Psychological Methods

A Method for Efficiently Sampling From Distributions With Correlated Dimensions

Brandon M. Turner, Per B. Sederberg, Scott D. Brown, and Mark Steyvers

Online First Publication, May 6, 2013. doi: 10.1037/a0032222

Psychological Methods

1082-989X/13/$12.00 DOI: 10.1037/a0032222

A Method for Efficiently Sampling From Distributions With Correlated Dimensions

Brandon M. Turner
Stanford University

Per B. Sederberg
The Ohio State University

Scott D. Brown
University of Newcastle

Mark Steyvers
University of California, Irvine

Bayesian estimation has played a pivotal role in the understanding of individual differences. However, for many models in psychology, Bayesian estimation of model parameters can be difficult. One reason for this difficulty is that conventional sampling algorithms, such as Markov chain Monte Carlo (MCMC), can be inefficient and impractical when little is known about the target distribution—particularly the target distribution's covariance structure. In this article, we highlight some reasons for this inefficiency and advocate the use of a population MCMC algorithm, called differential evolution Markov chain Monte Carlo (DE-MCMC), as a means of efficient proposal generation. We demonstrate in a simulation study that the performance of the DE-MCMC algorithm is unaffected by the correlation of the target distribution, whereas conventional MCMC performs substantially worse as the correlation increases. We then show that the DE-MCMC algorithm can be used to efficiently fit a hierarchical version of the linear ballistic accumulator model to response time data, which has proven to be a difficult task when conventional MCMC is used.

Keywords: differential evolution, optimal transition kernel, hierarchical Bayesian estimation, linear ballistic accumulator model, response time

Mathematical models often posit mechanisms designed to mimic the underlying processes involved in a behavior of interest. A model's behavior is most often specified by parameters that directly affect the mechanisms within the model. These parameters play an essential role for understanding how a behavior of interest might arise under the assumed model. To estimate the model parameters, researchers first fit the assumed model to experimental data. In psychology, parameter estimates can be used to better understand (human) behavior. They can even be used to provide evidence either for or against a research hypothesis of interest. Thus, properly estimating the parameters of a cognitive model is an important task.

Although there are many ways to estimate the parameters of a model, in this article, we focus exclusively on Bayesian estimation. Recently, Bayesian estimation has become a popular method of parameter inference in psychology (e.g., Klein Entink, Kuhn, Hornke, & Fox, 2009; Klugkist, Landy, & Hoijtink, 2010; McArdle, Grimm, Hamagami, Bowles, & Meredith, 2009; Muthén & Asparouhov, 2012; Prevost et al., 2007; Wirth & Edwards, 2007; Yuan & MacKinnon, 2009). For Bayesians, parameters are treated as random quantities along with the data. Inferences about parameters are based on the probability distributions of the parameters after some data are observed. These probability distributions are known as the posterior distributions, and they can provide remarkable insight into the architecture of a model.

The parameters of a model are generally specified to exclusively influence one aspect or another of a particular process, but the complexity of the models means that they can make similar predictions under very different selections of the parameter values. This is especially true when a subset of the parameters "trade off" with one another.

Take, for example, models of response time (RT). The most successful RT models are based on the idea that evidence for each choice accumulates throughout a trial. These models assume that observers have some initial amount of evidence in favor of each alternative. When this initial amount of evidence is equal across alternatives, the observer is said to be unbiased. Once the trial begins, the models assume that evidence accumulates for each alternative with a rate governed by a "drift rate" parameter. Once a threshold amount of evidence has accumulated for any alternative, the process stops and the observer issues the corresponding response.

Brandon M. Turner, Department of Psychology, Stanford University; Per B. Sederberg, Department of Psychology, The Ohio State University; Scott D. Brown, School of Psychology, University of Newcastle, Newcastle, New South Wales, Australia; Mark Steyvers, Department of Cognitive Sciences, University of California, Irvine.

This work was funded by National Institutes of Health Award F32GM103288. We thank Jeff Rouder and Patrick Shrout for helpful comments that improved an earlier version of this paper.

Correspondence concerning this article should be addressed to Brandon M. Turner, Department of Psychology, Stanford University, 342 Jordan Hall, Building 420, Stanford, CA 94305. E-mail: turner.826@gmail.com

The parameters of such models can trade off. For example, when the drift rate parameter increases, evidence accumulates more quickly, ultimately leading to a faster response. Similarly, when the threshold parameter is decreased, less evidence is required to make a decision; this also leads to a faster response. There are similar trade-offs between many other sets of parameters, such as the initial amount of evidence and the "nondecision time."

Models with highly correlated parameters exist in many other domains, such as structural equation modeling (e.g., Song & Lee, 2012), item factor analysis (e.g., Edwards, 2010; Patz & Junker, 1999a, 1999b), memory (e.g., Pooley, Lee, & Shankle, 2011; Turner, Dennis, & Van Zandt, 2011), and categorization (e.g., Vanpaemel, 2009). Models whose parameters are highly correlated make for a very challenging estimation problem for many current algorithms, especially in the Bayesian framework. We first begin by discussing a conventional method of estimating the posterior distribution, called the Metropolis–Hastings algorithm, and discuss why this method can be inefficient and impractical for estimating the posterior distributions of parameters from psychological models. We then argue that a population-based genetic algorithm, called differential evolution Markov chain Monte Carlo (DE-MCMC; Storn & Price, 1997; Ter Braak, 2006), is better suited for Bayesian estimation in many contexts. Although there are a number of alternative methods for obtaining samples from the posterior distribution that are much more efficient than conventional MCMC techniques (e.g., Andrieu & Thomas, 2008; Hoffman & Gelman, 2011; Neal, 2011), we focus on the DE-MCMC algorithm because of its high efficiency and ease of implementation. We illustrate the effectiveness of the algorithm in a small simulation study. We also show that the DE-MCMC algorithm's efficiency allows us to analyze a hierarchical (across subjects) version of the linear ballistic accumulator (LBA) model, which has not previously been possible.

Conventional Markov Chain Monte Carlo

MCMC sampling is a general technique that has been instrumental in estimating posterior distributions in Bayesian statistics (Gelman, Carlin, Stern, & Rubin, 2004; Robert & Casella, 2004). MCMC sampling has enjoyed widespread success because it is very simple to use and can be more efficient than other methods, such as rejection sampling, when the prior distribution differs substantially from the posterior distribution.

MCMC sampling is a random walk process that can be used to estimate a "target distribution." In the context of Bayesian estimation, the target distribution is usually the posterior distribution $\pi(\theta)$ for the parameter θ. MCMC is a general sampling technique with many variants. One of the oldest and most widely used variant is the Metropolis–Hastings algorithm, which we will refer to as "conventional MCMC" because of its common usage in psychology (see, e.g., Gershman, Blei, Pereira, & Norman, 2011; Lee, Steyvers, de Young, & Miller, 2011; Rouder & Lu, 2005; Rouder, Lu, Speckman, & Jiang, 2005; Rouder, Sun, Speckman, Lu, & Zhou, 2003, for examples in psychology). To use the Metropolis–Hastings algorithm to estimate the target distribution $\pi(\theta)$, we select an initial value θ_1 and then sample a candidate value θ^* from a proposal or kernel distribution $K(\cdot)$ conditioned on the initial value θ_1, so that

$$\theta^* \sim K(\theta_1).$$

For example, the kernel distribution could be Gaussian and centered at the initial value θ_1 with standard deviation σ, so that

$$\theta^* \sim N(\theta_1, \sigma^2),$$

where $N(a, b)$ is the Gaussian distribution with mean a and variance b. The parameter σ serves as a tuning parameter and can have a major impact on the efficiency of the algorithm, as we discuss below.

The new proposal θ^* will be "accepted" with Metropolis–Hastings probability (see below) and $\theta_2 = \theta^*$. If the proposal is not accepted, $\theta_2 = \theta_1$, and the chain will not move. This process continues with the proposal based on the current value of θ until a desired number of samples N have been obtained. Under certain assumptions, the sequence of proposals $\{\theta_1, \theta_2, \ldots, \theta_N\}$ "converges" to taking samples from the target distribution $\pi(\theta)$, which can then be used to estimate the parameter θ.

The Metropolis–Hastings probability of accepting a new proposal θ^* on the $(t + 1)$th iteration is given by

$$\alpha = \min\left(1, \frac{\pi(\theta^*)q(\theta_t|\theta^*)}{\pi(\theta_t)q(\theta^*|\theta_t)}\right), \quad (1)$$

where $q(a|b)$ is the *density* of the kernel distribution $K(\cdot)$ evaluated at a with parameters b. The proposal density function is a key component in guaranteeing that the stationary distribution is the intended target distribution. If we denote the probability of transitioning from θ_t to θ^* as

$$P(\theta_t \rightarrow \theta^*) = q(\theta^*|\theta_t)\alpha,$$

then if the kernel function is selected appropriately, so that

$$P(\theta^* \rightarrow \theta_t)\pi(\theta^*) = P(\theta_t \rightarrow \theta^*)\pi(\theta_t),$$

the stationary distribution is the target distribution $\pi(\theta)$.

Although this algorithm is powerful in theory, in practice its effectiveness depends heavily on the selection of the proposal distribution. The specification of the proposal distribution depends on (a) the selection of the tuning parameters and (b) the selection of the kernel function. We now discuss each of these in turn.

Selecting the Tuning Parameters

The selection of the tuning parameters can have a large influence of the performance of the sampler. For example, suppose the target distribution (i.e., the posterior) is the standard normal distribution centered at zero with a standard deviation of one. If we select a large value, such as $\sigma = 20$, we can expect many proposals to be far outside the target distribution, which will result in many of these proposals being rejected. However, we also should avoid selecting σ to be too small. For example, if we select $\sigma = 0.1$, nearly all proposals will be accepted but the chain will move within the posterior very slowly. The resulting samples will not be reflective of the target distribution.

To further illustrate this point, we performed a simple simulation using three values for $\sigma = \{0.1, 1, 20\}$ to sample from the standard normal target distribution. Each chain was first initialized to $\theta_1 = 0$, and 1,000 samples were taken using the Metropolis–Hastings algorithm with a normal transition kernel with standard

deviation equal to 0.1, 1, or 20. For this simple example, no burn-in period was required, because each chain was initialized within the true posterior. Figure 1 shows these results. The top panels show the estimates obtained using $\sigma = 0.1$ (left panel), $\sigma = 1$ (middle panel), or $\sigma = 20$ (right panel) along with the true target distribution (black density). The bottom panels show the "trace plots," or the path taken by each chain. The trace plots can be used to assess the "mixing" properties of the sampler. Ideally, the trace plots should look like completely random draws from a distribution. That is, a single path of the sampler should not be detectable, and the chain should bounce from one location in the distribution to another regularly. When a chain stays in one place for several consecutive iterations, it is said to be "stuck." This is a definitive sign of poor mixing.

The figure shows that when $\sigma = 0.1$, the chain moves far too slowly to accurately estimate the entire range of the target distribution. The trace plot shows a clear path from one location to the next, which indicates that the value for σ was too small. The middle panel shows the optimal transition kernel, which accurately estimates the target distribution. This trace plot displays good mixing properties—it covers the full target distribution, and the chain rarely sticks. Finally, when $\sigma = 20$ (right panel) the chain covers the target density well, but the rejection rate is too high and the chain gets stuck several times during the simulation.

