PSYCHOMETRIKA—VOL. 79, NO. 2, 185–209
APRIL 2014
DOI: 10.1007/S11336-013-9381-X

HIERARCHICAL APPROXIMATE BAYESIAN COMPUTATION

BRANDON M. TURNER
STANFORD UNIVERSITY

TRISHA VAN ZANDT
THE OHIO STATE UNIVERSITY

Approximate Bayesian computation (ABC) is a powerful technique for estimating the posterior distribution of a model's parameters. It is especially important when the model to be fit has no explicit likelihood function, which happens for computational (or simulation-based) models such as those that are popular in cognitive neuroscience and other areas in psychology. However, ABC is usually applied only to models with few parameters. Extending ABC to hierarchical models has been difficult because high-dimensional hierarchical models add computational complexity that conventional ABC cannot accommodate. In this paper, we summarize some current approaches for performing hierarchical ABC and introduce a new algorithm called Gibbs ABC. This new algorithm incorporates well-known Bayesian techniques to improve the accuracy and efficiency of the ABC approach for estimation of hierarchical models. We then use the Gibbs ABC algorithm to estimate the parameters of two models of signal detection, one with and one without a tractable likelihood function.

Key words: approximate Bayesian computation, hierarchical Bayesian estimation, signal detection theory, dynamic signal detection.

1. Introduction

Recently, there has been great interest in Bayesian estimation techniques, and our work (Turner & Van Zandt, 2012; Turner, Dennis, & Van Zandt, 2013; Turner & Sederberg, 2012) has focused on a particular approach called approximate Bayesian computation (ABC). The goal of this paper is to demonstrate how ABC can be applied to hierarchical models, in which the data-generating mechanisms for subjects are nested within a larger global structure that restricts the parameters for each subject.

To make the concepts that we will discuss more clear, we will orient our presentation around the classic model of signal detection theory (SDT; e.g., Green & Swets, 1966; Egan, 1958). We chose SDT as our working example because it is simple, well known in experimental psychology, and also because the classic SDT model can be contrasted with a more recent approach for which explicit predictions are difficult to derive (Turner, Van Zandt, & Brown, 2011). Such models, those without analytic expressions to describe their output, are usually explored by way of simulation and are the kinds of models for which ABC was designed. Note, however, that the techniques we present in this paper are applicable to a wide variety of models, and are not intended to be restricted to SDT alone.

1.1. Signal Detection Theory: Our Working Example

SDT is commonly applied to two-choice data in which signals (e.g., an auditory tone) are embedded in noise. For example, an auditory signal detection experiment might ask participants

Requests for reprints should be sent to Brandon M. Turner, Stanford University, Stanford, USA. E-mail: turner.826@gmail.com

to respond either "yes," indicating that they did hear a tone, or "no," indicating that they did not hear a tone after the presentation of a stimulus. The variability in the sensory effect of the stimulus is represented by two random variables. One variable represents the sensory effect of noise when no signal is presented, while the other variable represents the sensory effect of a signal.

The classic (equal-variance) SDT model has two parameters. The first parameter d is the standardized distance between the means of the signal and noise distributions. The parameter d represents the discriminability of the stimuli, such that higher values of d result in less overlap between the two distributions, and hence signals are more easily recognized. The model further assumes the presence of a fixed criterion c somewhere along the axis of sensory effect. Stimuli that have sensory effects greater than c are labeled signals and elicit a "yes" response, while stimuli that have sensory effects less than c are labeled noise and elicit a "no" response (see Macmillan & Creelman, 2005, for a review).

When the signal and noise representations have equal variance and the payoffs and penalties for correct and incorrect responses are the same for both signal and noise trials, an "optimal" observer should place their criterion c at $d/2$, the point at which the two representations cross or, equivalently, the point at which the likelihood that the stimulus is a signal equals the likelihood that it is noise. We can then write an observer's criterion c as $d/2 + b$, where b represents the observer's bias. Negative bias results in an downward shift of the criterion along the axis of sensory effect, whereas positive bias results in an upward shift.

Both d and b are psychologically meaningful in that they represent two critical ideas in perceptual decision making. The parameter d reflects the degree of difference between the two stimulus classes, and is assumed to change as the stimulus classes become more or less similar. The parameter b is a subject-specific parameter that reflects the subject's bias to respond either "yes" or "no." In an experiment, we manipulate the stimuli and observe changes in d, and manipulate stimulus frequencies or payoffs for correct and incorrect responses and observe changes in b. If the estimated values for d and b do not change with experimental conditions in ways that are theoretically sensible, then we can question the validity of the SDT model in that particular experimental context.

Figure 1 shows the equal-variance SDT model. The Gaussian distribution on the right represents the signal representation and the distribution on the left represents the noise representation. The criterion c is represented as the solid vertical line, which shows a slight positive bias (i.e., a tendency to say "no" more frequently than would an optimal observer). The light gray shaded region corresponds to the probability of a "yes" response when a signal stimuli is presented (i.e., the hit rate) whereas the dark gray shaded region corresponds to the probability of a "no" response when a noise stimulus is presented (i.e., the false alarm rate).

In equal-variance SDT, we can explicitly solve for d and b given the correct and incorrect response frequencies in the different stimulus categories. For more complex models, parameter estimates can be obtained in a number of ways, including maximum likelihood (e.g., Dorfman & Alf, 1969; Myung, 2003; Van Zandt, 2000) or least squares (e.g., Van Zandt, Colonius, & Proctor, 2000; McElree & Dosher, 1993; Nosofsky & Palmeri, 1997). These techniques are often limited in the extent to which parameters for subjects in the experiment are permitted to vary. For instance, we usually assume that the data-generating mechanism (such as SDT) is the same across all subjects (e.g., Nosofsky, Little, Donkin, & Fific, 2011).

Psychologists are often interested in systematic differences between groups or subjects. Subject-specific details such as age, demographic factors, or gender may be expected to influence a subject's performance on different tasks. For example, older observers might have lower ds than younger subjects. One naïve approach to understanding these subject differences is to assume that they manifest as differences in parameters across subjects, and so we might estimate model parameters (d and b) for each subject independently. We could then use these parameter

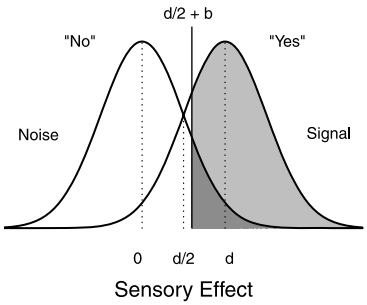

FIGURE 1.
The classic, equal-variance model of signal detection theory. Representations for signals and noise are represented as equal-variance Gaussian distributions, separated by a distance of d: the discriminability parameter. A criterion, shown as the vertical line, is used to determine the response. Any deviation from the optimal criterion placement (i.e., at $d/2$) is known as a "bias," and is measured by the parameter b.

estimates as subject measurements in the same way that we might treat the data. That is, we could perform inferential statistical analysis on the estimated parameters to draw conclusions about the influence of the experimental conditions on the underlying data-generating mechanism.

However, another approach is to assume that the subject-level parameters share some commonality, a relationship that is described by "group-level" parameters. One could then *simultaneously* estimate the parameters specific to each subject and the parameters that are common to the group in a hierarchical structure. For example, we might assume that each observer's d can be different, but that all of the ds over subjects are constrained in some theoretically interesting way. This approach is called hierarchical modeling.

1.2. Bayesian Hierarchical Modeling

Hierarchical modeling can be approached in a number of different ways. In this paper, we will discuss hierarchical modeling within the Bayesian framework (e.g., Efron, 1986; Lee, 2008; Shiffrin, Lee, Kim, & Wagenmakers, 2008; Rouder & Lu, 2005; Rouder, Lu, Speckman, Sun, & Jiang, 2005; Rouder, Sun, Speckman, Lu, & Zhou, 2003; Vandekerckhove, Tuerlinckx, & Lee, 2011). Bayesian inference treats the parameters of the model (which we assume generated the data) as random variables, just as the data are random variables. This randomness can be viewed either as reflecting the assumption that parameters fluctuate over time, subject or experimental conditions, or as reflecting our uncertainty about the true values of the parameters. The Bayesian approach provides a probability distribution for the possible values of the parameters of a model given the observed data. This probability distribution is called the posterior distribution of the parameters.

To acquire the posterior distribution we require two things: a prior distribution for the parameters and a likelihood function for the data. The prior distribution reflects our prior knowledge or beliefs about possible values for the parameters. For example, in classic SDT, values for d typically range as low as 0 and as high as 4, depending on the task. Because the classic SDT model is well established and we know the range of values the parameters may take, we can incorporate this previous knowledge into the analysis by selecting a prior that reflects this knowledge (Rouder & Lu, 2005; Lee, 2008; Lee & Wagenmakers, 2012; DeCarlo, 2012). For instance, in recognition memory, d might have an average of 1, and so we might select a normal prior for d with mean 1 and standard deviation 0.3.

The likelihood function, by contrast, can be more difficult to specify. The likelihood function relates the data to the model parameters by providing an estimate of how "likely" the observed data are to have been generated by different parameter values; this is the distribution of the

data under the model of interest. Specifically, let $\theta = \{\theta_1, \ldots, \theta_K\}$ be the model parameters and $Y = \{Y_1, Y_2, \ldots, Y_n\}$ be the measured variables. After the experiment is conducted, we observe that $Y = y$, so lower-case variables y indicate specific values for the random variables Y. We often make the assumption that the variables $Y = \{Y_1, Y_2, \ldots, Y_n\}$ are independent, and given the parameters θ, follow a probability distribution $f(y \mid \theta)$ provided by the model of interest. Then, given $Y = y$, the likelihood function is

$$L(\theta \mid Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n) = \prod_{i=1}^{n} f(y_i \mid \theta). \tag{1}$$

If the prior distribution of θ is given by $\pi(\theta)$, then using Bayes' rule, the posterior distribution of θ is

$$\pi(\theta \mid Y) = \frac{L(\theta \mid Y)\pi(\theta)}{\int L(\theta \mid Y)\pi(\theta)\, d\theta},$$

where the integral over θ in the denominator is the marginal distribution of Y called the prior predictive distribution. Because the prior predictive distribution is a normalizing constant of the posterior distribution, the posterior $\pi(\theta \mid Y)$ is proportional to the product of the likelihood $L(\theta \mid Y)$ and the prior $\pi(\theta)$:

$$\pi(\theta \mid Y) \propto L(\theta \mid Y)\pi(\theta).$$

Modern Bayesian methods have permitted us to estimate $\pi(\theta \mid Y)$ without having to deal with the intractable normalizing constant (Robert & Casella, 2004; Gelman, Carlin, Stern, & Rubin, 2004; Christensen, Johnson, Branscum, & Hanson, 2011).

Most modern methods for sampling from an unknown posterior distribution use some form of the Markov chain Monte Carlo (MCMC) algorithm. These methods rely on the theory of Markov chains, which describe the movement of a "particle" from one location to another. Under certain conditions and given enough time, the distribution of the possible locations of the particle will converge to a stationary distribution. In Bayesian analysis, the particle is a sample and the stationary distribution is the desired posterior distribution.

