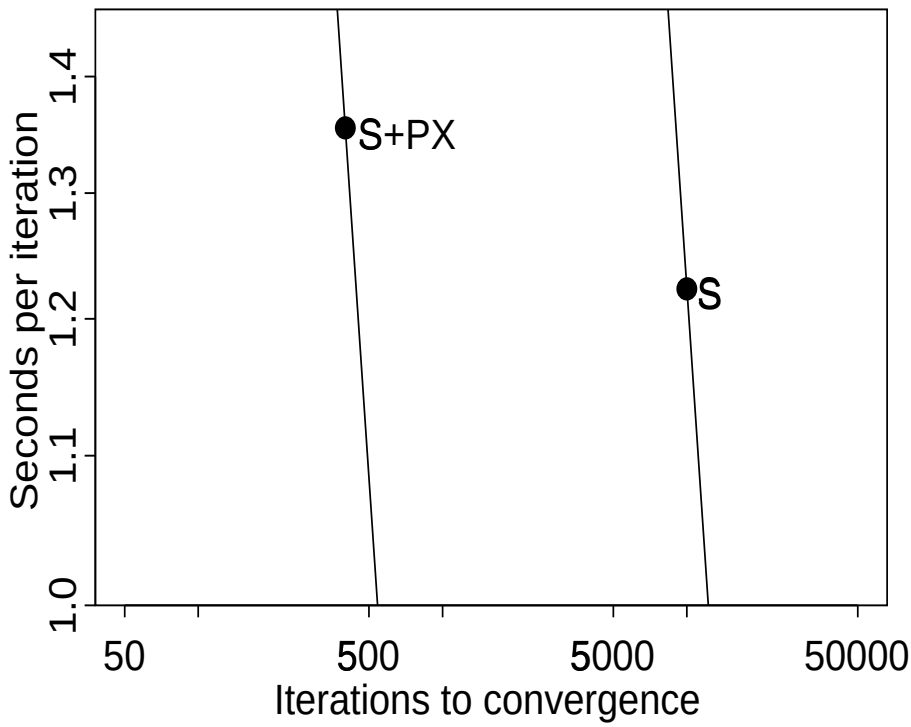

Figure 3: Computation time (on a logarithm scale) of the two algorithms for the climate modeling example. S: Scalar updating and S+PX: Scalar updating with parameter-expansion. The dots display the combination of the computation time per iteration and the iterations required for convergence for each algorithm. The lines are indifferent curves in total computation time.

time Gibbs or spherical Metropolis algorithm. This is a "blessing of dimensionality" to be balanced against the usual "curse." On the other hand, the "uncertainty principle" of the parameter-expanded Gibbs sampler keeps variance parameters from being trapped within a radius of approximately $1/\sqrt{m}$ from 0 (see the end of Section 4.3.2), so here it is helpful if m is low.

Further work is needed in several areas. First, it would be good to have a better approach to starting the Gibbs sampler. For large problems, the EM algorithm can be computationally expensive—and it also have the problem of zero estimates. It should be possible to develop a fast and reliable algorithm to find a reasonably overdispersed starting distribution without having to go through the difficulties of the exact EM algorithm. A second, and related, problem is the point estimate of (σ, τ) used to compute the estimated covariance matrix R_0 required for the scalar updating algorithms with transformation. Finally, it is not clear how far the parameter expansion idea can be expanded for even more improvements, as discussed in the EM context by van Dyk and Meng (1999).

