Likelihood-Free Bayesian Modeling

Dissertation

Presented in Partial Fulfillment of the Requirements for the Degree Doctor of Philosophy in the Graduate School of The Ohio State University

By

Brandon M. Turner, B.S., M.A., M.A.S.

Graduate Program in Department of Psychology

The Ohio State University

2011

Dissertation Committee:

Trisha Van Zandt, Advisor

Simon Dennis

Jay Myung

Abstract

Bayesian modeling has been influential in cognitive science. However, many psychological models of behavior have difficult or intractable likelihood functions. This poses a major problem for Bayesian inference, which requires a likelihood to provide estimates of the posterior distribution. In this dissertation, I provide a detailed examination of a new approach designed to avoid evaluating the likelihood. I provide an overview of current algorithms, and introduce a new algorithm for the estimation of the posterior distribution of the parameters of hierarchical models. I then apply the technique to two popular cognitive models of episodic memory. Finally, I show how the method can be used to perform model selection by means of mixture modeling.

Acknowledgments

I would like to thank

- Scott Brown for telling me how to run C from R, without this, I would probably still be hunched over my computer right now.
- Simon Dennis for help in developing the memory models and for encouraging me to participate in conferences, which was instrumental to my development.
- Jay Myung for helpful advice in writing (i.e.“selling”), submitting, and partitioning this dissertation into as many publications as possible.
- Trisha Van Zandt for three years of support, career development and friendship. She was always ready to drop whatever she was doing to listen to my ideas, regardless of how bizarre they were.
- My wife Katherine, for believing in me when I was incapable of believing in myself and for enduring the graduate school roller coaster ride as well as humanly possible.

Vita

November 18, 1985	Born - Bolivar, Missouri
2008	B.S. Mathematics, Missouri State University, Missouri
2008	B.S. Psychology, Missouri State University, Missouri
2009	M.A. Quantitative Psychology, The Ohio State University, Ohio
2011	M.A.S. Statistics, The Ohio State University, Ohio
2008-2011	Graduate Teaching Associate, The Ohio State University
2009-present	Psychology 100 Statistical Consultant, The Ohio State University
2011-Present	Psychology Department Statistical Consultant, The Ohio State University

Publications

Research Publications

B. M. Turner, T. Van Zandt, and S. Brown "A dynamic, stimulus-driven model of signal detection". *Psychological Review*, In Press

N. E. Betz and B. M. Turner "Using item response theory and adaptive testing in online career assessment". *Journal of Career Assessment*, *19 (3)* 274-286.

H. B. Rim, B. M. Turner, N. E. Betz, and T. E. Nygren "Studies of the dimensionality, correlates, and meaning of measures of the maximizing tendency". *Judgment and Decision Making*, *6 (6)* 565-579.

B. M. Turner, N. E. Betz and M. C. Edwards and F. H. Borgen "Psychometric examination of an inventory of self-efficacy for the Holland vocational themes using item response theory". *Measurement and Evaluation in Counseling and Development*, *43 (3)* 188-198.

B. M. Turner "The Mathematical Process of Classification". *The Pentagon*, *68 (2)* 5-16.

Fields of Study

Major Field: Psychology

Major Field: Statistics

Table of Contents

Page

List of Figures

Figure **Page**

Chapter 1: Introduction

1.1 Introduction

The major goal of mathematical modeling is to supply a model that provides insight into the underlying mechanisms guiding a behavior of interest. These underlying mechanisms are represented by some mathematical or statistical processes controlled by a number of parameters. For example, sequential sampling models represent the decision process as an accumulation of evidence for a response. Evidence is accumulated by sampling from a normal distribution with parameters governing the mean and variance of the amount of evidence to be accumulated. When these parameters are modified, the model makes different predictions about the behavior of interest. For example, when the mean of the normal is increased, the model will accumulate larger amounts of information, ultimately leading to a faster response. These model predictions can then be tested by experimental data. If a model predicts a behavior that is not validated by experimental data or if a model can not predict a behavior that is seen in experimental data, then this suggests that the processes used by the model may not be reflective of actual behavior. Psychologists are often interested in the systematic differences between groups or individuals. These differences may be explained by intrinsic details such as age, demographic information, or gender. Similarly, psychological models – when fit appropriately to data – convey information through the values of the parameters. One naive approach to understanding these differences is to estimate model parameters for each observer independently. One could then compare these estimates to the group, and draw inference regarding relative performance. However, experimental data are often highly structured and ripe with detailed information on both the individual and group levels. A better analytic strategy would be to estimate individual and group parameters *simultaneously*.

