Journal of Mathematical Psychology 56 (2012) 69–85

Contents lists available at SciVerse ScienceDirect

Journal of Mathematical Psychology

journal homepage: www.elsevier.com/locate/jmp

A tutorial on approximate Bayesian computation

Brandon M. Turner [a,*], Trisha Van Zandt [b]

[a] University of California, Irvine, United States
[b] The Ohio State University, United States

ARTICLE INFO

Article history:
Received 12 July 2011
Received in revised form
14 February 2012
Available online 26 March 2012

Keywords:
Approximate Bayesian computation
Tutorial
Bayesian estimation
Population Monte Carlo

ABSTRACT

This tutorial explains the foundation of approximate Bayesian computation (ABC), an approach to Bayesian inference that does not require the specification of a likelihood function, and hence that can be used to estimate posterior distributions of parameters for simulation-based models. We discuss briefly the philosophy of Bayesian inference and then present several algorithms for ABC. We then apply these algorithms in a number of examples. For most of these examples, the posterior distributions are known, and so we can compare the estimated posteriors derived from ABC to the true posteriors and verify that the algorithms recover the true posteriors accurately. We also consider a popular simulation-based model of recognition memory (REM) for which the true posteriors are unknown. We conclude with a number of recommendations for applying ABC methods to solve real-world problems.

1. Introduction

Following nearly a century of frequentist approaches to data analysis and model fitting, the "Bayesian revolution", together with the availability of powerful desktop computers and powerful algorithms to fit full Bayesian models, has allowed psychologists to exploit Bayesian methods in behavioral research. Bayesian methods are important not only because they circumvent the "ritualized exercise of devil's advocacy" (Abelson, 1995, p. 9) of null hypothesis testing, but also because they allow for statistical inference without compromising the theory motivating the experiments that generated the data (e.g., Lee, Fuss, & Navarro, 2006; Nilsson, Rieskamp, & Wagenmakers, 2011; Oravecz, Tuerlinckx, & Vandekerckhove, 2011; Vandekerckhove, Tuerlinckx, & Lee, 2011; Wetzels, Vandekerckhove, Tuerlinckx, & Wagenmakers, 2008). Thus, Bayesian techniques complement the development of statistical and mathematical models.

To understand the close link between Bayesian analyses and model development, consider the data $Y = \{Y_1, Y_2, \ldots, Y_n\}$ observed after conducting an experiment. The data could be anything, such as response times, ratings on a 1–7 scale, hit and false alarm rates, or EEG traces. The data from many experiments in cognitive psychology (as well as other areas of behavioral research) are assumed to arise from a specific mathematical or statistical model of the data-generating process. For example, if the data Y are response times, the data-generating process could be described by a two-boundary diffusion process (Ratcliff, 1978). If the data are hit and false alarm rates, the data-generating process could be described by signal detection theory (Green & Swets, 1966). Each of these models of the data-generating process depends on a set of parameters θ, such as the d', σ and β of signal detection theory, and the goal of statistical inference is to say something about how those parameters change under changes in experimental conditions.[1]

The fundamental difference between Bayesian statistics and frequentist techniques lies in how the parameters θ are conceived. For frequentists, parameters are assumed to be fixed within a group, condition or block of experimental trials and inference is therefore based on the sample space of hypothetical outcomes that might be observed by replicating the experiment many times. Inference about these unknown, fixed parameters takes the form of a null hypothesis test (such as a t-test), or estimating the parameters by determining the parameter values that minimize the difference between the model predictions and the data.

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—the *posterior distributions*. There are two requirements to compute or estimate these posterior distributions. First, we must be able to compute the likelihood of the data; that is, given a model with a set of parameters θ, we must specify the probability of each observation in the sample. For mathematical models

* Corresponding author.
E-mail address: turner.826@gmail.com (B.M. Turner).

[1] A word about notation is in order. Throughout this tutorial, an unadorned variable such as Y or θ should be permitted to take on vector values. If a variable is subscripted (e.g., ϵ_t), it is a scalar or an element (possibly vector-valued) of a vector. Capital Roman letters represent variable quantities, while lower-case letters represent fixed values.

0022-2496/$ – see front matter © 2012 Elsevier Inc. All rights reserved.
doi:10.1016/j.jmp.2012.02.005

(such as the diffusion model or signal detection theory), this requirement is simply that we be able to write down the theoretical probability density $f(y|\theta)$ for any observation y. Assuming that the observations $\{Y_1, Y_2, \ldots, Y_n\}$ are independent and identically distributed, the likelihood is defined as

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

Second, we must supply a *prior* distribution for θ. This prior distribution may be based on our previous understanding of likely values for θ. For example, in a diffusion model, we might place a distribution for the, say, drift rate at a location suggested by previous values of the drift rate estimated under different conditions (Wagenmakers, van der Maas, & Grasman, 2007). Alternatively, this prior may instead reflect the fact that we know nothing at all about θ. In this case, we might use a prior that is uninformative, or widely dispersed over the allowable range or *support* of θ.

Whether the prior is informative or not, after observing the data it is updated, by way of the likelihood, to produce the posterior distribution for θ. Using Bayes' Theorem, the posterior $\pi(\theta|Y)$ is

$$\pi(\theta|Y) = \frac{L(\theta|Y)\pi(\theta)}{\int L(\theta|Y)\pi(\theta)\, d\theta}. \quad (2)$$

With the posterior distribution of θ in hand, we can examine the random behavior of θ. For example, keeping in mind a frequentist alternative hypothesis such as $H_A : \theta > 0$, we can provide the probability that θ really is greater than zero, or, conversely, the probability that the null hypothesis $H_0 : \theta \leq 0$ is true. The posterior can be used to estimate a "credible set", the Bayesian counterpart to a confidence interval for θ. The central tendency of the posterior (mode, median or mean) can be used as a point estimate for θ. As an alternative to these approaches, one could designate a small interval such that any value within the interval is equivalent to the value of interest (e.g., values around $H_0 : \theta = 0$), for all practical purposes. This interval is referred to as the region of practical equivalence (Kruschke, 2011).

Although this framework is appealing and powerful in theory, exact evaluation of the posterior distribution can be very complicated, which until fairly recently restricted its utility to only a few problems. The difficulty in evaluating the posterior is due to the integral appearing as the denominator of Eq. (2), which is, for realistic models, usually intractable. However, this integral is simply a complicated normalizing constant. That is, the posterior distribution is proportional to the prior times the likelihood, or

$$\pi(\theta|Y) \propto L(\theta|Y)\pi(\theta). \quad (3)$$

If both the likelihood and the prior have analytic forms, Eq. (3) implies that the desired posterior is tantalizingly close at hand. If the distributional form of $\pi(\theta|Y)$ can be deduced from the product of the likelihood and the prior, we need only write down this distributional form (e.g., a gamma distribution) to compute the posterior probabilities of interest. If the posterior $\pi(\theta|Y)$ does not follow any convenient distributional form, there remains considerable computation before we can accurately estimate or obtain samples from it. The recent enthusiasm for Bayesian methods in the psychological community (and elsewhere) derives from the development of simulation methods (such as Markov chain and sequential Monte Carlo) and the availability of computers powerful enough to efficiently implement these methods to estimate the posterior $\pi(\theta|Y)$.

Monte Carlo methods make use of a "proposal" distribution, a simple distribution such as the Gaussian from which samples can be easily obtained. These samples are then filtered in such a way that the samples that are consistent with the desired posterior are retained and all others are discarded. When Monte Carlo methods are appropriately implemented, the theory of Markov chains guarantees that, in the limit (that is, with a large enough "chain" of samples), the distribution of the filtered samples approaches the distribution of the posterior $\pi(\theta|Y)$.

The prior $\pi(\theta)$ is always available, regardless of the model of interest, because it is selected by the researcher. However, there are many models for which a likelihood can be difficult or impossible to specify mathematically. This problem arises most frequently for computational or simulation-based models,[2] which generate predictions by simulating the data-generating mechanism. These models are very popular in the social sciences, and in cognitive psychology in particular. In these cases, the application of standard methods of Bayesian estimation, as well as classical maximum likelihood estimation (Myung, 2003), has not been possible.

Consider, for example, O'Reilly and colleagues' LEABRA model of learning (O'Reilly, 2001, 2006; O'Reilly & Munakata, 2000). LEABRA is a connectionist network in which different sets of individual computational units are organized into layers, and these layers communicate by way of weighted connections between the units. The network learns to produce certain patterns of activation in response to input patterns by modifying the connection weights.

The unique contribution of the LEABRA architecture is how its organization is tied to neural dynamics and neurophysiology. The parameters of the neural units are chosen to correspond to the electrophysiological constants controlling neural membrane potential. Learning occurs in different ways and different rates, corresponding to the Hebbian, error-monitoring, and reinforcement learning observed in biological systems. Different layers of neural units correspond to posterior cortex, hippocampus, and basal ganglia.

The model has been applied to a wide range of problems in cognition, including perception, language, attention, learning and memory. The behavior of the model in different circumstances is determined by simulating its behavior many times. There is no analytical form available that describes the probability of different model outputs. Therefore, like other simulation models in psychology, LEABRA has not been able to take advantage of progress in Bayesian computation. Similar problems have been encountered in biology, particularly in genetics. In this context, an approach called "approximate Bayesian computation" has been successfully applied to estimating the parameters of complex genetic models. Our tutorial presents this new approach and demonstrates how it can be applied to computational models of cognition.

2. Plan of the tutorial

We begin in Section 3 by presenting the ideas behind approximate Bayesian computation (ABC) and a number of algorithms that have been used to generate estimates of the posterior distribution. We start by demonstrating how ABC can be applied to a number of toy problems, problems for which the true posterior distribution can easily be derived and compared to the approximation provided by ABC (Sections 4 and 5).

Our first example considers a problem with binomially distributed data and a simple ABC rejection sampler (see Algorithm 1 in Fig. 1). Next, we move to an exponential model, which requires that we shift to a more general ABC algorithm, the ABC population Monte Carlo sampler (Algorithm 2 in Fig. 3). In Section 6, we

[2] In this paper we will make a distinction between "mathematical models", models for which a likelihood can be derived, and "simulation models", for which no closed-form likelihood exists.

generalize the ABC population Monte Carlo sampler for hierarchical models and apply it to simulated data from a hierarchical binomial model (see Algorithm 3 in Fig. 5). Finally, in Section 7 we apply Algorithm 3 to a popular computational model of recognition memory, Shiffrin and Steyver's (1997) Retrieving Effectively from Memory (REM) model. We conclude the tutorial with a number of practical suggestions for implementing the ABC approach.

3. Approximate Bayesian computation

Originally developed by Pritchard, Seielstad, Perez-Lezaun, and Feldman (1999), approximate Bayesian computation (ABC) replaces the calculation of the likelihood function $L(\theta|Y)$ in Eqs. (2) and (3) with a simulation of the model that produces an artificial data set X. The method then relies on some metric (a distance) to compare the simulated data X to the data Y that were observed.

Simulating the model to produce a data set that is then compared to the observed data is a technique that is used elsewhere to estimate parameters of computational models (e.g., Malmberg, Zeelenberg, & Shiffrin, 2004; Nosofsky, Little, Donkin, & Fific, 2011). It is common to use the sum of squared error between summary statistics of the simulated and observed data as a distance, and attempt to find point estimates of the parameters by minimizing the sum of squared error using standard optimization techniques: the method of least squares where simulation provides the "predicted" values for the model.

ABC is similar to this "approximate" method of least squares but has a much different goal. The goal of ABC is not to find point estimates of parameters that minimize some discrepancy function like the sum of squared error, but instead to obtain an estimate of the posterior distributions for those parameters.

Recall that the posterior of a parameter θ is the distribution of that parameter conditioned on the observed data Y. Without a likelihood, it is not possible to write down an expression for this posterior, or to estimate it using Monte Carlo methods. However, we can simulate data X using some $\theta = \theta^*$. We retain θ^* as a sample from the posterior if some pre-defined distance $\rho(X, Y)$ between the observed and simulated data is less than some small value ϵ_0. For small values of ϵ_0, the posterior $\pi(\theta|\rho(X, Y) \leq \epsilon_0)$ will approximate the posterior $\pi(\theta|Y)$ (Pritchard et al., 1999).

More formally, an ABC algorithm proceeds in the following way: first, we sample a candidate parameter value θ^* from some distribution. For the first candidate, a reasonable choice for this distribution will be the prior $\pi(\theta)$. We then use this candidate to simulate a data set X from the model of interest that has the same number of observations as the observed data set Y (so that the distributional properties of the simulated data X and any summary statistics computed from it can match those of the observed data Y).

