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

Andrew Gelman†, David A. van Dyk‡, Zaiying Huang§, W. John Boscardin¶

March 10, 2007

Abstract

Hierarchical linear and generalized linear models can be fit using Gibbs samplers and Metropolis algorithms; these models, however, often have many parameters, and convergence of the seemingly most natural Gibbs and Metropolis algorithms can sometimes be slow. We examine solutions that involve reparameterization and over-parameterization. We begin with parameter expansion using working parameters, a strategy developed for the EM algorithm by Meng and van Dyk (1997) and Liu, Rubin, and Wu (1998). This strategy can lead to algorithms that are much less susceptible to becoming stuck near zero values of the variance parameters than are more standard algorithms. Second, 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. We present an innovative evaluation of the algorithms in terms of how quickly they can get away 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 ultimate goal is to develop a fast and reliable method for fitting a hierarchical linear model as easily as one can now fit a non-hierarchical model, and to increase understanding of Gibbs samplers for hierarchical models in general.

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

1 Introduction

1.1 Background

Hierarchical linear models (also called multilevel models, random effects regressions, and mixed effects models) are important tools in many areas of statistics (see, for example, Robinson, 1991, Longford, 1993, and Goldstein, 1995, for reviews). In recent years, there has been increasing interest

*We thank Xiao-Li Meng and several reviewers for helpful discussion and the National Science Foundation for financial support.

†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, University of California, Irvine, dvd@ics.uci.edu

§Circulation Department, New York Times, huangz@nytimes.com

¶Department of Biostatistics, University of California, Los Angeles, jbosco@ucla.edu

in the Bayesian formulation (Lindley and Smith, 1972), which accounts 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 simulating draws of the parameters from their 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. Gibbs samplers (and also EM, Metropolis, and other algorithms) can get stuck because of dependence in the posterior distribution between batches of coefficients and their variance parameters. In this article, we consider computational methods that aim to improve convergence by embedding the target posterior distribution into a larger model. *Auxiliary variables* (Besag and Green, 1993) and *data augmentation* (Tanner and Wong, 1987) are two common methods that work in this way. Auxiliary variables are added to a model in such a way that conditioning on these variables allows a number of model parameters that would otherwise be updated one at a time to be blocked into one joint draw in a Gibbs sampler. Data augmentation, on the other hand, introduces additional variables (called augmented data or latent variables) to simplify the conditional draws of the Gibbs sampler. For example, the complete conditional distributions may become standard distributions in the larger model that includes the augmented data. Together these methods can simplify and reduce dependence in the conditional specifications in a joint posterior distribution.

In this article, we focus on a third strategy that builds on data augmentation. *Parameter expansion* introduces a specific sort of variable known as a working parameter. Such parameters are identifiable given the observed data and augmented data, but not given the observed data alone. This partial identifiability allows additional flexibility in constructing samplers. With care, it can be shown that this flexibility can be used to improve the convergence properties of the samplers. In this way, we are able to add structure that can be leveraged for improved computation without altering the fitted observed data model. The method was introduced by Meng and van Dyk (1997) and 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) introduced the terms *conditional* and *marginal* augmentation to indicate whether the working parameter is conditioned on or averaged over in the iteration; conditional augmentation essentially 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 van Dyk, 2000, van Dyk and Meng, 2001, and Imai and van Dyk, 2004). We evaluate our various proposed algorithms from the standpoint of computation time per iteration, number of iterations required for approximate convergence, and the autocorrelation in the resulting chains.

This paper proceeds as follows. The rest of Section 1 contains background material including discussion of the basic hierarchical linear models, standard Gibbs/Metropolis algorithms based on scalar or vector updating using linear regression computations, structured Markov chain Monte Carlo, and conditional and marginal augmentation. Section 2 present two improvements in computation: the parameter-expanded hierarchical model and the scalar updating algorithm with rotation, an example of conditional augmentation. Section 3 presents theoretical arguments on the computation time per iteration and convergence times for several proposed Gibbs sampler implementations, including an innovative argument based on how quickly a Markov chain can free itself from getting stuck in remote areas of the parameter space. Section 4 presents two examples, Section 5 discusses extensions to generalized linear model, and Section 6 concludes with recommendations and ideas for further research.

1.2 Notation for hierarchical 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:

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

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

Hierarchical linear models with more than two levels can be expressed in this canonical 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, Hodges, 1998, and Sargent, Hodges, and Carlin, 2006). For simplicity, we assume the mean vector β_0 is known.

The vector β can include modeled and unmodeled parameters (sometimes called "fixed and random effects," terminology we avoid for reasons described in Gelman, 2005, section 6). Elements of β that are unmodeled (for example, in a multilevel model, individual-level coefficients that do not vary by group) can be combined into a batch whose group-level variance is infinite. The notation

could equivalently be developed with modeled and unmodeled parameters treated separately (as in Laird and Ware, 1982).

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 $\sigma_z^2 = (\sigma_{zk}^2 : k = 1, \ldots, K_z)$, where we are using z as a floating subscript to stand for y or β:

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

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 vectors of indicator variables for each of the batches. (Depending on the details of the model, the actual computation may be done be selecting appropriate batches of coefficients rather than literally by matrix multiplication. The matrix notation, however, has the advantage of automatically including nonnested groupings which could not simply be handled by grouping the elements of β into ordered batches.)

1.2.2 Prior distributions on the variance parameters

We assign 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 noninformative prior distribution. There are several important special cases of this choice of prior distribution. First, the standard noninformative prior distribution, $p(\sigma_y, \sigma_\beta) \propto \Pi_{k=1}^{K_y} \sigma_y^{-1}$, 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 $\sigma_{\beta k} = 0$. See Gelman (2006) for further discussion of prior distributions for hierarchical models.

1.2.3 Extensions of the models

The model can be extended in various ways, including non-diagonal variances (Gelman et al., 1995), multivariate clustering of regression parameters (Barnard, McCulloch, and Meng, 1996; 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 the Gibbs or Metropolis algorithms. Rather than trying to present an ideal algorithm to handle all contingencies, our goal is to understand what works with an important and nontrivial basic family of models. In Section 5 we discuss extensions of our methods and algorithms to a class of generalized liner models.

1.3 Basic Gibbs samplers for the hierarchical normal model

There are two simple Gibbs sampler algorithms for the hierarchical linear model. In both algorithms, the variance parameters $\sigma = (\sigma_y, \sigma_\beta)$ 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. As Sargent, Hodges, and Carlin (2006) have shown, aspects of these two algorithms can be combined for greater efficiency; we discuss this further in Section 2. 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 $p(\beta|\sigma_y, \sigma_\beta, y)$ by running the regression of y_* on X_* with variance matrix Σ_*, where

$$y_* = \begin{pmatrix} y \\ \beta_0 \end{pmatrix}, \qquad X_* = \begin{pmatrix} X \\ I \end{pmatrix}, \qquad \text{and} \qquad \Sigma_* = \begin{pmatrix} \Sigma_y & 0 \\ 0 & \Sigma_\beta \end{pmatrix} \tag{5}$$

combine the data and prior information (Gelman et al., 1995; see also Dempster, Rubin, and Tsutakawa, 1991, and Hodges, 1998). Given Σ_*, the regression computation yields an estimate of the regression coefficient and an estimate of its variance matrix, which we label $\hat{\beta}$ and V_β, with the understanding that both are functions of the data and the variance parameters. (We sometimes write $\hat{\beta}(\sigma)$ and $R(\sigma)$ to emphasize the dependence of $\hat{\beta}$ and R on the variance components.) 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

$$\text{all-at-once Gibbs update:} \qquad \beta = \hat{\beta} + R^{-1} z, \tag{6}$$

where z is a vector of independent standard normal random variables.