Bayesian statistics provides a convenient framework for performing such an analysis (e.g., Efron, 1986, Lee, 2008, Shiffrin et al., 2008, Wagenmakers, 2007). Interest in Bayesian statistics has grown for a number of practical and theoretical reasons. Bayesian statistics was developed prior to classical statistics, but due to the complicated normalizing constant (see Chapter 2), it was set aside for nearly a century. Today, the availability of powerful desktop computers and algorithms to fit fully Bayesian models have inspired the adoption of Bayesian methods.

Although this framework is appealing and powerful in theory, it requires two things: a prior distribution and a likelihood function. The prior is always available, because it is selected by the researcher. By contrast, there are many models for which a likelihood can be difficult or impossible to specify mathematically. This problem arises most frequently for computational models, which generate predictions by simulating the data-generating mechanism. These models are very popular in the social sciences, and in cognitive psychology in particular. Simulation-based cognitive models with difficult likelihood functions were often developed by researchers who were interested primarily by explaining a behavior of interest, and only secondarily interested in estimation. Indeed, psychological plausibility should be the propelling force behind model development, especially when simplifying mathematical assumptions hinder a model's explanatory power. Regardless, situations in which a model's likelihood function is complicated or intractable currently results in a restricted set of available parameter estimation techniques. In particular, maximum likelihood and Bayesian estimation may not be possible.

In this scenario, one must resort to methods such as least squares estimation or fits "by-hand." To perform least squares estimation, we generate data under different combinations of the parameters and then compare this simulated data to the observed data through some discriminant function. The parameter values that minimize this discriminant function are the "best-fitting" values. The methods for exploring a parameter space may be complicated, such as a simplex, or they may be much more simple, such as a grid search. After the parameter space has been satisfactorily explored, the parameter values and their corresponding discriminant function values are compared. Other methods are more qualitative than quantitative, such as by-hand fits. These methods focus on determining whether or not a model can produce predictions similar in pattern to what was observed. While parameter values obtained with either of these methods may be appropriate, they generally have well-known shortcomings, and are not desirable (e.g., Lee, 2008, Myung, 2003, Rouder et al., 2005, Van Zandt, 2000).

Simulation-based models have not, up until now, been able to take advantage of progress in Bayesian computation because of the lack of an explicit likelihood function. However, a new approach called "approximate Bayesian computation" (ABC) has been successfully applied to estimating the parameters of complex or likelihood-free models in population genetics, where it still currently receives the most attention. There has been a recent surge of interest in other related areas, such as ecology, epidemiology and systems biology (see Beaumont, 2010, for a broad overview). Even more recently, ABC has spilled into psychology (Turner and Van Zandt, a,b, Turner et al., 2011a).

Originally developed by Pritchard et al. (1999), ABC replaces the calculation of the likelihood function with a simulation of the model, producing an artificial data set X. The method then relies on some metric (a distance) to compare the simulated data X to the observed data Y. In this way, ABC is similar to the 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.

In this dissertation, my goal is to examine the utility of the ABC approach. To do so, I will first investigate ABC in problems where standard Bayesian analyses can be performed. If the estimates obtained using ABC closely match the estimates obtained using standard Bayesian techniques, then the ABC approach can be validated. It will then be necessary to apply the ABC technique to psychological models.