We then compare the simulated data X to the observed data Y by computing a distance between them given by a distance function $\rho(X, Y)$. If $\rho(X, Y)$ is small enough, less than some ϵ_0, then the simulated data X is "close enough" to the observed data Y that the candidate parameter value θ^* has some nonzero probability of being in the approximate posterior distribution $\pi(\theta|\rho(X, Y) \leq \epsilon_0)$. Therefore, if $\rho(X, Y)$ is less than or equal to ϵ_0, we keep θ^* as a sample from the posterior, otherwise we discard it.

For computational ease, it is often convenient to define $\rho(X, Y)$ as a distance between summary statistics $S(X)$ and $S(Y)$ computed for the simulated and observed data. For example, $S(\cdot)$ could be the sample mean, so perhaps $\rho(X, Y) = (\overline{X} - \overline{Y})^2$, the squared distance between the sample means. However, some statistics contain more information about a parameter than others. Ideally, the summary statistic $S(\cdot)$ should be *sufficient* for the parameter θ. Briefly, sufficient statistics provide as much information about the parameter θ as the whole data set itself. Thus, if $S(\cdot)$ is a sufficient statistic for the parameter θ, then the posterior distribution can be written as

$$\pi(\theta \mid Y) = \pi(\theta \mid S(Y)).$$

More formally, to determine if a statistic $S(\cdot)$ is sufficient, we must be able to reexpress Eq. (1) as a function of the sufficient statistic and the data. By the Fisher–Neyman Factorization Theorem, if the probability distribution $f(y \mid \theta)$ can be factored as

$$f(y \mid \theta) = g\left(S(y) \mid \theta\right) h(y), \tag{4}$$

then $S(y)$ is sufficient for the parameter θ.

As an example, consider a series of n independent and identically-distributed Bernoulli trials and let $Y_i \in \{0, 1\}$ be the outcome on trial i. Then we can write

$$P(Y_i = y) = \begin{cases} \theta^y (1-\theta)^{1-y} & \text{for } y \in \{0, 1\} \\ 0 & \text{otherwise,} \end{cases}$$

where $\theta \in [0, 1]$ is the probability that $Y_i = 1$. Then the joint probability function for the set of outcomes $\{Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n\}$ is

$$\begin{aligned} f(y \mid \theta) &= \prod_{i=1}^{n} P(Y_i = y_i) \\ &= \prod_{i=1}^{n} \theta^{y_i} (1-\theta)^{1-y_i} \\ &= \theta^{\sum_{i=1}^n y_i} (1-\theta)^{n-\sum_{i=1}^n y_i} \\ &= \left(\frac{\theta}{1-\theta}\right)^{\sum_{i=1}^n y_i} (1-\theta)^n. \end{aligned}$$

Therefore, the function $f(y \mid \theta)$ can be written as a function of the unknown parameter θ and the statistic $S(y) = \sum_{i=1}^n y_i$. By Eq. (4), we can let $g\left(S(y) \mid \theta\right) = g\left(\sum_{i=1}^n y_i \mid \theta\right)$ and $h(y) = 1$, demonstrating that the statistic $S(y) = \sum_{i=1}^n y_i$ is sufficient for the parameter θ.

The premise behind ABC is that if $\rho(X, Y)$ is defined by way of sufficient statistics, then the resulting approximation to the posterior will be good as long as $\rho(X, Y)$ is less than some small ϵ. The issue is more complicated, however; $\rho(X, Y)$ must also be chosen in such a way that

$$\pi(\theta \mid Y) = \pi(\theta \mid S(Y)) \approx \pi(\theta \mid \rho(S(X), S(Y)) \leq \epsilon). \tag{5}$$

If ϵ and $\rho(S(X), S(Y))$ are chosen well, then the approximation given by an ABC sampler will be exact (Beaumont, 2010).

As one might imagine, choosing $\rho(X, Y)$ well can be tricky, in part because it will depend on the unknown likelihood $f(y \mid \theta)$. If the likelihood is unknown, it will be difficult (even impossible) to determine sufficient statistics for the parameter θ. However, as we will show in this paper, at least for some models, the choice of $\rho(X, Y)$ is fairly robust with respect to the particular summary statistics used.

ABC algorithms can take many forms. The simplest of these is the ABC rejection sampling algorithm (see Algorithm 1; Beaumont, Zhang, & Balding, 2002; Pritchard et al., 1999). The ABC rejection sampler simply discards the candidate value θ^* if it does not meet the criterion $\rho(X, Y) \leq \epsilon_0$, as we described above. For very small values of ϵ_0, the rejection rate can be dramatically high. As a result, Algorithm 1 can be very inefficient.

In the rest of this section, we present several different approaches to ABC, focusing in particular on those approaches most similar to the one we advocate for psychological models. This is not intended to be an exhaustive review of ABC algorithms. Interested readers may consult Beaumont (2010), Blum and François (2010), Hickerson and Meyer (2008), Hickerson, Stahl, and

Lessios (2006), Leuenberger and Wegmann (2010), Sousa, Fritz, Beaumont, and Chikhi (2009) and Wegmann, Leuenberger, and Excoffier (2009) for additional options and more mathematical background.

It is also worth noting that ABC is not the only approach to likelihood-free inference (e.g., Wood, 2010), nor are the algorithms presented in this article the most advanced ABC samplers available (Barthelme & Chopin, 2011; Bazin, Dawson, & Beaumont, 2010; Turner & Sederberg, submitted for publication; Turner & Van Zandt, submitted for publication; Wilkinson, submitted for publication). When selecting from the many ABC algorithms to present in this tutorial, we based our decision on computational efficiency, estimation accuracy, and accessibility to a general audience. Considering all of these issues, the algorithms and techniques described in this article are meant to provide a general familiarity with the topic, and they are not necessarily optimized for more advanced modeling problems.

3.1. Markov chain Monte Carlo sampling

Markov chain Monte Carlo (MCMC) sampling is a general technique that has been instrumental, as we discussed above, in Bayesian estimation (Gelman, Carlin, Stern, & Rubin, 2004; Robert & Casella, 2004). It has also been used in an ABC framework (Bortot, Coles, & Sisson, 2007; Marjoram, Molitor, Plagnol, & Tavare, 2003), and we discuss this work here.

MCMC sampling is a process that filters proposed values for θ to arrive at a sample of values drawn from the desired posterior distribution. There are a number of MCMC samplers, the most popular of which is the Metropolis–Hastings algorithm. We begin the Metropolis–Hastings algorithm by selecting some initial value θ_0 for θ. We then sample a candidate value θ^* from a proposal distribution $q(\cdot|\theta_0)$ conditioned on the initial value θ_0. For example, we could choose the proposal distribution q to be Gaussian. Formally,

$$\theta^* \sim \mathcal{N}(\theta_0, \sigma^2),$$

where the notation "$\sim$" means that θ^* has been sampled from or follows a distribution, in this case a Gaussian distribution with mean θ_0 and variance σ^2.

With some probability determined by the likelihood (called the "acceptance probability" or "rejection rate"; see below), we accept θ^* and set $\theta_1 = \theta^*$, or we reject it and set $\theta_1 = \theta_0$. We continue this procedure until, at the end of the sampling process, we have obtained a chain of values $\{\theta_0, \theta_1, \ldots, \theta_m\}$ that we can assume are a sample from the posterior distribution $\pi(\theta|Y)$.

The Metropolis–Hastings algorithm can be very efficient, especially when the prior distribution $\pi(\theta)$ differs substantially from the posterior distribution $\pi(\theta|Y)$. However, computing the acceptance probabilities to generate the chain $\{\theta_0, \theta_1, \ldots, \theta_m\}$ requires an expression for the likelihood.

3.1.1. The ABC MCMC algorithm

MCMC computations can be easily embedded within ABC algorithms. Focusing again on the Metropolis–Hastings algorithm, after sampling θ^*, instead of computing the acceptance probability from the likelihood, we use θ^* to produce simulated data X from the model. We then compute the distance $\rho(X, Y)$ between the observed data Y and the simulated data X and accept θ^* if $\rho(X, Y) \leq \epsilon_0$ and set $\theta_1 = \theta^*$. If $\rho(X, Y) > \epsilon_0$ we always reject θ^*, and $\theta_1 = \theta_0$.

Using the Metropolis–Hastings algorithm, the ABC MCMC acceptance probability for θ^* on iteration $i + 1$ is given by

$$\alpha = \begin{cases} \min\left(1, \dfrac{\pi(\theta^*)q(\theta_i|\theta^*)}{\pi(\theta_i)q(\theta^*|\theta_i)}\right) & \text{if } \rho(X, Y) \leq \epsilon_0 \\ 0 & \text{if } \rho(X, Y) > \epsilon_0, \end{cases}$$

where $\pi(\theta)$ is the prior distribution for θ and q is the proposal distribution. After computing α for θ^*, we draw a sample from a uniform [0, 1] distribution, and if this sample is less than α, we accept θ^*. If the proposal distribution q is symmetric, so $q(\theta_i|\theta^*) = q(\theta^*|\theta_i)$, then α depends only on the prior distribution and $\rho(X, Y)$.

The chain $\{\theta_0, \theta_1, \ldots, \theta_m\}$ must be evaluated for convergence (see Gelman et al., 2004; Robert & Casella, 2004). Convergence diagnostics are important because MCMC algorithms may suffer if the proposal distribution q is poorly chosen. For example, if σ^2 in the Gaussian proposal above is small, the chain is likely to get "stuck" in low-probability regions of the posterior. This occurs because, in low-probability regions, the candidate θ^* is unlikely to produce simulated data X close to the observed data Y. In this situation, the probability of the chain moving out of the low-probability region becomes effectively zero. This feature of the algorithm produces highly dependent samples, an undesirable result that can be remedied through thinning. Thinning is a procedure where only a subset of the chain consisting of equally spaced samples is retained as a sample from the posterior. For instance, we might decide to keep every 100th value from $\{\theta_0, \theta_1, \ldots, \theta_m\}$, which will require that we generate much longer chains.

While all MCMC chains are in danger of getting stuck, the ABC MCMC algorithm is particularly susceptible to this because of the two criteria that the proposal θ^* must meet: not only must it meet the acceptance probability of the standard Metropolis–Hastings sampler, it must also generate data that are sufficiently close to the observed data. Therefore, the rejection rate of ABC MCMC can be extraordinarily high, requiring inordinate computing cycles for even relatively simple problems. To make things worse, MCMC chains cannot be parallelized. As a consequence, we will not consider the ABC MCMC algorithm further.

3.2. Particle filtering

Sequential Monte Carlo sampling differs from the MCMC approach by its use of a particle filter. That is, rather than drawing candidates θ^* one at a time, these algorithms work with large pools of candidates, called particles, simultaneously. The particles are perturbed and filtered at each stage of the algorithm, bringing the pool closer and closer to a sample drawn from the desired posterior.

These algorithms begin by generating a pool of N candidate values for θ. Usually this pool is obtained by sampling from the prior distribution $\pi(\theta)$. Then, in subsequent iterations, particles are chosen randomly from this pool, and the probability of any particle being sampled depends on a weight assigned to that particle. For the first iteration, the probability of choosing any particle is equal to $1/N$; that is, the particles have equal weight. The different sequential Monte Carlo algorithms can be distinguished by how sampling weights are assigned to the particles in the pool in subsequent iterations.

The process of perturbing and filtering the particles requires that we choose what is called a transition kernel. The transition kernel serves the same purpose as the proposal distribution in the MCMC algorithm discussed above. To specify the transition kernel, we need to choose the distribution of a random variable η that will be added to each particle to move it around in the parameter space. For example, if a particle θ^* is sampled from the pool and perturbed by adding a Gaussian deviate $\eta \sim \mathcal{N}(0, \sigma^2)$ to it, then the new proposed value for θ is $\theta^{**} = \theta^* + \eta$. The transition kernel then describes the distribution for θ^{**} given θ^*: a Gaussian distribution with mean θ^* and variance σ^2.

Some algorithms also require that we specify a transition kernel that takes us back to θ^* from θ^{**}. If the distribution of θ^{**} given θ^* is a "forward" transition kernel, then the distribution of θ^* given

θ^{**} is a "backward" transition kernel. If the forward transition kernel is Gaussian as we just described, then, because $\theta^* = \theta^{**} - \eta$, one obvious choice for the backward transition kernel is again a Gaussian distribution with mean θ^{**} and variance σ^2. In general, the forward and backward kernels need not be symmetric or equal as in this example; in practice, however, they frequently are (e.g., Sisson, Fan, & Tanaka, 2007). The optimal choice for the backward kernel can be difficult to determine (Del Moral, Doucet, & Jasra, 2006). Symmetric kernels greatly simplify the algorithm, but may be a poor choice (see Toni, Welch, Strelkowa, Ipsen, & Stumpf, 2009).