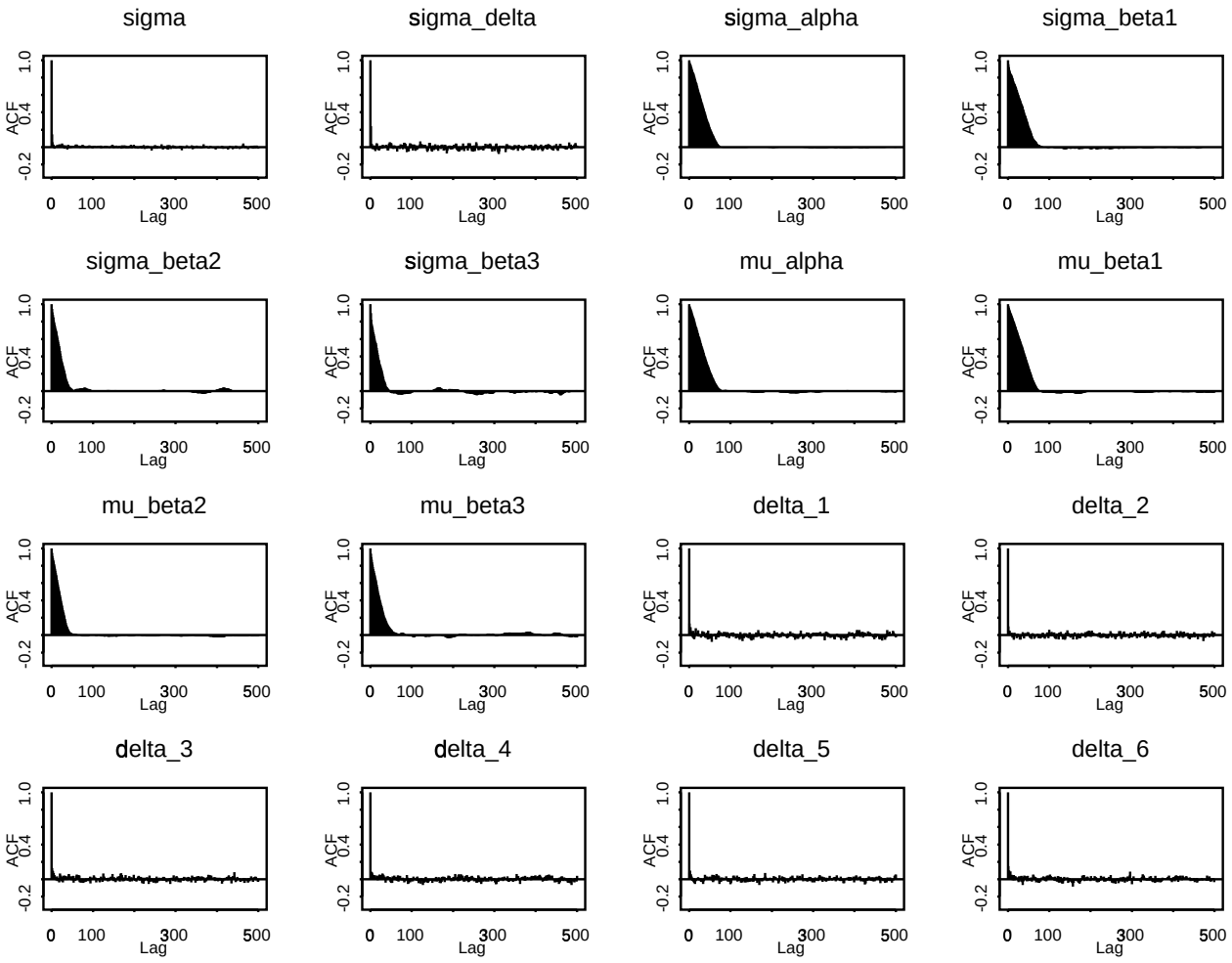

Figure 4: Simulation efficiency (measured by the autocorrelations) of the first 16 parameters for updating with parameter expansion for the climate modeling example.

References

Barnard, J., McCulloch, R., and Meng, X. L. (1997). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems (with discussion). *Journal of the Royal Statistical Society B* **36**, 192-236.

Besag, J., and Green, P. J. (1993) Spatial statistics and Bayesian computation (with discussion). *Journal of the Royal Statistical Society B* **55**, 25-102.

Boscardin, W. J. (1996). Bayesian analysis for some hierarchical linear models. Ph.D. thesis, Department of Statistics, University of California, Berkeley.

Boscardin, W. J., and Gelman, A. (1996). Bayesian regression with parametric models for het-