The most popular MCMC methods are random walks, which perturb a sample θ by some random amount and then decide whether or not the new value θ^* should be accepted as another sample from the posterior. A popular method for making this decision is called Metropolis–Hastings, which in its simplest form, directly evaluates the posterior probability of the new value relative to posterior probability of the old value. Because the unknown constant of proportionality cancels out,

$$\frac{\pi(\theta^* \mid Y)}{\pi(\theta \mid Y)} = \frac{L(\theta^* \mid Y)\pi(\theta^*)}{L(\theta \mid Y)\pi(\theta)}$$

can be computed exactly. If the probability of θ^* is greater than the probability of θ, as indicated by a probability ratio greater than one, we jump to the new value. If not, we jump to θ^* with probability equal to the probability ratio.

Another random walk method is Gibbs sampling, which forms the basis of the hierarchical modeling approach that we advocate in this paper. We discuss both Metropolis–Hastings and Gibbs sampling in more detail below. Interested readers may consult Robert and Casella (2004), Gelman et al. (2004), or Christensen et al. (2011) for a more thorough treatment of these and other computational Bayesian methods. For now, it is important to recognize that all of these methods require an analytic expression of the likelihood $L(\theta \mid Y)$.

1.3. Approximate Bayesian Computation

Although evaluation of the likelihood $L(\theta \mid Y)$ is essential to Bayesian estimation, for some interesting models it can be difficult or even impossible to mathematically specify. Consider, for example, the many simulation-based models in cognitive neuroscience (e.g., Usher & McClelland, 2001; Jilk, Lebiere, O'Reilly, & Anderson, 2008; O'Reilly & Frank, 2006; Mazurek, Roitman, Ditterich, & Shadlen, 2003). These models are frequently constructed from biologically-based mechanisms and have parameters that represent biological constructs such as long-term potentiation, membrane potential and spiking rates. Such constructs do not easily permit derivation of the probability function that describes the random behavior of the measured behavioral variables—the likelihood.

There are other models that are also difficult to fit to data because their likelihoods are intractable or poorly behaved. One example of such a model is Ratcliff's drift-diffusion model (Ratcliff, 1978; Ratcliff & Smith, 2004). Assuming constant parameters over trials, this model has an analytic likelihood function describing the random behavior of both response times and response accuracy. However, the likelihood is poorly behaved (for some parameter values and some data), and so parameter estimation can sometimes be numerically difficult. In particular, when the model's parameters are free to vary across trials (e.g., Ratcliff & Rouder, 1998), the likelihood must be numerically integrated, compounding issues of instability. An alternative approach is to specify a hierarchical model, where "trial" effects are modeled by separate parameters (Vandekerckhove et al., 2011). While this approach avoids the problems associated with numerical integration, the computational complexity remains high because it requires the estimation of a myriad number of parameters.

Some researchers have resorted to an approximation to least-squares estimation to fit simulation-based and intractable models (e.g., Ratcliff & Starns, 2009; Tsetsos, Usher, & McClelland, 2011; Malmberg, Zeelenberg, & Shiffrin, 2004). In this procedure, they simulate the model to generate data under different combinations of the parameters and then compare this simulated data to the observed data, each set of parameters providing a "distance" between the simulated and observed data. The distance is computed using a discriminant function such as the sum of squared errors. The parameter values that minimize this discriminant function are selected as the "best-fitting" values.

Methods for exploring the parameter space in approximate least-squares may rely on algorithms such as the simplex method (Nelder & Mead, 1965), or they may be little more than trial-and-error fits or fits "by hand." By-hand fits are more qualitative than quantitative, and focus on determining whether or not a model can produce predictions that are similar in pattern to those that were observed. More formal or exhaustive parameter searches are computationally very expensive: A large number of simulated data sets must be generated and compared to the data for each proposed set of parameters to obtain accurate parameter estimates. While such fits can appear to be quantitatively optimal, least-squares approaches do not always find the best parameter estimates (e.g., see Van Zandt, 2000; Rouder et al., 2005; Myung, 2003), and these approaches are not Bayesian.

The approximate Bayesian computation (ABC) technique provides a framework that is similar in concept to the approximate least-squares approach (Pritchard, Seielstad, Perez-Lezaun, & Feldman, 1999; Sisson, Fan, & Tanaka, 2007; Beaumont, Cornuet, Marin, & Robert, 2009; Toni, Welch, Strelkowa, Ipsen, & Stumpf, 2009). ABC originated in population genetics, where it still currently receives the most attention. However, there has been a recent surge of interest in ABC in other related areas such as ecology, epidemiology, and systems biology (see Beaumont, 2010, for a broad overview). Even more recently, ABC has been applied to models in psychology (Turner & Van Zandt, 2012; Turner et al., 2013).

To perform an ABC analysis, we first simulate the model under different combinations of the parameter values. Then, if this simulated data is close to the data that were observed (if the

value of a discriminant function is small), we can conclude that the parameters that generated the data must have some density in the posterior distribution. In this way, we can estimate the full posterior distribution without ever evaluating the likelihood function. Thus, using ABC, we can perform a Bayesian analysis for *any* model that can be simulated.

Despite its widening use, ABC is currently difficult to implement in hierarchical designs. The reason for this difficulty is mostly because ABC algorithms rely heavily on rejection samplers. In a rejection sampler parameter values are sampled from some "proposal" distribution, which may be quite far from the desired posterior distribution, and rejected if the simulated data they produce are too far from the observed data. When the number of parameters is small (a low-dimensional problem), ABC algorithms can be naïvely extended to hierarchical designs by jointly estimating the parameters across the tiers of the hierarchy; subject-level parameters are sampled and rejected at the same time as the group-level parameters. This idea has been implemented in the genetics literature to analyze mutation rate variation across specific gene locations (Excoffer et al., 2005; Pritchard, Seielstad, Perez-Lezaun, & Feldman, 1999). However, as dimensionality increases, as it would, for example, in an experimental design with a large number of subjects, the standard ABC algorithms can be very slow and even impractical because of the overwhelmingly higher rejection rate. This problem has been called the "curse of dimensionality" (Beaumont, 2010).

In the rest of this paper, we extend the powerful ABC approach to complex hierarchical designs, an advance that is crucial to Bayesian analysis of simulation-based models. First, we provide a brief review of ABC and Gibbs sampling. We then introduce the Gibbs ABC algorithm, which is a fast and accurate method for sampling from the posterior distributions of fully hierarchical models. We demonstrate the algorithm's effectiveness on a simple SDT example by comparing the true posteriors of the model to the estimated posteriors obtained by the algorithm applied to data simulated by that model. We then apply the algorithm to fit a hierarchical version of a computational SDT model.

2. Extending ABC to Hierarchical Models: The Gibbs ABC Algorithm

In previous work (Turner & Van Zandt, 2012; Turner et al., 2013; Turner & Sederberg, 2012), we have discussed ABC in some detail and demonstrated how ABC can be used to explore psychological models. For example, Turner and Van Zandt (2012) fit a simple version of the Retrieving Effectively from Memory (REM; Shiffrin & Steyvers, 1997) model and Turner et al. (2013) extended this approach to fit hierarchical versions of REM and the Bind, Cue, Decide Model of Episodic Memory (BCDMEM; Dennis & Humphreys, 2001). Turner and Sederberg (2012) showed in a simulation study that their algorithm could recover the true posterior distribution of a psychologically grounded model of simple response time: the Wald model (Wald, 1947; Matzke & Wagenmakers, 2009).[1] For the purposes of this paper, we will provide only a brief review of ABC, and focus on how Gibbs sampling can be used to extend ABC to hierarchical designs. Our approach uses the fact that the posterior of a set of hyperparameters depends on the data (that is, makes use of the likelihood) only through the lower-level parameters. This means we can employ Gibbs sampling at the level of the hyperparameters and bypass the problem of dimensionality. We will first briefly outline the ABC approach and then Gibbs sampling. Then we will present the Gibbs ABC algorithm.

[1]The Wald distribution describes the behavior of the first-passage time distribution of a single boundary diffusion process.

```
for 1 ≤ i ≤ N do
    Sample θ* from the prior:
        θ* ∼ π(θ)
    Generate data X* using the model:
        X* ∼ Model(θ*)
    Determine jump probability α.
    Generate p* ∼ U(0, 1)
    if p* < α then
        Store θ_i ← θ*
        Store X_i ← X*
    else
        Store θ_i ← θ_{i−1}
        Store X_i ← X_{i−1}
    end if
end for
```

```
for 1 ≤ i ≤ N do
    Sample ξ* from the prior: ξ* ∼ π(ξ)
    for 1 ≤ j ≤ J do
        Sample θ*_j from the prior:
            θ*_j ∼ π(θ | ξ*)
        Generate data X*_j using the model:
            X*_j ∼ Model(θ*_j)
    end for
    Determine jump probability α.
    Generate p* ∼ U(0, 1)
    if p* < α then
        Store θ_{1:J,i} ← θ*_{1:J}
        Store X_{1:J,i} ← X*_{1:J}
        Store ξ_i ← ξ*
    else
        Store θ_{1:J,i} ← θ_{1:J,i−1}
        Store X_{1:J,i} ← X_{1:J,i−1}
        Store ξ_i ← ξ_{i−1}
    end if
end for
```

FIGURE 2.
A basic ABC probabilistic sampling algorithm for estimating the posterior distribution of θ (*left*), and the same algorithm expanded for a hierarchical model (*right*). $U(0, 1)$ is the continuous uniform distribution over the interval $(0, 1)$.

2.1. The Approximate Bayesian Computational Approach

The concept behind ABC is that, if some candidate parameter θ^* can produce simulated data X^* that are close to the observed data Y, then there must be some nonzero probability that θ^* generated the observed data. This probability translates to some density in the posterior distribution at the location θ^*. The algorithms shown in Figure 2 are probabilistic sampling algorithms, in that the candidate θ^* is accepted as a sample from the desired posterior with a probability that depends on the distance between X^* and Y.

Consider the algorithm shown the left panel of Figure 2. Using the SDT model as an example, we define the vector-valued parameter $\theta = \{d, b\}$. Suppose we have 100 discrimination decisions (50 signal trials and 50 noise trials) from an observer who said "yes" on 40 of the signal trials (a "hit rate" of 0.80) and 20 of the noise trials (a "false-alarm rate" of 0.40). This defines our observed data set $Y = \{\text{number of false alarms, number of hits}\} = \{20, 40\}$.

To obtain N samples from the joint posterior distribution of θ, we generate candidate values for d and b by sampling random values from their prior distributions. For example, if the prior distribution of d is normal with mean 1 and standard deviation 1 and the prior distribution of b is normal with mean 0 and standard deviation 1, we might obtain the values $\theta^* = \{0.88, -0.03\}$. We then use these values to simulate a data set X^*.