We now present three sequential Monte Carlo sampling algorithms adapted for ABC. As we described above, each algorithm differs in the transition kernels they use and how weights are computed to control how particles are sampled from the pool. These algorithms are partial rejection control, population Monte Carlo, and sequential Monte Carlo. Our focus later in this paper will be on the population Monte Carlo algorithm.

3.2.1. Partial rejection control

The ABC partial rejection control (ABC PRC) algorithm was developed by Sisson et al. (2007) as a remedy for the problems associated with ABC MCMC discussed in the previous section. It was the first ABC algorithm to use a particle filter.

The ABC PRC algorithm requires that we choose both a forward and a backward transition kernel. We denote the forward kernel as a density function $q_f(\cdot|\theta^*)$ and the backward kernel as $q_b(\cdot|\theta^{**})$. We use $q_f(\cdot|\theta^*)$ to perturb the particle θ^* to θ^{**}, and then, with θ^{**}, we simulate data X and compare X to the observed data Y by computing $\rho(X, Y)$. If the particle θ^{**} passes inspection (if $\rho(X, Y)$ is less than some ϵ_0), then we keep it and give it a weight which will determine the probability of sampling it on subsequent iterations. If the particle does not pass inspection (if $\rho(X, Y) > \epsilon_0$), it is discarded, and the process is repeated until we obtain a particle that does pass inspection. The weight w given to the new particle θ^{**} is

$$w = \frac{\pi(\theta^{**}) q_b(\theta^*|\theta^{**})}{\pi(\theta^*) q_f(\theta^{**}|\theta^*)}.$$

This process is repeated until the pool consists of N new particles, each satisfying the requirement that $\rho(X, Y) \leq \epsilon_0$.

If we stop now, after recreating the pool once, then ABC PRC is equivalent to the ABC rejection sampler (Algorithm 1). However, we will repeat the process multiple times. On each iteration we sample particles with probabilities based on the weights they were assigned in the previous iteration. These weights allow us to discard particles from the pool in low-probability regions (particles said to be "performing poorly") and increase the number of particles in high-probability regions, finally resulting in a sample of particles that represent a sample from the desired estimate of the posterior $\pi(\theta|\rho(X, Y) \leq \epsilon_0)$.

This weighting scheme solves several of the problems of ABC MCMC, including the problem of a chain getting stuck in a low-probability region. However, the efficiency of the sampler relies heavily on the choices of the two kernels $q_f(\theta|\theta^*)$ and $q_b(\theta|\theta^{**})$ and the prior $\pi(\theta)$. Consider, for example, a situation with noninformative (i.e., flat) priors, so $\pi(\theta)$ is constant, and $q_b = q_f$. In this situation, the weights w assigned to the particles never change, and the algorithm reduces to an ABC rejection sampler. In addition, the ABC PRC produces biased estimates of the posterior (see Beaumont, Cornuet, Marin, & Robert, 2009): the distribution defined by the pool of particles and their weights does not converge to the true posterior. Beaumont et al. (2009) correct for this bias using a population Monte Carlo sampling scheme.

3.2.2. Population Monte Carlo sampling

ABC population Monte Carlo sampling (ABC PMC) has a different weighting scheme than ABC PRC (Beaumont et al., 2009). While the ABC PRC algorithm requires specifying both forward and backward transition kernels, the ABC PMC algorithm uses a single adaptive transition kernel $q(\cdot|\theta^*)$ that depends on the variance of the accepted particles in the previous iteration. This algorithm, shown in Algorithm 2, was inspired by the population Monte Carlo algorithm developed for standard Bayesian estimation by Cappé, Guillin, Marin, and Robert (2004).

Specifically, given the weight $w_{i,t-1}$ for particle $\theta_{i,t-1}$ on iteration $t-1$, the new weight $w_{i,t}$ for particle $\theta_{i,t}$ on iteration t is computed as

$$w_{i,t} = \frac{\pi(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1}\, q\left(\theta_{j,t-1}|\theta_{i,t}, \sigma_{t-1}\right)},$$

where $q\left(\cdot|\theta_{i,t}, \sigma_{t-1}\right)$ is a Gaussian kernel with mean $\theta_{i,t}$ and standard deviation σ_{t-1}. The variance σ_t^2 is given by

$$\sigma_t^2 = 2\frac{1}{N}\sum_{i=1}^{N}\left(\theta_{i,t} - \sum_{j=1}^{N}\theta_{j,t}/N\right)^2 = 2\mathrm{Var}(\theta_{1:N,t}).$$

One serious problem with many sampling schemes is the speed with which posterior estimates can be obtained. This speed is dictated by the particle acceptance rate, or the probability of accepting a proposal. Very low acceptance rates, which arise from poorly selected proposal distributions or transition kernels, result in a tremendous amount of computation wasted on evaluating proposals that have no chance of being selected. The importance of the ABC PMC weighting scheme is that it optimizes the acceptance probability. This happens because the weights minimize the Kullback–Leibler distance between the desired posterior and the proposal distribution. The Kullback–Leibler distance is a popular statistic that measures the discrepancy between two density functions (Beaumont et al., 2009; Kullback, Keegel, & Kullback, 1987). Minimizing the Kullback–Leibler distance in turn maximizes the acceptance probability (see Douc, Guillin, Marin, & Robert, 2007, for a proof).

Note that if $\epsilon_0 = 0$ and $\rho(X, Y)$ is a comparison between summary statistics that are sufficient for θ, then the ABC PMC algorithm produces exact posteriors (Beaumont, 2010). For continuous measures, because the probability that $\rho(X, Y)$ equals $\epsilon_0 = 0$ is zero, we choose some $\epsilon_0 > 0$, and the quality of the approximation will depend on the value of ϵ_0.

3.2.3. Sequential Monte Carlo sampling

Toni et al. (2009) derived the ABC sequential Monte Carlo sampling (ABC SMC) algorithm from a sequential importance sampling algorithm (Del Moral et al., 2006). The weights in ABC SMC are very similar to the weights in ABC PMC, except that the kernel $q(\cdot|\theta^*)$ is nonadaptive and not necessarily Gaussian. Thus, the weights assigned for the ith particle on the tth iteration in the ABC SMC algorithm are given by

$$w_{i,t} = \frac{\pi(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1}\, q(\theta_{j,t-1}|\theta_{i,t})}.$$

The ABC SMC algorithm is particularly useful when the transition kernel in ABC PMC cannot have infinite support (e.g., cannot be Gaussian). This might happen for certain models in which θ cannot be negative; consider, for example, the probability parameter p in the binomial distribution.

3.2.4. Summary

This section summarized some of the most popular and efficient ABC algorithms. We have experimented with all of these, and ABC PMC has consistently provided good results for the psychological models to which we have applied it. In addition, the ABC PMC algorithm requires the fewest specifications of tuning parameters by the user. Therefore, in the applications to follow we will focus primarily on the ABC PMC algorithm (Algorithm 2).

The first three examples that we present are toy problems where the true posteriors are known. This gives us the opportunity to demonstrate ABC, and also to demonstrate the accuracy of the posteriors estimated by ABC. We then show how ABC works with a more realistic model, where the true posteriors are unknown.

4. A binomial example

For our first example, we consider a signal detection experiment in which a subject is asked to respond "yes" when he or she hears a tone embedded in noise and "no" when he or she does not hear a tone. The sensory effects of signals and noise are assumed to follow, as in standard signal detection theory, normal distributions such that the mean of the signal distribution is greater than the mean of the noise distribution.

To simulate data from this experiment, we set the means of the signal and noise distributions at 1.50 and 0, respectively, with a common standard deviation of 1. Under these conditions, d' – the standard measure of discriminability – is equal to the mean of the signal distribution ($d' = 1.50$). With $d' = 1.50$, an ideal observer will correctly identify about 77% of the stimuli. Although we could estimate the signal detection theory parameters d' and β (see Lee, 2008; Rouder & Lu, 2005, for a fully Bayesian treatment of this problem) using ABC, for simplicity assume that we wish only to estimate the probability of a correct response made by the observer regardless of whether the stimulus was a signal or noise.

4.1. The model

Consider correct responses to be "successes" and incorrect responses to be "failures", and let a success be coded as $R = 1$ and a failure as $R = 0$. The outcome R on a single trial can then be modeled as a sample from the familiar Bernoulli distribution with parameter $p = P(R = 1)$. Further assuming that each trial is independent, we can model the number of correct responses Y with the binomial distribution. Recall that the binomial distribution gives the probability of $Y = y$ correct responses in a sequence of n independent and identically distributed Bernoulli trials as

$$f(y|n, p) = \binom{n}{y} p^y (1-p)^{n-y}$$

or

$$y|n, p \sim \text{Bin}(n, p),$$

where y takes on values in $\{0, 1, 2, \ldots, n\}$. Because n is determined by the experimenter, the focus of statistical inference centers on the parameter p.

Bayesian analysis of this model usually proceeds by assuming a beta prior for p, which allows p to range from 0 to 1. The beta distribution $\text{Beta}(\alpha, \beta)$ is given by

$$f(p|\alpha, \beta) = \begin{cases} \dfrac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} & \text{if } 0 < p < 1 \\ 0 & \text{otherwise,} \end{cases}$$

where $\alpha > 0$ and $\beta > 0$ are called the hyperparameters of the model. If we wish to specify an uninformative prior for p, it is convenient to use the fact that the beta distribution with parameters $\alpha = 1$ and $\beta = 1$ is the uniform [0, 1] distribution. For this example, a uniform prior is a convenient choice if we do not wish to speculate *a priori* about the performance of our subject.

The parameters of the beta distribution can be thought of as the number of successes (α) and the number of failures (β) for an earlier experiment. By letting $\alpha = 1$ and $\beta = 1$ for the prior, our experience with p is similar to having witnessed two outcomes, one a success and the other a failure. The uninformative Beta(1, 1) prior places equal probability on all possible values for p in (0, 1). Using the beta distribution as the prior will result in a beta posterior distribution for p. This equivalence relationship between the prior and the posterior is called "conjugacy", and is desirable because it eliminates the need to estimate the posterior. The posterior for this model is

$$p|\alpha, \beta, Y \sim \text{Beta}\,(\alpha = Y + \alpha_0, \beta = n - Y + \beta_0)\,, \tag{6}$$

where α_0 and β_0 denote the chosen values of the hyperparameters for the prior distribution, n denotes the number of trials, and $Y = \sum_{i=1}^{n} R_i$ is the number of correct responses. We will use this posterior distribution to assess the accuracy of the estimated posteriors produced by ABC.

4.2. Estimating the posterior using ABC

Having derived the posterior distribution of p, we could proceed immediately to evaluating hypotheses about p, such as the probability that $p > 0.5$ or computing a 95% credible set for p. However, our goal is to demonstrate the accuracy of the estimates of the posterior produced by the ABC approach, and so we pretend that the binomial likelihood is terribly difficult or impossible to work with. This unfortunate situation, which prevents us from obtaining the true posterior explicitly as in Eq. (6), forces us to simulate data from the binomial model and use the ABC approach.

We must first define a distance to compare our simulated data X with our observed data Y. For this example, we set this distance to

$$\rho(X, Y) = \frac{1}{n}|X - Y|,$$

the absolute difference between the proportions of observed and simulated correct responses. The distance $\rho(X, Y)$ measures the degree to which our simulated data X matches our observed data Y. When $\rho(X, Y) = 0$, the number of successes (failures) is exactly the same for both the observed and simulated data. Reaching this degree of precision can be quite costly in more complicated models. Later, we will allow for a monotonically decreasing set of tolerance thresholds ϵ meant to relieve the computational burden (see Section 5).

4.3. Results

We simulated the model under three sample sizes, each with $p = 0.7$. Treating each sample size as a set of observations from a different observer, the first observer performed $n = 10$ trials, the second observer performed $n = 100$ trials and the third observer performed $n = 1000$ trials. As n increases, the amount of information about the parameter p increases, resulting in posterior distributions that are more peaked (see Eq. (6)).

