SUPPLEMENTARY MATERIALS
Likelihood-free Bayesian Analysis of Memory Models

Brandon M. Turner
Department of Psychology
Stanford University

Simon Dennis, Trisha Van Zandt
Department of Psychology
The Ohio State University

Address correspondence to:
Brandon M. Turner
Department of Psychology
Stanford University
turner.826@gmail.com
Outline

In these supplementary materials, we present the technical details used in our article. We begin by discussing the models BCDMEM and REM in greater detail. Next, we provide a brief summary of some fundamentals of Bayesian statistics. We then discuss the basic ideas behind approximate Bayesian computation (ABC), and verify that the ABC approach can accurately recover the posterior distribution of the BCDMEM model in a simulation study. In the final two sections, we present the model details for Studies 1 and 2.

Two Models of Recognition Memory

In this section we present the details of the BCDMEM and REM models. For each model, we discuss how they explain recognition memory performance, and the parameters they require to do so.

The Bind Cue Decide Model of Episodic Memory

BCDMEM postulates that when a probe item is presented for recognition, the contexts in which that item was previously experienced are retrieved and matched against a representation of the context of interest. For example, suppose that you are trying to remember whether you ate toast for breakfast yesterday. You would make a false alarm if you say “yes” when you didn’t eat toast. BCDMEM argues that such false alarms occur primarily because you have eaten toast for breakfast many times, and when you try to make a decision about toast the contexts of these other breakfasts are retrieved. Whether you say “yes” or “no” to toast will depend on the degree of overlap between the reinstated representation of the context of interest (breakfast yesterday) and the set of contexts to which “toast” has been associated. Thus, the other breakfasts in which you ate toast will generate interference.
BCDMEM consists of two layers of nodes. The input layer represents items such that each node corresponds to one item – a local code. The output layer represents contexts as a pattern of activation over a set of nodes – a distributed code. When an item is studied, a random context pattern of length $v$ for that study episode is constructed by turning on nodes in the output layer with probability $s$ (the context sparsity parameter). The node in the input layer representing the studied item is connected to the nodes in the output layer through associative weights. These connections are established during study by connecting the active nodes on the input and the context layers with probability $r$ (the learning rate).

During the test phase, the presentation of a probe results in the activation of the corresponding node at the input layer. This node then activates a distributed pattern of activity at the output layer that includes both the pre-experimental contexts in which the item has been encountered, which activate nodes with probability $p$ (the context noise parameter), and the context created during study if the item appeared on the study list. This pattern is called the retrieved context vector.

The presentation of a probe also causes the reconstruction of a representation of the study list context called the reinstated context vector. The reconstruction process is unlikely to be completely accurate: nodes that were active during the study phase may become inactive with probability $d$ (the contextual reinstatement parameter).

A subject bases his or her “old” decision on a comparison between the activation patterns of the reinstated and retrieved context vectors. As in Dennis and Humphreys (2001), we let the $i$th node in the reinstated context vector be denoted by $c_i$ and the $j$th node in the retrieved context vector be denoted by $m_j$. Both $c_i$ and $m_j$ are binary, indicating that the nodes $i$ and $j$ are either inactive or active, so $c_i = 0$ or 1 and $m_j = 0$ or 1, respectively.

To evaluate the match between the reinstated and retrieved context vectors, we let $n_{i,j}$ denote the number of nodes in the reinstated context vector that are in state $i$ (0 or 1)
at the same time that the nodes in the retrieved context vector are in state $j$ (0 or 1). For example, $n_{1,1}$ denotes the number of nodes that are simultaneously active in both the reinstated and retrieved context vectors. Similarly, $n_{0,1}$ denotes the number of nodes that are inactive in the reinstated context vector but active in the retrieved context vector. We can then compute the probability that a probe item is a target and contrast that with the probability that a probe item is a distractor by computing a likelihood ratio given by

$$L(n \mid \theta) = \left[ \frac{1 - s + ds(1 - r)}{1 - s + ds} \right]^{n_{0,0}} \left[ \frac{r + p - rp}{p} \right]^{n_{1,1}} (1 - r)^{n_{1,0}} \times \left[ \frac{p(1 - s) + ds(r + p - rp)}{p(1 - s) + dsp} \right]^{n_{0,1}},$$

where $\theta = \{d, p, r, s, v\}$ is the set of parameters for BCDMEM and $n$ represents the vector of frequencies of node pattern matches and mismatches, so $n = \{n_{0,0}, n_{0,1}, n_{1,0}, n_{1,1}\}$.

Equation 1 is the probability that an item is a target divided by the probability that the item is a distractor. BCDMEM uses a decision rule such that the outcome (target or distractor) with the highest probability is selected. This means that the model will make an “old” response when $L(n \mid \theta) > 1$, otherwise it will make a “new” response.

The Retrieving Effectively from Memory Model

While many recent extensions of REM have included contextual noise (e.g., Lehman & Malmberg, 2012), in this article we use the (original) pure item noise variant (e.g., Shiffrin & Steyvers, 1997). REM is a global memory model, which means that recognition responses are based on an overall calculation of memory strength arising from a comparison between a probe item and all the items stored in memory. In contrast to the local representation used by BCDMEM, each stored item is assumed to be composed of a vector of $w$ features. The features of each item have some psychological interpretation (such as the extent to which the item “snail” is associated with the concept “slimy”). The values for each feature (e.g., “slimy”) are assumed to follow a geometric distribution, such that the probability that feature $K$ equals value $k$ is given by

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

The value of $g$ is assumed to be higher for high-frequency words than for low-frequency words. Because the mean and variance of the feature value $K$ are $1/g$ and $(1 - g)/g^2$, respectively, increasing $g$ will result in a smaller mean and a smaller variance. Thus, high-frequency words will be represented by traces with smaller feature values and will tend to have more features in common ($K$ values that are equal) than low-frequency words. This means that the value on one individual feature is less likely to help discriminate between a high-frequency target and a high-frequency distractor.