To simulate a data set, we note that, in the equal-variance SDT model, the parameters $\theta^* = \{0.88, -0.03\}$ imply that the observer says "yes" to any stimulus with perceived intensity greater than $c = 0.88/2 - 0.03 = 0.41$. The proportion of stimuli in the "noise" representation (which is normally distributed with mean 0 and standard deviation 1) giving rise to a "yes" response is therefore $1 - \Phi(0.41) = 0.34$, where Φ is the cumulative distribution function (CDF) for the

standard normal distribution. Similarly, the proportion of stimuli in the "signal" representation (which is normally distributed with mean 0.88 and standard deviation 1) giving rise to a "yes" response is therefore $1 - \Phi(0.41 - 0.88) = 0.68$. We can then generate a sample from a binomial distribution with number of trials equal to 50 and probability of success equal to 0.34 for the simulated false alarms, and another sample from a binomial distribution with number of trials equal to 50 and probability of success equal to 0.68 for the simulated hits.

Using the binomial probabilities 0.34 and 0.68, we might obtain the simulated data set $X^* = \{19, 30\}$. We must now determine whether θ^* has any reasonable chance of having been drawn from the desired posterior $\pi(\theta \mid Y)$. To do that, we must decide how close X^* is to Y. If X^* is very close to Y, we will have a high probability of accepting θ^*, and this probability will decrease as the distance between X^* and Y increases.

2.1.1. Defining Distance Between X^* and Y The success of ABC algorithms hinges on how we measure the distance $\rho(X, Y)$ between two data sets X and Y. For our running SDT example, $Y = \{\text{number of false alarms, number of hits}\} = \{20, 40\}$. For $\theta^* = \{0.88, -0.03\}$ we generated $X^* = \{19, 30\}$. One likely distance function might be the Euclidean metric

$$\rho\big(X^*, Y\big) = \big(X_1^* - Y_1\big)^2 + \big(X_2^* - Y_2\big)^2 = 101.$$

If $\rho(X, Y)$ is chosen well, then ABC algorithms allow us to obtain an approximation to the posterior distribution $\pi(\theta \mid Y)$ that is conditioned on the values of the discriminant function $\rho(X, Y)$ rather than the data Y alone. That is,

$$\pi(\theta \mid Y) \approx \pi\big(\theta \mid \rho(X, Y) \leq \epsilon\big). \tag{2}$$

This approximation is exact for certain algorithms under the appropriate choice of $\rho(X, Y)$. Specifically, $\rho(X, Y)$ should be a function of *sufficient* summary statistics $S(X)$ and $S(Y)$. A thorough discussion of this issue is beyond the scope of this paper. Interested readers should consult Turner and Van Zandt (2012) and Wilkinson (2011), and be assured that the algorithm we present here satisfies the conditions for exact posterior estimates.

The quantity ϵ is called a tolerance threshold. For many realistic problems, it is unlikely that we will be able to generate data X^* so that $\rho(X^*, Y) = 0$, especially if Y and X^* are continuous, or if the sample size of Y and X^* is very large. For this reason, some ABC algorithms use a fixed tolerance threshold ϵ such that if $\rho(X^*, Y) \leq \epsilon$ we keep θ^* as a sample of θ from an approximation of the posterior distribution $\pi(\theta \mid Y)$. However, if $\rho(X^*, Y) > \epsilon$ then we discard the proposed θ^*.

These rejection procedures face two problems. First, if ϵ is too small, it will be difficult to generate X^* close enough to Y to accept θ^*, which means that the rejection rate will be very high and computation time will be greatly increased. Second, if ϵ is not small enough, then the approximation $\pi(\theta \mid \rho(X, Y) < \epsilon)$ will not be very good. A number of solutions have been proposed to ameliorate these problems (see, e.g., Turner & Van Zandt, 2012, for a review), but an alternative is to use kernel-based ABC, a method that smoothly weights the "fitness" of θ^* based on the distance between X^* and Y (Turner & Sederberg, 2012; Wilkinson, 2011).

2.1.2. Sampling Values of θ Consider whether or not we should accept a proposal θ^* on step i of the algorithm shown in the left panel of Figure 2. We described earlier the Metropolis–Hastings method, which is one way of making an accept/reject decision in a MCMC algorithm. The method we describe here is very similar to the standard Metropolis–Hastings method, except that it is based on an evaluation of the distance $\rho(X^*, Y)$ instead of the posterior probability of θ^*.

We have in hand the value of θ_{i-1} from step $i-1$, along with the data X_{i-1} that were generated by θ_{i-1}. We define an acceptance probability α for θ^* as

$$\alpha = \min\left(1, \frac{\pi(\theta^*)\psi(\rho(X^*, Y) \mid \delta_{\text{ABC}})q(\theta_{i-1} \mid \theta^*)}{\pi(\theta_{i-1})\psi(\rho(X_{i-1}, Y)) \mid \delta_{\text{ABC}})q(\theta^* \mid \theta_{i-1})}\right),$$

where $\pi(\theta)$ is the prior for θ, $q(\theta^* \mid \theta)$ is the probability density function (PDF) of a "proposal distribution" from which θ^* is obtained, and $\psi(\rho(X, Y) \mid \delta_{\text{ABC}})$ is a kernel function that increases as the distance $\rho(X, Y)$ between X and Y decreases. The parameter δ_{ABC} is a "tuning parameter" that determines how quickly the function ψ increases with decreases in $\rho(X, Y)$.

A kernel function $\psi(x)$ is defined as a symmetric, nonnegative function that integrates to one. That is,

$$\begin{aligned} \psi(x) &> 0 \quad \text{for all } x \in (-\infty, \infty), \\ \psi(-x) &= \psi(x), \quad \text{and} \\ \int_{-\infty}^{\infty} \psi(x)\, dx &= 1. \end{aligned}$$

Kernel functions provide a way to weigh a variable (x) in terms of its distance from some central point. A common choice for $\psi(x)$ is $\phi(x)$, the PDF of the standard normal distribution, which we use in this paper. If the distance $\rho(X, Y)$ is a metric, then $\rho(X, Y) \geq 0$ for all X and Y, and $\rho(X, Y) = 0$ if and only if $X = Y$. Here, $\psi(\rho(X, Y) \mid \delta_{\text{ABC}}) = \phi(\rho(X, Y)/\delta_{\text{ABC}})$.

To understand why the acceptance probability α is computed in this way, consider first the special case outlined in Figure 2 where the PDF of the proposal distribution $q(\theta^* \mid \theta)$ is equal to the PDF of the prior distribution $\pi(\theta^*)$. In this case, the functions π and q cancel, leaving

$$\begin{aligned} \alpha &= \min\left(1, \frac{\psi(\rho(X^*, Y) \mid \delta_{\text{ABC}})}{\psi(\rho(X_{i-1}, Y) \mid \delta_{\text{ABC}})}\right) \\ &= \min\left(1, \frac{\phi(\rho(X^*, Y)/\delta_{\text{ABC}})}{\phi(\rho(X_{i-1}, Y)/\delta_{\text{ABC}})}\right). \end{aligned} \tag{3}$$

If $\rho(X^*, Y)$ is less than $\rho(X_{i-1}, Y)$, if θ^* produced a data set X^* that was closer to Y than the previously simulated data set X_{i-1}, then we want to keep that value of θ^*. Indeed, if $\rho(X^*, Y) < \rho(X_{i-1}, Y)$, then $\phi(\rho(X^*, Y)/\delta_{\text{ABC}}) > \phi(\rho(X_{i-1}, Y)/\delta_{\text{ABC}})$ or

$$\alpha = \min\left(1, \frac{\phi(\rho(X^*, Y)/\delta_{\text{ABC}})}{\phi(\rho(X_{i-1}, Y)/\delta_{\text{ABC}})}\right) = 1,$$

and θ^* is retained with probability 1 as a new sample from the posterior.

If $\rho(X^*, Y) > \rho(X_{i-1}, Y)$, if θ^* produced a data set X^* that was not as close to Y as the data set X_{i-1}, there is still a possibility that θ^* will be accepted as a sample from the desired posterior. Now, because $\phi(\rho(X^*, Y)/\delta_{\text{ABC}}) < \phi(\rho(X_{i-1}, Y)/\delta_{\text{ABC}})$,

$$\alpha = \min\left(1, \frac{\phi(\rho(X^*, Y)/\delta_{\text{ABC}})}{\phi(\rho(X_{i-1}, Y)/\delta_{\text{ABC}})}\right) = \frac{\phi(\rho(X^*, Y)/\delta_{\text{ABC}})}{\phi(\rho(X_{i-1}, Y)/\delta_{\text{ABC}})}.$$

As $\rho(X^*, Y)$ increases, α decreases, and it becomes less likely that we will retain θ^*. The value of the tuning parameter δ_{ABC} will determine how sensitive the sampling algorithm is to these variations in distance.

Assume for our running example that $\rho(X_{i-1}, Y) = 12$. Setting the tuning parameter $\delta_{\text{ABC}} = 50$, we obtain

$$\alpha = \min\left(1, \frac{\phi(101/50)}{\phi(12/50)}\right) \approx 0.13.$$

We now sample a p^* between 0 and 1 from a continuous uniform distribution ($p^* \sim U(0, 1)$), and if that p^* is less than 0.13, then we accept $\theta^* = \{0.88, -0.03\}$ as a sample from the desired posterior. We would set $\theta_i = \theta^*$, $X_i = X^*$, and generate a new proposal θ^* for the $i + 1$th step.

The dependence of the rejection rate (the number of θ^*s that are proposed and then rejected) on the tuning parameter δ_{ABC} is evident when we recompute α with a tuning parameter $\delta_{\text{ABC}} = 5$. In this case,

$$\alpha = \min\left(1, \frac{\phi(101/5)}{\phi(12/5)}\right) \approx 0.$$

It is now virtually impossible (probability less than 10^{-89}) that we would sample a p^* from a $U(0, 1)$ distribution that is less than α, and we would certainly reject $\theta^* = \{0.88, -0.03\}$ as a sample from the desired posterior. Instead, we would set $\theta_i = \theta_{i-1}$ and generate a new proposal θ^* for the $i + 1$th step. The ABC implementations in this paper used the normal kernel $\phi(x)$ with tuning parameter $\delta_{\text{ABC}} = 0.01$. The parameter δ_{ABC} was selected based on preliminary simulation studies where our goal was to select a δ_{ABC} that achieved accurate posterior estimates while maintaining reasonable acceptance rates (i.e., acceptance rates greater than 10 %).

In the right panel of Figure 2, proposals θ^* are chosen by sampling from the prior $\pi(\theta)$, which is a common practice in the simplest ABC algorithms. While these simple algorithms are appealing, many models are either too complex or the prior for θ is too diffuse, resulting in large rejection rates to obtain values of $\rho(X, Y)$ that are small enough. In these situations, we can sample instead from a proposal distribution $q(\theta^* \mid \theta)$, which is chosen so that q is in some way "close" to the desired posterior (also see Fearnhead & Prangle, 2012, for more principled rules). If $q(\theta^* \mid \theta)$ is symmetric, so that $q(\theta^* \mid \theta) = q(\theta \mid \theta^*)$, then the choice of $q(\theta^* \mid \theta)$ has no effect on the acceptance probability and α can be understood as we described above. In this paper, we chose $q(\theta^* \mid \theta)$ to be the normal distribution with mean θ and standard deviation 0.1, which is symmetric.