For the estimates of the posterior, we sampled $N = 10{,}000$ values for p for each observer using the rejection sampling algorithm (Algorithm 1) shown in Fig. 1 with tolerance threshold $\epsilon_0 = 0$. Fig. 2 shows the distributions of values for p for each of the three observers. Overlaying each histogram is the true posterior given by Eq. (6). Fig. 2 shows that as the number of trials increases, the posterior becomes more narrow around the true value of p.

1: Given data and assumed model $Y \sim \text{Model}(\theta)$, tolerance threshold ϵ, and prior distribution $\pi(\theta)$:
2: **for** $1 \leq i \leq N$ **do**
3: **while** $\rho(X, Y) > \epsilon$ **do**
4: Sample θ^* from the prior: $\theta^* \sim \pi(\theta)$
5: Generate data X from θ^*: $X \sim \text{Model}(\theta^*)$
6: Calculate discrepancy $\rho(X, Y)$
7: **end while**
8: Store $\theta_i \leftarrow \theta^*$
9: **end for**

Fig. 1. An ABC rejection sampling algorithm to estimate the posterior distribution of a parameter θ given data Y.

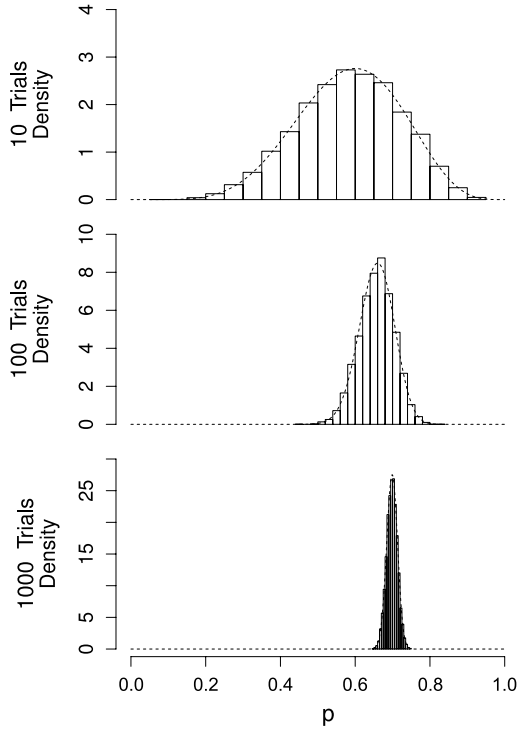

Fig. 2. The posterior distributions for three different subjects performing $n = 10$ (top panel), $n = 50$ (middle panel) and $n = 100$ trials. The dashed curve shows the true posterior distribution.

For each observer, the estimate of the posterior found using ABC is highly accurate, almost exactly equal to the true posterior.

The simplicity of this example allowed us to sample a great many values for p ($N = 10{,}000$) at a negligible cost. Fitting the data took only 5.13 s, 11.5 s, and 5.42 min on a standard Intel i7 processor (3.07 GHz) for each observer, respectively. To implement the algorithm, we used the software R (R Development Core Team, 2008) and distributed the particle evaluations across eight cores. A simplified version of this code is freely available on the first author's website.

5. An exponential example

While the binomial example demonstrates that the ABC approach can accurately estimate the posterior distribution of the probability parameter of the binomial distribution, the binomial variable Y is discrete, taking on only the values between 0 and n. This limited set of measurements and the simplicity of the model made exactly "matching" the observed data easy for the values of n that we examined. We should not expect things to be so easy for more complex models or continuous measurements.

Continuous measurements pose a more difficult modeling challenge because the probability of simulating exactly some value Y observed in the data (say, 2.99792458…) will be zero and perfect matches between X and Y will be impossible. In practice, we round continuous variables, so that 2.99792458… becomes 3.00 (or some other number measured to some acceptable degree of precision). This means we can still implement the ABC algorithm for continuous data, but we must be much more careful in how we select the set of tolerance thresholds ϵ.

For this example, we will apply an ABC algorithm to continuous data generated from an exponential model. The use of the exponential distribution in psychology is widespread. The exponential distribution often appears in modeling problems such as the distribution of response times via the ex-Gaussian (e.g., Farrell & Ludwig, 2008; Matzke & Wagenmakers, 2009; Rouder & Speckman, 2004), practice effects (e.g., Heathcote, Brown, & Mewhort, 2000), relating stimulus similarity to psychological distance (e.g., Nosofsky, 1986), predicting change (Brown & Steyvers, 2009) and memory decay (e.g., Lee, 2004; Liu & Aitkin, 2008; Rubin & Wenzel, 1996; Wixted, 1990). Here we will demonstrate that the ABC PMC extension of Algorithm 1 described above produces accurate estimates of the posterior of the exponential distribution's single parameter.

5.1. The model

The exponential distribution $\text{Exp}(\lambda)$ has the probability density function

$$f(y|\lambda) = \begin{cases} 0 & \text{if } y < 0 \\ \lambda \exp(-\lambda y) & \text{if } y \geq 0, \end{cases}$$

where the parameter $\lambda > 0$ is sometimes called the "rate", and $1/\lambda$ is the mean of Y. The gamma distribution $\Gamma(\alpha, \beta)$ has probability density function

$$f(y|\alpha, \beta) = \begin{cases} 0 & \text{if } y < 0 \\ \dfrac{\beta^\alpha}{\Gamma(\alpha)} y^{\alpha-1} \exp(-y\beta) & \text{if } y \geq 0, \end{cases}$$

where the hyperparameters $\alpha > 0$ and $\beta > 0$ are usually called the shape and rate parameters, respectively. The exponential distribution is a special case of the gamma distribution with $\alpha = 1$ and $\beta = \lambda$. The gamma distribution is a conjugate prior for exponential likelihood, so for observed data $Y = \{Y_1, Y_2, \ldots, Y_n\}$, and a gamma prior with $\alpha = \alpha_0$ and $\beta = \beta_0$, the posterior distribution of λ is

$$\lambda|\alpha, \beta, Y \sim \Gamma\left(\alpha = \alpha_0 + n, \beta = \beta_0 + \sum_{i=1}^{n} Y_i\right).$$

We will use this posterior to evaluate the accuracy of the ABC PMC algorithm. The values of the hyperparameters α_0 and β_0 were fixed at 0.1.

5.2. Estimating the posterior using ABC PMC

We face a problem at this point. How do we generate simulated data X that is sufficiently close to Y when (1) Y is continuous, and therefore cannot be perfectly matched, and (2) the proposal distribution may be very far from the posterior distribution, making it highly unlikely that any set of proposed parameter values can generate simulated data X that is close to the observed data Y? The use of the ABC PMC algorithm (Algorithm 2) described above and shown in Fig. 3 solves Issue 1, but Issue 2 is more complex.

Issue 2 is a concern especially when the target posterior distribution is very different from the proposal distribution, frequently the prior. If we begin by sampling from the prior, we will have to draw a very large number of samples before hitting on

one that results in $\rho(X, Y) \leq \epsilon_t$ for some reasonable value of ϵ_t. If we consider first ($t = 1$) a large value of ϵ_1, we can find satisfactory proposals much more quickly, but the resulting posterior estimates will not be very accurate. The goal, then, is to gradually reduce the value of ϵ_t, so that we "move" efficiently from the prior (or proposal) distribution to the desired posterior distribution.

We must therefore balance computational efficiency with the accuracy of the posterior estimate. To do this, we will specify a set of monotonic decreasing tolerance thresholds ϵ over which the ABC PMC algorithm will iterate. Starting with a large value for ϵ_1, we generate a sample of parameters from a distribution that is intermediate between the prior and the posterior. Assuming that the prior and the posterior distributions have the same support, as ϵ_t gets smaller, this intermediate distribution will more and more closely resemble the desired posterior.

One final issue concerns the interaction between the accuracy of the posterior estimate and the form of the distance function $\rho(X, Y)$. To explore the influence of the choice of $\rho(X, Y)$ on the accuracy of the estimated posteriors, we explored three different forms of $\rho(X, Y)$.

5.2.1. The distance function

Considering first the problem of selecting $\rho(X, Y)$, we retained (for the sake of comparison) the comparable distance as in the binomial example, or

$$\rho_1(X, Y) = \frac{1}{n}\left|\sum_{i=1}^{n} X_i - \sum_{i=1}^{n} Y_i\right| = \left|\bar{X} - \bar{Y}\right|.$$

We also examined

$$\rho_2(X, Y) = |\text{median}(X) - \text{median}(Y)|$$

and

$$\begin{aligned}\rho_3(X, Y) &= |[F^{-1}(0.75, X) - F^{-1}(0.25, X)] \\ &\quad - [F^{-1}(0.75, Y) - F^{-1}(0.25, Y)]| \\ &= |\text{IQR}(X) - \text{IQR}(Y)|,\end{aligned}$$

where $F^{-1}(q, X)$ denotes the qth quantile of the data X and IQR is the interquartile range. While both $\rho_1(X, Y)$ and $\rho_2(X, Y)$ reflect differences in the central tendency of X and Y, $\rho_3(X, Y)$ is the absolute difference between the interquartile ranges of the observed data Y and the simulated data X. Intuitively, for symmetric or nearly symmetric distributions, one may be able to obtain accurate posteriors on the basis of central tendency alone. However, for asymmetric distributions like the exponential, central tendency alone may not provide critical information about skewness or variability, and a distance function based on central tendency may produce inaccurate estimates of the posterior.

We examined other $\rho(X, Y)$ functions in addition to these three, such as the differences between the maximum (and minimum), the differences in the range, the Kolmogorov–Smirnov test statistic, and a probabilistic mixture of differences between the mean and variance. In general, the best $\rho(X, Y)$ functions incorporate all the observations in each sample X and Y (e.g., the sum of the data, the mean of the data). Sisson et al. (2007) demonstrated that the use of a single extreme order statistic, such as the maximum or minimum, results in poor estimates of the posterior. However, for models whose parameters reflect a limit on the range of measurement values that can be observed, a distance defined for the appropriate extreme statistic can yield quite good results. In these situations, a comparison between the maximum or minimum observations may be the best $\rho(X, Y)$ function available.

In sum, to choose an appropriate $\rho(X, Y)$, one strategy is to consider standard estimators of the parameters of the model and the statistical properties of those estimators. For example, a statistic such as $\bar{X}$ (used in $\rho_1(X, Y)$), which may be sufficient for a parameter reflecting central tendency, may provide the basis for a good choice of $\rho(X, Y)$. Maximum likelihood estimators when they are available, such as the minimum statistic for a lower limit, may also provide the basis for a good choice of $\rho(X, Y)$.

It is important to realize that we are not limited to using only one summary statistic. Indeed, many authors have combined several summary statistics in an attempt to efficiently connect the parameters of the model to the data that were observed (e.g., Bazin et al., 2010; Turner & Van Zandt, submitted for publication). When a likelihood is not available, the situation of most interest to anyone considering ABC, evaluating the statistical properties of estimators may not be straightforward. However, one benefit of a simulation-based model is that the parameters have psychological or mechanical interpretations that may be easier to relate to specific features of the data, and those features then can be incorporated into the choice of $\rho(X, Y)$.

5.2.2. Tolerance

We hinted above at the computational difficulties that can arise when tolerance thresholds ϵ are too small. This is a practical consideration, which must be resolved together with the number of tolerance criteria. The number of tolerance criteria determines the number of iterations of the ABC PMC algorithm, so a large number will result in a lengthy estimation procedure. However, too few criteria will result in substantial rejection rates, and again a lengthy estimation procedure. The goal, then, is to find a set of values ϵ that balances the number of iterations against the rejection rates within each iteration. One method for achieving this goal is to specify a set of monotonically decreasing values ϵ.

Currently, there are no good general guidelines for choosing such threshold criteria. The values ϵ will depend on, among other things, the scale of the data and the distance metric $\rho(X, Y)$. For example, using $\rho_1(X, Y)$ above for RT data, which ranges from 200 ms to 2000 ms depending on the task, an $\epsilon_0 < 1$ represents a very small distance indeed. However, for proportional data such as hit rates or subjective probabilities, an $\epsilon_0 < 1$ will not be at all useful. We will discuss some practical guidelines for selecting ϵ later, but until then the reader should recognize that we have selected ϵ somewhat arbitrarily.

To generate the data, we took $n = 500$ samples from an exponential distribution with $\lambda = 0.1$, so the observations ranged from 0 to around 70 with mean 10, standard deviation 10, and interquartile range of approximately 11. We chose the decreasing set of tolerances $\epsilon = \{3, 1, 10^{-1}, 10^{-3}, 10^{-4}, 10^{-5}\}$. We could have selected other values for ϵ; ultimately, only the last (smallest) element of ϵ matters. When the value of this element is small enough, reducing it further does not produce any additional changes in the approximate posterior distribution.