During study, the vector of features of an item from the study list are copied to a memory trace. This copying process is both error-prone and incomplete: features may or may not be copied into the trace, and, if copied, they may not be copied correctly. The memory trace is initially empty, consisting entirely of zeros for each feature. Each feature is copied into the trace with probability $u$, and, with probability $1 - u$, the feature in the trace will remain empty. If the feature is copied, it will be copied correctly with probability $c$. Otherwise, a random feature value is sampled from a geometric distribution with parameter $g$.\(^1\) After all the features of all items are copied into the trace, the result is 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 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 $j$th trace, and $n_{ijm}$ be the number of nonzero matching (“m”-type) features in the $j$th trace with a value of $i$. Then,

\(^1\)In the original formulation of REM, item features were generated from a geometric distribution with a word-frequency parameter $g_H$. Then, a different environmental parameter $g$ was used to generate feature values when features were stored incorrectly (see Equation 2). It is common to set $g_H = g$, which we do in this article.
the similarity $\lambda_j$ of the $j$th trace is

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

(2)

The average similarity across traces is then the overall memory strength $\Phi$ of the probe item given by

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

(3)

where $n$ is the number of traces in the episodic matrix. The memory strength $\Phi$ can be seen as a likelihood ratio: the probability that the probe is a target divided by the probability that the probe is a distractor. The optimal decision rule is then to respond “old” if $\Phi > 1$ and “new” otherwise.

Fundamentals of Bayesian Inference

One reason that the Bayesian approach is gaining traction in psychology is because potentially interesting hypotheses about the model parameters $\theta$ can be evaluated in the context of a theoretically meaningful structure by looking at the posterior distributions of those parameters.

To compute the posterior distribution we need two things. First, we must decide on a prior distribution $\pi(\theta)$ for $\theta$ that reflects our current understanding of $\theta$. If we know nothing or are unwilling to be specific about $\theta$, we can choose a prior distribution that is vague or noninformative. A noninformative prior places equal probability on all possible values in the parameter space. By contrast, an informative prior represents our a priori beliefs about the possible values of $\theta$. Because the prior represents our subjective beliefs as a probability distribution, we are always able to write down a prior, and this prior is likely to vary from one researcher to another.

Second, we must write down the probability function for the observed data given specific values for the parameters $\theta$. This function is the likelihood function, and it is determined by the model of interest. Formally, let the observed data be represented by the sample $Y = \{Y_1, Y_2, \ldots, Y_n\}$ and suppose that the model dictates that the probability
density function for the \(i\)th observation \(Y_i\) is \(f(y \mid \theta)\). If the \(n\) observations in the sample are independent and identically distributed, then the likelihood function is defined by

\[
L(\theta \mid Y_1 = y_1, Y_2 = y_2, \ldots, Y_n = y_n) = \prod_{i=1}^{n} f(y_i \mid \theta).
\] (4)

(Maximum likelihood estimation uses Equation 4 as the discrepancy function, where estimates of \(\theta\) are selected so that they maximize the probability of observing the data that were obtained.)

The posterior distribution \(\pi(\theta \mid Y)\) is an update of the prior distribution \(\pi(\theta)\) after having observed the data \(Y\). Using Bayes’ theorem, we write the posterior as

\[
\pi(\theta \mid Y) = \frac{L(\theta \mid Y)\pi(\theta)}{\int L(\theta \mid Y)\pi(\theta)d\theta}.
\] (5)

The integral denominator \(\int L(\theta \mid Y)\pi(\theta)d\theta\) is called the prior predictive distribution (Vanpaemel, 2010), and is a constant with respect to \(\theta\). Thus, we can rewrite Equation 5 as

\[
\pi(\theta \mid Y) \propto L(\theta \mid Y)\pi(\theta),
\] (6)

where the symbol “\(\propto\)” indicates that the posterior is proportional to the product of the likelihood and the prior.

The fact that the prior predictive distribution can be very difficult to compute prevented the widespread adoption of Bayesian methods until only very recently, with the advent of powerful computers and sophisticated estimation techniques such as Markov chain Monte Carlo (MCMC; e.g., Robert & Casella, 2004). Although most interesting problems do not permit us to write down exact posterior probabilities for \(\theta\), these estimation techniques use Equation 6 to obtain samples of \(\theta\) from the posterior. From these samples we can then estimate the probabilities of different hypotheses (such as the probability that \(\theta > 0\)) by computing the proportion of obtained samples that are consistent with those hypotheses. We can also compute “credible sets,” the Bayesian counterparts to confidence intervals and, if necessary, compute point estimates of the
parameters from measures of posterior central tendency. All these inferences are derived
directly from the model of interest through the likelihood $L(\theta|Y)$.

Simulation-based models, which are becoming more and more prevalent in cognitive
psychology and neuroscience, are often so complex that there is no way to derive the
likelihood function $L(\theta|Y)$. Until now, parameter estimation for these models (if such
estimation is done at all) usually relies on approximate least-squares methods, and there
has been no possibility of performing fully Bayesian analyses. Indeed, because our ability
to explore the parameter spaces of these models has been restricted, they perhaps have
not been as well understood as models with explicit likelihoods. Approximate Bayesian
computation (ABC) is a relatively new technique that allows fully Bayesian analysis while
circumventing the evaluation of the likelihood by simulating the model. Equipped with
only a prior distribution and an ABC algorithm, we can now estimate the posterior
distribution of the parameters of any model, including models that may not be
simulation-based, but whose likelihoods may be merely difficult to compute. In the next
section, we introduce this approach to posterior estimation.