2.1.3. Gibbs Sampling Having now considered the dual problems of proposing values for θ and evaluating their fitness, we must now consider the problem of scaling up into higher-dimensional parameter spaces. In our running SDT example, $\theta = \{d, b\}$ is two-dimensional, and it is not very difficult to imagine extending the algorithm in the left panel of Figure 2 to three, four, or even more dimensions. The ability to extend the algorithm will be limited when we begin to tackle the problem of hierarchical models, with the subject-level parameters for each subject, plus the hyperparameters in the upper levels of the hierarchy.

Estimating all the posteriors for the subject- and group-level parameters requires that we obtain a large number of n-dimensional samples from an n-dimensional joint probability distribution. Sampling from a joint distribution is more difficult than sampling from the (univariate) distribution of a single variable, but under general conditions, we can use a technique called Gibbs sampling to make this process more tractable.

Consider our SDT experiment with a single subject. We have been writing the subject's parameters as the vector $\theta = \{d, b\}$. Although it might be possible, we do not typically try to sample from the joint distribution of $\{d, b\}$. Instead, we initialize (perhaps by sampling from the priors) the sequence of samples (called the chain) to $\theta_1 = \{d_1, b_1\}$, and then begin a process where we select d_2 by sampling from the conditional distribution of $d \mid b = b_1$, and then select b_2 by sampling from the conditional distribution of $b \mid d = d_2$. This process is called Gibbs sampling.

Gibbs sampling requires that we know the conditional posteriors $\pi(b \mid d, Y)$ and $\pi(d \mid b, Y)$. If we know the priors for d and b and the probability of Y given d and b (i.e., the likelihood), then we can either sample directly from the appropriate (known) posteriors or, if the posteriors do not have an obvious analytic form, we can use a convenient MCMC technique, like Metropolis–Hastings or slice sampling. (These are the methods used by the popular Gibbs sampling program WinBUGS; see Lunn, Thomas, Best, & Spiegelhalter, 2000.)

More generally, for a fully hierarchical problem, we might consider a SDT experiment with J subjects. Each subject's parameters θ_j ($j = 1, \ldots, J$) can be written as the vector $\theta_j = \{d_j, b_j\}$. The model hierarchy assumes that each subject's d_j is sampled from a normal distribution with mean d_μ and standard deviation d_σ. Similarly, each subject's b_j is sampled from a normal distribution with mean b_μ and standard deviation b_σ. The group parameters can be written as the vector $\xi = \{d_\mu, d_\sigma, b_\mu, b_\sigma\}$. We might, for example, specify that the hyperparameter d_μ has a normal prior with mean 1 and standard deviation 1, b_μ has a normal prior with mean 0 and standard deviation 1, and the hyperparameters d_σ and b_σ have gamma priors with shape and scale equal to 1 (i.e., exponential with mean 1).

Our goal is to obtain a large number of samples from the joint distribution of

$$(\xi, \theta) = \{d_\mu, b_\mu, d_\sigma, b_\sigma, d_1, b_1, d_2, b_2, \ldots, d_J, b_J\},$$

which has dimension $KJ + M$, where $M = 4$ is the number of hyperparameters and $K = 2$ is the number of subject-specific parameters.

We use $\theta_{j,k}$ to denote the kth subject-level parameter for Subject j and $\theta_{j,k,i}$ to denote the ith sample of $\theta_{j,k}$ obtained on iteration i. Similarly, $\xi_{m,i}$ is the value of the mth hyperparameter ξ_m on iteration i. We will also write $\xi_{1:M,i}$ and $\theta_{1:J,1:K,i}$ to indicate all the values defining the vectors ξ and θ on the ith iteration. Using Gibbs sampling, we obtain samples from the joint posterior by first initializing the values of all the parameters to

$$\begin{aligned} \xi_{1:M,1} &= \{d_{\mu,1}, b_{\mu,1}, d_{\sigma,1}, b_{\sigma,1}\} \quad \text{and} \\ \theta_{1:J,1:K,1} &= \{d_{1,1}, b_{1,1}, d_{2,1}, b_{2,1}, \ldots, d_{J,1}, b_{J,1}\}. \end{aligned}$$

Then, on each iteration $i > 1$, we obtain samples from the conditional distributions of

$$\begin{aligned} &\xi_m \mid \xi_{-m}, \theta, \quad \text{and} \\ &\theta_j \mid \theta_{-j}, \xi \end{aligned} \tag{4}$$

for $j = 1, 2, \ldots, J$ and $m = 1, 2, \ldots, M$, where ξ_{-m} denotes the set of parameters ξ excluding the mth element, so that

$$\xi_{-m} = \{\xi_1, \ldots, \xi_{m-1}, \xi_{m+1}, \ldots, \xi_M\}.$$

To do this sampling in the Gibbs framework, we update each set of parameters separately by setting all of the other parameters to their current value in the chain. Thus, ξ_m is updated by setting

$$\begin{aligned} \theta &= \theta_{1:J,1:K,1}, \quad \text{and} \\ \xi_{-m} &= \{\xi_{1,2}, \xi_{2,2}, \ldots, \xi_{m-1,2}, \xi_{m+1,1}, \ldots, \xi_{M,1}\} \end{aligned}$$

in Equation (4), and θ_j is updated by setting

$$\begin{aligned} \xi &= \xi_{1:M,2}, \quad \text{and} \\ \theta_{-j} &= \{\theta_{1,1:K,2}, \theta_{2,1:K,2}, \ldots, \theta_{j-1,1:K,2}, \theta_{j+1,1:K,1}, \ldots, \theta_{J,1:K,1}\} \end{aligned}$$

in Equation (4). This is straightforward, if a little tedious, to implement for models with an analytic likelihood function.

If the model under consideration does not have an analytic likelihood function, then we must consider methods to adapt the algorithm in the left panel of Figure 2 to the hierarchical problem. The right panel of Figure 2 shows a naïve solution to this problem in which the hyperparameter vector ξ is updated in a single step together with the parameters θ.

The right panel of Figure 2, which does not use Gibbs sampling, breaks the sampling problem into two steps. First, we sample proposal hyperparameters ξ^* from the prior $\pi(\xi)$ and then we sample proposal parameters θ_j^* from the conditional prior $\pi(\theta_j \mid \xi^*)$. Writing the data $Y = \{Y_1, Y_2, \ldots, Y_J\}$, so that Y_j represents the observations taken on Subject j, each set of parameters $\{\xi^*, \theta_j^*\}$ must generate data X_j^* so that $\rho(X_j^*, Y_j) \leq \epsilon$. If a candidate hyperparameter ξ^* produces θ_j^*s that satisfy the criterion for all $j \in \{1, 2, \ldots, J\}$, then ξ^* and the θ_j^*s have some nonzero density in the approximate joint posterior distribution $\pi(\xi, \theta \mid \rho(X, Y) \leq \epsilon)$.[2] If it is not possible to find a θ_j^* that produces X_j^* close to Y_j, even if all the other $\theta_{l\neq j}^*$ produced X_l^*s close to their Y_ls, then the proposed ξ^* and all the proposed θ_j^*s must be discarded and the search for a sample of θ begins again with a new ξ^*. Therefore, this algorithm, while producing accurate estimates of the posterior is like the algorithm in the left panel of Figure 2, hopelessly inefficient for even moderately complex problems.

We could do several things to make the algorithm more efficient. First, we could use an empirical Bayes method. Empirical Bayes methods inform the choice of the priors by first using classical estimation techniques such as maximum likelihood. For example, given the maximum likelihood estimate $\hat{\xi}$ for ξ, we could generate the θ_j^*s from the conditional prior $\pi(\theta_j \mid \hat{\xi})$ (Line 5 of the right panel of Figure 2; see Pritchard et al., 1999, for an example). Second, we could allow the simulated data X_j^* to be arranged in any way possible to optimize the acceptance rate. That is, we might not want to restrict our comparison of X_j^* to data Y_j; perhaps data Y_l are closer to X_j^* and we could accept θ_j^* on that basis (see, e.g., Hickerson, Stahl, & Lessios, 2006; Hickerson & Meyer, 2008; Sousa, Fritz, Beaumont, & Chikhi, 2009). Finally, Bazin, Dawson, and Beaumont (2010) proposed a two-stage technique that can improve the naïve sampler. However, this method introduces additional error above and beyond the error encountered when $\rho(X, Y) \neq 0$ (Beaumont, 2010).

Gibbs sampling, however, provides a way to avoid the pitfalls of the naïve sampler entirely, and so we now present an alternative method that permits sampling from the posteriors of a fully hierarchical model with much greater computational efficiency.

2.2. *The Gibbs ABC Algorithm*

In this section, we show how we can sample directly from the conditional posterior distribution of the hyperparameters using well-accepted techniques. The key insight to this approach is the fact that the conditional distribution of the hyperparameters does not depend on the likelihood function. This, in combination with a mixture of Gibbs sampling and ABC sampling provides an algorithm that offers a significant improvement in accuracy and computation efficiency.

To implement the algorithm, we first consider the conditional posterior distribution of the subject-level parameters θ, which is

$$\begin{aligned}\pi(\theta \mid Y, \xi) &\propto L(\theta \mid Y, \xi)\pi(\theta \mid \xi) \\ &\propto \prod_{j=1}^{J} L(\theta_j \mid Y_j)\pi(\theta_j \mid \xi),\end{aligned}$$

[2]One can also use a kernel to weigh the fitness of the proposals ξ^* and θ_j^*.

given the conditional independence of the θ_js and Y_js. To see that the parameter ξ can be dropped from the likelihood, recall that the likelihood for Subject j can be seen as the probability of the data Y_j given the parameters θ_j; there is no role played by ξ in the PDF of Y_j, and so the likelihood Y_j depends on ξ only through the parameters θ_j. The conditional posterior distribution of θ_j given the data and all of the other parameters is therefore

$$\pi(\theta_j \mid Y, \xi) \propto L(\theta_j \mid Y_j)\pi(\theta_j \mid \xi). \tag{5}$$

Because the conditional posterior distribution of each of the θ_js depends on the partition of the data exclusive to the jth subject, the problem simplifies to performing ABC for each subject, and we can approximate each conditional posterior by

$$\pi(\theta_j \mid Y, \xi) \approx \psi\big(\rho(X_j, Y)|\delta_{\text{ABC}}\big)\pi(\theta_j \mid \xi). \tag{6}$$

Noting that $\pi(\xi \mid Y, \theta) \propto \pi(\theta \mid \xi)\pi(\xi)$, the joint conditional posterior distribution of the hyperparameters ξ is

$$\begin{aligned}\pi(\xi \mid Y, \theta) &\propto L(\theta \mid Y)\pi(\theta \mid \xi)\pi(\xi)\\ &\propto \pi(\theta \mid \xi)\pi(\xi)\\ &\propto \pi(\xi)\prod_{j=1}^{J}\pi(\theta_j \mid \xi).\end{aligned} \tag{7}$$

Because ξ influences the likelihood only through the parameter θ, the joint conditional distribution of $\xi = \{\xi_1, \ldots, \xi_m, \ldots, \xi_M\}$ does not depend on the likelihood; the likelihood is just a constant with respect to ξ. This means that we can sample from the conditional posterior distribution of ξ using standard techniques. If this distribution has a convenient form, we can directly sample from it. Otherwise, we can use any numerical technique, such as discretized sampling (e.g., Gelman et al., 2004), adaptive rejection sampling (Gilks & Wild, 1992), or MCMC (e.g., Robert & Casella, 2004).