The simplicity of this example illustrates a very practical problem with conventional MCMC. The setting of tuning parameters, like σ, can be difficult. In practice, one of the most practical solutions is a sort of trial and error approach, in which initial values of σ are chosen and modified based on the inspection of the mixing properties of the chains (see Gelman et al., 2004, for a detailed discussion of this procedure). The problem of tuning becomes much worse in realistic situations, such as when the target distribution contains multiple parameters; in that case, one must choose a separate value for σ for each dimension of the parameter space. Another problem surfaces when the parameters of a model are correlated. In the next section, we discuss how correlation can be combated with an appropriate selection of the transition kernel.

Correlation and the Selection of the Transition Kernel

Although the selection of the transition kernel can affect the performance of the sampler (see Gelman et al., 2004), the choice of the kernel is less important than the tuning parameters. When a model has highly correlated parameters, the least arduous solution is to ignore the correlation and select a separate σ for each dimension in the parameter space. In psychology, this is often done when the sampling is divided into "blocks" of the parameter space (see, e.g., Gelman et al., 2004; Rouder & Lu, 2005; Rouder et al.,

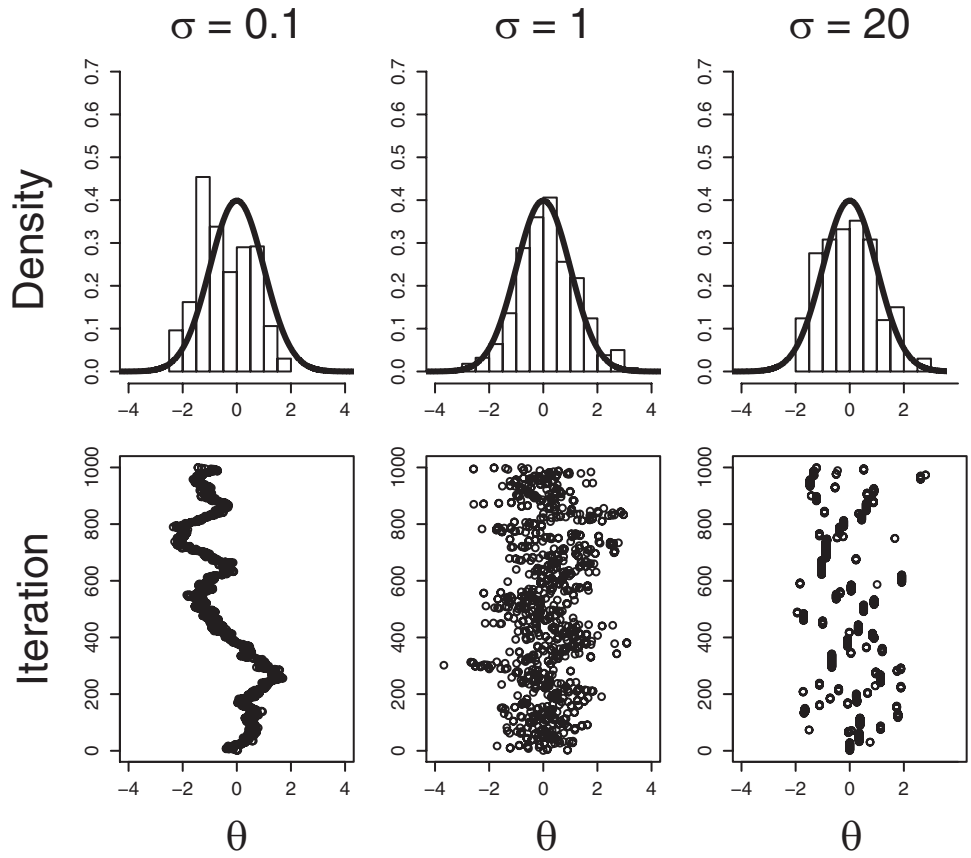

Figure 1. Estimates of the standard normal distribution obtained with different transition kernels: $\sigma = 0.1$ (left panel), $\sigma = 1$ (middle panel), and $\sigma = 20$ (right panel). The top panel shows the estimates obtained along with the target density (black lines), and the bottom panel shows the trace plots of the chains.

2003, 2005, for a more thorough discussion of the blocking procedure and examples in psychology).

Depending on how highly correlated the parameters in our model are, ignoring their correlation can lead to very poor sampling behavior. For example, suppose our target distribution contains two parameters that are moderately correlated. If we do not know how correlated these parameters are, we might naively choose a multivariate normal proposal distribution such that

$$\theta^* \sim N_2\left(\theta_t, \sum = \begin{bmatrix} \sigma_{11}^2 & 0 \\ 0 & \sigma_{22}^2 \end{bmatrix}\right), \quad (2)$$

where $N_p(a, b)$ represents the multivariate normal distribution of dimension p with mean vector a and covariance matrix b. These assumptions ignore the true parameter correlations and can result in substantial rejection rates. This is shown in Figure 2, which gives an example of using an independent transition kernel (e.g., Equation 2) when estimating a target density, shown as the gray cloud of points. The vector shows the current state of the chain θ_t and the proposal region for θ^*, shown as a circle. Here, assuming that the parameters are uncorrelated will result in many samples being drawn that are not in the target density, a situation that is illustrated by the proportion of white area inside the circle. Proposals that are generated in this area will almost certainly be rejected.

A second option is to specify a transition kernel that includes the correlation between the parameters; for example, one could specify a correlated multivariate normal distribution. This specification adds a parameter ρ to the previous example, such that

$$\theta^* \sim N_2\left(\theta_t, \sum = \begin{bmatrix} \sigma_{11}^2 & \rho\sigma_{11}\sigma_{22} \\ \rho\sigma_{11}\sigma_{22} & \sigma_{22}^2 \end{bmatrix}\right).$$

Although specifying the joint transition kernel in this way can improve the performance of the sampler, it can be very difficult to select the right value for the tuning parameter ρ.

The two examples above illustrate that the selection of the transition kernel can be both important and difficult, even in a very simple problem. In general, the construction of good transition kernels requires several runs of the algorithm in a type of trial and error approach until one is satisfied with the mixing qualities of the sampler. This process can be time consuming, and the choices made for one problem are unlikely to generalize to other problems even when the same model is used.

A practical consequence of these difficulties is that Bayesian estimation of response time models has been limited in psychology because our most sophisticated models of RT have highly correlated parameters (e.g., Lee, Fuss, & Navarro, 2006; Oravecz, Tuerlinckx, & Vandekerckhove, 2009; Vandekerckhove, Tuerlinckx, & Lee, 2011). Instead, Bayesian models of RT have mostly been limited to descriptive distributions that are simple and that fit well, such as the Weibull, ex-Gaussian, or lognormal (e.g., Craigmile, Peruggia, & Van Zandt, 2010;Farrell & Ludwig, 2008; Lee & Wagenmakers, 2010; Peruggia, Van Zandt, & Chen, 2002; Rouder et al., 2003, 2005). Even in these simplified cases, obtaining the Bayesian posterior distribution presents a very challenging modeling problem, the difficulty of which is evident in the technicality of the forementioned articles.

As a result of the implementation difficulties in estimation, some researchers have turned to general-purpose MCMC packages, particularly WinBUGS (Lunn, Thomas, Best, & Spiegelhalter, 2000). For example, Vandekerckhove, Tuerlinckx, and Lee (2008) and Vandekerckhove et al. (2011) have used WinBUGS to estimate the posterior distribution for the parameters of the diffusion model, and Donkin, Brown, and Heathcote (2011) have described a similar procedure for the LBA model (see also Dutilh, Vandekerckhove, Tuerlinckx, & Wagenmakers, 2009; Vandekerckhove, Verheyen, & Tuerlinckx, 2010, for additional applications). Although these applications speak to the utility of WinBUGS, WinBUGS can be slow and sometimes difficult to manage. Vandekerckhove et al. (2011) pointed out several issues with their program, such as long computation times (hours or days for real-world analyses), high autocorrelations, poor mixing, and computational instability.

We present an algorithm that can remedy the sampling problems outlined above. The method is a type of population-based MCMC, which uses a system of interacting Markov chains. This interaction produces high-quality proposals that automatically match the op-

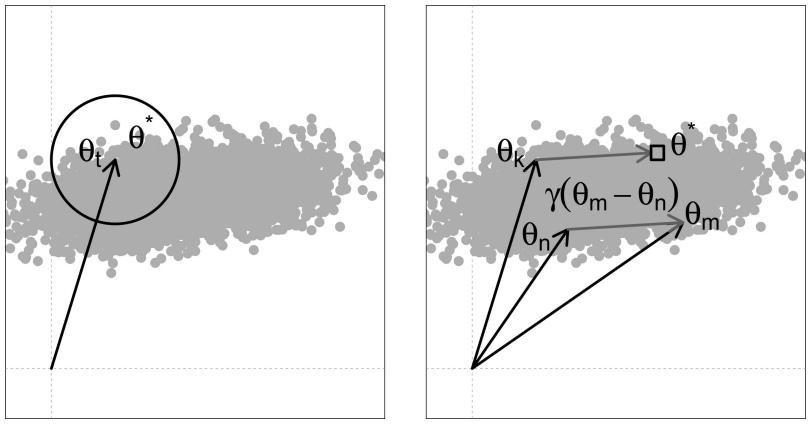

Figure 2. Graphical representation of the perturbation method assuming an independent bivariate normal transition kernel (left panel) used in conventional Markov chain Monte Carlo (MCMC) and the crossover method (right panel) used in differential evolution Markov chain Monte Carlo (DE-MCMC).

timal transition kernel, even in high-dimensional and highly correlated problems.

Differential Evolution Markov Chain Monte Carlo

Differential evolution (DE) was originally developed for optimization problems (Storn & Price, 1997) but was later combined with Markov chain Monte Carlo (MCMC) for Bayesian estimation by ter Braak (2006). Although there are a number of different variants of the DE algorithm (see Appendix A for a brief discussion), few are suitable for estimating posterior distributions. Because we seek a Bayesian solution, we focus on the DE-MCMC algorithm. The key to the DE-MCMC algorithm is more efficient generation of proposals; rather than simply adding random noise to the current state, DE-MCMC uses multiple interacting chains. The difference between some chains' current states is used to generate proposals for other chains. This means that the proposal for one chain is based on some weighted combination of other chains, which combines two steps that are usually distinct in population MCMC (crossover and mutation) and gives rise to a self-organizing system of chains.

Figure 2 (right panel) illustrates proposal generation in the DE-MCMC algorithm (Figure 3 describes the algorithm in pseudocode). The current state of the kth chain is shown by θ_k. The new proposal (θ^*) is generated by taking the difference between the current states of two other chains, chosen at random (θ_m and θ_n). This difference vector, shown by the gray line, sets the direction that the new proposal (θ^*) should move relative to θ_k. The distance to move along this direction is set by a tuning parameter γ, which can take any positive value. The parameter γ controls the magnitude of the jumping distribution. For example, setting $\gamma = 1$ results in a jumping magnitude that is equal to difference vector. Finally, a very small amount of random noise ε is added to the proposal (to avoid degeneracy problems). Algebraically, this process can be summarized as

$$\theta^* = \theta_k + \gamma(\theta_m - \theta_n) + \varepsilon. \quad (3)$$

As with conventional MCMC, the new proposal is accepted over the existing state with the Metropolis–Hastings probability given in Equation 1. The DE-MCMC algorithm has just one tuning parameter, which is much simpler than other MCMC methods. Typical settings for the tuning parameter include using $\gamma = 2.38/\sqrt{2d}$, where d is the number of dimensions of the parameter space (ter Braak, 2006, have shown this is an optimal setting under some conditions), or randomly sampling independent values for each proposal (e.g., $\gamma \sim CU[0.5, 1]$, where $CU[a, b]$ denotes the continuous uniform distribution on the interval $[a, b]$). The algorithm is also relatively insensitive to the distribution of the random noise term, ε. This is because the error term is designed to make the DE-MCMC algorithm probabilistic and not deterministic, so that transition probabilities can be calculated. Typically, uniformly distributed noise with zero mean and small variance is used, $\varepsilon \sim CU[-b, +b]$, with $b = 0.01$ or similar. Larger values of b relative to the variance of the target distribution will result in poor performance of the algorithm, because the benefits of the DE proposal mechanism will be masked by the uniform noise incurred by adding ε.