Approximate Bayesian Computation

Likelihood-free methods for Bayesian inference (Pritchard, Seielstad, Perez-Lezaun,
& Feldman, 1999; Marjoram, Molitor, Plagnol, & Tavare, 2003; Sisson, Fan, & Tanaka,
2007; Beaumont, Cornuet, Marin, & Robert, 2009; Toni, Welch, Strelkowa, Ipsen, &
Stumpf, 2009; Bazin, Dawson, & Beaumont, 2010) have been very influential in genetics
and are only now being applied to models in psychology (Turner & Van Zandt, 2012,
2011; Turner & Sederberg, 2012, 2013). In particular, ABC is an easy-to-use, flexible
method for approximating the posterior distributions for any model, but the power of the
method lies in its ability to estimate the posterior distributions of the parameters of
models with undefined or intractable likelihood functions (for a tutorial, see Turner &
Van Zandt, 2012).

There are several classes of ABC algorithms, including rejection sampling (Pritchard
et al., 1999; Beaumont, Zhang, & Balding, 2002), MCMC (Marjoram et al., 2003), and sequential Monte Carlo (Sisson et al., 2007; Beaumont et al., 2009; Toni et al., 2009). These algorithms, just like the non-ABC algorithms of standard Bayesian analysis, work by first generating “proposals” for $\theta$, proposals that may or may not have come from the desired posterior. These proposals are then filtered; some are kept and the others are rejected. The proposals that are kept form a sample of parameter values drawn (presumably) from the desired posterior distribution. Different algorithms filter the proposals in different ways.

As an example, consider a simple ABC rejection sampler shown in Algorithm 1. In this algorithm, proposal values for $\theta$ are drawn at random from the prior $\pi(\theta)$. Each proposal $\theta^*$ is then used to generate data $X$ from the model. That is, the model (which we call Model) is simulated, so $X \sim \text{Model}(\theta^*)$. To decide whether or not $\theta^*$ comes from the desired posterior $\pi(\theta|Y)$, we compare the simulated data $X$ to the observed data $Y$ by way of a discriminant function $\rho(X,Y)$. If the simulated data $X$ is similar enough to $Y$, producing a $\rho(X,Y)$ less than some tolerance threshold $\epsilon$, then we assume that the proposal $\theta^*$ must have some nonzero probability of coming from the posterior distribution $\pi(\theta|Y)$. Hence, we accept $\theta^*$ as a sample from the posterior. Otherwise, if $\rho(X,Y)$ is too large, we discard $\theta^*$ and repeat the process until we obtain a desired number $T$ of samples of $\theta$.

The ABC method is similar to approximate least squares estimation because we are evaluating parameter proposals by way of a discriminant function $\rho(X,Y)$ that compares the observed data to the simulated data. However, the intent of ABC is not to minimize $\rho(X,Y)$, but instead to obtain an estimate of the posterior distribution of $\theta$. This estimate is based on the assumption that

$$\pi(\theta|Y) \approx \pi(\theta|\rho(X,Y) \leq \epsilon)$$

(7)

for some small $\epsilon$ greater than or equal to zero. Appropriate values for $\epsilon$ will depend on the form of $\rho(X,Y)$ as well as the nature of the data $Y$. For example, continuous
1: for \(1 \leq t \leq T\) do
2: \hspace{1em} while \(\rho(X,Y) > \epsilon\) do
3: \hspace{2em} Sample \(\theta^*\) from the prior: \(\theta^* \sim \pi(\theta)\)
4: \hspace{2em} Generate data \(X\) using the model: \(X \sim \text{Model}(\theta^*)\)
5: \hspace{2em} Calculate \(\rho(X,Y)\)
6: \hspace{1em} end while
7: Store \(\theta_t \leftarrow \theta^*\)
8: end for

*Figure 1.* An ABC rejection sampling algorithm for estimating the posterior distribution of \(\theta\).

measurements will be impossible to reproduce exactly (i.e., produce data \(X\) such that \(\rho(X,Y) = 0\)), so we will need to decide how large \(\epsilon\) can be to obtain a good approximation to the desired posterior (see Turner & Van Zandt, 2012, for a discussion of this and other issues).

The simple rejection sampler presented in Figure 1 is computationally inefficient, making it difficult to implement for many interesting problems. For example, psychologists are often interested in modeling behavioral differences between individuals, groups of individuals, and even behavioral differences for one particular subject over different experimental conditions. Such analyses are naturally performed using hierarchical Bayesian modeling (see, e.g., Lee, 2011; Lee & Wagenmakers, 2012; Shiffrin, Lee, Kim, & Wagenmakers, 2008). However, hierarchical models are very complex: the number of parameters required is very large. We say that such models have high dimensionality, in considering their predictions as a function of their parameters.

To understand the dimensionality problem, consider that, in a hierarchical model, there is a set of individual-level parameters for each subject, plus the hyperparameters that describe or constrain the individual-level parameters. In general, the dimensionality
of the parameter space for a hierarchical model is greater than the number of model parameters times the number of subjects in the experiment – much, much higher than that for a non-hierarchical model.

The difficulties in extending the simple rejection ABC sampler to high-dimensional or hierarchical models is mostly due to the rate at which proposed values of θ must be rejected. As the number of dimensions increases, the probability of proposing a high-dimensional vector of parameters that produce simulated data X close enough to the observed data Y such that ρ(X, Y) ≤ ϵ becomes very small because the posterior is more difficult to locate in the parameter space (see Gelman, Carlin, Stern, & Rubin, 2004; Beaumont, 2010). This means that a very large number of proposals must be sampled, and simulations of the model with those proposals, before even a single proposal is accepted. When the rejection rate of proposals becomes very high, the computation time increases to such a degree that ABC is no longer practical. In addition, the stability and accuracy of the ABC approach decreases in high-dimensional spaces. These difficulties are generally referred to as the infamous “curse of dimensionality” (Beaumont, 2010).