This brings us to the Gibbs ABC algorithm shown in Figure 3, which is a mixture of standard and ABC estimation techniques. After initializing values for $\xi_{1:M,1}$ and $\theta_{1:J,1:K,1}$, on each iteration $i > 2$, we first draw samples of $\xi_{m,i}$ conditioned on all other parameters in the model, including all other values in the vector ξ. We use a Gibbs sampler to obtain values of $\xi_{1:M,i}$ by sampling directly from $\pi(\xi_m \mid Y, \theta_{1:J,1:K,i-1}, \xi_{-m,i})$ given by Equation (7).

Having obtained the values for ξ on iteration i, we then use those values to generate samples from the joint conditional posterior distribution of θ using ABC. If the distribution $q(\theta)$ from which the proposed values $\theta^*_{j,k}$ are drawn is equal to the prior distribution $\pi(\theta_{j,k} \mid \theta_{j,-k,i}, \xi_{1:M,i})$ then the jumping probability α is calculated in a similar manner as in Equation (3).

The Gibbs ABC algorithm is considerably more flexible than other hierarchical ABC algorithms. We can use any appropriate sampling method to estimate the posterior distribution of ξ and, if necessary, we could use different discriminant functions $\rho(X^*, Y)$ and tuning parameters δ_{ABC} for each subject. This might be useful when the model is misspecified, and so allowing for large distances for some subjects could improve convergence speed.

Consider, for example, a model that predicts a positive relationship between variables but the jth subject shows a negative relationship. In this situation, there will be no values for θ^*_j that could simulate data X^*_j close to the observed data Y_j. Only by increasing δ_{ABC} will $\rho(X^*_j, Y_j)$ be given a weight high enough that θ^*_j has a reasonable chance of being accepted.[3]

[3]Note that the posterior distributions of θ and ξ will still exist despite a misspecified model. The goal is to estimate the shapes of those posteriors by generating data that is close to the observed data as measured by $\rho(X_j, Y_j)$.

Initialize $\xi_{m,1}$ and each $\theta_{j,k,1}$.
for $2 \leq i \leq N$ **do**
  for $1 \leq m \leq M$ **do**
    Sample $\xi_{m,i}$ from the conditional posterior:
      $\xi_{m,i} \sim \pi(\xi_m \mid \theta_{1:J,1:K,i-1}, \xi_{-m,i})$
  end for
  for $1 \leq j \leq J$ **do**
    for $1 \leq k \leq K$ **do**
      Sample a value $\theta^*_{j,k}$ from a proposal distribution: $\theta^*_{j,k} \sim q(\theta)$
      Generate data $X^*_{j,k}$ using the model: $X^*_{j,k} \sim \text{Model}(\theta_{j,-k,i}, \theta^*_{j,k})$
      Determine jump probability α and sample $p^* \sim U(0,1)$.
      if $p^* < \alpha$ **then**
        Store $\theta_{j,k,i} \leftarrow \theta^*_{j,k}$
        Store $X_{j,k,i} \leftarrow X^*_{j,k}$
      else
        Store $\theta_{j,k,i} \leftarrow \theta_{j,k,i-1}$
        Store $X_{j,k,i} \leftarrow X_{j,k,i-1}$
      end if
    end for
  end for
end for

FIGURE 3.
The Gibbs ABC algorithm to estimate the posterior distributions for ξ and θ.

The Gibbs ABC algorithm also permits blocked sampling of parameters. Although the pseudocode in Figure 3 is written so that each of the M ξ parameters and JK θ parameters is sampled sequentially, this may lead to poor convergence. For example, parameters that are highly correlated should be blocked and sampled together from their joint posterior. In addition, the model structure itself may suggest that certain blocking strategies will lead to faster convergence. For example, consider the nondecision component of many models of response time. The contribution of this component is represented by a parameter t_0 which is assumed to reflect processes that are not of immediate interest (e.g., Ratcliff, 1978; Usher & McClelland, 2001; Brown & Heathcote, 2005, 2008). This parameter simply shifts the response times and otherwise does not affect the distribution. If simulating the model is time-consuming, it will be much more efficient on iteration i to perform ABC and obtain samples of all of the parameters using $t_{0,i-1}$. Conditioning on these obtained samples, we can generate a proposal t_0^* and then, without having to simulate X again, we can compute $\rho(X', Y)$ for $X' = X + t_0^*$ to evaluate the proposal t_0^*.

Finally, the hyperdistributions could also be poorly-behaved or undefined. In this case, the ABC algorithm could be extended to the hyperparameters by replacing the Gibbs sampling step in the algorithm with another ABC step.

3. An Illustrative Example: Fitting a Hierarchical SDT Model Using Gibbs ABC

The purpose of this section is to demonstrate how the Gibbs ABC algorithm can be used to fit a simple hierarchical model to data. We will continue by expanding on the SDT model,

adding hyperparameters that describe the distributions from which the subject-level SDT parameters were sampled. Because this model has an analytic likelihood function, we can contrast the estimated posteriors obtained using Gibbs ABC with those obtained using standard MCMC methods. We will show that the posteriors estimated in these two ways are very similar, and so argue that Gibbs ABC can be very accurate.

3.1. The Model

Each subject's probability of a "yes" response is determined by a discriminability parameter d_j and a bias parameter b_j. The discriminability parameters d_j follow a normal distribution with mean d_μ and standard deviation d_σ, and the bias parameters b_j follow a normal distribution with mean b_μ and standard deviation b_σ. The mean hyperparameter d_μ has a normal prior with mean 1 and standard deviation 1, and the hyperparameter b_μ has a normal prior with mean 0 and standard deviation 1. The standard deviation hyperparameters d_σ and b_σ have gamma priors with shape and scale equal to 1.[4]

We simulated the model by first setting $d_\mu = 1$, $b_\mu = 0$, $d_\sigma = 0.20$, and $b_\sigma = 0.05$. Using these parameter values, we drew a d_j and b_j for each of nine subjects from the normal hyperdistributions. We then used these subject-level parameters to generate "yes" responses for noise and signal trials by sampling from binomial distributions with probabilities equal to the areas under the normal curves to the right of $d_j/2 + b_j$ and N equal to 500.

3.2. Results

To apply the Gibbs ABC algorithm to the "observed" (simulated) data, we set $\rho(X, Y)$ equal to the Euclidean distance between the observed and simulated (using proposed values for θ) vectors of hit and false alarm rates. This distance was weighed with a Gaussian kernel using a tuning parameter $\delta_{\text{ABC}} = 0.01$. We generated 24 independent chains of 10,000 draws of each parameter, discarding the first 1,000 iterations chain as a "burn-in" period. We did not thin the chains, and so we obtained 216,000 samples to form an estimate of the joint posterior distributions for each parameter.

Figure 4 shows the estimated posterior distributions for the model's hyperparameters (d_μ, b_μ, d_σ, and b_σ) as histograms plotted behind solid lines. These lines are the posterior density estimates obtained using a likelihood-informed method (MCMC), and the vertical lines represent the true values of the hyperparameters. The left panel of Figure 4 shows these estimates for the hyper mean parameters for the bias parameter b_μ (top) and the discriminability parameter d_μ (bottom). The right panel shows the estimates for the hyper standard deviation parameters for the bias parameter b_σ (top) and the discriminability parameter d_σ (bottom) on the log scale. The estimates obtained using our Gibbs ABC algorithm closely match the estimates obtained using conventional MCMC.

While Figure 4 shows that the estimates using Gibbs ABC and likelihood-informed MCMC methods match closely at the group level, it is also important to show that the Gibbs ABC algorithm provides estimates that closely match standard MCMC methods at the subject level. Figures 5 and 6 show the estimated posterior distributions for the bias (b_j) and discriminability (d_j) parameters, respectively, for each of the nine subjects. The vertical dashed lines in each panel show the values used to generate the data. Together, the figures show that the Gibbs ABC algorithm also provides accurate estimates at the subject level.

[4]We investigated a range of priors and determined that the choice of priors, if reasonably variable, had little effect on the final estimated posterior. The priors that we selected permit a range of values for d and b that reflect those that are reported in the perceptual and memory literature (Rouder & Lu, 2005; Lee, 2008).

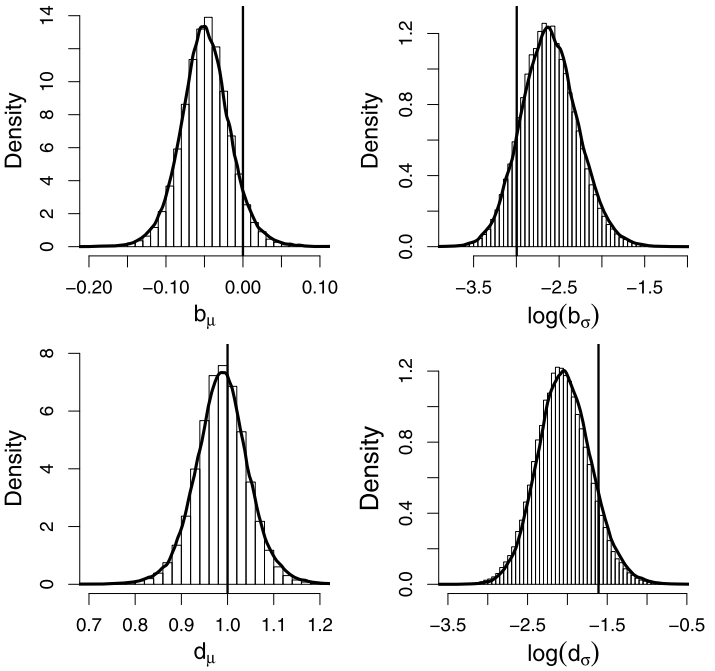

FIGURE 4.
The estimated posterior distributions obtained using likelihood-informed methods (*black densities*) and the Gibbs ABC algorithm (*histograms*) for the hyperparameters of the classic SDT model. The true values used to generate the data are shown as the *vertical lines*. The *rows* correspond to group-level parameters for the bias parameter b (*top*) and the discriminability parameter d (*bottom*). The *columns* correspond to the hyper means (*left*) and the hyper standard deviations on the log scale (*right*).