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

$$\text{one-at-a-time Gibbs update: for each } j, \text{ sample} \quad \beta_j \sim p(\beta_j|\beta_{-j}, \Sigma_y, \Sigma_\beta, y), \tag{7}$$

where β_{-j} is all of β except β_j. Expression (7) 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 (7) 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 zeroes in the X matrix.

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

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

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:

$$\begin{aligned} SS_y &= W_y \text{Diag}((y-X\beta)\Sigma_y^{-1}(y-X\beta)^t), \qquad \text{or} \\ SS_\beta &= W_\beta \text{Diag}((\beta-\beta_0)\Sigma_\beta^{-1}(\beta-\beta_0)^t), \end{aligned} \tag{8}$$

with the variance matrices Σ_y and Σ_β defined in (3).

1.4 Structured Markov chain Monte Carlo

If the components of β are highly correlated in their posterior distribution, then the one-at-a-time Gibbs sampler 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 3.1.)

This tradeoff is well known and is discussed at length by Sargent, Hodges, and Carlin (2006), who advocate the all-at-once Gibbs sampler to avoid a slowly converging chain. To reduce the per-iteration computational costs, they suggest replacing Σ_* by an approximate value when updating β. The approximation may be updated periodically during the pre-convergence burn-in period, but at some point it is fixed and no longer updated to preserve the Markov property of the chain. The approximation to $\Sigma_\star$ is used in the jumping rule of a Metropolis-Hastings sampler, thus preserving the target posterior distribution as the stationary distribution of the sampler.

Optimally, we would combine the computational speed of the one-at-a-time algorithm with the fast convergence of the all-at-once algorithm. We purse two strategies that aim to accomplish this without a Metropolis-Hastings correction, at least in the hierarchical linear model. Our strategies are based on the the methods on conditional and marginal augmentation, and we conclude this section with a general introduction and example of these methods.

1.5 Conditional and marginal augmentation

Conditional and marginal augmentation (Meng and van Dyk, 1999) are a set of techniques that alter the joint distribution of the observed data and the augmented data in order to improve computation. This is done while preserving the marginal distribution of the observed data and thus the target posterior distribution. In the hierarchical linear model, β is treated as augmented data, and we consider alternative forms of $p(y, \beta|\sigma)$ that preserve $p(y|\sigma)$. We alter the joint distribution in a very particular way, and it can be shown that the resulting joint posterior distribution has the target posterior distribution of the parameter as its marginal distribution but has less posterior correlation between the parameter and the augmented data (Meng and van Dyk, 1999, van Dyk and Meng, 2001). The result is a Markov chain with reduced autocorrelation but with the target posterior distribution of the model parameters as its stationary distribution.

In order to alter the joint distribution of y and β, we introduce a one-to-one transformation $\xi = \mathcal{D}_\alpha(\beta)$ that is parameterized in terms of the so-called working parameter α. A working parameter indexes a class of joint distributions of the observed and augmented data given the model parameters in such a way that the marginal distribution of the observed data does not depend on the working parameter. In particular, the working parameter indexes a class of equivalent models, in the sense that

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

that is, $p(y|\sigma, \alpha)$ does not depend on α. The working parameter does not affect inference based on $p(y|\sigma)$. We use the notation ξ to emphasize that we are using a different joint distribution of the observed and augmented data, with the understanding that this does not affect the marginal distribution of the observed data. In the context of the hierarchical linear model we transform from β to ξ via pre-multiplication by a matrix depending on α that rescales and potentially decorrelates the components of β.

Conditional and marginal augmentation differ in their treatment of working parameters. Conditional augmentation constructs a Gibbs sampler based on the conditional distributions of

$$p(\xi, \sigma|y, \alpha) \propto p(y|\xi, \sigma, \alpha)p(\xi|\sigma, \alpha)p(\sigma),$$

where, as usual, we assume the working and model parameters are a priori independent. The working parameter is chosen to optimize or at least to improve convergence properties. It is this step that effectively reduces the posterior correlation between the parameters and and the augmented data. In contrast, marginal augmentation averages over the working prior density, $p(\alpha)$ and uses a Gibbs

sampler based on the conditional distributions of

$$\int p(\xi, \sigma, \alpha|y)d\alpha \propto p(\sigma) \int p(y|\xi, \sigma, \alpha)p(\xi|\sigma, \alpha)p(\alpha)d\alpha.$$

By eliminating the conditioning on α that is implicit in a standard Gibbs sampler, we increase the conditional posterior variance of the augmented data. This in turn allows bigger jumps and reduces the autocorrelation of the Markov chain. We illustrate how to set up and use conditional and marginal augmentation samplers in our hierarchical models in Section 2, illustrating with a simple example in Section 1.6.

When using marginal augmentation, the choice of the working prior density, $p(\alpha)$, affects the convergence behavior of the algorithm; generally more diffuse densities improve mixing. Unfortunately, improper working prior densities lead to non-positive recurrent Markov chains since $p(\alpha|y) = p(\alpha)$ and hence $p(\xi, \sigma, \alpha|y)$ is improper. With judicious choice of the improper working prior density, however, we can ensure that the 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). The basic strategy is to construct a sequence of Markov transition kernels on σ or (β, σ) each with the target posterior distribution as their stationary distribution. The sequence is further constructed so that its limiting transition kernel is the proper Markovian kernel constructed with the improper working prior density. This strategy is illustrated in the appendix.

1.6 Example: a simple hierarchical model

We illustrate the basic idea in a simple example to which we shall return in Section 4.1: a hierarchical linear model of the form,

$$y_j|\mu, \beta_j \overset{\text{indep.}}{\sim} \text{N}(\mu + \beta_j, \sigma_j^2), \quad \text{with} \quad \beta_j \overset{\text{iid.}}{\sim} \text{N}(0, \sigma_\beta^2), \tag{9}$$

with σ_j known for each j. A Gibbs sampler for this model can get stuck near $\sigma_\beta = 0$: once this group-level variance is set near zero in the chain, the β_j's are likely to be drawn near zero in their step of the Gibbs updating, at which point σ_β is likely to be sampled near zero, and so forth. We formalize this argument in Section 3.3.2; what is relevant here is that this pattern of poor convergence can be a particular problem when the true group-level variance is near zero, which is common in multilevel models after group-level predictors have been included.

The key parameter expansion step is to include a redundant multiplicative parameter:

$$y_j|\mu, \xi_j \overset{\text{indep.}}{\sim} \text{N}(\mu + \alpha\xi_j, \sigma_j^2), \quad \text{with} \quad \xi_j \overset{\text{iid.}}{\sim} \text{N}(0, \sigma_\xi^2), \tag{10}$$

where the regression parameters of interest are now $\beta_j = \alpha\xi_j$, with group-level standard deviation $\sigma_\beta = |\alpha|\sigma_\xi$. As has been shown by van Dyk and Meng (2001) and Liu (2003), and also by Liu, Rubin, and Wu (1998) in the context of the EM algorithm, computation of model (10) is straightforward.