For each of the model fits we used $N = 500$ particles.

5.3. Results

The top panel of Fig. 4 shows the estimated posteriors for three elements of ϵ (columns) for each of the three distance functions (rows). The dashed curves on each panel show the true posteriors and the histograms show the estimated posteriors obtained using ABC PMC. The major finding is that as ϵ_t decreases, the accuracy of the estimated posterior increases. When ϵ_t is small enough (10^{-3}) the approximate posterior distribution will not change very much with further decrease in $\epsilon_{s>t}$. This provides a check on whether or not the estimated posterior has been obtained: if reductions in ϵ_t do not produce changes in the estimated posterior, then the estimate has converged to its final target.

Each panel in Fig. 4 shows, in the upper right corner, the Kullback–Leibler distance between the estimated and actual posteriors. Using this distance as a measure of accuracy of the

Table 1
Computation times for the exponential example (in minutes).

Iteration	Tolerance ϵ	Mean $\rho_1(X, Y)$	Median $\rho_2(X, Y)$	IQR $\rho_3(X, Y)$
1	3	0.07	2.74	0.12
2	1	0.07	1.18	0.08
3	10^{-1}	0.13	3.19	0.68
4	10^{-3}	0.42	2.96	4.84
5	10^{-4}	3.72	16.21	45.76
6	10^{-5}	34.40	143.40	454.71
Total		38.81	169.67	506.18

estimated posterior, we can see that the accuracy under $\epsilon_2 = 1$ is poorer than under $\epsilon_4 = 10^{-3}$ or $\epsilon_6 = 10^{-5}$, and that there is not much change in the accuracy for $\epsilon_t \leq 10^{-3}$. Furthermore, the estimates are more accurate for the distance function $\rho_1(X, Y)$ than for $\rho_2(X, Y)$ or $\rho_3(X, Y)$.

The computation times were considerably longer in this example when compared to the binomial example. Table 1 shows the total time in minutes to complete each iteration, for each distance function. The most obvious trend, to which we have already alluded, is that as ϵ_t is decreased, the computation time grows very fast. However, Table 1 also shows that there are considerable differences across the distance functions. For the mean, the total time (bottom row) is manageable—just less than 40 min. The second distance function took about three hours while the third distance function took about eight hours. While these times may seem unreasonably long, we argue that this is a small price to pay to circumvent the likelihood function. Furthermore, the bulk of the computation time occurs on Iteration 6. Without this final and unnecessary iteration, the finishing times for the simulations would range from five minutes to less than an hour.

The mean difference distance function $\rho_1(X, Y)$ produced more accurate posterior estimates than the other functions because the mean is a sufficient statistic for the parameter λ. However, even the other functions, $\rho_2(X, Y)$ and $\rho_3(X, Y)$, produced estimates that were close to the true posterior. We must note, however, that none of these distance functions, chosen for their simplicity, are necessarily the best that we could have used. A distance based on the entire distribution, such as the Kullback–Leibler distance itself or a Pearson-type discrepancy function (that is, a chi-squared statistic), may produce more accurate posteriors. We compared these alternative distance functions to the results using $\rho_1(X, Y)$ and found that the degree of improvement was very small. This demonstrates that, although selecting an appropriate $\rho(X, Y)$ may be difficult, there may be a range of $\rho(X, Y)$ functions that lead to similar – possibly even exactly the same – results.

6. A hierarchical binomial model

An important extension of Bayesian procedures is to hierarchical models (e.g., Lee, 2011; Shiffrin, Lee, Kim, & Wagenmakers, 2008). A hierarchy is a system of groupings of elements (e.g., subjects in experimental conditions) where lower levels of groupings (e.g., subjects) are subsets of the higher levels (e.g., conditions). Hierarchies are very important to mathematical modelers because they allow inferences to be made at different levels, which is essential to the study of individual and group differences. For instance, in the binomial example, we inferred the probability of correct detections by a single subject. But, if we had collected data from a large number of subjects, we would expect that some of these subjects will have higher (or lower) probabilities than others for reasons that may be more or less interesting.

A hierarchical model allows us to infer not only the probability of correct responses for each subject, but also the probability of correct responses for the groups, taking into account any fixed or random factors of interest such as age, culture, or gender. The estimates of the effects of experimental factors at the higher levels of the hierarchy are informed by the effects of these factors at the level of each individual. In this way the posteriors of the hyperparameters (the parameters at the highest levels of the hierarchy) "learn" from the individual-level parameters, providing pictures of both overall experimental effects and individual differences. This learning also occurs at the individual level because each individual-level parameter informs the hyperparameters, which then distribute this information to the other individual-level parameters in the model. The individual-level parameters then use this information to "borrow" strength from the estimates of other individual-level parameters, an effect known as *shrinkage*. The example in this section will extend the binomial model in Section 4 to a hierarchical design to demonstrate the capabilities of the ABC algorithm further.

We will again consider a simple signal detection experiment similar to the one previously discussed, except this time we will be drawing inferences about four subjects who each complete one block of 100 trials. We are not only interested in determining the posterior distribution of the probability of a correct response at the subject level, but we are also interested in the experiment-level hyperparameters of the distribution from which these probabilities are drawn.

6.1. The model

We assume that all individual parameter values p_i come from a common beta distribution with parameters α and β. The p_is are the subject-specific parameters, while α and β are the group-level hyperparameters. In moving from the simple binomial model to a hierarchical binomial model, we run into the problem of how best to sample from the posteriors of the hyperparameters α and β. Because the mean of a beta distribution is $\alpha/(\alpha + \beta)$, the parameters α and β are not conditionally independent given the values for p_i. Therefore, estimating the posteriors for α and β requires sampling from their joint distribution—we cannot separate them and sample each independently.

To simplify matters, we can consider the posteriors of the subject-level parameters p_i transformed by the logit transformation, or

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right).$$

The logit function is useful because it transforms the probability space from $p_i \in (0, 1)$ to $\text{logit}(p_i) \in (-\infty, \infty)$. If we also assume that

$$\text{logit}(p_i)|\mu, \sigma^2 \sim \mathcal{N}(\mu, \sigma^2),$$

the new hyperparameters μ and σ^2, which take the place of the old hyperparameters α and β, can be modeled independently (see, e.g., Christensen, Johnson, Branscum, & Hanson, 2011; Gelman et al., 2004).

We choose the prior for the new hyperparameter μ to be Gaussian. Specifically,

$$\mu \sim \mathcal{N}(\mu_\mu, \xi_\mu^2)$$

and, because variances are always positive, we choose an inverse gamma prior for σ, or

$$\sigma \sim \Gamma^{-1}(\alpha_\sigma, \beta_\sigma).$$

These choices ensure a proper posterior distribution and a straightforward approach to parameter estimation (see Gelman et al., 2004, for more details).

For our simulations, we set $\mu_\mu = 0$ and $\xi_\mu^2 = 10{,}000$. This is a diffuse prior that gives approximately equal weight to values

Given data Y and model $Y \sim \text{Model}(\theta)$, a set of tolerance thresholds ϵ, and prior distribution $\pi(\theta)$:
At iteration $t = 1$,
for $1 \leq i \leq N$ **do**
  while $\rho(X, Y) > \epsilon_1$ **do**
    Sample θ^* from the prior: $\theta^* \sim \pi(\theta)$
    Generate data X from θ^*: $X \sim \text{Model}(\theta^*)$
    Calculate discrepancy $\rho(X, Y)$
  end while
  Set $\theta_{i,1} \leftarrow \theta^*$
  Set $w_{i,1} \leftarrow \frac{1}{N}$
end for
Set $\sigma_1^2 \leftarrow 2\text{Var}(\theta_{1:N,1})$
At iteration $t > 1$,
for $2 \leq t \leq T$ **do**
  for $1 \leq i \leq N$ **do**
    while $\rho(X, Y) > \epsilon_t$ **do**
      Sample θ^* from the previous iteration: $\theta^* \sim \theta_{1:N,t-1}$ with probabilities $w_{1:N,t-1}$
      Perturb θ^* by sampling $\theta^{**} \sim N(\theta^*, \sigma_{t-1}^2)$
      Generate data X from θ^{**}: $X \sim \text{Model}(\theta^{**})$
      Calculate discrepancy $\rho(X, Y)$
    end while
    Set $\theta_{i,t} \leftarrow \theta^{**}$
    Set $w_{i,t} \leftarrow \frac{\pi(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q_f\left(\theta_{j,t-1} | \theta_{i,t}, \sigma_{t-1}\right)}$
  end for
  Set $\sigma_t^2 \leftarrow 2\text{Var}(\theta_{1:N,t})$
end for

Fig. 3. The ABC PMC algorithm to estimate the posterior distribution of a parameter θ given data Y.

for μ ranging from -5000 to 5000. It is important to note that being more vague (e.g., setting $\xi_\mu^2 = 10^{10}$) is unnecessary. Consider the approximate endpoints of the parameter space for p_i, 0.01 and 0.99. The logit transformation of these points is $\text{logit}(0.99) = -\text{logit}(0.01) = 4.5951$. Thus, a still reasonable prior could be much more narrow (e.g., $\xi_\mu^2 = 100$), without greatly affecting the estimate of the posterior. We then set $\alpha_\sigma = \beta_\sigma = 0.1$. This also is an overly diffuse specification, following the same argument.[3]

6.2. Estimating the posteriors using ABC PMC

We simulated data for four subjects, each providing 100 detection responses. For each subject, we sampled a p_i by first sampling from the distribution of $\text{logit}(p_i)$, which was Gaussian with mean $\mu = -1.0$ and variance $\sigma^2 = 0.5$, and then transforming the sampled value back to the probability scale. We used the obtained p_i as the parameter for generating 100 Bernoulli(p_i) random variables to simulate the detection responses Y_{ij} for Subject i.

We implemented Algorithm 3, shown in Fig. 5, which is the ABC PMC algorithm modified for the hierarchical model. This modification samples from the posterior distributions of the individual-level parameters $\text{logit}(p_i)$ given the values sampled from the posterior distributions of the hyperparameters μ and σ. For a distance metric we selected

$$\rho(X, Y) = \frac{1}{Sn} \sum_{i=1}^{S} \left| \sum_{j=1}^{n} X_{ij} - \sum_{j=1}^{n} Y_{ij} \right|$$

where Y_{ij} (X_{ij}) denotes the jth observed (simulated) response for the ith observed (simulated) subject, $S = 4$ is the number of subjects, and $n = 100$ is the number of observations per subject. In other words, this statistic is the mean absolute difference between the observed and simulated correct response proportions over subjects. Another way to see this distance is as the average of the distances for each subject computed as for the binomial example of Section 4.

Note that if we simulate data for each subject that exactly matches the number of correct responses observed, then $\rho(X, Y) = 0$. The largest that $\rho(X, Y)$ can be is 1. We therefore set $\epsilon = \{0.1, 0.05, 0.03, 0.01, 0.005\}$, emphasizing again that the selection of ϵ is somewhat arbitrary. We try to converge to a distance not much larger than 0, choosing each ϵ_t to reduce the computational burden associated with this goal. (We will discuss a more

[3] We tried other specifications for the prior, such as $\mu_\mu = 0, \xi_\mu^2 = 2.5, \alpha_\sigma = 1$, and $\beta_\sigma = 4$, but these more concentrated priors had little effect on the posteriors we obtained.