A number of modifications to the simple rejection sampler have been proposed to overcome these problems. These include empirical Bayes methods (Pritchard et al., 1999) that inform the choice of hyperparameter prior distributions with classical estimation techniques (e.g., maximum likelihood performed on individual-level parameters obtained by ABC), breaking the estimation problem into two steps, the first for the posteriors of the hyperparameters and the second for the posteriors of the individual-level parameters (Bazin et al., 2010), and strategies for matching subjects’ data to different sets of individual level parameters (e.g., Hickerson, Stahl, & Lessios, 2006; Hickerson & Meyer, 2008; Sousa, Fritz, Beaumont, & Chikhi, 2009). Although these algorithms improve the situation somewhat, they still leave much to be desired (see Turner & Van Zandt, 2011, for a discussion).

In this article, we make use of a better approach. Writing the hyperparameters as
the vector $\phi$ and the individual-level parameters as the vector $\theta$, we can show that

$$
\pi(\phi|Y, \theta) \propto \prod_{i=1}^{S} \pi(\theta_i|\phi) \pi(\phi),
$$

(8)

where $S$ is the number of subjects and $\theta_i$ is the set of parameters specific to Subject $i$.

The priors $\pi(\phi)$ and $\pi(\theta_i|\phi)$ are known because we select them at the time that we specify the model, and so we can obtain samples directly from the posterior distribution of the hyperparameters $\phi$ using Gibbs sampling (see Gelman et al., 2004).

In addition to using Gibbs sampling, we use ABCDE (Turner & Sederberg, 2012) to generate efficient proposals. As we discussed, ABCDE uses a kernel-based weighing scheme (see Wilkinson, 2011), which is much more efficient than rejection-based samplers such as Algorithm 1. In contrast to basing the accept/reject decision for a proposal $\theta^*$ on a fixed tolerance threshold $\epsilon$, kernel-based ABC evaluates the fitness of $\theta^*$ by assigning it a weight that can be used to determine the probability with which that $\theta^*$ will be accepted.

A kernel $K(x|\delta_{ABC})$ is a non-negative, real-valued function of $x$ that is symmetric around 0 and integrates to 1. These conditions imply that $K(x|\delta_{ABC})$ is a probability density function of some random variable with mean (and median) equal to 0. Statistical applications frequently use kernel functions as a way to weigh observations in a sample as a function of their distance from a point, so that observations farther away from that point carry less weight than observations very close to that point. Like all probability density functions, the kernel $K(x|\delta_{ABC})$ has some dispersion parameter $\delta_{ABC}$ (e.g., the standard deviation), which for weighing purposes we call a tuning parameter.

For kernel-based ABC, we choose $K(x|\delta_{ABC})$ to be the probability density function of a normally-distributed random variable with mean 0 and standard deviation $\delta_{ABC}$. We then use this kernel function to weigh each proposal $\theta^*$ by how close the data $X$ are to the observed data $Y$ by calculating

$$
\psi(\theta^*|X, Y) = \prod_{j} K(\rho(X_j, Y_j)|\delta_{ABC}),
$$

(9)

for each $(X_j, Y_j)$ pair. Our discrepancy function $\rho(X, Y)$, which should be very close to 0
when \( X \approx Y \), will give high weights for values of \( \theta^* \) that produce \( X \approx Y \) and lower weights for values of \( \theta^* \) that produce \( X \) far from \( Y \).

We now present the results of a simulation study meant to test the accuracy of the ABC approach for the BCDMEM model. The equations derived in Myung, Montenegro, and Pitt (2007) make conventional Bayesian estimation of the parameters of the BCDMEM model possible and as a result, we can use these “true” estimates to evaluate the estimates obtained using likelihood-free techniques. A close agreement between the two estimates suggests that the ABC method is a legitimate approach to Bayesian estimation for this type of model.

Validating the ABC Approach

Myung et al. (2007) analytically derived two sets of expressions for the likelihood for BCDMEM. The first set of equations (i.e., Equation 10 in Myung et al.) provides the explicit likelihood function, which we will refer to as the “exact” equations. Unfortunately, the exact equations can be difficult to evaluate precisely for all values of \( \theta \). For this reason, Myung et al. also derived asymptotic expressions (i.e., Equations 15 and 16 in Myung et al.) that approximate the exact solution. We will refer to this second set of equations as the “asymptotic” equations.

The exact and asymptotic expressions for the hit and false alarm rates allow us to fully specify the likelihood function and obtain the posterior distribution for the parameter set \( \theta \) when \( v \) is fixed to some positive integer (\( v \) must be fixed or the other parameters are not identifiable; Myung et al., 2007). Because exact expressions for the likelihood function are known, BCDMEM is an excellent simulation-based memory model with which to validate the ABC approach.

To perform the simulation study, we began by generating data from BCDMEM for a single subject in a recognition memory experiment with four conditions. In each of the four conditions, the simulated subject was given a 10-item list during the study phase. At test, the subject responded “old” or “new” to presented probes according to whether it
was more likely or not that the probe was a target or distractor. The test lists consisted of 10 targets and 10 distractors.

To both generate and fit the data, we fixed the vector length $v$ at 200 and the context sparsity parameter $s$ at 0.02. We then generated 20 “old”/“new” responses for each condition using $d = 0.3$, $p = 0.5$, and $r = 0.75$. With $v$ and $s$ fixed, our goal was to estimate the joint posterior distribution for the parameters $d$, $p$, and $r$.

The Standard Bayesian Approach

To construct the Bayesian model we must compute the likelihood function and choose priors for the parameters $d$, $p$, $r$ and $s$. Beginning with the likelihood, we note that a response to a probe (a target or distractor) is a Bernoulli variable, with the probability of an “old” response given by either the exact or asymptotic equations. Therefore, assuming independent responses over trials, the number of hits in a sequence of $N_T$ targets follows a binomial distribution with parameters $N_T$ and probability of a “success” (i.e., an “old” response) equal to $P(H|\theta)$. Similarly, the number of false alarms in a sequence of $N_D$ distractors follows a binomial distribution with parameters $N_D$ and probability of a “success” equal to $P(FA|\theta)$. It follows then, that the likelihood is obtained by multiplying the two binomial densities together.