Equation 3 shows that the probability of transitioning from the current state θ_k to the proposal θ^* is given by

$$q(\theta^*|\theta_k) = \sum_i \sum_j \mathrm{MCU}_p(\theta_k + \gamma(\theta_i - \theta_j), b),$$

where $\mathrm{MCU}_p(a, b)$ is a p-variate continuous uniform distribution centered at a with boundaries $[a - b, a + b]$, and the probability of the reverse move is given by

$$\begin{aligned} q(\theta_k|\theta^*) &= \sum_i \sum_j \mathrm{MCU}(\theta^* - \gamma(\theta_i - \theta_j), b) \\ &= \sum_i \sum_j \mathrm{MCU}(\theta^* + \gamma(\theta_j - \theta_i), b). \end{aligned}$$

Here, $q(\theta^*|\theta_k) = q(\theta_k|\theta^*)$, so the stationary distribution of this chain will be the target distribution $\pi(\theta)$. Furthermore, because these transition probabilities are equal, there is no need to evaluate

1: Initialize $\theta_{k,1}$ such that $\pi(\theta_{k,1}) > 0 \; \forall \; k \in \{1, 2, \ldots, K\}$.
2: **for** $2 \leq i \leq N$ **do**
3: Specify γ.
4: **for** $1 \leq k \leq K$ **do**
5: Sample θ_m and θ_n without replacement from the set $\{\theta\} \backslash \{\theta_{k,i-1}\}$.
6: Sample ϵ from a symmetric distribution with small variance.
7: Propose $\theta^* = \theta_{k,i-1} + \gamma(\theta_m - \theta_n) + \epsilon$.
8: Sample $\alpha \sim CU[0, 1]$.
9: **if** $\alpha < \pi(\theta^*)/\pi(\theta_{k,i-1})$ **then**
10: Store $\theta_{k,i} \leftarrow \theta^*$.
11: **else**
12: Store $\theta_{k,i} \leftarrow \theta_{k,i-1}$.
13: **end if**
14: **end for**
15: **end for**

Figure 3. The differential evolution Markov chain Monte Carlo (DE-MCMC) algorithm for obtaining N samples with K chains of the target density $\pi(\theta)$.

them in Equation 1, and the Metropolis–Hastings probability[1] reduces to

$$\alpha = \min\left(1, \frac{\pi(\theta^*)}{\pi(\theta_k)}\right).$$

Simulation Study

In this section, we compare the sampling efficiency of DE-MCMC against conventional MCMC in a simulation. Suppose the target distribution is a bivariate normal distribution with density function

$$f(\theta|\mu, \Sigma) = \frac{1}{2\pi|\Sigma|^{1/2}}\exp\left[-\frac{1}{2}(\theta-\mu)^T\Sigma^{-1}(\theta-\mu)\right], \quad (4)$$

where

$$\Sigma = \begin{bmatrix} \sigma_{11}^2 & \rho\sigma_{11}\sigma_{22} \\ \rho\sigma_{11}\sigma_{22} & \sigma_{22}^2 \end{bmatrix},$$

and μ is the mean vector containing two elements. Our claim is that the DE-MCMC sampler is a more efficient approach to estimating a target distribution with highly correlated dimensions, so our simulations will focus on the correlation parameter $\rho \in [-1, 1]$.

To examine the sampling behavior of both the conventional MCMC and DE-MCMC approaches, we will draw samples from the bivariate normal distribution and evaluate two measures of efficiency. The first measure is the rejection rate and is calculated by recording the number of times a proposal is generated that does not result in a change of state (i.e., the proposal is rejected). The second measure is the Kullback–Leibler distance. The Kullback–Leibler distance is a popular statistic that measures the discrepancy between two density functions (Kullback, Keegel, & Kullback, 1987). For our purposes, the two densities correspond to the true posterior distribution, which is known for this simple example, and the estimated posterior distribution, which is formed by constructing a kernel density estimate of the distribution of all of the values obtained from the sampling process (see Silverman, 1986).

For simplicity, we set $\mu = [0\ 0]^T$ and $\sigma_{11} = \sigma_{22} = 1$. In our simulation, we will vary the correlation parameter ρ across the interval (0, 0.99). We explore only half of the support of ρ, because the sampling behavior will be similar regardless of the sign of the correlation parameter.

First, to implement the conventional MCMC sampler, we used a bivariate normal proposal function (see Equation 4), where $\sigma_{11} = \sigma_{22} = 1$ and $\rho = 0$. That is, the conventional MCMC sampler was given the advantage of having the optimal (true) settings for two of its three tuning parameters. For the third tuning parameter, we naively assumed that θ_1 and θ_2 were uncorrelated ($\rho = 0$). Implementing the DE-MCMC sampler requires effectively only one tuning parameter to be set: We sampled γ randomly on each step from a uniform distribution between 0.5 and 1. We set $b = 0.001$ for the small-variance random noise.

For both samplers, we ran 16 chains for 1,000 iterations with no burn-in period, resulting in 16,000 samples from the target distribution. For the conventional MCMC sampler, the 16 chains did not interact, and so the structure of the sampler is said to be "embarrassingly parallel." This is not true of the DE-MCMC sampler, which uses all the current states of the chains to generate a new proposal for each chain. In both samplers, each chain was initialized with a random sample from the target density. The above procedure was replicated for each sampler 10 times for each value of $\rho \in \{0, 0.01, 0.02, \ldots, 0.99\}$, resulting in 2,000 simulations. At the close of the each simulation, we recorded the rejection rate and the Kullback–Leibler divergence statistic between the posterior estimate and the true marginal distributions of both θ_1 and θ_2.

Figure 4 shows the results for both the conventional MCMC sampler (gray lines) and the DE-MCMC sampler (black lines). The left panel plots the rejection rate for each sampler as a function of the correlation parameter. This shows that, though the rejection rate of the DE-MCMC sampler stays roughly constant across all values of the correlation parameter, the rejection rate of the conventional MCMC sampler increases exponentially as ρ increases from zero to one. The figure also shows that the conventional MCMC sampler has a slightly better rejection rate than the DE-MCMC sampler when $\rho < 0.25$, that their rejection rates are very similar when $0.25 < \rho < 0.4$, and that the DE-MCMC sampler performs much better when $\rho > 0.4$. Although this is the case when $\gamma \sim CU[0.5, 1]$, when we reduced the upper bound of the continuous uniform distribution to 0.8 (results not shown), the DE-MCMC sampler outperformed the conventional MCMC sampler across all values of ρ with an average rejection rate of 42%.

The right panel of Figure 4 shows the mean Kullback–Leibler distance for each sampler for the marginal distribution of θ_1 because the results were identical for θ_2. Although it is somewhat difficult to interpret these results because the Kullback–Leibler statistic does not follow any known distribution, we can see that the DE-MCMC sampler recovers the target density more accurately than the conventional MCMC sampler when $\rho > 0.7$.

Visual Assessment of Samples

We discussed earlier that one could assess the mixing properties of the chains by visually examining the trace plots. Figure 5 shows the trace plots (rows) for four randomly selected chains (for visual clarity) at three selected values of ρ: 0.0, 0.5, and 0.9. The columns represent three different methods of sampling. The first column corresponds to DE-MCMC, and the second column corresponds to conventional MCMC. The third column corresponds to a blocked version of the conventional MCMC algorithm, in which we used the same tuning parameters as in the unblocked version. Recall that the unblocked MCMC algorithm (as well as the DE-MCMC algorithm) proposes two parameters at once via the bivariate normal distribution. On the other hand, the blocked MCMC algorithm proposes one parameter value at a time, resulting in two samples being drawn per iteration. In general, the blocking procedure can dramatically improve the mixing properties of the sampler.

Figure 5 shows that the three methods compare favorably, with each trace plot displaying excellent mixing qualities. However, at $\rho = 0.9$ the conventional MCMC algorithm has some difficulty mixing properly. This defective mixing is a result of the systematic mismatch between proposal distribution and target distribution. This observation is consistent with the increased rejection rates

[1] When the kernel function is symmetric, the algorithm is often simply called the Metropolis algorithm.

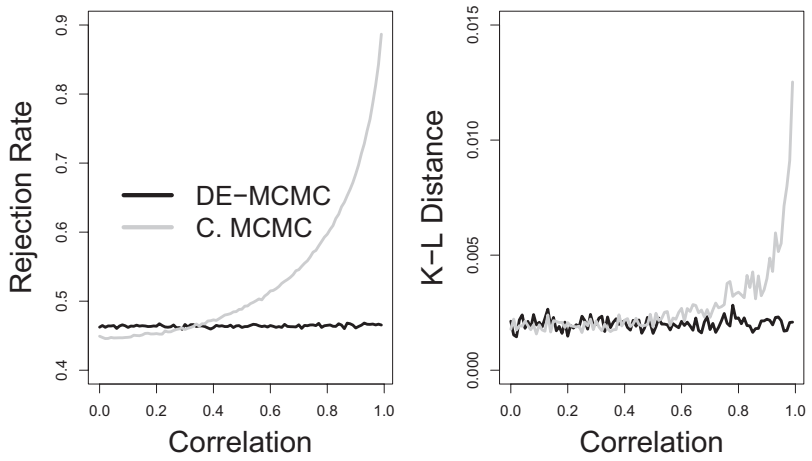

Figure 4. Performance results of the conventional MCMC sampler (gray lines) and the DE-MCMC sampler (black lines). The left panel shows the rejection rate and the right panel shows the Kullback–Leibler divergence statistic, both as functions of the correlation of the bivariate normal distribution. C.MCMC = conventional Markov chain Monte Carlo; DE-MCMC = differential evolution Markov chain Monte Carlo.

and the increasing Kullback–Leibler distance measurement shown in Figure 4.

Another way to assess the mixing properties of the chains is through the autocorrelation functions (ACFs). For a given lag value *l,* the ACFs calculate the correlation between the set of all values in the chain and the set of all values in the chain *l* iterations behind the first set. Thus, if the chain tends to stick frequently, consecutive values in the chain will tend to be similar. This will result in high values in the ACF.

Figure 6 shows the ACFs for three values of $\rho = 0.0, 0.5, 0.9$ (rows, respectively) and the three different methods of sampling: DE-MCMC (left column), unblocked conventional MCMC (middle column), and the blocked conventional MCMC (right column). The *y*-axes correspond to the values of the ACF, and the *x*-axes correspond to the values for the lag. The first row shows that each algorithm has a moderate but acceptable degree of autocorrelation for lags less than 15. However, when the correlation is increased to $\rho = 0.5$, the ACFs for the unblocked conventional MCMC sampler increase for larger values of the lag, indicating poorer mixing. In the last row, when $\rho = 0.9$, the values of the ACF rise sharply for all values of the lag (i.e., 1 through 40). However, both the DE-MCMC and blocked conventional MCMC algorithms remain stable in their mixing abilities.