These results demonstrate that the Gibbs ABC algorithm can recover the true posterior distributions of the hierarchical SDT model accurately and efficiently. In the next section, we demonstrate the utility of the algorithm on the Dynamic, Stimulus-Driven (DSD) model of signal detection (Turner et al., 2011) by fitting it to data presented in Van Zandt and Jones (2011) and also Turner et al.

4. The Dynamic, Stimulus-Driven Model of Signal Detection

The classic SDT model assumes fixed stimulus representations and response criteria, and so is unable to account for any change in discrimination performance over time. Such changes include between-trial effects such as sequential dependencies, changes in discriminability with experience, or the detection of a change (and subsequent adaptation to that change) in the stimulus stream. Since SDT's inception, there have been many modifications to the basic SDT framework to explain these changes (e.g., Erev, 1998; Kubovy & Healy, 1977; Mueller & Weidemann, 2008; Treisman & Williams, 1984; Brown & Steyvers, 2005; Benjamin et al., 2009), but most maintain that the performance differences from trial to trial are a function of the criterion, and not changes in the stimulus representation. As a result, these approaches are incapable of explaining how an observer might establish stimulus representations for a novel task, or how an observer might adapt these representations in response to changes in the stimulus stream.

Instead of explaining the trial-by-trial differences in the decision rule as changes in the criterion, Turner et al. (2011) proposed a dynamic version of SDT in which the stimulus representations are altered after the presentation of each stimulus. The representations are maintained by

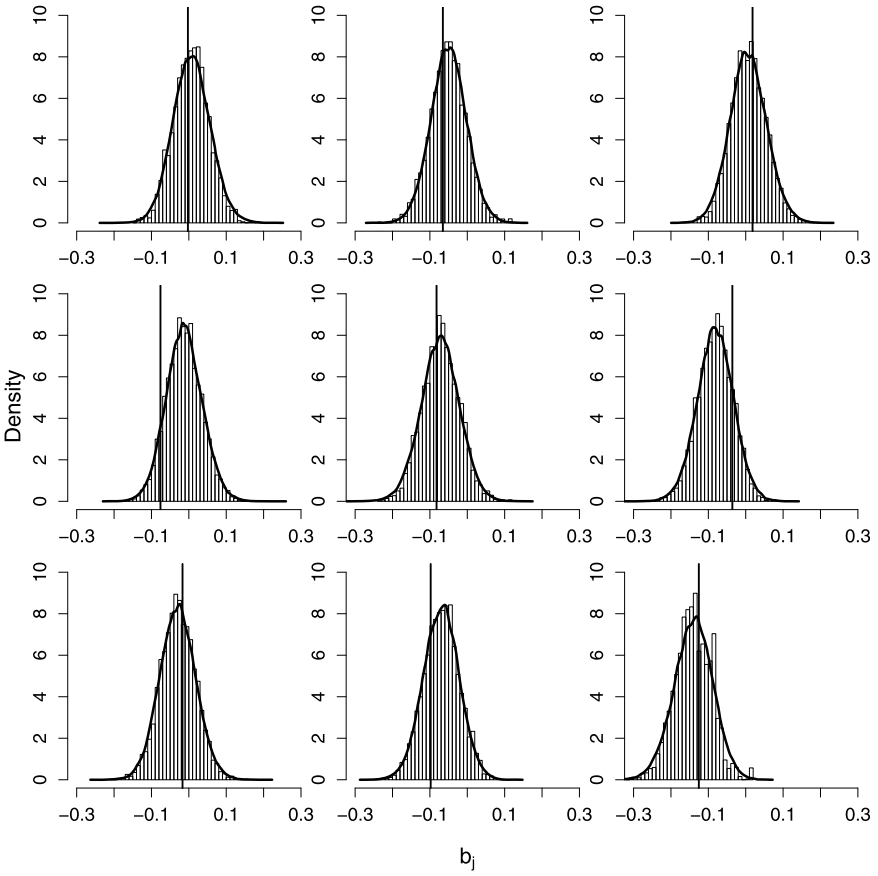

FIGURE 5.
The estimated posterior distributions obtained using likelihood-informed methods (*black densities*) and the Gibbs ABC algorithm (*histograms*) for the subject-level bias parameters b_j of the classic SDT model. The true values used to generate the data are shown as the *vertical lines*. Each panel represents a different subject.

a number of points located along a perceptual axis. Each representation point carries information about the likelihoods of both signal and noise at that location. At first, the model uses only a few representation points (e.g., two). Then, with probability γ, new representation points are placed following the stimulus and corresponding feedback. This process continues until η representation points are contained in the system, at which time new representation points replace the oldest representation points in the system.

The model assumes that if a stimulus was a signal, then the signal representation is updated at all of the representation points within a bandwidth δ of the presented stimuli. All other representation points outside of the bandwidth decay (i.e., their likelihoods are decreased) a small amount. A learning rate parameter λ determines how quickly representations change to match the statistical properties of the stimulus stream. The parameter λ can range from zero to one, with larger values resulting in more dynamic stimulus representations whereas smaller values result in more stable representations.

As in classic SDT, a response is determined by the likelihoods that a presented stimulus is a signal or noise. A "yes" response is made if the representation point has a higher signal likelihood estimate, and a "no" response is made if the representation point has a higher noise likelihood

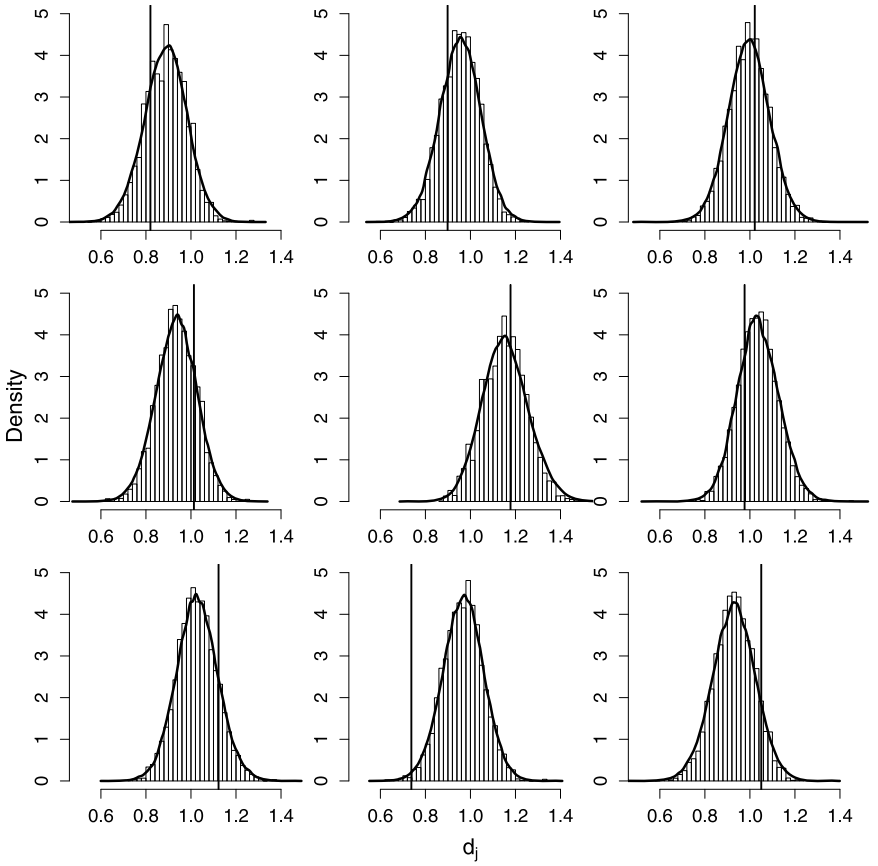

FIGURE 6.
The estimated posterior distributions obtained using likelihood-informed methods (*black densities*) and the Gibbs ABC algorithm (*histograms*) for the subject-level discriminability parameters d_j of the classic SDT model. The true values used to generate the data are shown as the *vertical lines*. Each panel represents a different subject.

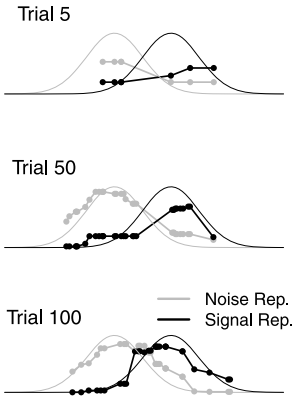

FIGURE 7.
An example of how the DSD model evolves the representations (*dotted lines*) for both signal (*black*) and noise (*gray*) to match the true stimulus-generating distributions (*solid lines*). The *top*, *middle*, and *bottom panels* show the DSDT model's representations after 5, 50, and 100 trials.

estimate, or if the two likelihoods are equal, a guess is made by choosing either the "yes" or "no" response with equal probabilities.

Figure 7 shows how the representations produced by the DSD model change over time. The dotted lines represent the noise (gray) and signal (black) representations used by the model, and each dot corresponds to a representation point. At first, the model uses only a few points (top panel), but after the model is presented with more stimuli, the representations begin to look more like the true stimulus-generating distributions (solid lines). Finally, after many stimulus presentations, the representations closely resemble the true stimulus generating distribution, as shown in the bottom panel.

Turner et al. (2011) showed how the dynamic updating process was a useful extension of the basic SDT framework. However, the dynamic representations, which change from trial to trial, makes generating model predictions difficult. To compute the probability of a "yes" response on Trial t, one would first need to know the probabilities of the different possible representations on Trial t, which would depend on the stimuli presented on Trials 1 to $t-1$, as well as the responses to them. The derivation of these probabilities is a very difficult problem, and to avoid it, Turner et al. resorted to hand-held fits obtained by approximate least-squares.

Hand-held fitting procedures severely limit the extent to which inference can be made about a model's parameters. In particular, one cannot assess how one subject differs from another and how that subject's performance might differ from the group of subjects in the experiment. The HABC approach allows for full hierarchical Bayesian inference despite the lack of explicit expressions for a model's likelihood. We now use Gibbs ABC to fit the DSD model to the data from Turner et al. (2011).

4.1. The Model

We used the data from the low d' condition of Experiment 1 reported in Turner et al. (2011). In this experiment, subjects were presented with 340 patient blood assays and asked to determine whether the patient had been infected with a deadly disease or not. If a subject indicated that the patient had been infected (by means of a "yes" response) then that patient would receive treatment for the disease. However, subjects were told that healthy (i.e., uninfected) patients who received treatment would die as a consequence of the treatment. By contrast, if a subject indicated that a patient did not have the disease (by means of a "no" response) then that patient would not receive treatment. If a sick patient (i.e., an infected patient) was not treated, that patient would die as a consequence of the disease.

The blood assays were presented in the form of numbers randomly drawn from Gaussian distributions with means of 40 and 60 for healthy and sick patients, respectively, and with common standard deviations of 10. The 340 blood assays were completed over 5 blocks of 68 trials each, for 31 subjects. Additional experimental details can be found in Turner et al. (2011) and Van Zandt and Jones (2011).