To choose the priors for the BCDMEM model, we note that each of the parameters in the model are probabilities and so they are bounded between zero and one. In ordinary circumstances we would consult existing research to inform our decisions about the priors for these parameters. However, because our data were simulated and because there are no previous Bayesian treatments of BCDMEM, there is no appropriate previous research. Consequently, we will assume non-informative priors such that each of the parameters may fall anywhere in the interval $(0,1)$. A mathematically convenient way to specify this prior is to use a beta distribution. The parameterization of the beta distribution that we use has two parameters called the sample size $\xi$ and the mean $\omega$, and has a probability
density function given by

\[
f(p|\omega, \xi) = \begin{cases} 
\frac{\Gamma(\xi)}{\Gamma(\omega \xi) \Gamma(\xi (1-\omega))} p^{\xi \omega - 1} (1 - p)^{\xi (1-\omega) - 1} & \text{if } 0 < p < 1 \\
0 & \text{otherwise.} 
\end{cases}
\]

(10)

In this function, \(\Gamma(x)\) is the gamma function, \(\omega\) takes on values in the interval \((0, 1)\), and \(\xi\) ranges over the positive real numbers. When \(\xi = 2\) and \(\omega = 0.5\), the beta distribution is equal to one over the interval from zero to one. Note that the usual parameterization of the beta distribution sets \(\alpha = \xi \omega\) and \(\beta = \xi (1 - \omega)\), so the uniform \((0, 1)\) distribution is obtained when \(\alpha = \beta = 1\). Hence, we set

\[d, p, r \sim \text{Beta}(0.5, 2).\]

We used differential evolution with Markov chain Monte Carlo (DE-MCMC) to sample from this joint posterior (see Braak, 2006; Turner, Sederberg, Brown, & Steyvers, 2012). After a burn-in period of 1,000 samples, we obtained 9,000 samples from the joint posterior and collapsed across 12 chains. We did this twice, once using the exact equations for \(P(H | \theta)\) and \(P(FA | \theta)\) and once using the asymptotic equations (see Myung et al., 2007). We used standard techniques to assess convergence of the chains (using the coda package in \text{R}; Plummer, Best, Cowles, & Vines, 2006; R Core Team, 2012).

The ABC Approach

In contrast to the standard Bayesian approach, the ABC approach does not require that we derive the likelihood function. Instead, we must choose an appropriate discriminant function \(\rho(X, Y)\), an appropriate kernel function \(K\), and a tolerance parameter \(\delta_{ABC}\).

The choice of \(\delta_{ABC}\) depends on both \(\rho(X, Y)\) and \(K\), and convenient statistics on which to base \(\rho(X, Y)\) are the hit and false alarm rates in each condition. We set

\[\rho(X, Y) = ||X - Y|| = \sum_i \sqrt{(X_i - Y_i)^2},\] the standard Euclidean distance metric. For
our kernel function, we selected a Gaussian kernel such that

$$\mathcal{K}(x|\delta_{ABC}) = \frac{1}{\sqrt{2\pi}\delta_{ABC}} \exp \left[ -\frac{x^2}{2\delta_{ABC}^2} \right], \quad (11)$$

Letting $Y_j = \{O_{j,T}, O_{j,D}\}$ for the observed data and $X_j = \{\hat{O}_{j,T}, \hat{O}_{j,D}\}$ for the simulated data in the $j$th condition, proposals $\theta^*$ were evaluated by the equation

$$\psi(\theta^*|X,Y) = \prod_{j=1}^{J} \mathcal{K} \left( \rho(\hat{O}_{j,T}, O_{j,T})|\delta_{ABC} \right) \mathcal{K} \left( \rho(\hat{O}_{j,D}, O_{j,D})|\delta_{ABC} \right). \quad (12)$$

Given our choices of kernel function in Equation 11 and $\rho(X,Y)$, we set $\delta_{ABC} = 0.1$.

We could have used more advanced methods for establishing an estimate for $\delta_{ABC}$ (see Turner & Sederberg, 2012), but we found that settings of $\delta_{ABC}$ in the range from 0.05 to 0.25 resulted in reasonably accurate posterior estimates.

To facilitate a comparison across likelihood-free and likelihood-informed posterior estimates, we implemented the simplest version of the ABCDE algorithm\(^2\) to sample from the joint posterior distribution (Turner & Sederberg, 2012). Other than the way proposals were evaluated, all other algorithmic parameters were identical to those that we used for the DE-MCMC sampling procedure in the standard Bayesian approach.

**Results**

Figure 2 shows the estimated posterior distributions for $d$, $p$, and $r$ using ABCDE (the gray densities) and likelihood-based DE-MCMC using the exact (solid black densities) and the asymptotic (dashed black densities) expressions. In each panel, the dashed vertical lines show the true parameter values used to generate the data.

The estimated posterior distributions we obtained using ABC are very close to the posterior distributions obtained using DE-MCMC and the exact expressions for the likelihood. We can therefore conclude that the combination of $\rho(X,Y)$, $\mathcal{K}$, and $\delta_{ABC}$ that we selected produced accurate ABC posterior estimates. However, the estimates we

\(^2\)We used only a single group of particles with no mutation or migration steps, and only ran the algorithm in “sample” mode.
Figure 2. The approximate marginal posterior distributions for the parameters $d$ (left panel), $p$ (right panel), and $r$ (right panel) in BCDMEM using ABCDE (gray lines), the exact likelihood expressions (solid black lines), and the asymptotic likelihood equations (dashed black lines). The dashed vertical lines represent the true value of the parameters used to generate the data.

obtained using the asymptotic expressions are very different from those we obtained using the exact expressions and ABC. In particular, the posterior estimates for the parameters $p$ and $r$ are appreciably different from the true solutions. This inaccuracy suggests that the asymptotic expressions will not be very useful for Bayesian analyses of BCDMEM.