Although a more formal investigation of the ABC approach and its usefulness when applied to psychological models would be interesting, it would be incomplete without a thorough examination of ABC methods for hierarchical designs. Indeed, the analysis of individual differences is of great importance to a psychologist. Not only is the behavior important on an individual level, it is also important on an experimental level. However, ABC is currently difficult – even impractical in some cases – to implement in hierarchical designs. The reason for this is mostly due to the accept/reject nature of the algorithms. When the number of individual-level parameters is small, the standard ABC algorithm can be easily extended into the hierarchical case by considering the joint estimation of the parameters in the tiers of the hierarchy. This idea has been implemented in the genetics literature in order to analyze mutation rate variation across specific locations of genes, known as loci (Excoffer et al., 2005, Pritchard et al., 1999). However, these approaches can be very slow and even impractical when the number of parameters increases.

One recent attempt at solving this problem, discussed in Chapter 3, divides the estimation procedure into two parts. First, we decide upon a set of group-level parameters, called hyperparameters, based on using ABC on the marginal data. We then use the group-level parameters from the first stage to generate individual-level parameters, which are also estimated using ABC. This technique requires data simulation at two stages, which can be time consuming. In addition, as noted by Bazin et al. (2010) and Beaumont (2010), this approach involves an approximation of the posterior distribution for the hyperparameters, over and above the approximation due to ABC alone (e.g., using a false model, using summary statistics).

In Chapter 3, I develop a new algorithm, based on Gibbs sampling, designed for the estimation of hierarchical models using ABC. This algorithm embeds the ABC technique into standard Bayesian estimation, which maximizes accuracy and minimizes computation time. This algorithm is highly flexible, giving way to user- and situation-specific constraints.

1.2 Organization

In Chapter 2, I present a manuscript submitted to the Journal of Mathematical Psychology as a tutorial paper on ABC methods. This manuscript is joint work with Trisha Van Zandt. In this manuscript, we first review the existing ABC algorithms and delineate between them. We demonstrate the usefulness of the ABC approach through a variety of examples. These examples are compared, whenever possible, to the true posterior distributions, in order to demonstrate the ABC algorithm's ability to recover the true posterior.

Although the manuscript in Chapter 2 includes an example of using ABC to estimate the parameters of a hierarchical model, I emphasize in Chapter 3 that the algorithms presented in the tutorial are not suited for estimation of more complicated hierarchical designs. I argue that a more in-depth exploration of hierarchical ABC algorithms is necessary. In Chapter 3, I first review available algorithms for performing hierarchical ABC, and then introduce a new mixture algorithm. These algorithms are contrasted with one another in a simple example. In this example, posterior estimates obtained using the the two-stage algorithm developed by Bazin et al. (2010) and the mixture algorithm are compared to a true posterior distribution. In a comparison of the Kullback-Leibler statistics, the mixture algorithm outperforms the two-stage algorithm. I then provide some suggestions for optimizing the mixture algorithm.

Chapter 4 focuses on applying ABC techniques presented in Chapters 2 and 3 to simulation-based psychological models. First, the Bind Cue Decide model (BCDMEM; Dennis and Humphreys, 2001) of episodic memory is used to compare posterior estimates obtained using standard and approximate Bayesian techniques.

The standard Bayesian fits are obtained using both the exact and asymptotic expressions for the likelihood provided in Myung et al. (2007). Once convinced that ABC provides an accurate estimate of the "true" posteriors, I proceed into a hierarchical version of BCDMEM. This model is used to fit the data presented in Dennis et al. (2008). The results show close agreement between the posterior predictive distributions and the data. Next, the Retrieving Effectively from Memory model (REM; Shiffrin and Steyvers, 1997) is used in a variety of simulation studies meant to investigate the relationships between model parameters and model predictions through inspection of the joint posterior distribution, obtained using ABC. Several interesting conclusions are drawn from these results. Finally, I fit a hierarchical version of REM to the data presented in Dennis et al. The results once again show close agreement between the posterior predictive densities and the data. In the final chapter, I demonstrate how the ABC approach can be used to perform model selection by means of mixture modeling. The technique allows one to determine, for a given data set, the probability that data was generated by a particular model. I use this technique to choose between the models REM and BCDMEM for an exhaustive set of hypothetical data in a simulation study. The results suggest that certain regions of the data space are consistently fit better by one model or the other. In addition, I show how the model selection space is manipulated as the number of items presented at study increases in a recognition memory task. These results have drastic implications for global memory models. I then fit a mixture model to the data presented in Dennis et al. (2008).