Our simulations demonstrate the benefits of DE-MCMC over conventional MCMC. The results are especially promising, given that the conventional MCMC sampler was nearly optimal for the target under consideration. In a more applied setting, the researcher would have to choose suitable values for the tuning parameters σ_1, σ_2, and ρ, while possibly having very little information about the target distribution. On the other hand, the DE-MCMC sampler requires specification of just the parameters *b* and γ. Reasonably small values of *b* have very little effect on either the accuracy or the rejection rate of the sampler. Although γ may be more difficult to specify, the above example has shown that sampling γ randomly from a continuous uniform distribution is a simple and effective way of estimating this particular density.

Examining the trace plots and the ACFs has provided further evidence that the DE-MCMC algorithm is performing favorably. First, the DE-MCMC algorithm is unaffected by the increase in the correlation of the target distribution, whereas the unblocked conventional MCMC algorithm is. The blocked conventional MCMC algorithm is also unaffected by the increase in the correlation; however, it is considerably slower than both of the unblocked algorithms, because the blocked version requires multiple evaluations of the likelihood function. Furthermore, both conventional MCMC algorithms were given the advantage of optimal tuning parameters, whereas the DE-MCMC required almost no tuning. Considering this, we argue that DE-MCMC is the most efficient of the three algorithms because it was fast, unaffected by the increase in ρ, and required almost no tuning.

Fitting Models of Choice Response Time to Real Data

We now examine the effectiveness of the DE-MCMC sampler by using it to fit a cognitive model to experimental data. There are many excellent models of choice RT that we could explore, but we chose the LBA because of its analytic simplicity. Although the LBA model has been fit in a Bayesian framework (Donkin, Averell, Brown, & Heathcote, 2009), current approaches suffer from severe practical limitations, such as computational instability, and the impracticality of examining hierarchical versions. Below, we fit a hierarchical version of the LBA to data first presented by Forstmann et al. (2011).

Experiment

The data were presented in Forstmann et al. (2011) and consisted of responses from 20 young subjects and 14 older subjects. The study was a moving dots task in which subjects were asked to decide whether a cloud of semirandomly moving dots appeared to move to the left or to the right. Subjects indicated their response by pressing one of two spatially com-

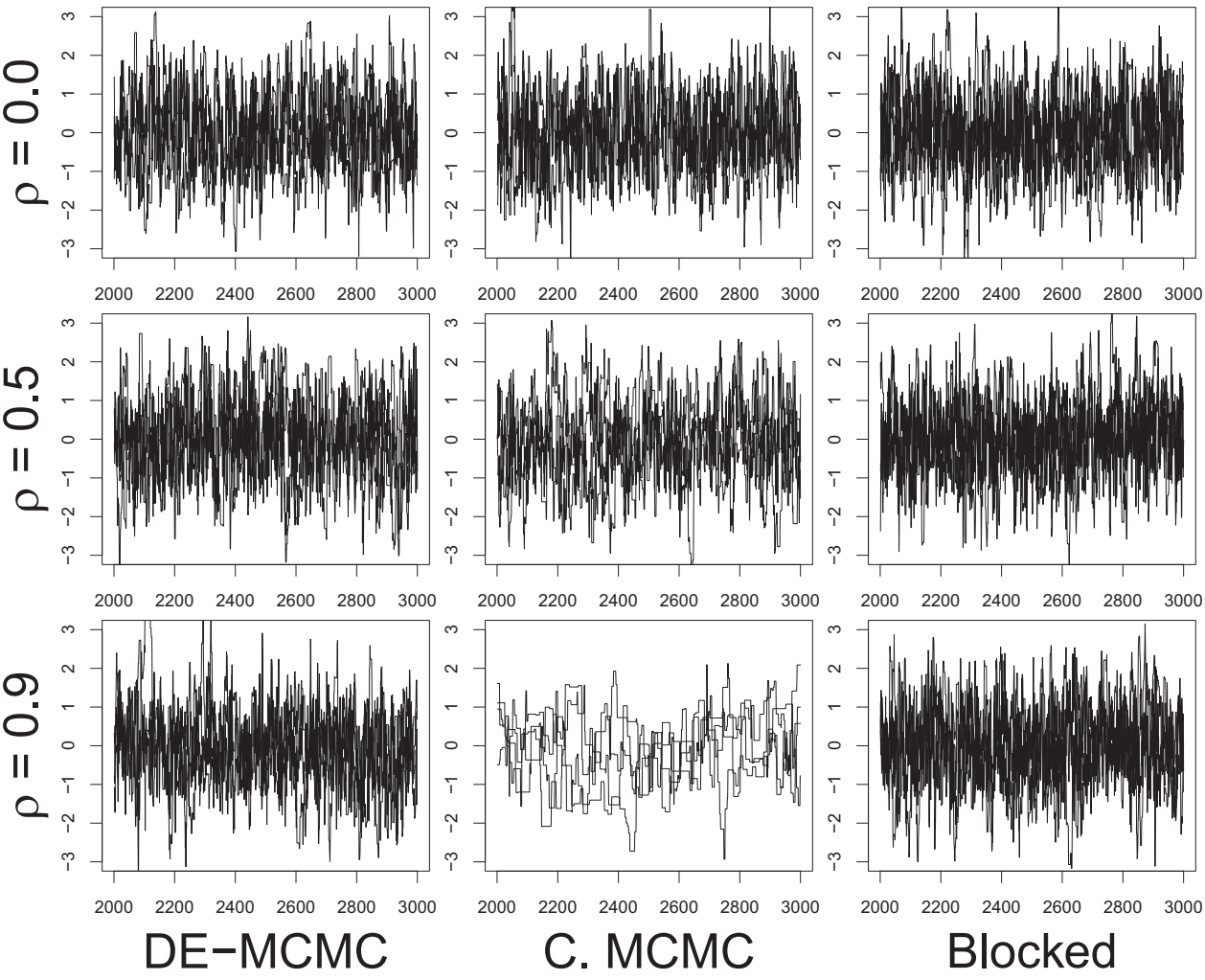

Figure 5. Trace plots for three values of $\rho = \{0.0, 0.5, 0.9\}$ (rows) and three methods of sampling: DE-MCMC (left column), conventional MCMC (middle column), and blocked (right column). C.MCMC = conventional Markov chain Monte Carlo; DE-MCMC = differential evolution Markov chain Monte Carlo.

patible buttons with either the left or the right index finger. Before each decision trial, subjects were instructed whether to respond quickly (the speed condition), accurately (the accuracy condition), or at their own pace (the neutral condition). Following the trial, subjects were provided feedback about their performance. In the speed and neutral conditions, subjects were told that their responses were too slow whenever they exceeded a RT of 400 and 750 ms, respectively, for the young subjects and 470 and 820 ms for the older subjects, respectively. In the accuracy condition, subjects were told when their responses were incorrect. Each subject completed 840 trials, equally distributed over the three conditions.

The Linear Ballistic Accumulator Model

The LBA model provides a simple explanation of choice and RT (Brown & Heathcote, 2008). It eliminates many complexities assumed by previous models, such as competition between alternatives (e.g., Brown & Heathcote, 2005; Usher & McClelland, 2001), passive decay of evidence ("leakage"; e.g., Usher & McClelland, 2001), and even within-trial variability (e.g., Ratcliff, 1978; Stone, 1960). The model's simplicity allows closed-form expressions for the first-passage-time distributions for each accumulator. With these equations, one can specify the likelihood function for the model parameters. This has been instrumental in the LBA model's success (e.g., Donkin, Averell, et al., 2009; Donkin et al., 2011; Donkin, Heathcote, & Brown, 2009; Forstmann et al., 2008, 2010, 2011).

Donkin, Averell, et al. (2009) provided a detailed technical account of how to fit the LBA model to data. Among other things, Donkin, Averell, et al. showed how to use the program WinBUGS (Lunn et al., 2000) to fit the LBA model in a Bayesian framework. Although the WinBUGS program is very user friendly, it relies on general-purpose MCMC methods. As we have shown, standard MCMC methods can suffer dramatically for models with highly correlated parameters (e.g., the LBA). Although one could fit a hierarchical version of the LBA using a modified version of the

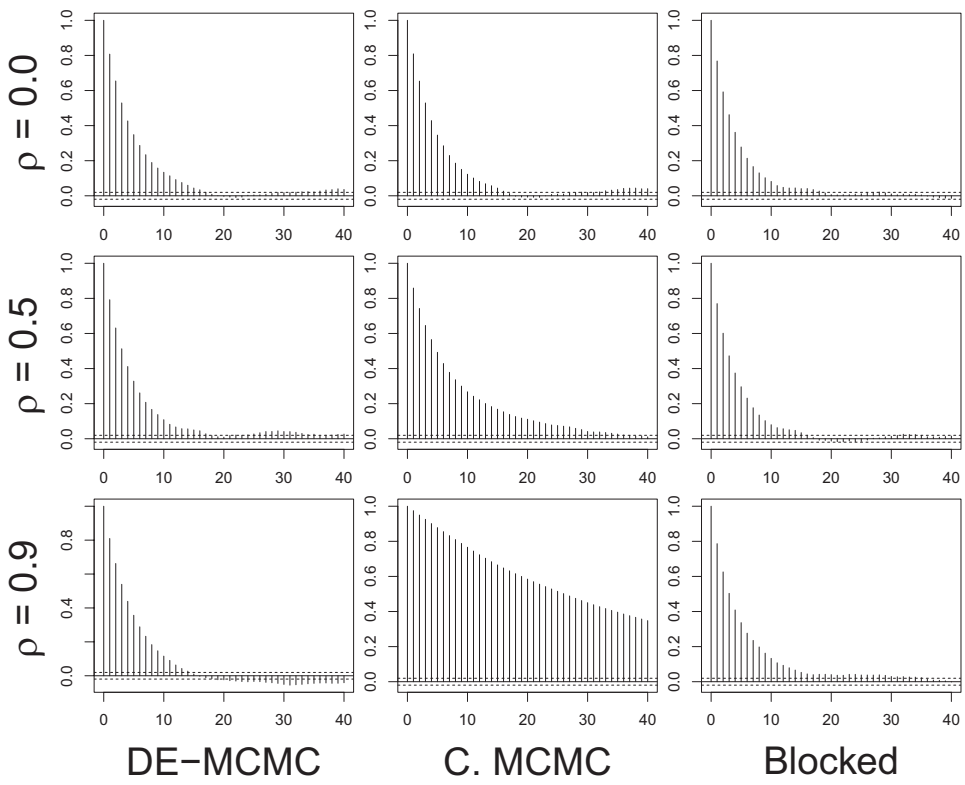

Figure 6. Autocorrelation functions (ACFs) for three values of ρ = {0.0, 0.5, 0.9} (rows) and three methods of sampling: DE-MCMC (left column), conventional MCMC (middle column), and blocked (right column). The *y*-axes represent the values of the ACFs, and the *x*-axes represent the magnitude of the lag setting from 0 to 40. C.MCMC = conventional Markov chain Monte Carlo; DE-MCMC = differential evolution Markov chain Monte Carlo.

WinBUGS code provided by Donkin, Averell, et al., we have not found published fits of such a model.