The likelihood function, $p(y_j|\mu,\sigma_\beta)$ obtained by integrating out the ξ_j's, is the same as that obtained by integrating out the β_j's under (9). Since we do not alter the prior distribution on (μ,σ_β), we can fit either model to obtain a Monte Carlo sample from the same posterior distribution $p(\mu,\sigma_\beta|y)$. The added parameter α is not necessarily of any statistical interest, but expanding the model in this way can allow faster convergence of the Gibbs sampler.

2 Algorithms using marginal and conditional augmentation

2.1 Motivation and notation for the expanded model

As we discuss in detail in Section 3.3.2, any of the Gibbs samplers discussed in Section 1 can be slow to converge when the 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 deviation 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 (8)). Ultimately, the stochastic nature of the Gibbs sampler allows it to escape this trap but this may require many iterations.

Van Dyk and Meng (2001) show how the method of marginal augmentation can substantially improve convergence in this setting; see also Meng and van Dyk (1999), Liu, Rubin, and Wu (1998), van Dyk (2000), and Liu (2003). In order to introduce the working parameter into the standard model given in (2), we follow the parameterization of Liu, Rubin, and Wu (1998), which in our notation is

$$A = [\text{Diag}(W_\beta\alpha)]^{-1}, \tag{11}$$

where W_β is the matrix of 0's and 1's from (3) that indicates the batch corresponding to each β_j, and α is a $(K_\beta \times 1)$ vector of working parameters. Setting $\xi = A\beta = [\text{Diag}(W_\beta\alpha)]^{-1}\beta$ yields the reformulated model,

$$\begin{aligned} y|\xi,\alpha &\sim \text{N}(XA^{-1}\xi, \Sigma_y) && (12)\\ \xi|\alpha &\sim \text{N}(\xi_0, \Sigma_\xi) && (13)\\ \alpha &\sim \text{N}(\alpha_0, \Sigma_\alpha), && (14) \end{aligned}$$

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: $\Sigma_\xi = \text{Diag}(W_\beta(\sigma_\beta/\alpha)^2)$, where * represents componentwise multiplication. In the limit as $\Sigma_\alpha \to \infty$, the parameter α becomes non-identified, but the marginal distribution (β,σ_β) is still defined. As we discuss in Section 1.5, care must be taken when constructing samplers under this limiting working prior distribution.

The transformation $\xi = A\beta$ introduces one parameter α_k for each of the hierarchical variance parameters $\sigma^2_{\beta k}$. We expect samplers constructed using the appropriate conditional distributions of (12)–(14) to perform better than the corresponding sampler constructed with (1)–(2). Indeed, we shall see that, with this working parameter, the Gibbs sampler is less prone to move slowly in regions of the parameter space with $\sigma_{\beta k}$ near zero.

Adding the new parameter α potentially generalizes the hierarchical model in an interesting direction—allowing interactions among the parameters (which is different than interactions among the regression predictions); see Gelman (2004). Our primary purpose in this paper, however, is to use the expanded model to improve computational efficiency.

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

The expanded model can be viewed as a generalization of the usual hierarchical regression model on β, expanding the class of prior distributions on the variance parameter σ_β (as described in Gelman, 2006, for the one-variance-parameter case). When the prior distribution on the additional parameter α is proper, one can construct a Gibbs sampler on the expanded space and marginalize by simply focusing on the inferences for the β, σ_β. When α is assigned an improper prior distribution, it will also have an improper posterior distribution, and we have to be more careful in defining a Gibbs sampler with the appropriate marginal distribution on the identified parameter.

The sampling distribution $p(y|\sigma)$ implied by (12)–(14) is the same as that implied by (1)–(2) for any value *fixed* value of α or if we average over any *proper* working prior distribution on α. In this sense the models are equivalent and α meets the definition of a working parameter. When drawing inferences under model (12)–(14), however, we must account for the parameters' transformation:

Model (1)–(2)	Model (12)–(14)
the vector β	the vector $(W_\beta \alpha) * \xi$
a single 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 (12)–(14) for computational efficiency, then the quantities on the right side of the above table should be reported. For each batch k of regression coefficients, the coefficients ξ_j and the variance parameters $\sigma_{\xi k}$ sampled under model (12)–(14) must be multiplied by the corresponding sampled scale parameter α_k so that they can be given their proper interpretations as β_j's and $\sigma_{\beta k}$ in the original model (1)–(2).

In fact, with these prior distributions the parameters α_k and $\sigma_{\xi k}$ cannot be separately identified and only affect the final inferences under the model through their product. If proper prior distributions are assigned, then α and σ_ξ can be interpreted separately as part of a multiplicative-parameter

model (see Gelman, 2004, section 5.2).

2.3 Gibbs sampler computation for the extended model

Under a proper working prior distribution, there are many possible implementations of the Gibbs sampler under the expanded model. For example, we can iteratively sample $(\alpha, \sigma_y, \sigma_\xi)$ and ξ from their respective complete conditional distributions. Given ξ, we can express equation (12) as

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

This is a normal linear regression of y on the columns of the known design matrix, $X\mathrm{Diag}(\xi)W_\beta$ with regression coefficient α. In our model as presented with a diagonal data variance matrix (and, more generally, with sufficient structure on Σ_y) we can sample σ_y from $p(\sigma_y|\xi, y)$. (Likewise, we can independently sample σ_ξ using (14).) Given Σ_y, a conditional posterior simulation of α can be obtained by running a standard regression program (augmenting the data vector, design matrix and the variance matrix to include the Gaussian prior distribution) to obtain an estimate $\hat{\alpha}$ and covariance matrix, then drawing from the multivariate normal distribution centered at that estimate and with that covariance matrix (see, e.g., Gelman et al., 1995). The multiplicative parameter α is typically a relatively short vector, and so we are not bothered by simply updating it in vector fashion using linear regression computations.

Similarly, to update ξ, we express (12) as

$$y \sim \mathrm{N}\left(X\mathrm{Diag}(W_\beta\alpha)\xi,\, \Sigma_y\right) \tag{16}$$

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

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

Rather than sampling α (or equivalently, A) in the iteration, the method of conditional augmentation aims to fix the working parameter at some optimal or nearly optimal value in terms of the convergence properties of the resulting sampler. For conditional augmentation, rather than sampling $A = [\mathrm{Diag}(W_\beta\alpha)]^{-1}$, we simply allow it to be any fixed matrix. Thus, α amounts to an index on a class of transformations, and, we can simply consider ξ to be a transformation of β.

Suppose we do not sample A in the iteration, but draw the variance components jointly given ξ and y and draw the components of ξ one at a time given the variance components and y. If we take A to be an identity matrix, we recover the one-at-a-time Gibbs sampler, so this formulation constitutes a generalization of that in Section 1.3. Since we choose A to 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 (6)) at each

iteration, which orthogonalizes the elements of ξ and results 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—that is, 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) or an initial run of the Gibbs sampler. Given $\hat{\sigma}_0$, we can run the regression just once, to compute A, and then run the Gibbs sampler conditional on 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. (We expect this to typically be a useful strategy since the design matrix X and the fixed structure of the covariance matrices Σ_* do not change as the simulation progresses, and this induces stability in R.)