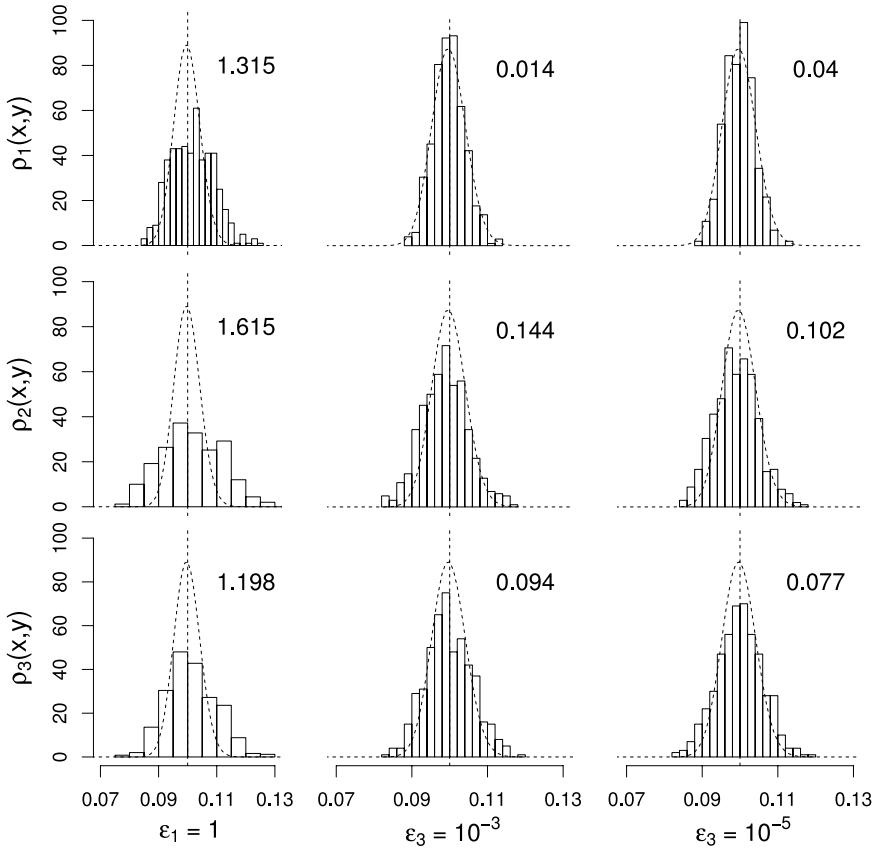

Fig. 4. The posterior distribution of λ at three different tolerance thresholds (columns; $\epsilon = 1, 10^{-3}, 10^{-5}$) and three different $\rho(X, Y)$ functions (rows; see text for details). The dashed curve shows the true posterior distribution and the dashed vertical lines shows the true parameter value. The numbers in the upper right-hand corner of each panel are the Kullback–Leibler distances between the estimated and true posteriors.

efficient approach to this problem in the General Discussion.) For this algorithm we used 1000 particles.

6.3. Results

Fig. 6 shows the estimated posteriors for p_i, μ and $\log(\sigma)$ for three selected levels of ϵ (rows; specifically, $\epsilon_1 = 0.1$, $\epsilon_2 = 0.05$, and $\epsilon_5 = 0.005$). We selected these values of ϵ to display because the posteriors for the other values were not substantially different from ϵ_5. That is, the estimates converged on the true posteriors for $\epsilon \leq 0.03$. The left panel of Fig. 6 shows the approximate marginal posterior distributions of p_i for each of the four subjects, together with the true marginal posteriors (dashed curves) for each of the subjects, as well as the true sampled values for p_i (dashed vertical lines in order corresponding to the curves). As ϵ_t gets smaller the approximate marginal posteriors for p_i approach the true marginal posteriors.

The right panels of Fig. 6 show the approximate joint posterior distribution of the hyperparameters μ and $\log(\sigma)$. We plot the parameter σ on a log scale to reduce it to a scale comparable to that of μ. The darker areas in these smoothed estimates correspond to regions of higher density. The figure shows that as ϵ_t goes to 0 the variance of the joint distribution decreases. The true sampled values of μ and $\log(\sigma)$ are shown as the dashed lines in the figures.

The true values of the hyperparameters do not appear to reflect the central tendency of the estimated posteriors, even for the smallest element of ϵ. Although the true values of the hyperparameters are contained within the marginal 95% credible sets for μ and $\log(\sigma)$, this apparent "inaccuracy" in the posteriors arises from the quite small number of subjects in the experiment. Even with a larger number of subjects, an entirely accurate posterior estimate may not be centered exactly on the true parameter values. Observe, for instance, the differences in the true parameter values and the central tendencies of the true marginal posterior distributions of the p_is of Fig. 6 (left panel). This happens because, for small numbers of subjects (or observations), there will be a stronger influence of the prior on the shape of the posterior.

This example demonstrates some of the mathematical complexities that arise as a result of a hierarchical design. However, the extension of the ABC PMC algorithm to hierarchical designs requires little innovation. Algorithm 3 is equivalent to Algorithm 2 if hyperparameters and lower-level parameters (δ and θ, respectively, in Algorithm 3) are contained within a single vector (θ in Algorithm 2). We present the hierarchical algorithm separately because the distinction between the levels of parameters is an important one which will become critical with more complicated models.

The extension of Algorithms 2–3 comes at a high computational cost. One serious problem with Algorithm 3 is the time required to obtain the posterior estimates, which was around three days. However, several new algorithms have recently been developed (Bazin et al., 2010; Turner & Sederberg, submitted for publication; Turner & Van Zandt, submitted for publication), that attenuate the computation time for hierarchical models.

While this and the previous examples demonstrate the ability of the ABC approach to recover true posterior distributions, all of these posteriors could be easily recovered using standard likelihood-based techniques such as MCMC. We now turn our attention to a popular psychological model of episodic recognition memory, the Retrieving Effectively from Memory model (REM; Shiffrin & Steyvers, 1997). REM is a simulation model without an explicit likelihood.[4] This model serves as our final example.

[4] An unpublished manuscript by Montenegro, Myung, and Pitt (2011) derives the likelihood for REM. The likelihood is very complex and we will not discuss it here.

Given data and model $Y \sim \text{Model}(\delta, \theta_{1:S})$ for S subjects, set of tolerance thresholds $\epsilon_{1:t}$, hyperprior distribution $\pi_H(\delta)$, and lower-level prior distribution $\pi_L(\theta|\delta)$:
At iteration t=1,
for $1 \leq i \leq N$ **do**
  while $\rho(X, Y) > \epsilon_1$ **do**
    Sample δ^* from the hyper prior: $\delta^* \sim \pi_H(\delta)$
    Generate $\theta^*_{1:S}$ given the hyperparameter δ^*: $\theta^*_{1:S} \sim \pi_L(\theta|\delta^*)$
    Generate data X: $X \sim \text{Model}(\delta^*, \theta^*_{1:S})$
    Calculate discrepancy $\rho(X, Y)$
  end while
  Set $\delta_{i,1} \leftarrow \delta^*$, $\theta_{i,1,1:S} \leftarrow \theta^*_{1:S}$, and $w_{i,1} \leftarrow 1/N$
end for
Set $\sigma_1^2 \leftarrow 2\text{Var}(\delta_{1:N,1})$
for $2 \leq t \leq T$ **do**
  for $1 \leq i \leq N$ **do**
    while $\rho(X, Y) > \epsilon_t$ **do**
      Sample δ^* from the previous iteration $\delta^* \sim \delta_{1:N,t-1}$ with probabilities $w_{1:N,t-1}$
      Perturb δ^* by sampling $\delta^{**} \sim N(\delta^*, \sigma^2_{t-1})$
      Generate $\theta^*_{1:S}$ given the hyperparameter δ^{**}: $\theta^*_{1:S} \sim \pi_L(\theta|\delta^{**})$
      Generate data X: $X \sim \text{Model}(\delta^{**}, \theta^*_{1:S})$
      Calculate discrepancy $\rho(X, Y)$
    end while
    Set $\delta_{i,t} \leftarrow \delta^{**}$, $\theta_{i,t,1:S} \leftarrow \theta^*_{1:S}$, and $w_{i,t} \leftarrow \dfrac{\pi_H(\delta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q_f\left(\delta_{j,t-1}|\delta_{i,t}, \sigma_{t-1}\right)}$.
  end for
  Set $\sigma_t^2 \leftarrow 2\text{Var}(\delta_{1:N,t})$
end for

Fig. 5. A hierarchical ABC PMC algorithm to estimate the posteriors of the hyperparameters δ and the individual-level parameters θ given data Y.

7. Retrieving Effectively from Memory (REM)

The REM model can be used to explain performance in a number of episodic memory tasks. In this section, we will focus on recognition memory. In a recognition memory task, a subject is given a list of study items (e.g., words) during a study phase and is instructed to commit them to memory. After the study phase, the subject might perform some filler task, such as completing a puzzle. Following these two phases is a test phase. During the test phase, a subject is presented with a "probe" item and asked to respond either "old", meaning that the subject believes the probe was on the previously studied list, or "new", meaning that the subject believes the probe was not on the previously studied list. The probe word could have been on the previously studied list (in which case it is a "target") or it could be a new word (in which case it is a "distractor").

Given the two possible types of probes and the two possible types of responses, there are four possible stimulus–response outcomes on each trial. We focus on hits and false alarms. A hit occurs when a target is presented and the subject responds "old", and a false alarm occurs when a distractor is presented and the subject incorrectly responds "old". The hit rates can be plotted as a function of the false alarm rates, producing the receiver operating characteristic (ROC; e.g., Egan, 1958; Green & Swets, 1966).

At the time of REM's inception, there were a number of regularities in recognition memory data that were not easily explained by the then-current memory models (see Glanzer, Adams, Iverson, & Kim, 1993, for a review). These regularities included the absence of a list strength effect, the mirror effect and the slope of the ROC curve. The list strength effect is the finding that strengthening a subset of the study list (e.g., presenting some items more often than others during study) influences memory for the remaining (nonstrengthened) items in the list. The list strength effect is not evident in most recognition memory experiments. The mirror effect occurs when two types of items with different recognition rates are presented, such as high- and low-frequency words. More easily recognizable items (e.g., low-frequency words) show both higher hit rates and lower false alarm rates than less easily recognizable items (e.g., high-frequency words). Finally, the ROC curves constructed from hit and false alarm rates indicate that the variance of perceived memory strength for targets is greater than that of distractors. This difference in the variance stays fairly constant over manipulations such as word frequency, list length and list strength (e.g., Egan, 1958; Ratcliff, McKoon, & Tindall, 1994; Ratcliff, Sheu, & Gronlund, 1992). In developing the REM model, Shiffrin and Steyvers (1997) attempted to explain these regularities within a single framework.

REM is a global memory model, which means that recognition responses are based on a calculation of familiarity, which in turn is based on the representation of all the items on the study list. Each item is assumed to be composed of a list of features. The number of features w for each item is assumed to be equal, and each item

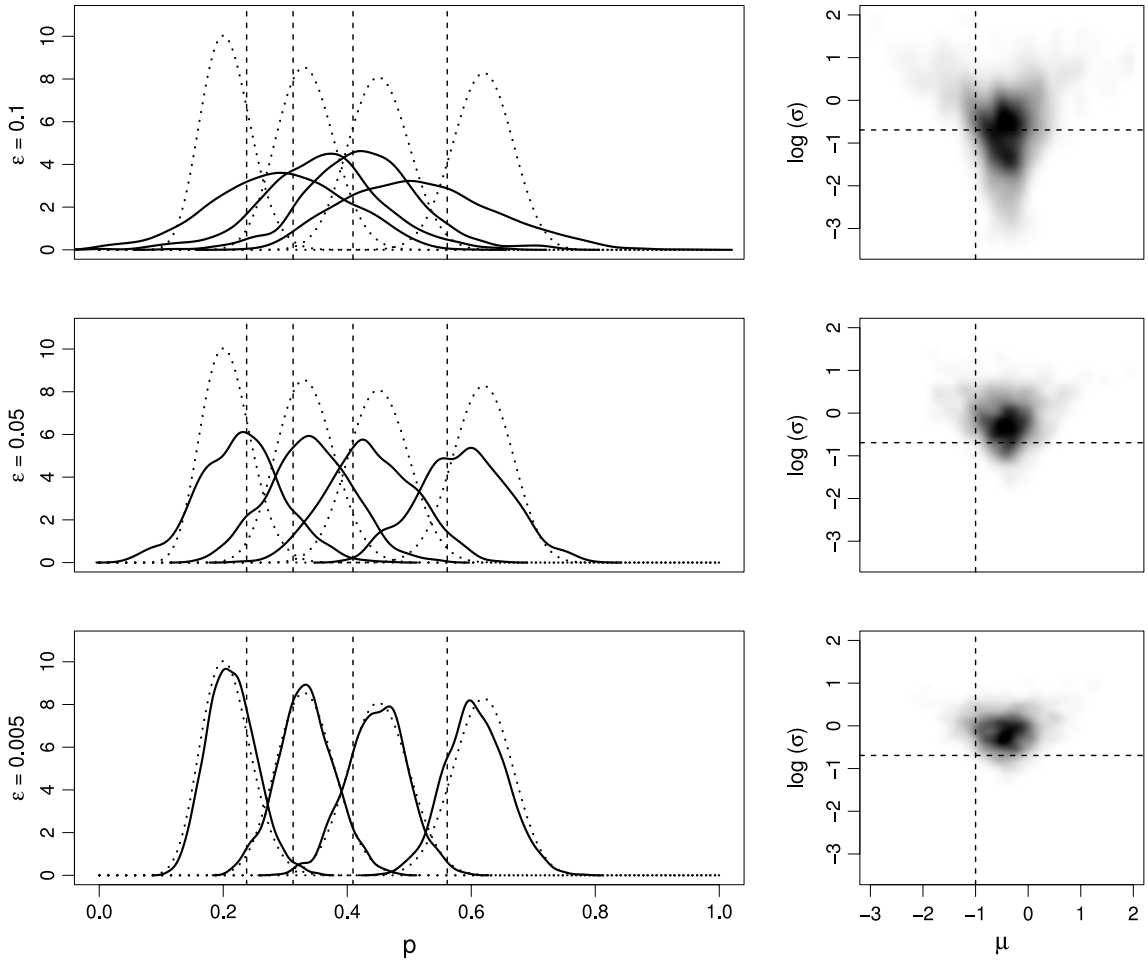

Fig. 6. The posterior distributions for the probability of a correct response for four subjects (left panel; solid lines) at three levels of ϵ_t (rows). The true posterior distributions are shown by the dashed distributions and the true sampled values are shown by the vertical lines (left panel). The right panel shows contours of the approximate joint posterior distribution for the hyperparameters μ and $\log(\sigma)$.

is stored as a vector called a "trace". Although the features of each item are assumed to have some psychological interpretation (such as the extent to which the item "bear" is associated with the concept "fur"), the values for each feature (e.g., "fur") are generated randomly. In particular, the features follow a geometric distribution, such that the probability that feature K equals value k is given by

$$P(K = k) = (1 - g)^{k-1} g,$$

where k takes on values in $\{1, 2, \ldots, \infty\}$ and the parameter $g \in (0, 1)$ is called the environmental base rate.