For ease of exposition, we first consider a special case without individual differences, so that a single set of parameters describes the behavior of all responses in the data. The *i*th single observation for the *j*th subject in a typical choice experiment will contain two pieces of information. The first piece is the response choice, which we denote $RE_{i,j}$ where $RE_{i,j} \in \{1, \ldots, C\}$, where C is the number of response alternatives. The second piece is the response time (RT), which we denote $RT_{i,j} \in (0, \infty)$. The LBA model assumes that evidence accumulates for each of the C alternatives at the beginning of a decision trial. Each accumulator begins with an independent amount of starting evidence k_c, which is sampled independently for each accumulator from a continuous uniform distribution $k_c \sim [0, A]$. The evidence for accumulator c then increases at a rate d_c (the drift rate for the cth response alternative), which is sampled independently for each accumulator from a normal distribution with mean v_c and standard deviation s, so $d_c \sim N(v_c, s)$. Each accumulator gathers evidence until one accumulator reaches a response threshold b. Finally, the LBA model assumes that the observed RT is the sum of the decision time plus some extra time τ for the nondecision process, such as motor execution. For simplicity, τ is usually assumed to be constant across trials. Thus, the final observed RT is given by

$$RT = \min_c \left(\frac{b - k_c}{d_c} \right) + \tau.$$

It is common to set the variance of the sampled drift rates to one ($s = 1$) to satisfy scaling conditions of the model, but other constraints are possible.

In a Bayesian framework, the data are expressed as a function of the model parameters, which is known as the likelihood function. To derive the likelihood function, we begin by identifying the probability density function (PDF). To simultaneously explain both RT and choice, we require the "defective" distribution for the cth accumulator, which is the probability of the cth accumulator reaching the threshold and the other accumulators not reaching the threshold. The density function for this distribution is given by

$$\text{LBA}(c, t|b, A, v, s, \tau) = f(c, t) \prod_{k \neq c} [1 - F(k, t)], \qquad (5)$$

where $v = \{v_1, \ldots, v_C\}$, $f(c, t)$ and $F(c, t)$ are the PDF and cumulative density function (CDF) for the time taken for the cth accumulator to reach the threshold, respectively (see Brown and Heathcote, 2008, for details). To incorporate the nondecision time parameter into the PDF, we substitute $(t - \tau)$ for t in Equation 5. Finally, for a vector of N responses RE with

corresponding response times *RT,* the likelihood function is given by

$$L(b,\ A,\ v,\ s,\ \tau|RT,\ RE) = \prod_{i=1}^{N} \mathrm{LBA}(RE_i,\ RT_i|b,\ A,\ v,\ s,\ \tau).$$

The Model for the Experiment

Because there are three speed conditions in the experiment, we use a vector of response threshold parameters $b = \{b^{(1)}, b^{(2)}, b^{(3)}\}$, so that $b^{(1)}$, $b^{(2)}$, and $b^{(3)}$ are used for the accuracy, neutral, and speed conditions, respectively. In addition, we include an index k to reflect the condition number $k = \{1, 2, 3\}$, so that $RE_{i,j,k}$ and $RT_{i,j,k}$ denote the ith response from the jth subject in the kth condition. We can also index the drift rates to reflect each response alternative so that $v = \{v^{(1)}, v^{(2)}\}$, where $v^{(1)}$ is the drift rate for the accumulator corresponding to the incorrect response choice and $v^{(2)}$ is the drift rate for the accumulator corresponding to the correct response choice.

To extend the LBA model to a hierarchical version, we represent each of the $j = \{1, 2, \ldots, S\}$ subjects using separate parameter vectors

$$(A_j,\ b_j^{(1)},\ b_j^{(2)},\ b_j^{(3)},\ s_j,\ v^{(1)}{}_j,\ v^{(2)}{}_j,\ \tau_j).$$

We make the assumption that the data were independent and identically distributed and the likelihood functions for Subject j in Condition 1 (accuracy), Condition 2 (neutral), and Condition 3 (speed) are

$$L(b_j^{(1)},\ A_j,\ s_j,\ v_j,\ \tau_j|RT,\ RE) = \prod_{i=1}^{N} \mathrm{LBA}(RE_{i,j,1},\ RT_{i,j,1}|b_j^{(1)},\ A_j,\ v_j,\ s_j,\ \tau_j),$$

$$L(b_j^{(2)},\ A_j,\ s_j,\ v_j,\ \tau_j|RT,\ RE) = \prod_{i=1}^{N} \mathrm{LBA}(RE_{i,j,2},\ RT_{i,j,2}|b_j^{(2)},\ A_j,\ v_j,\ s_j,\ \tau_j), \text{ and}$$

$$L(b_j^{(3)},\ A_j,\ s_j,\ v_j,\ \tau_j|RT,\ RE) = \prod_{i=1}^{N} \mathrm{LBA}(RE_{i,j,3},\ RT_{i,j,3}|b_j^{(3)},\ A_j,\ v_j,\ s_j,\ \tau_j),$$

respectively, and the likelihood function for the entire data set is

$$L(b,\ A, s,\ v,\ \tau|RT,\ RE) = \prod_{k=1}^{3} \prod_{j=1}^{S} \prod_{i=1}^{N} \mathrm{LBA}\left(RE_{i,j,k},\ RT_{i,j,k}|b_j^{(k)},\ A_j,\ v_j,\ s_j,\ \tau_j\right).$$

To fit the model in a Bayesian framework, we must now specify prior distributions for each of the lower level parameters. Each of these lower level parameters is restricted to be positive, so we specified truncated normal distributions for each parameter, given by

$$\begin{aligned} b_j^{(k)} &\sim TN\left(b_\mu^{(k)},\ b_\sigma^{(k)},\ 0,\ \infty\right),\\ A_j &\sim TN\left(A_\mu\, A_\sigma,\ 0,\ \infty\right),\\ v_j^{(c)} &\sim TN\left(v_\mu^c,\ v_\sigma^c,\ 0,\ \infty\right), \text{ and}\\ \tau_j &\sim TN\left(\tau_\mu,\ \tau_\sigma,\ 0,\ \infty\right), \end{aligned}$$

where TN (a, b, c, d) denotes a truncated normal distribution with mean parameter *a,* standard deviation *b,* lower bound *c,* and upper bound *d.* There are many other priors that one could choose to facilitate the restriction that all model parameters must be positive. We chose the truncated normal distribution because it provides a convenient and interpretable way to specify the mean and dispersion of the lower level parameters. One could further constrain the parameters by observing the restrictions $A < b^{(k)}\ \forall\ k \in \{1, 2, 3\}$ and $\tau_j < \min(RT_j)$, where RT_j represents the response times for the jth subject; however, we found the constraints applied above to be sufficient. To identify the model, we fixed $s_j = 1\ \forall\ j \in \{1, 2, \ldots, S\}$.

We specified mildly informative priors for each of the hypermean parameters, so that

$$\begin{aligned} b_\mu^{(k)} &\sim TN(1,\ 0.5,\ 0,\ \infty),\\ A_\mu &\sim TN(1,\ 0.5,\ 0,\ \infty),\\ v_\mu^{(c)} &\sim TN(2,\ 1,\ 0,\ \infty), \text{ and}\\ \tau_\mu &\sim TN(0.5,\ 0.5,\ 0,\ \infty), \end{aligned}$$

and mildly informative priors for the hyperstandard deviations parameters, so that

$$\begin{aligned} b_\sigma^{(k)} &\sim \Gamma(1,\ 1),\\ A_\sigma &\sim \Gamma(1,\ 1),\\ v_\sigma^{(c)} &\sim \Gamma(1,\ 1), \text{ and}\\ \tau_\sigma &\sim \Gamma(1,\ 1). \end{aligned}$$

Results

We fit the model to data from both the younger and the older subjects separately by implementing a "blocked" version of the DE-MCMC algorithm.[2] To do so, we first paired the mean and standard deviation hyperparameters governing each lower level parameter together and cycled through each of the seven blocks. Figure 7 shows pseudocode for the algorithm we used to fit the hierarchical LBA model. Our first block (see Line 3 of Figure 7) consisted of sampling from the joint conditional distribution of $(b_\mu^{(1)}, b_\sigma^{(1)})$ by updating the K chains with the algorithm presented in Figure 3. The conditional distribution here will depend on the lower level parameters $b_j^{(1)}$. We then turned to the next block (see Line 4 of Figure 7) consisting of the parameters $(b_\mu^{(2)}, b_\sigma^{(2)})$ and so on, until the position of the chain had been updated for every hyperparameter set. We then grouped all of the lower level parameters together by subject (see Line 11 of Figure 7) and updated them by conditioning on the current state of the hyperparameters constructed in the first set. Although its inclusion was not essential, to improve the mixing and convergence properties of the chains, we included a migration step (see Appendix B) with probability 0.05.

We used 24 chains and obtained 2,500 samples after a burn-in period of 500 samples for each chain. We set $\gamma = 2.38/\sqrt{2d}$, where d is the dimensionality of the parameter space (which was either $d = 2$ for the hyperparameters or $d = 7$ for the lower level parameters), and set $b = 0.001$. For data sets for both the young and older subjects, we excluded observations for which the RT was less than 250 ms.

[2] We initially attempted a blocked version of the conventional MCMC algorithm but were unsuccessful. In fact, it was the difficulties we encountered in this attempt that provoked exploration of alternative solutions. See the Discussion for more details.

1: Initialize each chain for the hyper and lower-level parameters.
2: **for** $2 \leq i \leq N$ **do**
3: Update $(b_\mu^{(1)}, b_\sigma^{(1)})$ conditioned on $b_j^{(1)}$ obtained on the $(i-1)$th iteration.
4: Update $(b_\mu^{(2)}, b_\sigma^{(2)})$ conditioned on $b_j^{(2)}$ obtained on the $(i-1)$th iteration.
5: Update $(b_\mu^{(3)}, b_\sigma^{(3)})$ conditioned on $b_j^{(3)}$ obtained on the $(i-1)$th iteration.
6: Update (A_μ, A_σ) conditioned on A_j obtained on the $(i-1)$th iteration.
7: Update $(v_\mu^{(1)}, v_\sigma^{(1)})$ conditioned on $v_j^{(1)}$ obtained on the $(i-1)$th iteration.
8: Update $(v_\mu^{(2)}, v_\sigma^{(2)})$ conditioned on $v_j^{(2)}$ obtained on the $(i-1)$th iteration.
9: Update (τ_μ, τ_σ) conditioned on τ_j obtained on the $(i-1)$th iteration.
10: **for** $1 \leq j \leq J$ **do**
11: Update parameters $(b_j^{(1)}, b_j^{(2)}, b_j^{(3)}, A_j, v_j^{(1)}, v_j^{(2)}, \tau_j)$ with crossover step by conditioning on the hyperparameters obtained on the ith iteration.
12: **end for**
13: **end for**

Figure 7. Pseudocode for fitting the hierarchical linear ballistic accumulator model to data.