As with structured MCMC, we aim to develop a sampler with the low autocorrelation of the all-at-once Gibbs sampler but with the quick per-iteration computation time of the one-at-a-time Gibbs sampler. Also as with structured MCMC, our strategy involves an approximation to the conditional posterior variance matrix of the regression coefficients. An advantage of our proposal, however, is that it does not require a Metropolis-Hastings correction to adjust for the approximation. The relative computational performance of the two strategies, nonetheless, depends on the quality of the approximations.

A natural modification is to set A to the posterior mean of R, which can be computed for example by updating occasionally (for example, every 200 steps) on line; for example, set $A^{(0)} = R(\hat{\sigma}_0)^{-1}$ and then, for $t = 200, 400, \ldots,$

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

This updating would take place only during the burn-in period (typically half the total length of the chain; see Gelman and Rubin,1992) so as not to interfere with the Markov property of the ultimate chain.

2.5 Setting up the computation

We perform all computations in terms of the regression model (12)–(14), without assuming any special structure in the design matrix X. The algorithms must begin with estimates of the variance parameters σ 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 initial values that are "over dispersed": That is, from a distribution that includes and is more variable than the target 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). When using over-dispersed starting values, it becomes particularly important to consider the behavior of the algorithms when they are started far from the posterior mode, as we discuss in Section 3.3.

3 Theoretical arguments

The previous section introduced augmentation algorithms and suggested why they should be able to speed convergence by reducing dependence among regression coefficients (by a matrix multiplication that is equivalent to rotating the parameter space) and reducing dependence between batches of coefficients and their variance parameters (using multiplicative redundant parameters). After a brief look at computation time, we provide theoretical arguments to show the potential gains from these ideas in Gibbs sampling. The most detailed part of our treatment is in Section 3.3 of the convergence of the expanded algorithm, because this is where we are providing a novel discussion of the algorithm's ability to avoid being stuck near the boundary the of parameter space.

3.1 Computation time per iteration

The QR decomposition of the W matrix and backsolving the two upper-triangular systems that are required for the vector updating algorithms take $O(2nm^2)$ 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 transforming 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 negligible 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.

3.2 Convergence 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 (5) also increase, 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 2.4 will do a good job of making the components of ξ close to independent. Thus, *increasing the number of regression coefficients and the number of parameters per batch (the $n_{\beta k}$) 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)). In this discussion we focus on the rotation-to-approximate-independence described in Section 2.4. Thus ξ refers to the transformations of β described in Section 2.4.

We can understand the efficiencies of vector and scalar updating by considering three scenarios involving inference about the variance parameters.

First suppose all the variance parameters are known. Then the vector updating of ξ converges in a single step, meaning that the Gibbs sampler draws are equivalent to independent draws from the posterior distribution of ξ. In contrast, the speed of the untransformed scalar updating algorithm 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), but in general finding a good transformation can require matrix operations that can be slow in high dimensions. Without such a rotation, the computational savings from the scalar updating algorithm may be lost if they require many more iterations to converge. Finally, the transformed scalar updating algorithm will converge in one step, and be just as efficient as the vector algorithm.

Second, consider a setting where the variance parameters are unknown but are accurately determined by the posterior distribution. This should be the case in large data sets with many observations per group and many groups. 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 uncertainty in Σ_y and Σ_ξ.

Finally, 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 during the burn-in period based on current values of Σ_y and Σ_ξ in the simulation, as discussed in Section 2.4.

3.3 Effect of parameter expansion on convergence rate

Because the parameter expansion operates independently for each hierarchical variance component k, we consider its effect on the convergence of these components separately. Ideally, the marginal augmentations methods can outperform the standard Gibbs sampler in three settings:

- $\sigma_{\beta k}$ near the posterior mode (relevant in problems for which $\sigma_{\beta k}$ is well estimated in the posterior distribution),
- $\sigma_{\beta k}$ near 0 (relevant because of possible extreme starting points and also for problems in which $\sigma_{\beta k}$ has a non-trivial posterior probability of being near 0),
- $\sigma_{\beta k}$ near ∞ (relevant because of possible extreme starting points and also for problems in which $\sigma_{\beta k}$ has a non-trivial posterior probability of being large).

For each of these cases, we consider first the speed of the corresponding EM algorithms and then consider the Gibbs sampler. Because EM algorithms are deterministic computing their speed of convergence is easier than for Gibbs samplers. Due to the similarity in the structure of the two classes of algorithms, however, the convergence properties of EM algorithms can emulate those of Gibbs samplers (e.g., van Dyk and Meng, 2001). 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, $\sigma_\beta = \alpha\sigma_\xi$, 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:

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

The Gibbs sampler for this model is:

$$\begin{aligned} \xi_j &\leftarrow \text{N}\left(\frac{1}{\alpha}\frac{y_j}{1+\frac{1}{\sigma_\beta^2}}, \frac{1}{\alpha^2}\frac{1}{1+\frac{1}{\sigma_\beta^2}}\right), \qquad \text{for } j = 1, ..., m \\ \alpha &\leftarrow \text{N}\left(\frac{\sum_{j=1}^m \xi_j y_j}{\sum_{j=1}^m \xi_j^2}, \frac{1}{\sum_{j=1}^m \xi_j^2}\right) \\ \sigma_\xi^2 &\leftarrow \sum_{j=1}^m \xi_j^2/\chi_{m-1}^2. \end{aligned} \tag{17}$$

(In this case, the vector and the componentwise algorithms are identical since the ξ's are independent conditional on the hyperparameters.) The EM updating is simply,

$$\begin{aligned} \alpha &\leftarrow \alpha\frac{s_y^2}{\frac{1}{1+\frac{1}{\sigma_\beta^2}}s_y^2 + 1} \\ \sigma_\xi^2 &\leftarrow \frac{1}{\alpha^2}\left(\frac{1}{1+\frac{1}{\sigma_\beta^2}}\left(\frac{1}{1+\frac{1}{\sigma_\beta^2}}s_y^2 + 1\right)\right), \end{aligned} \tag{18}$$

where

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

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

3.3.1 Speed of the algorithms near the posterior mode

For $\sigma_{\beta k}$ near the posterior mode (assuming that mode is not 0), the speed of the two-step Gibbs sampler is roughly the same as the corresponding EM algorithm (Liu, 1994), with convergence rate depending on the covariance matrix of the approximate Gaussian distribution. 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_k fixed at 1).

3.3.2 Speed of the algorithms for σ_β near 0

For $\sigma_{\beta k}$ 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 parameterization, α is fixed at 1, and only $\sigma_\beta = \sigma_\xi$ is updated. For σ_β near 0, we can express (18) as

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

Thus, even the *relative* change in σ_β at each step approaches 0 as $\sigma_\beta \to 0$. If $s_y^2 \leq 1$, this means that σ_β will approach 0 at a slower-than-linear rate; if $s_y^2 > 1$ and σ_β happens to be started near 0, then it will move away from 0 hopelessly slowly.