The computational time required to obtain the estimates in Figure 2 varied considerably over the three different techniques. For the DE-MCMC estimates using the exact likelihoods, numerical integration with high precision required two hours and 20 minutes of computation. By contrast, we obtained estimates using the asymptotic likelihoods in only 36 seconds. The ABC approach gave results in two minutes and 33 seconds, and the estimated posterior distribution closely matched the true posterior. Therefore, although ABCDE was slower than DE-MCMC using the asymptotic likelihoods, the ABCDE posteriors were nearly as accurate as those obtained with DE-MCMC using the exact likelihoods but, were obtained nearly 55 times faster.
Summary

In this section, we illustrated the utility of the ABC approach by fitting the model BCDMEM to simulated data. The derivations in Myung et al. (2007) provide expressions for the model’s likelihood, which allowed us to compare estimates of the posterior distribution obtained with standard Bayesian techniques to the estimates obtained with ABC. We showed that the ABC estimates were very close to the estimates obtained using the exact expressions, but that the estimates obtained using the asymptotic expressions did not closely match either the ABC or exact expression estimates.

Methods for Fitting the Models to Study 1

We now present the mathematical details of the two hierarchical models in turn, beginning with the BCDMEM model.

BCDMEM

To fit BCDMEM to the data from Dennis, Lee, and Kinnell (2008) we must first recall the role that each parameter in the model plays and their relationship to the experimental design used by Dennis et al.. The vector length $v$ is fixed at 200 and assumed to be uninfluenced by experimental factors. The context sparsity parameter $s$ determines the probability that context nodes become active during study, and the learning rate $r$ determines the probability that a study item becomes associated with any particular context node. We set $s = 0.02$ as suggested by current literature using approximate least-squares methods, while $r$ was permitted to vary over subjects, reflecting individual differences in encoding strength.

The remaining parameters are the contextual reinstatement parameter $d$ and the context noise parameter $p$. The contextual reinstatement parameter $d$ is the probability that a context node that was active during study becomes inactive in the reconstructed context. This probability will be lower in conditions that encourage contextual reinstatement, that is, when additional filler activity is present and for the short lists with
no additional filler. However, for long lists with no additional filler activity, where
contextual reinstatement is less likely, \( d \) will be higher.

The context noise parameter \( p \) is the probability that nodes become active in the
retrieved context vector because of exposure to an item in pre-experimental contexts.
Therefore, we expect that high-frequency words will be associated with higher values of \( p \)
than low-frequency words (Steyvers & Malmberg, 2003).

To model the effects of word frequency, list length and filler, we define two new
independent parameters \( \delta \) and \( \tau \). These parameters are related to the model parameters
by way of three binary indicator variables identifying the Condition \( j \) on a test trial.
First, we let \( F_j \) represent the conditions in which the additional filler activity was absent
(\( F_j = 0 \)) or present (\( F_j = 1 \)), and we let \( L_j \) represent the short (\( L_j = 0 \)) or long (\( L_j = 1 \))
lists. Therefore, over list length and additional filler conditions, we can write the
contextual reinstatement probability as \( \delta L_j (1 - F_j) + d (1 - L_j (1 - F_j)) \), where \( \delta \) is the
contextual reinstatement probability for long lists with no additional filler activity and \( d \)
is the contextual reinstatement probability for all other conditions.

Next, we let \( W_j \) represent whether the studied words were of high (\( W_j = 1 \)) or low
(\( W_j = 0 \)) frequency. Therefore, the context noise parameter is \( p (1 - W_j) + \tau W_j \), where \( \tau \)
is the probability that nodes become active in the retrieved context vector for
high-frequency words and \( p \) is the probability that nodes become active in the retrieved
context vector for low-frequency words. We can then write the parameter vector \( \theta_{i,j} \) for
the \( i \)th subject in the \( j \)th condition as

\[
\theta_{i,j} = \{ \delta_i L_j (1 - F_j) + d_i (1 - L_j (1 - F_j)), p_i (1 - W_j) + \tau_i W_j, r_i, s, v \}.
\] (13)

To complete the model, we must now select priors for each of the parameters.
Because they all represent probabilities, they all are restricted to be between zero and one,
and we can again use the beta distribution (Equation 10) as a common prior for each
parameter. We therefore assume that each of the individual-level parameters come from a
common beta distribution, or

\[ d_i \sim \text{Beta}(\omega_d, \xi_d), \]
\[ p_i \sim \text{Beta}(\omega_p, \xi_p), \]
\[ r_i \sim \text{Beta}(\omega_r, \xi_r), \]
\[ \delta_i \sim \text{Beta}(\omega_\delta, \xi_\delta), \quad \text{and} \]
\[ \tau_i \sim \text{Beta}(\omega_\tau, \xi_\tau), \]

(14)

where the hyperparameters \( \omega_k \) and \( \xi_k \) (for \( k = d, p, r, \delta, \) or \( \tau \)) define the hierarchical model structure. BCDMEM has never been fit hierarchically to data, so we have no information about the likely ranges of the hyperparameters. We use mildly informative priors for the hyperparameters to reflect this uncertainty. Because the hyper-priors are the same for each of the five parameters, we will group the hyper-means

\[ \omega = \{ \omega_d, \omega_p, \omega_r, \omega_\delta, \omega_\tau \} \]

and the hyper-scales \( \xi = \{ \xi_d, \xi_p, \xi_r, \xi_\delta, \xi_\tau \} \) and write the hyperpriors as

\[ \omega \sim \text{Beta}(0.5, 2), \quad \text{and} \]
\[ \xi \sim \Gamma(0.1, 0.1) \]

for each element in the vectors \( \omega \) and \( \xi \).