Our goal is to make inferences about each of the model parameters for each subject individually while simultaneously making inferences about the group-level parameters. The parameters of interest are: γ, the probability of adding/replacing a representation point; λ, the learning rate; δ, the bandwidth; and η, the maximum number of representation points. The jth subject's parameters are γ_j, λ_j, δ_j, and η_j. We specified a hyperdistribution from which each of these subject-level parameters were drawn, and the hyperparameters (e.g., the mean and variance) of

each hyperdistribution formed the basis of our group-level analysis. For this model, we set

$$\begin{aligned}
\gamma_j &\sim \mathcal{TN}(\gamma_\mu, \gamma_\sigma, 0, 1),\\
\lambda_j &\sim \mathcal{TN}(\lambda_\mu, \lambda_\sigma, 0, 1),\\
\delta_j &\sim \mathcal{TN}(\delta_\mu, \delta_\sigma, 0, \infty), \quad \text{and}\\
\eta_j &\sim \mathcal{TN}(\eta_\mu, \eta_\sigma, 2, \infty),
\end{aligned}$$

where $\mathcal{TN}(s, t, u, v)$ denotes a truncated normal distribution with mean parameter s, standard deviation parameter t, lower bound u and upper bound v. The truncated normal distribution is a convenient choice for defining boundaries for the space of each subject parameter. For example, δ cannot go below zero, and we suspected that the variability between parameter values for each subject would be approximately normally distributed. For η, we chose a lower bound of two to force the model to maintain at least two representation points for each subject.

To complete the hierarchical Bayesian model, we specified mildly informative priors for each of the hypermeans, such that

$$\begin{aligned}
\gamma_\mu &\sim \text{Beta}(1, 1),\\
\lambda_\mu &\sim \text{Beta}(1, 1),\\
\delta_\mu &\sim \mathcal{TN}(10, 5, 0, \infty), \quad \text{and}\\
\eta_\mu &\sim \mathcal{TN}(20, 10, 2, \infty),
\end{aligned}$$

and hyper standard deviations, such that

$$\begin{aligned}
\gamma_\sigma &\sim \Gamma(1, 1),\\
\lambda_\sigma &\sim \Gamma(1, 1),\\
\delta_\sigma &\sim \Gamma(5, 1), \quad \text{and}\\
\eta_\sigma &\sim \Gamma(5, 1).
\end{aligned}$$

Because we have never fit the DSD model in a Bayesian framework, we had little guidance in selecting the priors. As such, we specified priors to be consistent with the parameter estimates obtained in Turner et al. (2011), but we maintained a great deal of uncertainty to reflect our inexperience with the model's parameters.

4.2. Results

To implement the Gibbs ABC algorithm, we used the Euclidean distance between the observed and simulated hit and false alarm rates for each subject, weighed by a Gaussian kernel with a standard deviation of $\delta_{\text{ABC}} = 0.01$ to assess the fitness of each proposal. We ran 24 independent chains for 4,000 iterations, and discarded the first 1,000 iterations. This gave 72,000 samples to form an estimate of the joint posterior distribution of each parameter.

Figure 8 shows the estimated posterior distributions for each of the hyperparameters of the model. The top row shows the hypermean posterior estimates whereas the bottom row shows the hyper standard deviation posterior estimates on the log scale. The vertical lines in the top row represent the values obtained by Turner et al. (2011) to fit the data. The posterior estimates show that the parameter estimates obtained by Turner et al. were reasonable, in the sense that they are, for the most part, contained within a highest-density interval of the estimated posterior distributions. However, the likelihood-free Bayesian approach provides substantially more information in the form of parameter estimates at both the group and subject levels.

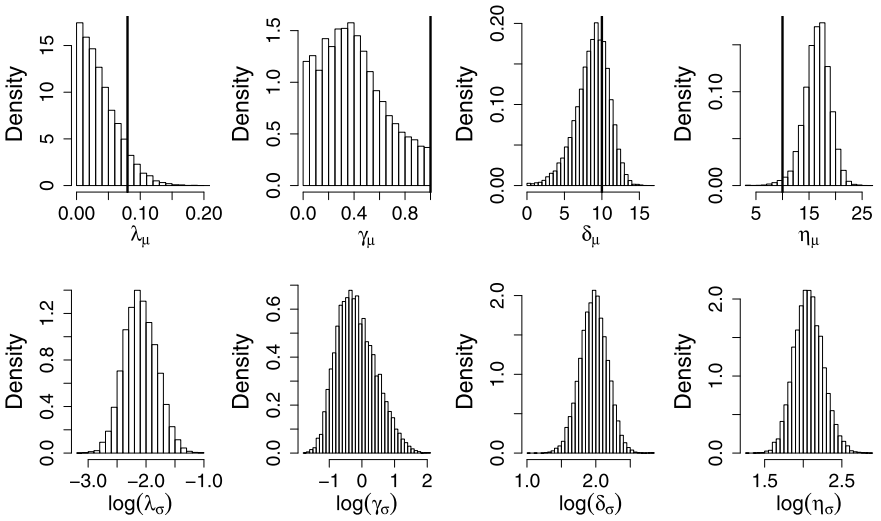

FIGURE 8.
Estimated posterior distributions for each of the hyperparameters in the DSD model. The *top row* corresponds to the hyper mean parameters whereas the *bottom row* corresponds to the hyper standard deviation parameters (on the log scale). The *left column* corresponds to the recency parameter λ, the *middle left column* corresponds to the probability of representation point replacement γ, the *middle right column* corresponds to the bandwidth parameter δ, and the *right column* corresponds to the number of representation points η. The *vertical lines in the top row* represent the values used by Turner et al. (2011) used to fit the model to these data, collapsed across subjects.

The posterior estimates of the parameters of the DSD model shown in Figure 8 give insight into how stimulus representations are established and maintained. For example, the estimate of λ_μ, the mean learning rate, is concentrated on smaller values (e.g., 0.0–0.10), suggesting that the subjects in this experiment tended to rely on representations formed earlier in their experiences with the stimulus set. This is sensible, because the statistical properties of the stimuli were fixed over the course of the experiment and the representations did not need to adapt to changes in the stimulus stream.

The posterior estimate for the node replacement probability γ_μ has a mode of approximately 0.4, which is much smaller than the estimate of 1.0 reported in Turner et al. (2011). A small value for γ_μ, in combination with a small value for λ_μ, suggests that the representations used by the subjects in this experiment were stable and did not vary appreciably from trial to trial.

Finally, the posterior estimate of the mean number of representation points η_μ is centered between 15 and 20, values that are greater than the parameter estimate of 10 reported by Turner et al. (2011). The stimuli themselves were drawn from distributions that ranged from 20 to 80, and so, dividing this range by the mean of η_μ, we can estimate that subjects placed representation points approximately 3–4 units apart along the decision axis. Thus, the subjects in this experiment made use of a sparse representation of the decision axis (i.e., every 3–4 units) rather than the full decision axis assumed by the classic SDT model.

Although the estimated posterior distributions provide detailed information about the parameters in the model, they provide little information about the fit of the model. One approach to assessing the fit of the model to the data is through the posterior predictive distribution. The posterior predictive distribution is obtained by generating predictions from the model conditional on the parameter estimates that were obtained. The result of generating predictions from the model in this way is a probability density function for hit and false alarm rates, which can then be compared to the actual hit and false alarm rates observed from the experiment. Figure 9 shows the posterior predictive distribution of the model (gray cloud) along with the data from the ex-

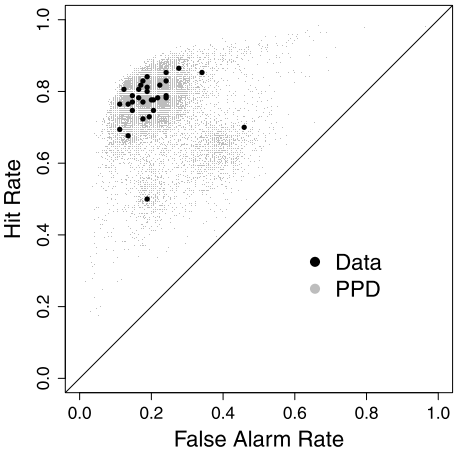

FIGURE 9.
The posterior predictive distribution of the DSD model (*gray cloud*) along with the data of Turner et al. (2011; *black points*).

periment (black points). The figure shows a close alignment of the model predictions and the observed data, indicating that the model fit is at least not inconsistent with the data.

Figure 9 shows that most of the subjects clustered together, having very similar values of discriminability and bias as measured by the classic SDT model (see Macmillan & Creelman, 2005). However, there were two subjects who performed very differently from the rest, having much smaller values of discriminability. In Turner et al. (2011), the data from these two subjects were difficult to capture using approximate least-squares. However, the hierarchical version of the DSD model captured these patterns much better, with the posterior predictive density extending to the areas in the ROC space in which these two subjects' data were found.

5. Summary and Conclusions

We began this paper with a discussion of approximate Bayesian computation (ABC). Although ABC can be applied to any model, it is particularly useful for those models that do not have explicit likelihoods. Such models, frequently being strictly computational or simulation-based, are common in psychology and cognitive neuroscience, and so ABC methods are an important advance for testing and evaluating them. However, ABC methods have not, up to this point, been applied to hierarchical models in psychology. Hierarchical models have parameters that describe the wide range of individual differences across subjects within experimental conditions as well as the effects of experimental conditions at a group level. The increased computational demands associated with high-dimensional hierarchical models has prevented the application of ABC except in simple cases (Beaumont, 2010).

We briefly discussed a naïve extension of ABC to hierarchical models that did not distinguish between subject-level parameters and hyperparameters (see the right panel of Figure 2). We argued that this approach is not practical for even moderately-challenging problems because of its overwhelming rejection rates. We then presented a new algorithm, called Gibbs ABC, which combines the ABC approach for the subject-level parameters with standard Bayesian techniques for the hyperparameters. In an illustrative example, we then used the Gibbs ABC algorithm to estimate the parameters of the classic SDT model.

The application of ABC to the classic SDT model accomplished two things. First, the likelihood function for the SDT model is known and very simple, and so we could estimate the

true posterior distributions easily with standard MCMC techniques. The estimates under MCMC were very similar to the estimates we obtained using Gibbs ABC for both the group- and subject-level parameters. Therefore, we can conclude that the ABC estimates were accurate. Second, this exercise demonstrates that ABC is not restricted to simulation-based models. The likelihood of the SDT model, the binomial, is of a simple closed form, and so lends itself well to MCMC methods (Rouder & Lu, 2005; Lee, 2008). The efficiency of the two methods, ABC and MCMC, was comparable; each fit was obtained in less than 10 minutes.

After this demonstration, we used Gibbs ABC to estimate the parameters of the dynamic, stimulus driven (DSD) model of signal detection. Unlike the classic SDT model, the DSD model constructs evolving representations of the two stimulus classes, and the changes in the representations from trial to trial result in an intractable likelihood function. As a result, previous estimation of the model parameters was limited to approximate least-squares (Turner et al., 2011) on a restricted version of the full model. Using ABC, we easily fit a complex, hierarchical version of the full DSD model containing 132 parameters. We were then able to use the parameter estimates we obtained to gain better insight into our model as well as the representations that subjects may have used during the experiment. We assessed the model fit by plotting the predictions of the fitted model against the data that were observed. We concluded that the Gibbs ABC approach provided reasonably accurate posterior estimates because the model predictions matched the location and spread of the observed data.