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

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

which is linearly convergent, which should be acceptable in practice as long as σ_β 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) $\mathrm{E}(\sum_{j=1}^m \xi_j^2)$ is replaced by the random variable $\sum_{j=1}^m \xi_j^2$, and (2) division by a χ_m^2 random variable in the update of σ_β^2. (Under the conventional parameterization, the updating of α in (18) does not occur since α is fixed at 1.) When the current value of σ_β 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 σ_β^2; so combined they give the random updating of σ_β a coefficient of variation of $1/m$. Within the context of the usual parameterization, with σ_β near 0, this variation acts like a random walk, and it implies that the Gibbs sampler will require on the order of $m(\log(\sigma_{\beta 0}))^2$ iterations to "escape" from a starting $\sigma_{\beta 0}$ near 0. (See Rosenthal, 1995, for more formal versions of this argument.)

For the parameter-expanded algorithm, the Gibbs sampler from (17) adds another random component through the updating of α. The updated $\sigma_\beta = |\alpha|\sigma_\xi$ is essentially a least-squares regression coefficient plus noise (the Gibbs update for α), divided by the square root of a χ_{n-1}^2 random variable (the Gibbs update for σ_ξ). This added random component causes the draw to be less dependent on the previous iteration. The updated σ_β can then be written as

$$\sigma_\beta \leftarrow \left| \frac{\sum_{j=1}^m \gamma_j y_j}{\sqrt{\sum_{j=1}^m \gamma_j^2}} + z \right| / x, \tag{19}$$

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

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

and the following random variables are independent:

$$z \sim \mathrm{N}(0,1), \qquad x^2 \sim \chi_{m-1}^2, \qquad z_j \sim \mathrm{N}(0,1), \qquad \text{for } j = 1, ..., m. \tag{20}$$

The random variables in (19) induce a distribution for σ_β on the left of (19) that depends only on m, $s_y^2 = \frac{1}{m}\sum_{j=1}^m y_j^2$, and the previous value of σ_β. For σ_β near 0, this distribution has a standard deviation on the order of $1/\sqrt{m}$ *on the absolute scale*. Therefore, it is impossible for σ_β to get stuck at values less than $1/\sqrt{m}$, no matter how small its value was in the previous iteration. This is a stunning improvement over the Gibbs sampler for the conventional parameterization.

This jumpy behavior of the parameter-expanded algorithm near $\sigma_\beta = 0$ is reminiscent of the nonzero lowest energy state in quantum mechanics, which is related to the uncertainty principle—if σ_β is precisely localized near 0, then the "momentum" of the system will be high (in that σ_β 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 σ_β is determined by the number of coefficients m in the batch.

3.3.3 Speed of the algorithms for σ_β near ∞

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

4 Examples

We consider two examples that have arisen in applied research. For each example and each algorithm, we compute computation time per iteration, number of iterations required for 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).

4.1 Examples and algorithms

Our first example is a hierarchical 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 β_j in that school; a regression analysis applied to the data from that school alone yielded an estimate of the treatment effect and its standard error, which we label y_j and σ_j, respectively. We model the data using the two-level Gaussian model (9). The sample size within each school is large enough to justify the Gaussian assumption in level one of (9). We use a noninformative uniform hyperprior density, $p(\mu, \sigma_\beta) \propto 1$ (see Gelman, 2006, for discussion of alternative prior distributions). Interest lies in the individual school parameters β_j and also in the hyperparameters (μ, σ_β); we label the set of unknown parameters $\gamma = (\beta, \mu, \sigma_\beta)$. 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 non-iteratively by first computing the marginal posterior density for σ_β at a grid of points and then simulating the parameters in the order σ_β, μ, β (Rubin, 1981).

We consider two standard Gibbs samplers that can be constructed to sample the posterior distribution under this model. The first updates the mean parameters jointly, or as a vector; we refer to this sampler as the V-Sampler.

V-Sampler:

Step 1: Sample $(\mu^{(t+1)}, \beta^{(t+1)})$ jointly from $p\left(\mu, \beta \;\middle|\; \sigma_\beta = \sigma_\beta^{(t)}, y\right)$.

Step 2: Sample $\sigma_\beta^{(t+1)}$ from $p\left(\sigma_\beta \mid \mu = \mu^{(t+1)}, \beta = \beta^{(t+1)}, y\right)$.

A second sampler, updates each of the mean components separately from their respective complete conditional distributions; we call this sampler the S-Sampler.

S-Sampler:

Step 1: Sample $\mu^{(t+1)}$ from $p\left(\mu \;\middle|\; \beta = \beta^{(t)}, \sigma_\beta = \sigma_\beta^{(t)}, y\right)$.

Step 2: Sample $\beta_j^{(t+1)}$ from $p\left(\beta_j \;\middle|\; \mu = \mu^{(t+1)}, \beta_{-j} = \beta_{-j}^{(t+1)}, \sigma_\beta = \sigma_\beta^{(t)}, y\right)$, for $j = 1, \ldots, J$, where $\beta_{-j}^{(t+1)} = \left(\beta_1^{(t+1)}, \ldots, \beta_{j-1}^{(t+1)}, \beta_{j+1}^{(t)}, \ldots, \beta_J^{(t)}\right)$

Step 3: Sample $\sigma_\beta^{(t+1)}$ from $p\left(\sigma_\beta \;\middle|\; \mu = \mu^{(t+1)}, \beta = \beta^{(t+1)}, y\right)$.

A possible difficulty for both of these Gibbs samplers is that the maximum likelihood estimate of the hierarchical scalar parameter, σ_β, is zero, There is also a substantial portion of the posterior density near $\sigma_\beta = 0$, and there is the possibility that the algorithm may move slowly in this region.

To improve computational efficiency, we consider two marginal Gibbs samplers that are based on a working parameter α that is introduced via $\xi = \beta/\alpha$. Introducing this transformation into (9), the model becomes (10), where $\sigma_\xi = \sigma_\beta/|\alpha|$. We can fit either model (9) or (10) to obtain a Monte Carlo sample from the same posterior distribution $p(\mu, \sigma_\beta|y)$. In the appendix we further show that with the improper prior distribution specified below on α, all the samplers return a sample from the same target joint posterior distribution $p(\beta, \mu, \sigma_\beta|y)$.

To derive the samplers under model (10), we begin with the proper working prior distribution, $\alpha \sim$ Inv-gamma(ξ_0, ξ_1), which implies $p(\mu, \sigma_\xi^2, \alpha) \propto \alpha^{-\xi_0} \exp(-\xi_1/\alpha)/\sigma_\xi$. We then construct two Gibbs sampler on the joint proper posterior distribution $p(\mu, \xi, \sigma_\xi, \alpha|y)$; the two samplers use updating schemes that correspond to the V-Sampler and the S-Sampler, respectively. The chains are constructed so that the marginal chains $\{\gamma^{(t)}, t = 1, 2, \ldots\}$ with $\gamma = (\mu, \beta, \sigma_\beta)$ are themselves Markovian. We then take the limit of the transition kernels for the two chains as the working prior distribution becomes improper. We call these limiting kernels the V+PX-Sampler an the S+PX-Sampler, respectively. Here we state these limiting samplers; we prove that their stationary distributions are both the target posterior distribution, $p(\gamma|y)$ in the appendix.

V+PX-Sampler:

Step 1: Sample $(\mu^{(t+1)}, \beta^\star)$ jointly from $p\left(\mu, \beta \;\middle|\; \sigma_\beta = \sigma_\beta^{(t)}, y\right)$; this is the same distribution as is sampled in Step 1 of the V-Sampler.