Figure 3 shows a graphical diagram for this model. These types of diagrams are often very useful for illustrating how the parameters in the model (white nodes) are connected via arrows to the observed data (gray nodes; see Lee, 2008; Shiffrin et al., 2008; Lee & Wagenmakers, 2012). When the variables are discrete they are shown as square nodes, whereas when they are continuous they are shown as circular nodes. A double-bordered variable indicates that the quantity is deterministic rather than stochastic. Specifically, the node corresponding to \( \theta_{i,j} \) is double-bordered because it is always determined by evaluating Equation 13. Finally, “plates” show how vector-valued variables are interconnected. For example, the node for the parameter \( \omega_{ij} \) is not on the
Figure 3. A graphical diagram for the hierarchical BCDMEM model for the data of Dennis, Lee and Kinnell (2008).

plate, which indicates that this parameter is fixed across both subjects and test trials, while there are separate $d_i$ nodes for every subject.

REM

As for BCDMEM, we allow the environmental base rate to vary with word frequency. For mathematical convenience, we specify two environmental base rate parameters: $g$ for low-frequency words and $\gamma$ for high-frequency words (Steyvers & Malmberg, 2003). We again make use of an indicator variable $W_j$ for high ($W_j = 1$) and low ($W_j = 0$) frequency words, so we can write the rate parameter for traces in Condition
To model the effects of filler activity, we assume that spurious traces are added to memory prior to the test phase. These extra traces distort memory by introducing interference, increasing the memory strength of unstudied probes. While there are many ways to distort memory in REM, we chose this approach because it added only a single parameter and so matched the number of parameters in the BCDMEM model. We redefine the indicator variable $F$ so that $F_j = 1$ for the filler conditions and $F_j = 0$ for the no filler conditions. The parameter $\eta$ is used to model the number of spurious features added to the episodic matrix. Therefore, we can write the number of traces in Condition $j$ as $N_{STUDY} + F_j \eta$.

We represent the number of hits for Subject $i$ in Condition $j$ as $O_{i,j,T}$ and, similarly, false alarms as $O_{i,j,D}$. Montenegro, Myung, and Pitt (2011) showed that $w$ must be fixed to some positive integer to ensure that the model is identifiable, that is, that one unique set of parameter values can be found to best fit the data. By convention, we set $w$ equal to 20 (e.g., Shiffrin & Steyvers, 1997; Criss & McClelland, 2006). We denote the number of items presented at study as $N_{STUDY}$, so there will be $N_{STUDY} + \eta$ item traces in the episodic memory matrix for the filler conditions. For notational convenience, we again define $\theta_{i,j}$ as the vector of parameters for the $i$th subject and $j$th condition, so

$$\theta_{i,j} = \{ N_{STUDY} + \eta F_j, g_i(1 - W_j) + \gamma W_j, u_i, c_i, w \}. \quad (15)$$

Four of the parameters in this model, $g$, $\gamma$, $u$ and $c$, are restricted to be in the range (0, 1). This means that, for these parameters, we can again use the beta distribution given in Equation 10 as their prior. The individual-level parameters for each subject are assumed
to arise from a common beta distribution with the appropriate hyperparameters, or
\[ c_i \sim \text{Beta}(\omega_c, \xi_c), \]
\[ g_i \sim \text{Beta}(\omega_g, \xi_g), \]
\[ u_i \sim \text{Beta}(\omega_u, \xi_u), \text{ and} \]
\[ \gamma_i \sim \text{Beta}(\omega_\gamma, \xi_\gamma). \]

The number of spurious features \( \eta \) must be positive and can only take on discrete values.

For these reasons, we chose the prior for \( \eta \) to be negative binomial distribution with hyperparameters \( \omega_\eta \) (the size parameter) and \( \xi_\eta \) (the probability parameter), or
\[ \eta_i \sim \text{NegBin}(\omega_\eta, \xi_\eta). \]

The probability density function of the negative binomial distribution is
\[ \pi(\eta \mid \omega_\eta, \xi_\eta) = \frac{\Gamma(\eta + \omega_\eta)}{\Gamma(\omega_\eta)\eta!} (\xi_\eta)^\omega_\eta (1 - \xi_\eta)^\eta. \]

Because REM has never been fit in a Bayesian framework, we have no information to guide our selection of the priors for the group-level parameters. As a consequence, we used mildly informative priors for each of them. We defined the hyper-means as \( \omega = \{\omega_g, \omega_u, \omega_c, \omega_\gamma\} \) and the hyper-scales as \( \xi = \{\xi_g, \xi_u, \xi_c, \xi_\gamma\} \), and then we selected values for the hyperparameters that placed (approximately) equal probability over the possible values in the parameter space. Thus we chose
\[ \omega \sim \text{Beta}(0.5, 2), \]
which is the continuous uniform density on the interval \((0,1)\), and
\[ \xi \sim \Gamma(0.1, 0.1). \]

For \( \omega_\eta \) and \( \xi_\eta \), we assumed
\[ \omega_\eta \sim \Gamma(0.1, 0.1) \text{ and} \]
\[ \xi_\eta \sim \text{Beta}(0.5, 2). \]
Figure 4. A graphical diagram for the hierarchical REM model for the data of Dennis, Lee and Kinnell (2008).

Figure 4 shows a graphical diagram for this model. As with Figure 3, there are five lower-level parameters to measure subject-specific effects and ten hyperparameters to model the group-level effects.

Implementing the ABC Algorithm

To fit the models, we first separated the hyperparameters $\phi = \{\omega, \xi\}$ from the individual-level parameters $\theta_{i,j}$, which are defined in Equation 13 for BCDMEM and Equation 15 for REM, respectively. We then sampled the parameters $\phi$ conditioned on the parameters $\theta_{i,j}$ using DE-MCMC (Braak, 2006; Turner et al., 2012), and sampled the
parameters $\theta_{i,j}$ conditioned on the parameters $\phi$ using the ABCDE algorithm (Turner & Sederberg, 2012).