We should note that the DSD model lacks a likelihood only at the subject level. The subject-level parameters were drawn from simple, well-known hyperdistributions. This need not be the case: the subject-level parameters themselves may have poorly-behaved hyperdistributions. In this case, the ABC algorithm could be extended to the hyperparameters.

In previous efforts, we have shown that the ABC approach accurately recovers the posterior distribution for different models of varying complexity (Turner & Van Zandt, 2012; Turner & Sederberg, 2012). However, these previous applications have been limited because they were either not hierarchical or very inefficient. We have shown that the Gibbs ABC algorithm allows for accurate estimation of hierarchical model parameters without the use of a likelihood function. As such, the present paper marks an advance toward a fully-Bayesian analysis of hierarchical simulation-based models.

References

Bazin, E., Dawson, K.J., & Beaumont, M.A. (2010). Likelihood-free inference of population structure and local adaptation in a Bayesian hierarchical model. *Genetics*, *185*, 587–602.

Beaumont, M.A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, *41*, 379–406.

Beaumont, M.A., Cornuet, J.-M., Marin, J.-M., & Robert, C.P. (2009). Adaptive approximate Bayesian computation. *Biometrika*, *asp052*, 1–8.

Benjamin, A.S., Diaz, M., & Wee, S. (2009). Signal detection with criterior noise: applications to recognition memory. *Psychological Review*, *116*, 84–115.

Brown, S., & Heathcote, A. (2005). A ballistic model of choice response time. *Psychological Review*, *112*, 117–128.

Brown, S., & Heathcote, A. (2008). The simplest complete model of choice reaction time: linear ballistic accumulation. *Cognitive Psychology*, *57*, 153–178.

Brown, S., & Steyvers, M. (2005). The dynamics of experimentally induced criterion shifts. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *31*, 587–599.

Christensen, R., Johnson, W., Branscum, A., & Hanson, T.E. (2011). *Bayesian ideas and data analysis: an introduction for scientists and statisticians*. Boca Raton: CRC Press, Taylor and Francis Group.

DeCarlo, L.T. (2012). On a signal detection approach to m-alternative forced choice with bias, with maximum likelihood and Bayesian approaches to estimation. *Journal of Mathematical Psychology*, *56*, 196–207.

Dennis, S., & Humphreys, M.S. (2001). A context noise model of episodic word recognition. *Psychological Review*, *108*, 452–478.

Dorfman, D.D., & Alf, E. (1969). Maximum-likelihood estimation of parameters of signal-detection theory and determination of confidence intervals—rating-method data. *Journal of Mathematical Psychology*, *6*, 487–496.

Efron, B. (1986). Why isn't everyone a Bayesian? *American Statistician*, *40*, 1–11.

Egan, J.P. (1958). *Recognition memory and the operating characteristic* (Tech. Rep. AFCRC-TN-58-51). Hearing and Communication Laboratory, Indiana University, Bloomington, Indiana.

Erev, I. (1998). Signal detection by human observers: a cutoff reinforcement learning model of categoriation decisions under uncertainty. *Psychological Review*, *105*, 280–298.
Excoffer, L., Estoup, A., & Cornuet, J.-M. (2005). Bayesian analysis of an admixture model with mutations and arbitrarily linked markers. *Genetics*, *169*, 1727–1738.
Fearnhead, P., & Prangle, D. (2012). Constructing summary statistics for approximate Bayesian computation: semi-automatic approximate Bayesian computation. *Journal of the Royal Statistical Society. Series B*, *74*, 419–474.
Gelman, A., Carlin, J.B., Stern, H.S., & Rubin, D.B. (2004). *Bayesian data analysis*. New York: Chapman and Hall.
Gilks, W.R., & Wild, P. (1992). Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, *41*, 337–348.
Green, D.M., & Swets, J.A. (1966). *Signal detection theory and psychophysics*. New York: Wiley.
Hickerson, M.J., & Meyer, C. (2008). Testing comparative phylogeographic models of marine vicariance and dispersal using a hierarchical Bayesian approach. *BMC Evolutionary Biology*, *8*, 322.
Hickerson, M.J., Stahl, E.A., & Lessios, H.A. (2006). Test for simultaneous divergence using approximate Bayesian computation. *Evolution*, *60*, 2435–2453.
Jilk, D.J., Lebiere, C., O'Reilly, R.C., & Anderson, J.R. (2008). SAL: an explicitly pluralistic cognitive architecture. *Journal of Experimental and Theoretical Artificial Intelligence*, *20*, 197–218.
Kubovy, M., & Healy, A.F. (1977). The decision rule in probabilistic categorization: what it is and how it is learned. *Journal of Experimental Psychology. General*, *106*, 427–446.
Lee, M.D. (2008). Three case studies in the Bayesian analysis of cognitive models. *Psychonomic Bulletin & Review*, *15*, 1–15.
Lee, M. D., & Wagenmakers, E.-J. (2012). A course in Bayesian graphical modeling for cognitive science. Available from http://www.ejwagenmakers.com/BayesCourse/BayesBookWeb.pdf; last downloaded January 1, 2012.
Lunn, D., Thomas, A., Best, N., & Spiegelhalter, D. (2000). WinBUGS—a Bayesian modelling framework: concepts, structure and extensibility. *Statistics and Computing*, *10*, 325–337.
Macmillan, N.A., & Creelman, C.D. (2005). *Detection theory: a user's guide*. Mahwah: Lawrence Erlbaum Associates.
Malmberg, K.J., Zeelenberg, R., & Shiffrin, R. (2004). Turning up the noise or turning down the volume? On the nature of the impairment of episodic recognition memory by Midazolam. *Journal of Experimental Psychology. Learning, Memory, and Cognition*, *30*, 540–549.
Matzke, D., & Wagenmakers, E.-J. (2009). Psychological interpretation of the ex-Gaussian and shifted Wald parameters: a diffusion model analysis. *Psychonomic Bulletin & Review*, *16*, 798–817.
Mazurek, M.E., Roitman, J.D., Ditterich, J., & Shadlen, M.N. (2003). A role for neural integrators in perceptual decision making. *Cerebral Cortex*, *13*, 1257–1269.
McElree, B., & Dosher, B.A. (1993). Serial retrieval processes in the recovery of order information. *Journal of Experimental Psychology. General*, *122*, 291–315.
Mueller, S.T., & Weidemann, C.T. (2008). Decision noise: an explanation for observed violations of signal detection theory. *Psychonomic Bulletin & Review*, *15*, 465–494.
Myung, I.J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, *47*, 90–100.
Nelder, J.A., & Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, *7*, 308–313.
Nosofsky, R.M., Little, D.R., Donkin, C., & Fific, M. (2011). Short-term memory scanning viewed as exemplar-based categorization. *Psychological Review*, *118*, 280–315.
Nosofsky, R.M., & Palmeri, T. (1997). Comparing exemplar-retrieval and decision-bound models of speeded perceptual classification. *Perception and Psychophysics*, *59*, 1027–1048.
O'Reilly, R., & Frank, M. (2006). Making working memory work: a computational model of learning in the prefrontal cortex and basal ganglia. *Neural Computation*, *18*, 283–328.
Pritchard, J.K., Seielstad, M.T., Perez-Lezaun, A., & Feldman, M.W. (1999). Population growth of human Y chromosomes: a study of Y chromosome microsatellites. *Molecular Biology and Evolution*, *16*, 1791–1798.
Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*, 59–108.
Ratcliff, R., & Rouder, J.N. (1998). Modeling response times for two-choice decisions. *Psychological Science*, *9*, 347–356.
Ratcliff, R., & Smith, P.L. (2004). A comparison of sequential sampling models for two-choice reaction time. *Psychological Review*, *111*, 333–367.
Ratcliff, R., & Starns, J. (2009). Modeling confidence and response time in recognition memory. *Psychological Review*, *116*, 59–83.
Robert, C.P., & Casella, G. (2004). *Monte Carlo statistical methods*. New York: Springer.
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.
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.
Rouder, J.N., Sun, D., Speckman, P., Lu, J., & Zhou, D. (2003). A hierarchical Bayesian statistical framework for response time distributions. *Psychometrika*, *68*, 589–606.
Shiffrin, R.M., Lee, M.D., Kim, W., & Wagenmakers, E.-J. (2008). A survey of model evaluation approaches with a tutorial on hierarchical Bayesian methods. *Cognitive Science*, *32*, 1248–1284.
Shiffrin, R.M., & Steyvers, M. (1997). A model for recognition memory: REM—retrieving effectively from memory. *Psychonomic Bulletin & Review*, *4*, 145–166.
Sisson, S., Fan, Y., & Tanaka, M.M. (2007). Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States of America*, *104*, 1760–1765.
Sousa, V.C., Fritz, M., Beaumont, M.A., & Chikhi, L. (2009). Approximate Bayesian computation without summary statistics: the case of admixture. *Genetics*, *181*, 1507–1519.

Toni, T., Welch, D., Strelkowa, N., Ipsen, A., & Stumpf, M.P. (2009). Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, *6*, 187–202.

Treisman, M., & Williams, T. (1984). A theory of criterion setting with an application to sequential dependencies. *Psychological Review*, *91*, 68–111.

Tsetsos, K., Usher, M., & McClelland, J.L. (2011). Testing multi-alternative decision models with non-stationary evidence. *Frontiers in Neuroscience*, *5*, 1–18.

Turner, B.M., Dennis, S., & Van Zandt, T. (2013). Likelihood-free Bayesian analysis of memory models. *Psychological Review*, *120*, 667–678.

Turner, B.M., & Sederberg, P.B. (2012). Approximate Bayesian computation with differential evolution. *Journal of Mathematical Psychology*, *56*, 375–385.

Turner, B.M., & Van Zandt, T. (2012). A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, *56*, 69–85.

Turner, B.M., Van Zandt, T., & Brown, S.D. (2011). A dynamic, stimulus-driven model of signal detection. *Psychological Review*, *118*, 583–613.

Usher, M., & McClelland, J.L. (2001). On the time course of perceptual choice: the leaky competing accumulator model. *Psychological Review*, *108*, 550–592.

Van Zandt, T. (2000). How to fit a response time distribution. *Psychonomic Bulletin & Review*, *7*, 424–465.

Van Zandt, T., Colonius, H., & Proctor, R.W. (2000). A comparison of two response time models applied to perceptual matching. *Psychonomic Bulletin & Review*, *7*, 208–256.

Van Zandt, T., & Jones, M.R. (2011). *Stimulus rhythm and choice performance*. Unpublished manuscript.

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

Wald, A. (1947). *Sequential analysis*. New York: Wiley.

Wilkinson, R.D. (2011). *Approximate Bayesian computation (ABC) gives exact results under the assumption of model error*. Manuscript submitted for publication.

Manuscript Received: 18 MAR 2013
Published Online Date: 3 DEC 2013