Figure 8 shows the estimated hypermean parameters for both the younger (left, light gray densities) and the older (right, dark gray densities) subject groups. The top left panel of Figure 8 shows the estimated posterior distributions for the group mean threshold parameter b_μ for the accuracy (left density), neutral (middle density), and speed (right density) conditions. Across age, the estimates for the speed condition are much lower than the other conditions, whereas there is only a slight difference in these estimates between the neutral and accuracy conditions. The top right panel shows the estimates for the group mean of the upper bound of the start point distribution A_μ. The relative estimates of the ratio of the start point to the threshold parameter (i.e., A/b, not separately plotted) do not differ appreciably across age. The bottom left column shows the group mean drift rate parameters for the incorrect (left curve) and the correct (right curve) accumulators. The relative estimates of the drift rates suggest that correct responses were faster than incorrect responses. Finally, the right column of Figure 8 shows the estimated mean nondecision time parameters for the two data sets. Because the estimates are for the mean parameter of a truncated normal distribution bounded by zero, the estimates are difficult to compare when significant mass is truncated, as with the nondecision parameter. To examine this more carefully, we generated a posterior predictive distribution with the obtained posterior estimates for both data sets. We found that, on average, the best estimate for the nondecision time parameter at the individual level was 0.16 for the elderly subjects and 0.17 for the young subjects. Although the difference in means suggests that the younger subjects had a larger nondecision time parameter, the posteriors were variable enough to conclude that there was not an appreciable difference in the parameter estimates across the different age groups.

In all, 266 posterior estimates were obtained: 238 individual-level posteriors and 28 hyper-level posteriors. Although for brevity we do not present them here, plots of model predictions against the data confirm that the average parameter values provide a good fit to the data. Furthermore, the parameter estimates provide a reasonable explanation of the data that is consistent with previous findings (e.g., Forstmann et al., 2011; Ratcliff, Thapar, & McKoon, 2007; Starns & Ratcliff, 2010).

Discussion

In this article, we have advocated for using DE-MCMC as a method to draw samples from joint posterior distributions. The DE-MCMC algorithm is highly efficient, particularly when the dimensions of a posterior distribution are highly correlated. The problem of estimating the parameters of a model with highly correlated parameters has been studied extensively, and there are many alternatives to the DE-MCMC approach. Robert and Sahu (1997) proposed blocked sampling as a Gibbs step with a multivariate normal draw over effect parameters. As we showed in the simulated example, blocked sampling is an excellent approach that reduces the dimensionality of the parameter space, which in turn increases the acceptance rate of the sampler (see Gelman et al., 2004, for further discussion). Another Gibbs-style approach is to use decorrelating steps as proposed by Liu and Sabatti (2000), which has been effectively used in psychology (Morey, Rouder, & Speckman, 2008, 2009). Many other approaches, such as adaptive rejection sampling (Gilks & Wild, 1992), adaptive rejection Metropolis sampling (Gilks, Best, & Tan, 1995), Hamiltonian MCMC (Hoffman & Gelman, 2011), and blocked Metropolis–Hastings sampling (Patz & Junker, 1999a, 1999b), have been proposed as alternative solutions to estimation of high-dimensional, highly correlated parameter spaces.

However, many of these methods require significant additional work for implementation (e.g., calculating the derivative of the log likelihood function in adaptive rejection sampling) or can be characterized as "multiple try" methods. Multiple try methods are methods that require the evaluation of the likelihood function multiple times per iteration in the algorithm. For many models, the evaluation of the likelihood can be costly, resulting in poor performance of the algorithm. Gibbs and, more generally, blocked sampling have a related problem, because they sample from the conditional distributions for a block of parameters. Such procedures are generally slower than a single evaluation of the likelihood per iteration.

One of the most compelling reasons to use the DE-MCMC algorithm is that the information about the structure of the posterior distribution is used to generate informative proposals. The

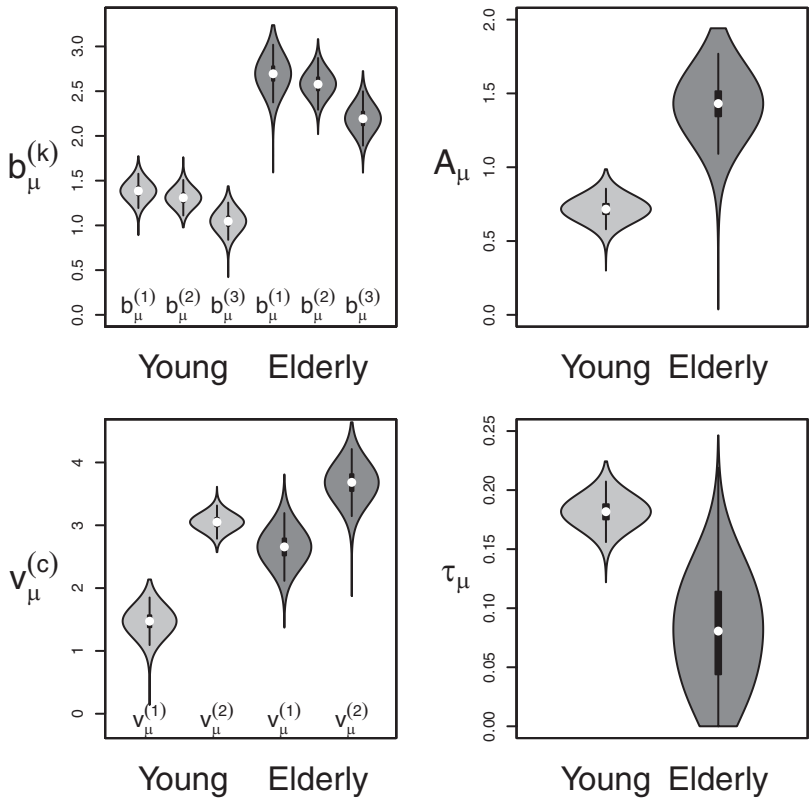

Figure 8. The estimated posterior distributions for each mean hyperparameter for the young subjects' data (left, light gray) and the older subjects' data (right, dark gray). For mean threshold parameters b_μ (top left panel), the left, middle, and right densities correspond to the accuracy, neutral, and speed conditions, respectively. For the mean drift rate parameters v_μ (bottom left panel), the left and right densities correspond to the incorrect and correct responses, respectively. The estimated posterior distribution of the upper bound of the starting point A_μ is shown in the top right panel, and the estimated posterior distribution for the mean parameter τ_μ of the nondecision time is shown in the bottom right panel. For each violin plot, the thin black lines are the effective range of the data (minimum and maximum data points that are still within 1.5 times the interquartile range), the thick black lines represent the range from the 0.25 and 0.75 quantiles, and the white dots represent the median.

DE-MCMC algorithm uses the difference between two randomly selected chains to approximate the derivative information used by multiple try methods.

We also argued that some algorithms, such as conventional MCMC, are difficult to tune properly. By contrast, the DE-MCMC algorithm requires only one tuning parameter, γ, regardless of the dimensionality of the posterior. However, other specifications can play an important role in the performance of the sampler. First, though the selection of the distribution of the random error term ε has little influence, the variance of ε is important. A good rule of thumb is to select b so that the variance of the distribution of ε is very small with respect to the target distribution of interest but large enough so that the sampler is not deterministic. Second, one must select enough chains to allow the sampler to mix properly. In our studies, we have found that setting the number of chains to at least one greater than twice the dimensionality of the posterior to be adequate. When blocked sampling is used, the dimensionality of the highest block becomes the critical number.

It is important to emphasize that poor mixing behavior is a direct result of the complexity of the model. Mixing behavior of chains can be greatly improved by reparameterizing or rethinking the parameters of the model, so that each parameter operates independently. Such a reparameterization would ameliorate the problem of a highly correlated parameter space, which, in turn, would allow for efficient use of conventional MCMC algorithms (see Figure 4). However, reparameterization can be difficult, and solutions that work well for one model may not be possible for other models. Considering this, we instead focused on presenting a generalizable method for fitting unrefined models to data.

Although the DE-MCMC algorithm worked very well for the LBA example, one may wonder why conventional MCMC would not work. We attempted a number of sampling schemes but were largely unsuccessful as a result of the heavy correlations between the individual-level parameters. To illustrate this, the top right portion of Figure 9 shows the joint posterior distributions for each pair of the seven parameters for Subject 13 of the older group, who was randomly chosen. The bottom left portion of Figure 9 shows the correlations of the corresponding joint posterior. Of the 21 correlations, only four of them have magnitudes below 0.40, and nine of them are greater than or equal to 0.70. From our simulation study (see Figure 4), we can see that under these extreme correlations, conventional MCMC will suffer severely from high rejec-

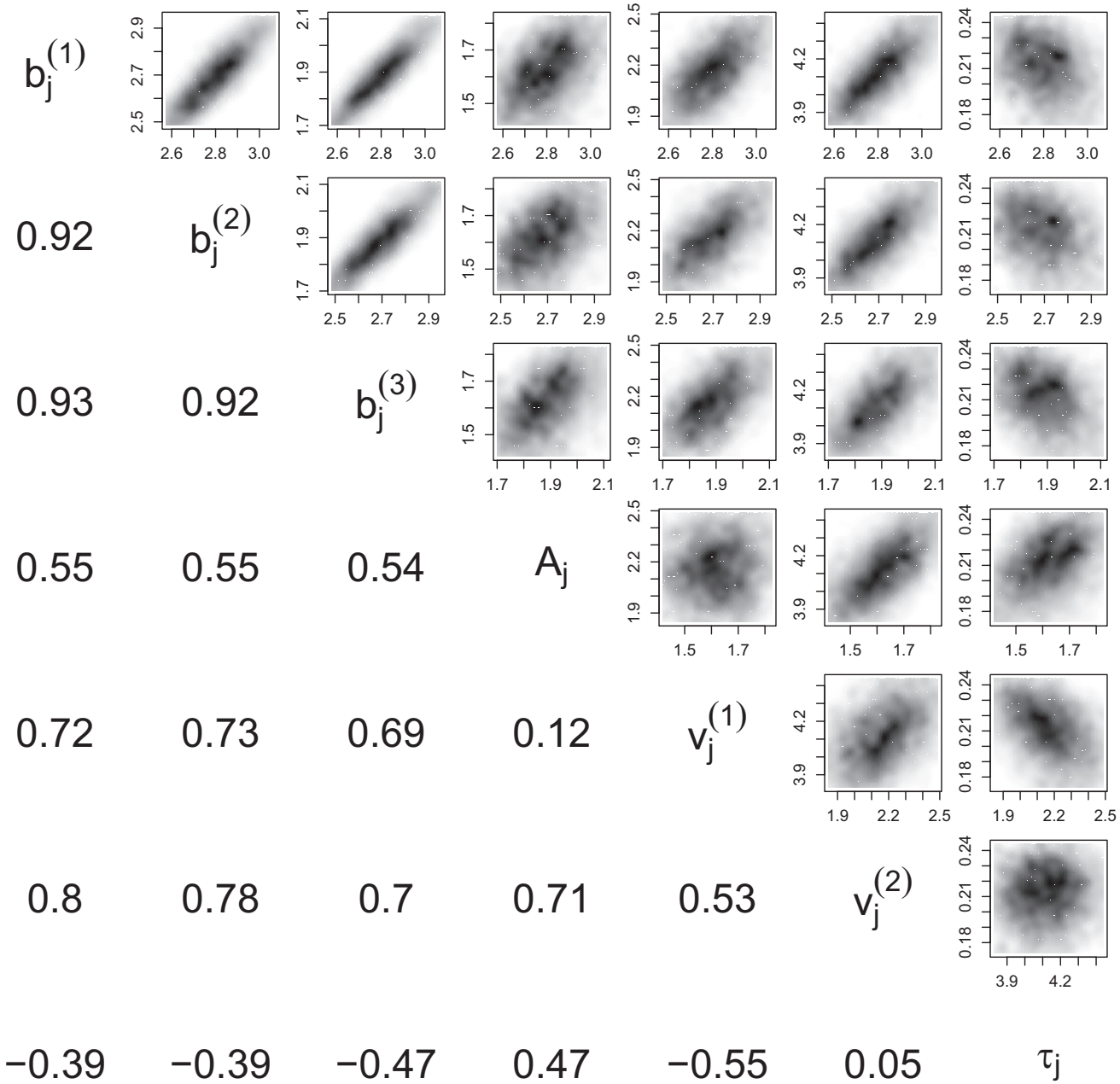

Figure 9. Joint posterior distributions for each of the seven individual-level parameters for Subject 13 (top right corner) along with the correlations (bottom left corner).

tion rates. As a result, the quality of the estimated posteriors will be poor.