To understand the role that g plays, consider the difference in recognition performance between low-frequency and high-frequency words. The value of g is assumed to be higher for high-frequency words than for low-frequency words. Because the variance of the feature value K is $(1 - g)/g^2$, increasing g will result in smaller variance. Thus, high-frequency words will have more common features (K values that are equal) than low-frequency words and the individual features will be less diagnostic (i.e., harder to recognize). Furthermore, when g increases, the mean of K, $1/g$, decreases, resulting in a drop in overall discriminability, which we discuss below.

During study, the features of an item from the study list are copied to a memory trace. This copying process is both error-prone and incomplete. The item representation in the trace is initially empty, consisting entirely of zeros for each feature. The copying process operates in two steps. First, a feature is copied into the trace with probability u. Thus, with probability $1 - u$, the feature will remain empty. If the feature is copied, it may be copied correctly with probability c or it may be replaced with a random value. If the feature is replaced, its value will be drawn again from a geometric distribution with parameter g.[5] This process is repeated over all features of all studied items, resulting in an "episodic matrix", the dimensions of which are determined by w, the number of features, and the number of items n on the study list.

At test, when a probe item (again consisting of a vector of w features) is presented, the probe is compared to each trace in the episodic matrix. Following the notation in Shiffrin and Steyvers (1997), we let n_{jq} be the number of nonzero mismatching ("q"-type) features in the jth trace, and n_{ijm} be the number of nonzero matching ("m"-type) features in the jth trace with a value of i. Then, the similarity λ_j of the jth trace is

$$\lambda_j = (1 - c)^{n_{jq}} \prod_{i=1}^{\infty} \left[\frac{c + (1 - c) g (1 - g)^{i-1}}{g (1 - g)^{i-1}} \right]^{n_{ijm}}. \tag{7}$$

These similarities are then averaged across traces to produce the overall familiarity Φ of the probe item:

$$\Phi = \frac{1}{n} \sum_{j=1}^{n} \lambda_j, \tag{8}$$

[5] Although this parameter could vary over subjects, it is common to set it equal to the environmental base rate parameter. When this assumption is made, the model is called "fully informed" (Criss & McClelland, 2006).

where n is the number of traces in the episodic matrix. The familiarity Φ is a likelihood ratio: the probability that the probe is a target divided by the probability that the probe is a distractor. Once Φ has been computed, a Bayesian decision rule is used such that if $\Phi > 1$, then the probability that the probe is a target is higher, and the model elicits an "old" response. Otherwise, it elicits a "new" response.

It is not obvious that we can write down an expression for the probability of responding "old" or "new" as a function of the model parameters g, w, c, and u—the likelihood is not available analytically (but see Footnote 4). Estimates of REM's parameters have been obtained by "hand-held" fits in which parameter values have been adjusted manually over a restricted range (Shiffrin & Steyvers, 1997), or by simulating the model and using least-squares procedures that rely on the match between simulated and observed data (e.g., Malmberg et al., 2004). These procedures severely limit the extent to which inference can be made about the parameters, in particular, how these parameters vary with changes in experimental conditions. The ABC approach allows full Bayesian inference despite the lack of an expression for the REM likelihood.

7.1. The model

Our goal is to make inferences about the parameters g, u, and c for a single simulated subject in a recognition memory experiment over two list-length conditions. In two study phases, the subject sees a 10- and a 20-item word list in the short and long list conditions, respectively. The test lists consist of the entire previously-studied list plus 10 or 20 distractor items for the short and long list conditions, respectively. For the purposes of this demonstration, we will not use a hierarchical model, but see Turner, Dennis, and Van Zandt (manuscript in preparation) for a hierarchical REM model fit to the data of Dennis, Lee, and Kinnell (2008).

We simulated REM in three stages. After selecting values for g, u and c, we generated a stimulus set using the parameter g. Next, we filled in the episodic matrix during the study phase using the parameters g, u and c. Finally, we completed the test phase by using the same parameters g, u and c and Eq. (7). Using this three-step procedure allows the posteriors to reflect variance from both the stimulus set and the memory process.

Each of the parameters in REM are probabilities, bounded by zero and one, which makes selecting the priors straightforward. Because REM has never been fit in a Bayesian framework, we have no reason to believe that the parameters are located at any particular point in the parameter space. Therefore, we use noninformative priors that weigh equally all of the values in the set (0, 1), that is,

$$g, u, c \sim \text{Beta}(1, 1).$$

We use the same parameter values over each condition of the experiment; the only quantity that changes is n, the size of the study list.

7.2. Estimating the posterior

The data we observe in a recognition memory experiment are the numbers of hits and false alarms across the different conditions. The numbers of hits Y_{HIT} and false alarms Y_{FA} follow binomial distributions. More specifically, for list-length condition j, $Y_{j,\text{ HIT}} \sim \text{Bin}(n_{j,\text{ OLD}}, p_{\text{HIT}})$ and $Y_{j,\text{ FA}} \sim \text{Bin}(n_{j,\text{ NEW}}, p_{\text{FA}})$. The likelihood of the joint event $(Y_{j,\text{ HIT}}, Y_{j,\text{ FA}})$ is then the product of these two binomial probabilities (see Turner et al., manuscript in preparation). The difficulty with REM and other simulation-based memory models is that the probabilities p_{HIT} and p_{FA}, which are functions of the model parameters, are not easily determined (but see again Footnote 4 and also Myung, Montenegro, & Pitt, 2007).

Using again the ABC PMC algorithm (Algorithm 2), we set

$$\rho(X, Y) = \frac{1}{2C}\left[\sum_{j=1}^{C}\left|(X_{j,\text{ FA}} - Y_{j,\text{ FA}})/N_{\text{NEW}}\right| + \sum_{j=1}^{C}\left|(X_{j,\text{ HIT}} - Y_{j,\text{ HIT}})/N_{\text{OLD}}\right|\right], \tag{9}$$

where the number of conditions C equals 2. This $\rho(X, Y)$ is zero when the observed hit and false alarm rates equal the simulated hit and false alarm rates (and also the miss and correct rejection rates) for each condition. The maximum value of $\rho(X, Y)$ is one.

Given the range of $\rho(X, Y)$, we set $\epsilon = \{0.2, 0.1, 0.06, 0\}$. As before, this selection is determined by practical considerations. We wish to balance the number of iterations required to accept a given set of parameters with the number of iterations required to filter those parameters. The smallest value of ϵ is zero, which means we are converging to a perfect match between the simulated and observed data. We are also fitting all of the data, in contrast to our earlier exponential example where $\rho(X, Y)$ was a function of only summary statistics such as the mean or interquartile range. Obtaining a perfect match between the observed and simulated data in this way ensures the accuracy of the estimated posteriors.

We used 1000 particles to estimate the posteriors.

7.3. Results

To generate the data, we simulated 20 and 40 responses using REM for the two conditions with $n = 10$ and $n = 20$ items at study, respectively. For each condition, we set $g = 0.6$, $u = 0.335$, and $c = 0.7$. These values are shown in Fig. 7 as the dashed lines. The simulated subject had hit rates of 0.80 and 0.60 and false alarm rates of 0.40 and 0.15 for the two conditions.

Fig. 7 shows the estimated joint posterior distributions for each pair of the parameters: c versus u (left panel), g versus u (middle panel) and g versus c (right panel). Not surprisingly, the figure shows a negative curvilinear relationship between the parameters c and u, representing the trade-off between the probability u of copying a feature and the probability c of copying it correctly. To produce accurate responses, both c and u will need to be reasonably high. However, when c and u are both near one, we would expect almost perfect performance. Similarly, when c and u are both near zero, we would expect near chance performance. Our subject was neither perfect nor at chance, so the joint posterior does not extend to the upper right nor the lower left corners of the joint sample space for c and u.

There are a number of other noteworthy features of the joint posterior estimates. First, like the negative correlation between u and c, the positive correlation between c and g is quite strong, as shown in the right panel of Fig. 7. Small values of g result in large values of the feature K. These large features, which are unlikely to have arisen from an incorrect copying, contribute to very high levels of similarity when they are matched (see Eq. (7)). Assuming a fixed value of u, familiarity will also be higher if c is high, resulting in a larger number of accurately copied features. Therefore, c and g can trade off against each other, such that a given level of familiarity requires either fewer high feature values copied correctly (low g and low c), or more low feature values copied correctly (high g and high c).

Second, the correlation between u and g is not as strong as for u and c and c and g. Like the correlation between c and g, for a fixed value of c, a given level of familiarity can be produced by higher feature values with a smaller probability of being copied (low g and low u) or by lower feature values with a higher probability of

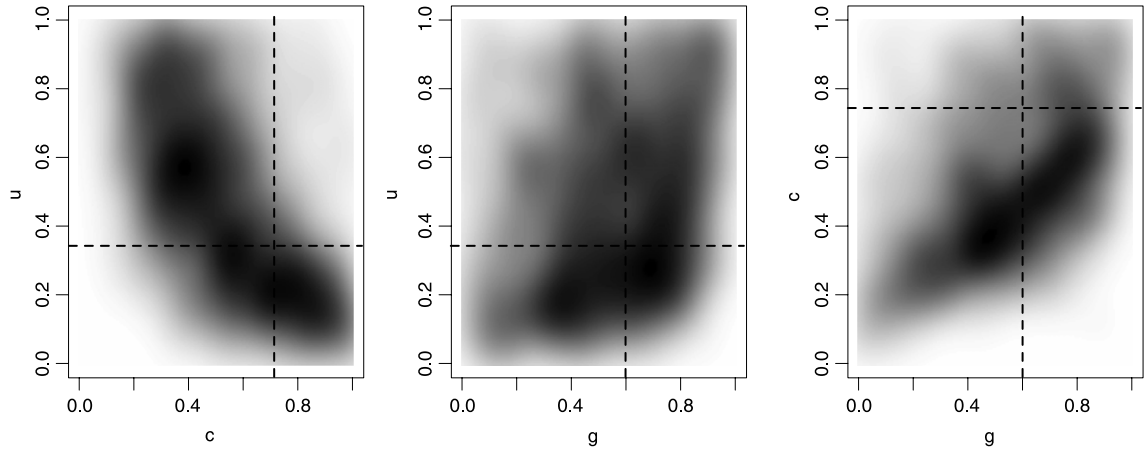

Fig. 7. The estimated joint posterior distributions for each pair of the parameters in REM: c versus u (left panel), g versus u (middle panel) and g versus c (right panel). The dashed lines show the parameter values used to generate the data.

being copied (high g and high u). However, the posterior estimates are highly variable. This reflects the extent to which four data points (the hit and false alarm rates from the short and long list conditions) can move the uninformative Beta(1, 1) priors to any particular location in the parameter space. Given the low level of information contributing to these posteriors, it is actually surprising how precise they are. The values of c, u and g that generated the data are within the equal-tail 95% credible intervals of the posterior estimates.