Step 2: Sample $\sigma_\beta^\star$ from $p\left(\sigma_\beta \;\middle|\; \mu = \mu^{(t+1)}, \beta = \beta^\star, y\right)$; this is the same distribution as is sampled in Step 2 of the V-Sampler.

Step 3: Sample α from $p\left(\alpha \;\middle|\; \mu = \mu^{(t+1)}, \xi = \beta^\star, \sigma_\xi = \sigma_\beta^\star, y\right)$; that is, sample

$$\alpha \sim \mathrm{N}\left[\frac{\sum_{j=1}^J \beta_j^\star (y_j - \mu^{(t+1)})/\sigma_j^2}{\sum_{j=1}^J (\beta_j^\star)^2/\sigma_j^2}, \left(\sum_{j=1}^J (\beta_j^\star)^2/\sigma_j^2\right)\right] \tag{21}$$

Step 4: Compute $\beta^{(t+1)} = \alpha\beta^\star$ and $\sigma_\beta^{(t+1)} = |\alpha|\sigma_\beta^\star$.

Here we use a star in the superscript to represent intermediate quantities that are not part of the transition kernel.

The S+PX-Sampler begins with Step 1 and Step 2 of the S-Sampler, but records what is recorded as $\beta^{(t+1)}$ and $\sigma_\beta^{(t+1)}$ in the S-Sampler as $\beta^\star$ and $\sigma_\beta^\star$, respectively. The iteration is completed with Step 3 and Step 4 of the V+PX-Sampler.

In the second example, we consider the forecast skill of three 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 let y_{jt} and $x_{jt}^{m,k}$ represent observed and the k^{th} predicted precipitation anomaly for the m^{th} GCM ensemble (out of $M = 3$ models) in box j 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 a Bayesian framework:

$$y_{jt} = \beta_j + \delta_t + \sum_{m=1}^M \beta_j^m \bar{x}_{jt}^m + \epsilon_{jt}, \quad \text{for } j = 1, \ldots, 527 \text{ and } t = 1, \ldots, 41. \tag{22}$$

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.

- β_j is an offset for location j; we assign it a normal population distribution with mean μ_β and standard deviation σ_β.
- 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

In this example, our parameter expansion model is

$$y_{jt} = \alpha_\beta \tilde{\beta}_j + \alpha_\delta \tilde{\delta}_t + \sum_{m=1}^{M} \alpha_{\beta^m} \tilde{\beta}_j^m \bar{x}_{jt}^m + \epsilon_{jt}, \quad \text{for } j = 1, \ldots, 527 \text{ and } t = 1, \ldots, 41, \tag{23}$$

using the notation $\tilde{\beta}$, etc., for unscaled parameters (so that $\beta_j = \alpha_\beta \tilde{\beta}_j$ for each j; $\delta_t = \alpha_\delta \tilde{\delta}_t$ for each t; and so forth). We assign uniform prior distributions on the (unidentifiable) multiplicative parameters α and the mean and scale hyperparameters for the prior distributions on the $\tilde{\beta}$'s, $\tilde{\delta}$'s, and so forth. We compute the samplers for the expanded models and then save the simulations of the parameters as defined in (22). For this problem it was most direct to simply set up the model in the expanded parameter space.

4.2 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 represents scalar updating without parameter expansion, V+PX represents vector updating with parameter expansion, and S+PX represents scalar updating with parameter expansion. The dots display the combination of the average computation time per iteration and the average number of iterations required for convergence for each algorithm. The lines represent indifference curves for total computation time until convergence to the target distribution. These are straight 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 per Gibbs sampler chain; the second is the vector updating with parameter expansion, which takes 0.69 seconds; the

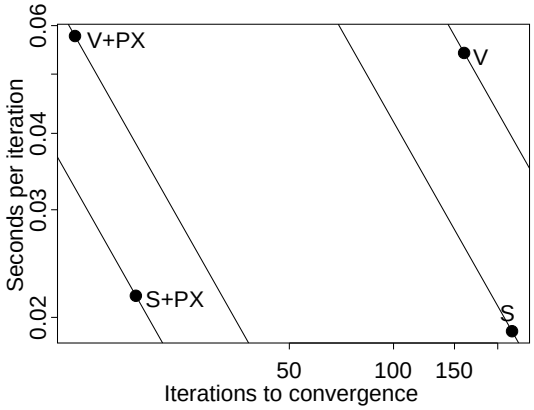

Figure 1: Computation time of the four algorithms for the eight schools example (plotted on a logarithmic scale). V: vector updating, S: scalar updating, V+PX: vector updating with parameter expansion, and S+PX: scalar updating with parameter expansion. The dots display the combination of the computation time per iteration and the average number of iterations required for approximate convergence of each algorithm. The lines are indifference curves in total computation time for convergence to the stationary distribution.

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 22 times faster than the traditional algorithms.

Figure 2 shows the simulation efficiency (measured by the autocorrelations and the sum of their absolute values) for the variance component σ_β. 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 large, 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 per iteration in our simulation was similar with or without parameter expansion. However, the scalar updating with parameter expansion only needed 400 iterations to reach convergence, compared 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). The most efficient algorithm is scalar updating with parameter expansion, which takes an average of 500 seconds per chain in total computation time compared to 13,000 seconds per chain for scalar updating.

5 Generalized linear models

In this section we discuss how the methods developed in Sections 2 and 3 for hierarchical linear models can be extended to hierarchical generalized linear models, so that (1) is replaced by

$$y|\beta \sim \text{glm}(X\beta) \quad \text{or} \quad y|\beta \sim \text{glm}(X\beta, \Sigma_y); \tag{24}$$

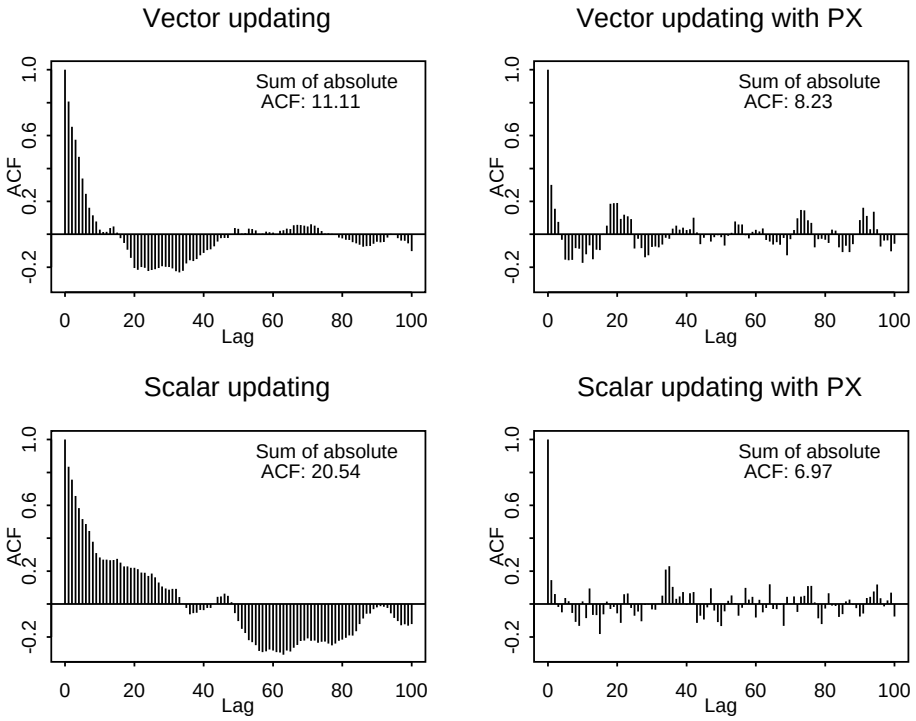

Figure 2: Simulation efficiency (measured by the autocorrelations) of variance component σ_β for the eight schools example. The sum of the absolute autocorrelation functions (ACF) are given and are roughly equal to the expected sample sizes (ESS).