We chose a Gaussian kernel function as in Equation 11, specified

$$\rho(X, Y) = \|X - Y\|,$$

and set $\delta_{ABC} = 0.2$. Similar to Equation 12, we evaluate the fitness of the proposal $\theta^{*}_{i,j}$ by calculating

$$\psi(\theta^{*}_{i,j} \mid X, Y) = \prod_{i=1}^{S} \prod_{j=1}^{C} \mathcal{K} \left( \rho(\hat{O}_{i,j,T}, O_{i,j,T}) \mid \delta_{ABC} \right) \mathcal{K} \left( \rho(\hat{O}_{i,j,D}, O_{i,j,D}) \mid \delta_{ABC} \right),$$

where we denote the number of simulated hits as $\hat{O}_{i,j,T}$ and the number of simulated false alarms as $\hat{O}_{i,j,D}$.

We ran the algorithm in parallel on 8 processors to obtain 24 sequences (chains) of 10,000 samples from the joint posterior distribution of $\{\phi, \theta\}$. We discarded the first 2,000 samples from each chain, and then assessed each chain for convergence. We then collapsed across all 24 chains, resulting in 192,000 samples from the joint posterior distributions.

### Evaluating the Quality of Model Fit

To compare the relative fit of the two models to the data, we computed the DIC (Spiegelhalter, Best, Carlin, & Linde, 2002) and BPIC (Ando, 2007) values. Because we held $\delta_{ABC}$ constant across the two models, the DIC and BPIC measures can be used in a similar manner as they are in standard Bayesian applications. To compute the DIC and BPIC values, we first defined the deviance as $D = -2 \log(L(\theta \mid Y))$, where $L(\theta \mid Y)$ is approximated by Equation 16. We then evaluated the expectation $\bar{D}$ of the deviance by taking the mean of $D$ over all sampled values of $\theta$. Subtracting from this expectation the best log-likelihood value obtained, $\hat{D} = \min(D)$ (Celeux, Forbes, Robert, & Titterington, 2006; Spiegelhalter et al., 2002), we obtain a measure of the effective number of parameters $p_D = \bar{D} - \hat{D}$.\textsuperscript{3} As $p_D$ increases, the model becomes more flexible, making it easier for the model to fit the data.

\textsuperscript{3}The choice of $\hat{D} = \min(D)$ rather than $\hat{D} = E(D)$, where $E$ denotes the expected value, is justified here because the posterior distributions are non-normal and are not symmetric (Celeux et al., 2006).
The DIC is obtained by summing the number of effective parameters and the expected deviance, so

\[ DIC = p_D + \bar{D}. \]

Models with smaller (i.e., more negative) DIC values are preferred over models with larger DIC values. Recently, the DIC statistic has been the subject of some scrutiny because the DIC metric tends to prefer models that over-fit the data (e.g., Ando, 2007). As an alternative, Ando (2007) proposed the Bayesian predictive information criterion (BPIC), given by

\[ BPIC = \bar{D} + 2p_D. \]

Both of these evaluation metrics are simple to calculate and have the advantage over other measures of model fit (e.g., the BIC or the Bayes factor) because they can be calculated from MCMC output alone.

Methods for Fitting the Models to Study 2

We fit both models to the data from each stimulus type condition independently because our primary research objective was to evaluate the relative fits of the two models to list length data on a variety of stimuli types. Fitting the three stimulus types independently – rather than fitting all of the data in a single, three-level hierarchical model – allowed such a comparison to be made.

BCDMEM

For BCDMEM, we begin again by fixing the vector length parameter \( v \) at 200, and the context sparsity parameter \( s \) at 0.02. Thus, we can write the parameter vector \( \theta_i \) for the \( i \)th subject as

\[ \theta_i = \{d_i, p_i, r_i, s, v\}. \]
We again assume that each of the individual-level parameters come from a common beta distribution, or
\[
\begin{align*}
d_i &\sim \text{Beta}(\omega_d, \xi_d), \\
p_i &\sim \text{Beta}(\omega_p, \xi_p), \text{ and} \\
r_i &\sim \text{Beta}(\omega_r, \xi_r),
\end{align*}
\]
where the hyperparameters \(\omega_k\) and \(\xi_k\) (for \(k = d, p,\) or \(r\)) define the hierarchical model structure. We use the same mildly informative priors for the hyperparameters as in Study 1.

**REM**

For REM, we again fix the vector length parameter \(w\) at 20. Thus, we write the parameter vector \(\theta_i\) for the \(i\)th subject as
\[
\theta_i = \{c_i, g_i, u_i, w\}. \tag{18}
\]
We again assume that each of the individual-level parameters come from a common beta distribution, or
\[
\begin{align*}
c_i &\sim \text{Beta}(\omega_c, \xi_c), \\
g_i &\sim \text{Beta}(\omega_g, \xi_g), \text{ and} \\
u_i &\sim \text{Beta}(\omega_u, \xi_u),
\end{align*}
\]
where the hyperparameters \(\omega_k\) and \(\xi_k\) (for \(k = c, g,\) or \(u\)) define the hierarchical model structure. We use the same mildly informative priors for the hyperparameters as in Study 1.

**Implementing the ABC Algorithm**

All of the implementation details of the ABC algorithm were the same as when we fit the models to Dennis et al.’s (2008) data in the previous section.
References

Ando, T. (2007). Bayesian predictive information criterion for the evaluation of hierarchical

structure and local adaptation in a Bayesian hierarchical model. *Genetics, 185*, 587-602.


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

Differential Evolution: easy Bayesian computing for real parameter spaces. *Statistics and
Computing, 16*, 239-249.

for missing data models. *Bayesian Analysis, 1*, 651-673.

of the retrieving effectively from memory model (rem) and the subjective likelihood model


the list-length effect. *Journal of Mathematical Psychology, 59*, 361-376.

NY: Chapman and Hall.

vicariance and dispersal using a hierarchical Bayesian approach. *BMC Evolutionary Biology,
8*, 322.

approximate Bayesian computation. *Evolution, 60*, 2435-2453.


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