The success of the DE-MCMC sampler on such a difficult problem as the hierarchical LBA speaks to the advantages of the approach. The high correlations exhibited by the LBA are typical in models of response time. For example, Turner and Sederberg (2012) have shown that the posterior distributions of the parameters of the Wald model are similarly correlated, and the use of a DE variant was central to the success of their application of a likelihood-free sampler. In addition, by using WinBUGS (see Vandekerckhove et al., 2011) or similar methods, one can show that the joint posterior distributions of the diffusion model (Ratcliff, 1978) show patterns similar to those observed in Figure 9.

Conclusions

In this article, we have advocated the use of DE-MCMC as a method for estimating a target density $\pi(\theta)$. Although there are a

number of different sampling methods that could outperform conventional MCMC and perhaps even DE-MCMC, we have shown that the DE-MCMC algorithm is a highly efficient sampler that requires minimal tuning and is very simple to implement. We have argued that the conventional approach is both difficult to successfully implement and unlikely to generalize to different models or data sets. On the other hand, the DE-MCMC approach is easy to set up. Even in its simplest form, it can easily handle difficult problems, such as fitting a hierarchical LBA model. We have shown in a simulation study that the DE-MCMC approach is as good as or much better than the standard Metropolis–Hastings algorithm, depending on the degree of correlation between the parameters of a model. Given the success of the DE-MCMC algorithm in the present article, we believe that the DE-MCMC approach will make obtaining the Bayesian posterior a much more practical endeavor in the field of mathematical psychology.

References

Andrieu, C., & Thomas, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing, 18,* 343–373. doi:10.1007/s11222-008-9110-y

Brown, S., & Heathcote, A. (2005). A ballistic model of choice response time. *Psychological Review, 112,* 117–128. doi:10.1037/0033-295X.112.1.117

Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice reaction time: Linear ballistic accumulation. *Cognitive Psychology, 57,* 153–178. doi:10.1016/j.cogpsych.2007.12.002

Craigmile, P. F., Peruggia, M., & Van Zandt, T. (2010). Hierarchical Bayes models for response time data. *Psychometrika, 75,* 613–632. doi:10.1007/s11336-010-9172-6

Donkin, C., Averell, L., Brown, S., & Heathcote, A. (2009). Getting more from accuracy and response time data: Methods for fitting the linear ballistic accumulator. *Behavior Research Methods, 41,* 1095–1110. doi:10.3758/BRM.41.4.1095

Donkin, C., Brown, S., & Heathcote, A. (2011). Drawing conclusions from choice response time models: A tutorial. *Journal of Mathematical Psychology, 55,* 140–151. doi:10.1016/j.jmp.2010.10.001

Donkin, C., Heathcote, A., & Brown, S. (2009). Is the linear ballistic accumulator model really the simplest model of choice response times: A Bayesian model complexity analysis. In A. Howes, D. Peebles, & R. Cooper (Eds.), *Proceedings of the 9th International Conference on Cognitive Modeling*. Retrieved from http://iccm-conference.org/iccm-archive/2009/sideshow.psyc.bbk.ac.uk/rcooper/iccm2009/proceedings/

Dutilh, G., Vandekerckhove, J., Tuerlinckx, F., & Wagenmakers, E.-J. (2009). A diffusion model decomposition of the practice effect. *Psychonomic Bulletin & Review, 16,* 1026–1036. doi:10.3758/16.6.1026

Edwards, M. C. (2010). A Markov chain Monte Carlo approach to confirmatory item factor analysis. *Psychometrika, 75,* 474–497. doi:10.1007/s11336-010-9161-9

Farrell, S., & Ludwig, C. J. H. (2008). Bayesian and maximum likelihood estimation of hierarchical response time models. *Psychonomic Bulletin & Review, 15,* 1209–1217. doi:10.3758/PBR.15.6.1209

Forstmann, B. U., Anwander, A., Schäfer, A., Neumann, J., Brown, S., Wagenmakers, E.-J., . . . Turner, R. (2010). Cortico-striatal connections predict control over speed and accuracy in perceptual decision making. *Proceedings of the National Academy of Sciences, USA, 107,* 15916–15920. doi:10.1073/pnas.1004932107

Forstmann, B. U., Dutilh, G., Brown, S., Neumann, J., von Cramon, D. Y., Ridderinkhof, K. R., & Wagenmakers, E.-J. (2008). Striatum and pre-SMA facilitate decision-making under time pressure. *Proceedings of the National Academy of Sciences, USA, 105,* 17538–17542. doi:10.1073/pnas.0805903105

Forstmann, B. U., Tittgemeyer, M., Wagenmakers, E.-J., Derrfuss, J., Imperati, D., & Brown, S. (2011). The speed–accuracy tradeoff in the elderly brain: A structural model-based approach. *Journal of Neuroscience, 31,* 17242–17249. doi:10.1523/JNEUROSCI.0309-11.2011

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2004). *Bayesian data analysis*. New York, NY: Chapman & Hall.

Gershman, S. J., Blei, D. M., Pereira, F., & Norman, K. A. (2011). A topographic latent souce model for fMRI data. *NeuroImage, 57,* 89–100. doi:10.1016/j.neuroimage.2011.04.042

Gilks, W. R., Best, N. G., & Tan, K. K. C. (1995). Adaptive rejection Metropolis sampling within Gibbs sampling. *Applied Statistics, 44,* 455–472. doi:10.2307/2986138

Gilks, W. R., & Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics, 41,* 337–348. doi:10.2307/2347565

Hoffman, M. D., & Gelman, A. (2011). *The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo*. Manuscript submitted for publication.

Hu, B., & Tsui, K.-W. (2005). Distributed evolutionary Monte Carlo with applications to Bayesian analysis (Technical Report No. 1112). Retrieved from http://www.stat.wisc.edu/techreports/tr1112.pdf

Klein Entink, R. H., Kuhn, J.-T., Hornke, L. F., & Fox, J.-P. (2009). Evaluating cognitive theory: A joint modeling approach using responses and response times. *Psychological Methods, 14,* 54–75. doi:10.1037/a0014877

Klugkist, I., Laudy, O., & Hoijtink, H. (2010). Bayesian evaluation of inequality constrained hypotheses for contingency tables. *Psychological Methods, 15,* 281–299. doi:10.1037/a0020137

Kullback, S., Keegel, J. C., & Kullback, J. H. (1987). *Topics in statistical information theory*. New York, NY: Springer-Verlag.

Lee, M. D., Fuss, I. G., & Navarro, D. J. (2006). A Bayesian approach to diffusion models of decision-making and response time. In B. Scholkopf, J. Platt, & T. Hoffman (Eds.), *Advances in Neural Information Processing 19* (pp. 809–815). Cambridge, MA: MIT Press.

Lee, M., Steyvers, M., de Young, M., & Miller, B. (2011). A model-based approach to measuring expertise in ranking tasks. In L. Carlson, C. Hoelscher, & T. F. Shipley (Eds.), *Proceedings of the 33rd Annual Conference of the Cognitive Science Society* (pp. 1304–1309). Austin, TX: Cognitive Science Society.

Lee, M. D., & Wagenmakers, E.-J. (2010). *A course in Bayesian graphical modeling for cognitive science*. Retrieved from http://www.ejwagenmakers.com/BayesCourse/BayesBookWeb.pdf

Liu, J. S., & Sabatti, C. (2000). Generalised Gibbs sampler and multigrid Monte Carlo for Bayesian computation. *Biometrika, 87,* 353–369. doi:10.1093/biomet/87.2.353

Lunn, D., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS—A Bayesian modelling framework: Concepts, structure and extensibility. *Statistics and Computing, 10,* 325–337. doi:10.1023/A:1008929526011

McArdle, J. J., Grimm, K. J., Hamagami, F., Bowles, R. P., & Meredith, W. (2009). Modeling life-span growth curves of cognition using longitudinal data with multiple samples and changing scales of measurement. *Psychological Methods, 14,* 126–149. doi:10.1037/a0015857

Morey, R. D., Rouder, J. N., & Speckman, P. (2008). A statistical model for discriminating between subliminal and near-liminal performance. *Journal of Mathematical Psychology, 52,* 21–36. doi:10.1016/j.jmp.2007.09.007

Morey, R. D., Rouder, J. N., & Speckman, P. (2009). A truncated-probit item response model for estimating psychophysical thresholds. *Psychometrika, 74,* 603–618. doi:10.1007/s11336-009-9122-3

Muthén, B., & Asparouhov, T. (2012). Bayesian structural equation modeling: A more flexible representation of substantive theory. *Psychological Methods, 17,* 313–335. doi:10.1037/a0026802

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. In S. Brooks, A. Gelman, G. Jones, & X.-L. Meng (Eds.), *Handbook of Markov chain Monte Carlo* (pp. 113–162). Boca Raton, FL: CRC Press.

Oravecz, Z., Tuerlinckx, F., & Vandekerckhove, J. (2009). A hierarchical Ornstein–Uhlenbeck model for continuous repeated measurement data. *Psychometrika, 74,* 395–418. doi:10.1007/s11336-008-9106-8

Patz, R. J., & Junker, B. W. (1999a). Applications and extensions of MCMC in IRT: Multiple item types, missing data, and rated responses. *Journal of Educational and Behavioral Statistics, 24,* 342–366.

Patz, R. J., & Junker, B. W. (1999b). A straightforward approach to Markov chain Monte Carlo methods for item response models. *Journal of Educational and Behavioral Statistics, 24,* 146–178.

Peruggia, M., Van Zandt, T., & Chen, M. (2002). Was it a car or a cat I saw? An analysis of response times for word recognition. *Case Studies in Bayesian Statistics 6,* 319–334.

Pooley, J. P., Lee, M. D., & Shankle, W. R. (2011). Understanding memory impairment with memory models and hierarchical Bayesian analysis. *Journal of Mathematical Psychology, 55,* 47–56. doi:10.1016/j.jmp.2010.08.003

Prevost, A. T., Mason, D., Griffin, S., Kinmonth, A.-L., Sutton, S., & Spiegelhalter, D. (2007). Allowing for correlations between correlations in random-effect meta-analysis of correlation matrices. *Psychological Methods, 12,* 434–450. doi:10.1037/1082-989X.12.4.434

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review, 85,* 59–108. doi:10.1037/0033-295X.85.2.59

Ratcliff, R., Thapar, A., & McKoon, G. (2007). Application of the diffusion model to two-choice tasks for adults 75–90 years old. *Psychology and Aging, 22,* 56–66. doi:10.1037/0882-7974.22.1.56

Robert, C. P., & Casella, G. (2004). *Monte Carlo statistical methods*. New York, NY: Springer.