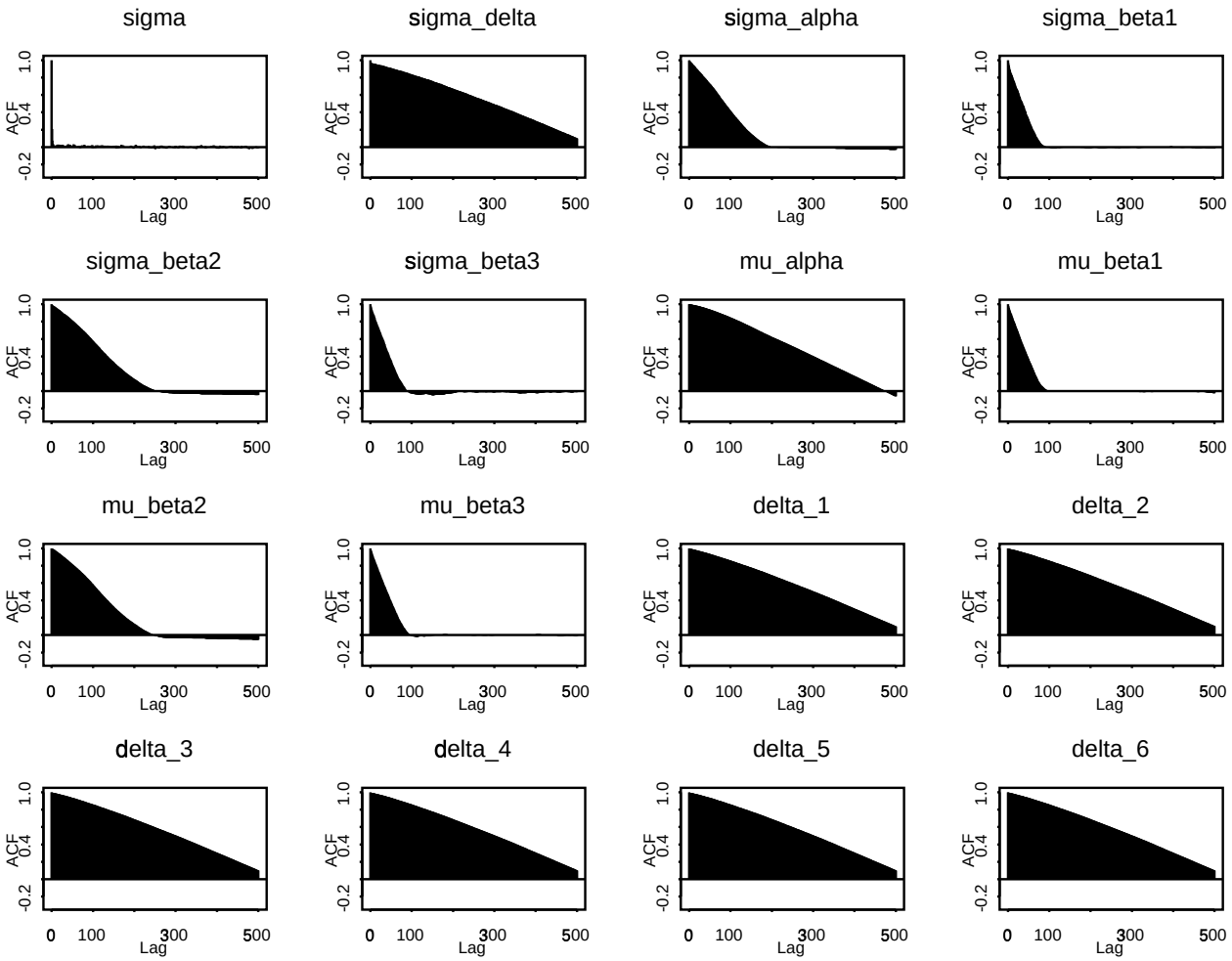

Figure 5: Simulation efficiency (measured by the autocorrelations) of the first 16 parameters for scalar updating *without* parameter expansion for the climate modeling example.

eroscedasticity. *Advances in Econometrics* **11A**, 87-109.

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

Daniels, M. J., and Kass, R. E. (1999). Nonconjugate Bayesian estimation of covariance matrices and its use in hierarchical models. *Journal of the American Statistical Association.*

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society B* **39**, 1-38.

Dempster, A. P., Rubin, D. B., and Tsutakawa, R. K. (1981). Estimation in covariance components models. *Journal of the American Statistical Association* **76**, 341-353.

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.

Gelfand, A. E., Sahu, S. K., and Carlin, B. P. (1995). Efficient parameterization for normal linear mixed models. *Biometrika* **82**, 479-488.

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

Gelman, A., and Little, T. C. (1997). Poststratification into many categories using hierarchical logistic regression. *Survey Methodology* **23**, 127-135.

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

Gilks, W. R., Best, N., and Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statistics* **44**, 455-472.

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

Gilks, W. R., and Roberts, G. O. (1996). Strategies for improving MCMC. In *Practical Markov Chain Monte Carlo*, ed W. Gilks, S. Richardson, and D. Spiegelhalter, 89-114. London: Chapman and Hall.

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.

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

Hills, S. E., and Smith, A. F. M. (1992). Parametrization issues in Bayesian inference (with discussion). In *Bayesian statistics 4,* ed. J. M. Bernado, J. O. Berger, A. P. Dawid, and A. F. M. Smith, 227-246. New York: Oxford University Press.

Hodges, J. H. (1998). some algebra and geometry for hierarchical models, applied to diagnostics. *Journal of the Royal Statistical Society B.*

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., Rubin, D. B., and Wu, Y. N. (1998). Parameter expansion to EM accelerate EM: the PX-EM algorithm. *Biometrika* **85**, 755-770.

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

Liu, J., and Wu, Y. N. (1999). Parameter expansion for data augmentation. *Journal of the American Statistical Association* **94**, 1264-1274.

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

McCullagh, P., and Nelder, J. A. (1989). *Generalized linear models*, second edition. Londom: 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.

Raftery, A. E. (1996). Hypothesis testing and model selection via posterior simulation. In *Practical Markov Chain Monte Carlo*, ed. W. Gilks, S. Richardson, and D. Spiegelhalter, 163-187. New York: Chapman and Hall.

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

Rosenthal, J. S. (1995). Minorization conditions and convergence rates for Markov chain Monte Carlo. *Journal of the American Statistical Association* **90**, 558-566.

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

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.