The total computation time for the simulation was about 45 min. As in the exponential example, the bulk of the computation time was on the last iteration, which took about 34 min. This means that we were able to obtain suitable estimates of the joint posterior distributions in about 10 min.

8. General discussion

In this tutorial, we have discussed an approach to Bayesian analysis called approximate Bayesian computation (ABC). This approach is particularly beneficial when the model of interest has a difficult or intractable likelihood function. This situation arises frequently in more complex models of cognitive processes, such as those that are found in memory, problem solving, and cognitive neuroscience research. ABC algorithms are very easy to use and, once developed, the basic algorithm can be easily applied to new models.

Although the ABC approach provides a method to circumvent intractable or ill-behaved likelihood functions, this approach is certainly not without a cost. As we mentioned in the introduction, the ABC approach is computationally more expensive than standard Bayesian samplers. However, with modern multi-core computers and graphics processing units (GPUs), computation time is becoming less of an issue. One important feature of the ABC PMC algorithm (and particle filters in general) is that particle evaluations can be completely parallelized, potentially reducing computation time even more.

We have a number of recommendations for users of the ABC algorithms we have presented in this tutorial. First, there is little need to use a rejection sampler (Algorithm 1). The ABC PMC algorithm (Algorithms 2 and 3) will be much more effective for most problems in cognitive modeling. Second, the choice of the distance function $\rho(X, Y)$ will be determined at least in part by the data to be modeled. Accuracy data can be modeled adequately using a function that compares the means, but distributional analyses such as those implemented for RT data will require a distance function based on the entire distribution. For this purpose, we recommend a Kolmogorov–Smirnov statistic or a Pearson-type discrepancy function such as that used for the chi-squared test. Pearson-type discrepancy functions could also be used for frequency data (e.g., accuracy and Likert-type ranking data).

Finally, the choice for the tolerance thresholds ϵ will depend on the selected distance function $\rho(X, Y)$, among other things. Most of the distance functions we presented for our examples were limited in their range, and so our specifications for ϵ were not hard to choose. In the case where $\rho(X, Y)$ is unbounded, such as for the exponential example, we have to be more careful.

In practice, we will have no idea what the (random) parameter values for a model will be, so we have no idea what the appropriate values for ϵ might be. However, we should have some idea of the potential range of values for ϵ given the selected distance function $\rho(X, Y)$. For example, the distance function based on the Kolmogorov–Smirnov statistic is constrained between 0 and 1. We might, then, select 0.5 for ϵ_1. After performing the first iteration of the ABC PMC algorithm, each accepted value for the parameters will have associated with it a value of $\rho(X, Y) < 0.5$. The distribution of these $\rho(X, Y)$ values can then be inspected to determine the next value for ϵ.

Given monotonic decreasing ϵ_t, as t gets large, the variance of $\rho_t(X, Y)$ will approach 0. For continuous measures, we could continue iterating, choosing ϵ_t to be smaller than ϵ_{t-1}, until the software eventually rounds ϵ_t to 0. In practice, this may be very inefficient. Instead, we recommend that iterations end when the variance of $\rho(X, Y)$ at iteration t reaches some sufficiently small value. We have found this method of determining the ϵ values useful in other investigations of more complicated models (e.g., Turner et al., manuscript in preparation).

We have applied these methods in our own work, and found that we are able to fit models and perform Bayesian analyses in areas where parameter estimation is traditionally very difficult (e.g., Turner et al., manuscript in preparation). Furthermore, this method provides opportunities to explore models that are currently neglected (or perhaps avoided) because of their computational complexity and the associated difficulties encountered during attempts to estimate their parameters. One example of this is the neurophysiologically plausible leaky competing accumulator model (Usher & McClelland, 2001) which does not have a closed-form likelihood but has the potential to explain a very wide range of choice data, including data from tasks with more than two alternative responses.

There is a tendency among mathematical modelers to view simulation-based models as less valuable than mathematical models. Mathematical models, with their closed-form expressions, provide a clear way to evaluate limits on parameters and the

influence of each parameter on the predictions. By contrast, it is not always clear what the predictions are for a simulation-based model nor which component of the model is responsible for producing a given effect. It is also more difficult to isolate variance within the components of a simulation-based model. ABC, while it does not completely eliminate all of these problems, permits researchers to choose models that, for reasons of complexity or computation, they may not have considered previously.

Acknowledgments

This work was made possible by NSF grants BCS-0738059 and SES-1024709. Portions of this work were presented at the 43rd Annual Meeting of the Society for Mathematical Society, Portland and the 2011 Annual Context and Episodic Memory Symposium, Philadelphia, and were submitted by Brandon Turner in partial fulfillment of the requirements for the Ph.D. in Psychology at the Ohio State University.

References

Abelson, R. P. (1995). *Statistics as principled argument*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Barthelme, S., & Chopin, N. (2011). ABC-EP: expectation propagation for likelihood-free Bayesian computation. In *Proceedings of the 28th international conference on machine learning. Bellevue, WA*.

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.

Beaumont, M. A., Zhang, W., & Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, *162*, 2025–2035.

Blum, M. G. B., & François, O. (2010). Non-linear regression models for approximate Bayesian computation. *Statistics and Computing*, *20*, 63–73.

Bortot, P., Coles, S. G., & Sisson, S. A. (2007). Inference for stereological extremes. *Journal of the American Statistical Association*, *102*, 84–92.

Brown, S., & Steyvers, M. (2009). Detecting and predicting changes. *Cognitive Psychology*, *58*, 49–67.

Cappé, O., Guillin, A., Marin, J. M., & Robert, C. P. (2004). Population Monte Carlo. *Journal of Computational and Graphical Statistics*, *13*, 907–929.

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

Criss, A. H., & McClelland, J. L. (2006). Differentiating the differentiation models: A comparison of the retrieving effectively from memory model (REM) and the subjective likelihood model (SLiM). *Journal of Memory and Language*, *55*, 447–460.

Del Moral, P., Doucet, A., & Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society. Series B*, *68*, 411–432.

Dennis, S., Lee, M., & Kinnell, A. (2008). Bayesian analysis of recognition memory: the case of the list-length effect. *Journal of Mathematical Psychology*, *59*, 361–376.

Douc, R., Guillin, A., Marin, J.-M., & Robert, C. (2007). Convergence of adaptive mixtures of importance sampling schemes. *Annals of Statistics*, *35*, 420–448.

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

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

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

Glanzer, M., Adams, J. K., Iverson, G. J., & Kim, K. (1993). The regularities of recognition memory. *Psychological Review*, *100*, 546–567.

Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics*. New York: Wiley Press.

Heathcote, A., Brown, S. D., & Mewhort, D. J. K. (2000). The power law repealed: the case for an exponential law of practice. *Psychonomic Bulletin and Review*, *7*, 185–207.

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.

Kruschke, J. K. (2011). *Doing Bayesian data analysis: a tutorial with R and BUGS*. Burlington, MA: Academic Press.

Kullback, S., Keegel, J. C., & Kullback, J. H. (1987). *Lecture notes in statistics*: *Vol. 42. Topics in statistical information theory*. New York: Springer-Verlag.

Lee, M. D. (2004). A Bayesian analysis of retention functions. *Journal of Mathematical Psychology*, *48*, 310–321.

Lee, M. D. (2008). Three case studies in the Bayesian analysis of cognitive models. *Psychonomic Bulletin and Review*, *15*, 1–15.

Lee, M. D. (2011). Special issue on hierarchical Bayesian models. *Journal of Mathematical Psychology*, *55*, 1–118.

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* (19th ed.) (pp. 809–815). Cambridge, MA: MIT Press.

Leuenberger, C., & Wegmann, D. (2010). Bayesian computation and model selection without likelihoods. *Genetics*, *184*, 243–252.

Liu, C. C., & Aitkin, M. (2008). Bayes factors: prior sensitivity and model generalizability. *Journal of Mathematical Psychology*, *52*, 362–375.

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.

Marjoram, P., Molitor, J., Plagnol, V., & Tavare, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States*, *100*, 324–328.

Matzke, D., & Wagenmakers, E.-J. (2009). Psychological interpretation of the ex-Gaussian and shifted wald parameters: a diffusion model analysis. *Psychonomic Bulletin and Review*, *16*, 798–817.

Montenegro, M., Myung, J. I., & Pitt, M. A. (2011). REM integral expressions. Unpublished Manuscript.

Myung, I. J. (2003). Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, *47*, 90–100.

Myung, J. I., Montenegro, M., & Pitt, M. A. (2007). Analytic expressions for the BCDMEM model of recognition memory. *Journal of Mathematical Psychology*, *51*, 198–204.

Nilsson, H., Rieskamp, J., & Wagenmakers, E.-J. (2011). Hierarchical Bayesian parameter estimation for cumulative prospect theory. *Journal of Mathematical Psychology*, *55*, 84–93.

Nosofsky, R. M. (1986). Attention, similarity, and the identification–categorization relationship. *Journal of Experimental Psychology: General*, *115*, 39–57.

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.

Oravecz, Z., Tuerlinckx, F., & Vandekerckhove, J. (2011). A hierarchical latent stochastic differential equation model for affective dynamics. *Psychological Methods*, *16*, 468–490.

O'Reilly, R. C. (2001). Generalization in interactive networks: the benefits of inhibitory competition and Hebbian learning. *Neural Computation*, *13*, 1199–1242.

O'Reilly, R. C. (2006). Biologically based computational models of cortical cognition. *Science*, *314*, 91–94.

O'Reilly, R., & Munakata, Y. (Eds.). (2000). *Computational explorations in cognitive neuroscience: understanding the mind by simulating the brain*. Cambridge, MA: MIT Press.

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.

R Development Core Team (2008). R: a language and environment for statistical computing. R Foundation for Statistical Computing. Vienna, Austria. ISBN: 3-900051-07-0. URL: http://www.R-project.org.

Ratcliff, R. (1978). A theory of memory retrieval. *Psychological Review*, *85*, 59–108.

Ratcliff, R., McKoon, G., & Tindall, M. H. (1994). Empirical generality of data from recognition memory receiver-operating characteristic functions and implications for the global memory models. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *20*, 763–785.

Ratcliff, R., Sheu, C.-F., & Gronlund, S. D. (1992). Testing global memory models using ROC curves. *Psychological Review*, *99*, 518–535.

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

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

Rouder, J. N., & Speckman, P. L. (2004). An evaluation of the vincentizing method of forming group-level response time distributions. *Psychonomic Bulletin and Review*, *11*, 419–427.

Rubin, D. C., & Wenzel, A. E. (1996). One hundred years of forgetting: a quantitative description of retention. *Psychological Review*, *4*, 734–760.

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 and 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*, *104*, 1760–1765.

Sousa, V. C., Fritz, M., Beaumont, M. A., & Chikhi, L. (2009). Approximate Bayeisian 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.

Turner, B. M., Dennis, S., & Van Zandt, T. (2011). Bayesian analysis of memory models. Manuscript in preparation.
Turner, B. M., & Sederberg, P. B. (2012). Approximate Bayesian computation with differential evolution. Manuscript (submitted for publication).
Turner, B. M., & Van Zandt, T. (2012). Hierarchical approximate Bayesian computation. Manuscript (submitted for publication).
Usher, M., & McClelland, J. L. (2001). On the time course of perceptual choice: the leaky competing accumulator model. *Psychological Review*, *108*, 550–592.
Vandekerckhove, J., Tuerlinckx, F., & Lee, M. D. (2011). Hierarchical diffusion models for two-choice response time. *Psychological Methods*, *16*, 44–62.
Wagenmakers, E.-J., van der Maas, H. J. L., & Grasman, R. P. P. P. (2007). An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin and Review*, *14*, 3–22.
Wegmann, D., Leuenberger, C., & Excoffier, L. (2009). Efficient approximate Bayesian computation coupled with Markov chain Monte Carlo without likelihood. *Genetics*, *182*, 1207–1218.
Wetzels, R., Vandekerckhove, J., Tuerlinckx, F., & Wagenmakers, E.-J. (2008). Bayesian parameter estimation in the Expectancy Valence model of the Iowa gambling task. *Journal of Mathematical Psychology*, *54*, 14–27.
Wilkinson, R. D. (2012). Approximate Bayesian computation (ABC) gives exact results under the assumption of model error. Manuscript (submitted for publication).
Wixted, J. T. (1990). Analyzing the empirical course of forgetting. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *16*, 927–935.
Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, *466*, 1102–1104.