Robert, G. O., & Sahu, S. (1997). Updating schemes, correlation structure, blocking and parameterisation for the Gibbs sampler. *Journal of the Royal Statistical Society, Series B: Statistical methodology, 59,* 291–317. doi:10.1111/1467-9868.00070

Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review, 12,* 573–604. doi:10.3758/BF03196750

Rouder, J. N., Lu, J., Speckman, P., Sun, D., & Jiang, Y. (2005). A hierarchical model for estimating response time distributions. *Psychonomic Bulletin & Review, 12,* 195–223. doi:10.3758/BF03257252

Rouder, J., Sun, D., Speckman, P., Lu, J., & Zhou, D. (2003). A hierarchical Bayesian statistical framework for response time distributions. *Psychometrika, 68,* 589–606. doi:10.1007/BF02295614

Silverman, B. W. (1986). *Density estimation for statistics and data analysis*. London, England: Chapman & Hall.

Song, X.-Y., & Lee, S.-Y. (2012). A tutorial on the Bayesian approach for analyzing structural equation models. *Journal of Mathematical Psychology, 56,* 135–148. doi:10.1016/j.jmp.2012.02.001

Starns, J. J., & Ratcliff, R. (2010). The effects of aging on the speed–accuracy compromise: Boundary optimality in the diffusion model. *Psychological Aging, 25,* 377–390. doi:10.1037/a0018022

Stone, M. (1960). Models for choice reaction time. *Psychometrika, 25,* 251–260. doi:10.1007/BF02289729

Storn, R. (2008). Differential evolution research—Trends and open questions. In U. K. Chakraborty (Ed.), *Advances in differential evolution* (pp. 1–31). New York, NY: Springer.

Storn, R., & Price, K. (1997). Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization, 11,* 341–359. doi:10.1023/A:1008202821328

Tanese, R. (1989). Distributed genetic algorithms. In J. D. Schaffer (Ed.), *Proceedings of the Third International Conference on Genetic Algorithms* (pp. 434–439). San Mateo, CA: Kaufmann.

ter Braak, C. J. F. (2006). A Markov chain Monte Carlo version of the genetic algorithm differential evolution: Easy Bayesian computing for real parameter spaces. *Statistics and Computing, 16,* 239–249. doi:10.1007/s11222-006-8769-1

ter Braak, C. J., & Vrugt, J. A. (2008). Differential evolution Markov chain with snooker updater and fewer chains. *Statistics and Computing, 18,* 435–446. doi:10.1007/s11222-008-9104-9

Turner, B. M., Dennis, S., & Van Zandt, T. (in press). Likelihood-free Bayesian analysis of memory models. *Psychological Review*. doi:10.1037/a0032458

Turner, B. M., & Sederberg, P. B. (2012). Approximate Bayesian computation with differential evolution. *Journal of Mathematical Psychology, 56,* 375–385. doi:10.1016/j.jmp.2012.06.004

Usher, M., & McClelland, J. L. (2001). On the time course of perceptual choice: The leaky competing accumulator model. *Psychological Review, 108,* 550–592. doi:10.1037/0033-295X.108.3.550

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2008). A Bayesian approach to diffusion process models of decision-making. In V. M. Sloutsky, B. C. Love, & K. McRae (Eds.), *Proceedings of the 30rd Annual Conference of the Cognitive Science Society* (pp. 1429–1434). Austin, TX: Cognitive Science Society.

Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response time. *Psychological Methods, 16,* 44–62. doi:10.1037/a0021765

Vandekerckhove, J., Verheyen, S., & Tuerlinckx, F. (2010). A crossed random effects diffusion model for speeded semantic categorization data. *Acta Psychologica, 133,* 269–282. doi:10.1016/j.actpsy.2009.10.009

Vanpaemel, W. (2009). BayesGCM: Software for Bayesian inference with the generalized context model. *Behavior Research Methods, 41,* 1111–1120. doi:10.3758/BRM.41.4.1111

Vrugt, J. A., ter Braak, C. J. F., Diks, C. G. H., Robinson, B. A., Hyman, J. M., & Higdon, D. (2009). Accelerating Markov chain Monte Carlo simulation by differential evolution with self-adaptive randomized subspace sampling. *International Journal of Nonlinear Sciences & Numerical Simulation, 10,* 271–288.

Wirth, R. J., & Edwards, M. C. (2007). Item factor analysis: Current approaches and future directions. *Psychological Methods, 12,* 58–79. doi:10.1037/1082-989X.12.1.58

Yuan, Y., & MacKinnon, D. P. (2009). Bayesian mediation analysis. *Psychological Methods, 14,* 301–322. doi:10.1037/a0016972

(*Appendices follow*)

Appendix A

Variants of DE-MCMC

Since its inception in 1995, DE has seen tremendous growth both in its applications and in variants to improve the efficiency of the algorithm (see Storn, 2008). For example, a number of variants select a chain at random or use the best individual in the population as the base for the crossover step instead of simply using the current chain θ_k. Two of the most widely used of these variants are the DE/best/1/bin and the DE/current-to-best/1/bin crossover and mutation methods. The notation here is common in the DE literature, whereby the first value, DE, stands for differential evolution, the second value in the name describes the base particle, the third value describes how many differences between two random particles are taken (in this case just 1 difference), and the final value in the name refers to the use of a binomial probability for determining whether each parameter of an individual changes from the previous generation to the new proposed value. As an explicit example, the proposal generation process in the current-to-best algorithm follows:

$$\theta^* = \theta_k + \gamma_1(\theta_b - \theta_k) + \gamma_2(\theta_m - \theta_n).$$

where θ_k is the chain for which the proposal is being generated, θ_b is the best performing chain from the previous iteration (e.g., the chain with the highest probability density), θ_m and θ_n are other chains sampled with uniform probability, and γ_1 and γ_2 scale the mutation.

Guiding the proposals with the best performing chains can give rise to much higher acceptance rates and faster convergence to the maximum of the target distribution; however, these improvements come with a cost. Very few of these variants produce symmetric transition probabilities, which requires that we derive the probability distribution for the proposal and reverse moves. Deriving this distribution can be complicated. Even when it has been derived, it adds considerable computation for each proposal evaluation, even for a small number of chains. Solving for θ_k in the current-to-best algorithm above demonstrates that incorporating the best chain renders the probability of transitioning to the proposal θ^* much higher than the reverse transition from the proposal back to θ_k. Consequently, although these improved algorithms are perfectly suited for generating maximum a posteriori (MAP) estimates, they are not appropriate for estimating the target distribution.

Many of the DE variants that guide proposals based on the best chains can, however, be employed during a burn-in period. After the burn-in period, one can use the symmetric DE algorithms to estimate the full target distribution. As a result, the chains converge very quickly to the high-density regions of the target distribution; then, after the burn-in period, one can completely map out the target distribution with the symmetric DE algorithm. Turner and Sederberg (2012) employed such a method by focusing on the current-to-best variant, because simply setting $\gamma_1 = 0$ shifts the algorithm back to the original symmetric proposal scheme. This makes it easy to incorporate in the simulation.

Not all the DE algorithmic advances pose such problems for inclusion in a Bayesian framework. For example, another strategy for optimizing the crossover move is to sample the full set of chains and use the sum of the difference vectors to generate a proposal of the form

$$\theta^* = \theta_k + \gamma\left(\sum_i^Q \theta_m^{(i)} - \sum_j^Q \theta_n^{(j)}\right) + \varepsilon,$$

where Q is the total number of pairs to generate, and $\theta_m{}^{(i)}$ and $\theta_n{}^{(j)} \in \{1, 2, \ldots, Q\}$. This type of generation stabilizes the variability in the directional vector and can lead to better performance (Vrugt et al., 2009). Algorithms of this form can also be used to generate proposals that rely on previous states of the chains (ter Braak and Vrugt, 2008).

Finally, more complex modifications to the basic DE algorithm, such as the snooker updater algorithm proposed by ter Braak and Vrugt (2008) and the distributed evolutionary Monte Carlo algorithm proposed by Hu and Tsui (2005), have proved useful for Bayesian estimation in high dimensional spaces. In the latter algorithm, Hu and Tsui (2005) incorporated a migration step from a distributed genetic algorithm framework that, when used at random intervals in the basic DE algorithm, has been shown to improve the behavior of the sampler (Tanese, 1989). Given that the migration is deterministic and simply involves moving particles between groups, it does not introduce a bias in the proposals and can be included in the evolution algorithm to help diversify the Markov chains. We now discuss the migration step in more detail.

(*Appendices continue*)

Appendix B

The Migration Step

Although the crossover step is a very efficient means of proposal generation, it can perform quite poorly when the individual chains are initialized poorly, such as specifying the chains to values that are very far from the target density. A similar problem occurs when a single chain or a minority of the chains are far from the target density (referred to as "outlier" chains). When this occurs, the outlier chains are still selected with uniform probability, but they will tend to generate proposals that are far from the target density because the difference vector will be large. It is clear that these outlier chains may have a significant impact on the convergence and mixing properties of the sampler.

To remedy the problem of outlier chains, Hu and Tsui (2005) proposed a migration step taken from the distributed genetic algorithm framework (Tanese, 1989) to efficiently circulate the states of the chains. The idea is to propose a jump from one chain's current state to another chain's current state. This proposal often includes multiple chain states being swapped in a cyclical fashion, so that Chain 1 moves to Chain 2 and Chain 2 moves to Chain 3 and Chain 3 moves to Chain 1.

More formally, to perform the migration step, we must first determine the number of chains that will be involved in the swapping by sampling a number η uniformly from the set $K = \{1, 2, \ldots, K\}$. Then, to determine which of the chains to use, we sample η numbers without replacement from K, forming the group set $G = \{G_1, G_2, \ldots, G_\eta\}$. Finally, we swap the states of the chains from each of the sets in a cyclical fashion, so that

$$\left\{\theta_{G_1}, \theta_{G_2}, \ldots, \theta_{G_{\eta-1}}\, \theta_{G_\eta}\right\} \rightarrow \left\{\theta_{G_\eta}, \theta_{G_1}, \ldots, \theta_{G_{\eta-2}}\, \theta_{G_{\eta-1}}\right\}.$$

When the cost associated with evaluating the target density function is high, it is most efficient if this swap is a deterministic one and does not rely on the Metropolis–Hastings probability (e.g., Turner and Sederberg, 2012). Although a deterministic transition rule works to diversify groups of particles, it does not solve the problem of outlier chains. However, we can modify the migration step by adding a small amount of noise to each proposal, so that

$$\begin{aligned}
\theta^*_{G_\eta} &= \theta_{G_\eta} + \varepsilon \\
\theta^*_{G_1} &= \theta_{G_1} + \varepsilon \\
&\vdots \\
\theta^*_{G_{\eta-1}} &= \theta_{G_{\eta-1}} + \varepsilon.
\end{aligned}$$

If ε is sampled from a symmetric distribution (e.g., a continuous uniform distribution), then the forward and backward transition probabilities may again be safely ignored and the probability of, say θ_{G_1} jumping to $\theta^*_{G_\eta}$, reduces to

$$\alpha = \min\left(1, \frac{\pi(\theta^*_{G_\eta})}{\pi(\theta_{G_1})}\right).$$

When the migration step is carried out in this way, outlier chains will quickly be pulled into higher density regions of the posterior and will not be replaced by other chains (as in the deterministic rule). In our own studies, we have found the migration step to be an extremely useful tool when good initial values are difficult to determine.

Received May 16, 2012
Revision received October 11, 2012
Accepted February 5, 2013 ■