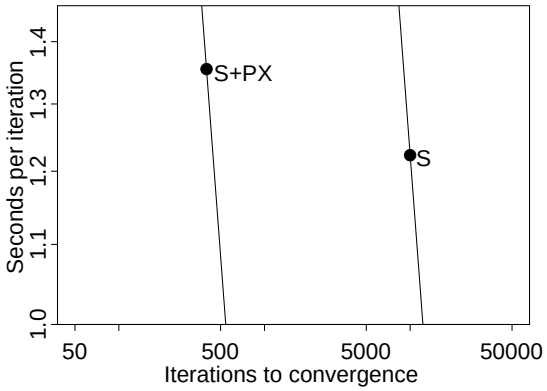

Figure 3: Computation time (on a logarithmic scale) of the two algorithms for the climate modeling example. S: scalar updating and S+PX: scalar updating with parameter expansion. The dots display the combination of the computation time per iteration and the iterations required for convergence for each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

the simpler form is used for models such as the binomial and Poisson with no free variance parameters. We use the general "glm" notation to include the necessary probability models and link functions (McCullagh and Nelder, 1989).

5.1 A Metropolis adaptation to the simple Gibbs samplers

For a hierarchical generalized linear model, the simple all-at-once and one-at-a-time Gibbs samplers described in Section 1.3 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 linearizion 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.

5.2 Parameter expansion for generalized linear models

We can use marginal augmentation for the generalized linear model, if we simply replace (12) by

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

with the equivalences to the original model as described in Section 2.2. As in Section 2.3 we use a proper working prior distribution and again the Gibbs sampler for the new model must include a step to update α. Since the model is non-conjugate, this can be performed by Metropolis jumping, or via a linearizion of the model $y \sim \text{glm}(X((W_\xi\alpha)*\xi),\Sigma_y)$—considered as a likelihood for α—followed by a draw from the corresponding approximate Gaussian conditional prior distribution for α and a Metropolis-Hastings accept/reject step. Computation for ξ can similarly be performed using a Metropolis jump or a Metropolis-approximate-Gibbs. In either case, we want an approximate transformation to independence (as in Section 2.4), whether for scaling the Metropolis proposal distribution or approximating the Gibbs sampler. Finally, the variance parameters can be updated using the Gibbs sampler as with the normal model, since they are linked to the other model parameters through the prior distribution, not the likelihood.

5.3 Adaptation of the one-at-a-time algorithms

If the components of β are highly correlated in their posterior distribution, then the Metropolis-Hastings sampler corresponding to the one-at-a-time Gibbs sampler can move slowly. To improve the

sampler, we can adapt the rotation method based on an approximation to the posterior covariance matrix of the regression parameters. In particular, 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 non-conjugate 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).

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 also suspect that the general ideas of approximate rotation for correlated parameters and parameter expansion for variance components will be useful in more elaborate settings such as multivariate and nonlinear models.

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, and more recently called "redundant parameterization" in the context of multilevel models (Gelman and Hill, 2006). 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 2.3. 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 is another useful model generalization that was originally motivated for computational reasons (see Gelman, 2006).

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-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 3.3.2), 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 has the problem of zero estimates. It should be possible to develop a fast and reliable algorithm to find a reasonably over-dispersed 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, σ_β) used to compute the estimated covariance matrix R_0 required for the scalar updating algorithms with transformation. Third, the ideas presented here should be generalizable to multivariate models as arise, for example, in decompositions of the covariance matrix in varying-intercept, varying-slope models (O'Malley and Zaslavsky, 2005, MacLehose et al., 2006). Finally, as discussed by van Dyk and Meng (2001) and Liu (2003), the parameter expansion idea appears open-ended, which makes us wonder what further improvements are possible for simple as well as for complex hierarchical models.

References

Barnard, J., McCulloch, R., and Meng, X. L. (1996). Modeling covariance matrices in terms of standard deviations and correlations, with application to shrinkage. *Statistica Sinica* **10**, 1281–1311.

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 heteroscedasticity. *Advances in Econometrics* **11A**, 87–109.

Carlin, B. P., and Louis, T. A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis*, second 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* **94**, 1254–1263.

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. (2004). Parameterization and Bayesian modeling. *Journal of the American Statistical Association* **99**, 537–545.

Gelman, A. (2005). Analysis of variance: why it is more important than ever (with discussion). *Annals of Statistics* **33**, 1–53.

Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*.

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

Gelman, A., and Hill, J. (2006). *Data Analysis Using Regression and Multilevel (Hierarchical) Models*. Cambridge University Press.

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 (with discussion). *Journal of the Royal Statistical Society B* **60**, 497–536.

Laird, N. M., and Ware, J. H. (1982). Random-effects models for longitudinal data. *Biometrics* **38**, 963–974.

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 American Statistical Association*.

Liu, J. S. (1994). The Fraction of Missing Information and Convergence Rate for Data Augmentation. In *Computing Science and Statistics. Computationally Intensive Statistical Methods. Proceedings of the 26th Symposium on the Interface*, Interface Foundation of North America (Fairfax Station, VA), 490–497.

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.

MacLehose, R. F., Dunson, D. B., Herring, A., and Hoppin, J. A. (2006). Bayesian methods for highly correlated exposure data. Technical report, Department of Epidemiology, University of North Carolina.

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.

O'Malley, A. J., and Zaslavsky, A. M. (2005). Cluster-level covariance analysis for survey data with structured nonresponse. Technical report, Department of Health Care Policy, Harvard Medical School.

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.

Sargent, D. J., Hodges, J. S., and Carlin, B. P. (2000). Structured Markov chain Monte Carlo. *Journal of Computational and Graphical Statistics* **9**, 217–234.

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.

A Verifying the stationary distribution of samplers based on marginal augmentation

We consider Example 1 of Section 4.1. To prove that $p(\gamma|y)$ is the stationary distribution of both the V+PX-Sampler and the S+PX-Sampler we use Lemma 2 of Liu and Wu (1999). The Lemma states that if a sequence of proper Markovian transition kernels each have the same stationary distribution, and if the sequence converges to a proper Markovian transition kernel, the limiting kernel has the same stationary distribution as the sequence. We construct the sequence of proper Markovian transition kernels using a sequence of proper prior distributions on α, $p_n(\alpha)$, that converge to the improper prior distribution, $p_\infty(\alpha) \propto 1/\alpha$. Namely, we use $\alpha \sim$ Inv-gamma(ω_n, ω_n) where $\omega_n \to 0$ as $n \to \infty$.

Consider the following transition kernel constructed under some proper $p_n(\alpha)$.

Proper V+PX-Sampler

Step 1: Sample $(\mu^{(t+1)}, \xi^\star, \alpha^\star)$ from $p_n\left(\mu, \xi, \alpha \;\middle|\; \sigma_\beta = \sigma_\beta^{(t)}, y\right)$.

Step 2: Sample $\sigma_\xi^\star$ from $p\left(\sigma_\xi \;\middle|\; \mu = \mu^{(t+1)}, \xi = \xi^\star, \alpha = \alpha^\star, y\right)$.

Step 3: Sample $\alpha^{(t+1)}$ from $p_n\left(\alpha \;\middle|\; \mu = \mu^{(t+1)}, \xi = \xi^\star, \sigma_\xi = \sigma_\xi^\star, y\right)$

Step 4: Set $\beta^{(t+1)} = \alpha^{(t+1)}\xi^\star$ and $\sigma_\beta^{(t+1)} = |\alpha^{(t+1)}|\sigma_\xi^\star$.

Here we use the subscript n to emphasize the dependency of certain conditional distributions on the choice of $p_n(\alpha)$. For any proper $p_n(\alpha)$, the stationary distribution of this sampler is $p(\gamma, \alpha|y) = p(\gamma|y)p_n(\alpha)$. Moreover, the marginal chain $\{\gamma^{(t)}, t = 1, 2, \ldots\}$ is Markovian with stationary distribution $p(\gamma|y)$ *for any* proper $p_n(\alpha)$. Thus, in order to establish that the stationary distribution of V+PX-Sampler is also $p(\gamma|y)$, we need only show that the limit of the sequence of

transition kernels constructed using the PROPER V+PX-SAMPLER is the Markovian kernel given in the V+PX-SAMPLER.

PROOF: STEP 1 and STEP 2 of the PROPER V+PX-SAMPLER can be rewritten as

- Sample $\alpha^\star$ form $p_n\left(\alpha \mid \sigma_\beta^{(t)}, y\right) = p_n(\alpha)$.
- Sample
$$(\mu^{(t+1)}, \beta^\star) \text{ jointly from } p\left(\mu, \beta \mid \sigma_\beta^{(t)}, y\right). \tag{25}$$
- Sample
$$\sigma_\beta^\star \text{ from } p\left(\sigma_\beta \mid \mu = \mu^{(t+1)}, \beta = \beta^\star, y\right). \tag{26}$$
- Set $\xi^\star = \beta^\star/\alpha^\star$ and $\sigma_\xi^\star = \sigma_\beta^\star/|\alpha^\star|$.

Only the draw of $\alpha^\star$ depends on $p_n(\alpha)$.

To analyze STEP 3 in the limit, we note that because $p_n(\alpha) \to p_\infty(\alpha)$, $p_n(\mu, \xi, \sigma_\xi, \alpha|y)$ converges to $p_\infty(\mu, \xi, \sigma_\xi, \alpha|y)$, the (improper) posterior distribution under $p_\infty(\alpha)$. Thus, by Fatou's lemma, the corresponding conditional distributions also converge, so that, $p_n(\alpha|\mu, \xi, \sigma_\xi, y) \to p_\infty(\alpha|\mu, \xi, \sigma_\xi, y)$. Thus, given $(\mu^{(t+1)}, \xi^\star, \alpha^\star)$, STEP 3 converges to the sampling from the proper distribution

$$\begin{aligned}
\alpha^{(t+1)} &\sim \mathrm{N}\left[\frac{\sum_{j=1}^J \xi_j^\star (y_j - \mu^{(t+1)})/\sigma_j^2}{\sum_{j=1}^J (\xi_j^\star)^2/\sigma_j^2}, \left(\sum_{j=1}^J (\xi_j^\star)^2/\sigma_j^2\right)\right] \\
&= \alpha^\star \mathrm{N}\left[\frac{\sum_{j=1}^J \beta_j^\star (y_j - \mu^{(t+1)})/\sigma_j^2}{\sum_{j=1}^J (\beta_j^\star)^2/\sigma_j^2}, \left(\sum_{j=1}^J (\beta_j^\star)^2/\sigma_j^2\right)\right],
\end{aligned} \tag{27}$$

Notationally, we refer to the normal random variable in (27) as α, i.e., $\alpha^{(t+1)} = \alpha^\star \alpha$.

Finally, in STEP 4, we compute $\beta^{(t+1)}$ and $\sigma_\beta^{(t+1)}$ which in the limit simplifies to

$$\beta^{(t+1)} = \alpha^{(t+1)}\xi^\star = \alpha\beta^\star \text{ and } \sigma_\beta^{(t+1)} = |\alpha^{(t+1)}|\sigma_\xi^\star = |\alpha|\sigma_\beta^\star. \tag{28}$$

Thus, under the limiting kernel, $\gamma^{(t+1)}$ does not depend on $\alpha^\star$ and we do not need to compute $\alpha^\star$, $\xi^\star$, $\sigma_\xi^\star$, or $\alpha^{(t+1)}$. Thus the iteration consists of sampling steps given in (25), (26), and (21), and computing (28). But this is exactly the transition kernel given by the V+PX-SAMPLER. ■

A similar strategy can be used to verify that the S+PX-SAMPLER is the proper Markovian limit of a sequence of proper Markovian transition kernels each with stationary distribution equal to $p(\gamma|y)$. In particular, we use the same sequence of proper prior distributions, $p_n(\alpha)$ to construct the following sequence of transition kernels.

PROPER S+PX-SAMPLER

STEP 1: Sample $(\alpha^\star, \xi^\star, \sigma_\xi^\star)$ from $p_n\left(\alpha, \xi, \sigma_\xi \mid \gamma = \gamma^{(t)}, y\right)$; i.e., sample $\alpha^\star \sim p_n(\alpha)$ and set $\xi^\star = \beta^{(t)}/\alpha^\star$ and $\sigma_\xi^\star = \sigma_\beta^{(t)}/|\alpha^\star|$.

STEP 2: Sample $\mu^{(t+1)}$ from $p\left(\mu \mid \xi = \xi^\star, \sigma_\xi = \sigma_\xi^\star, \alpha = \alpha^\star, y\right)$.

STEP 3: Sample ξ_j from $p\left(\xi_j \mid \mu = \mu^{(t+1)}, \xi_{-j}, \sigma_\xi = \sigma_\xi^\star, \alpha = \alpha^\star, y\right)$ for $j = 1, \ldots, J$.

STEP 4: Sample σ_ξ from $p\left(\sigma_\xi \mid \mu = \mu^{(t+1)}, \xi, \alpha = \alpha^\star, y\right)$.

STEP 5: Sample $\alpha^{(t+1)}$ from $p_n\left(\alpha \mid \mu = \mu^{(t+1)}, \xi, \sigma_\xi, y\right)$

STEP 6: Set $\beta^{(t+1)} = \alpha^{(t+1)}\xi$ and $\sigma_\beta^{(t+1)} = |\alpha^{(t+1)}|\sigma_\xi$.

In the limit, we find that $\gamma^{(t+1)}$ does not depend on α_0, thus the limiting transition kernel is proper and Markovian. Analysis similar to that given for the V+PX-SAMPLER shows that the limiting transition kernel is the kernel described by the S+PX-SAMPLER.