Chapter 2: An Introduction to Approximate Bayesian Computation

In this chapter, I present a manuscript submitted to the Journal of Mathematical Psychology as a tutorial paper on ABC techniques. This manuscript is joint work with Trisha Van Zandt and serves to formally introduce the basics of the ABC approach. In later chapters, I build upon these ideas for more complicated modeling.

2.1 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" (p. 9 Abelson, 1995) of null hypothesis testing, but also because they allow for statistical inference without compromising the theory that motivated the experiments that generated the data (e.g., Lee et al., 2006, Nilsson et al., 2011, Vandekerckhove et al., 2011). 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 and 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. 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 their probability distributions after some

[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 letters represent variable quantities, while lower-case letters represent fixed values.

data are observed. Computation or estimation of these probability distributions requires two things. 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 (such as the diffusion model or signal detection theory), this requirement is simply that we be able to write down the proposed probability density $f(y|\theta)$ for each observation from the theoretical mechanism that generates the data. Then, assuming that the observations $\{Y_1, Y_2, \ldots, Y_n\}$ are independent and identically distributed, the likelihood is defined as

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

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 et al., 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 a probability distribution for θ, called the *posterior* distribution. Using Bayes' Theorem, the posterior $\pi(\theta|Y)$ is

$$\pi(\theta|Y) = \frac{L(\theta|Y)\pi(\theta)}{\int L(\theta|Y)\pi(\theta)\, d\theta}. \tag{2.1}$$

With the posterior distribution of θ in hand, we can say things about 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 the probability that the alternative hypothesis is true (or, conversely, that the null hypothesis is false). 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.

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 a few toy problems. The difficulty arises in the integral appearing as the denominator of Equation 2.1, 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

$$\pi(\theta|Y) \propto L(\theta|Y)\pi(\theta). \tag{2.2}$$

If both the likelihood and the prior have analytic forms, Equation 2.2 implies that the desired posterior is tantalizingly close at hand. However, unless the distributional form of $\pi(\theta|Y)$ can be deduced from the product of the likelihood and the prior, there remains considerable computation before we can accurately estimate the posterior 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 models, 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, standard methods of Bayesian estimation, as well as classical maximum likelihood estimation (Myung, 2003), are not possible.

Consider, for example, O'Reilly and colleagues' LEABRA model of learning (O'Reilly, 2001, 2006, O'Reilly and 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 neuroanatomy. 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 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, and learning and memory. The behavior of the model in different circumstances is determined by simulating its behavior many times. It does not have a likelihood that describes the probability of different model outputs. Therefore, like other simulation-based 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.

2.2 Plan of the Tutorial

We begin in Section 2.3 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 2.4 and 2.5).

Our first example considers a beta-binomial problem and a simple rejection sampler (see Algorithm 1). Next, we move to an exponential model, which requires that we shift to a more general ABC algorithm, the population Monte Carlo sampler

(Algorithm 2). In Section 2.6 we generalize the ABC population Monte Carlo sampler for hierarchical models and apply it to simulated data from a hierarchical binomial model (see Algorithm 3). Finally, in Section 2.7 we apply the algorithm 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.

2.3 Approximate Bayesian Computation

Originally developed by Pritchard et al. (1999), approximate Bayesian computation (ABC) replaces the calculation of the likelihood function $L(\theta|Y)$ in Equations 2.1 and 2.2 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 (Malmberg et al., 2004, Nosofsky et al., 2011). In these papers, the authors 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 the 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 the 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 $\pi(\theta|\rho(X, Y) < \epsilon_0)$ 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. Initially, this distribution will be the prior $\pi(\theta)$. We then use this candidate parameter value to simulate data X from a model. We then compare the simulated data X to the observed data Y by computing a distance between them given by some 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 having generated the observed data. If $\rho(X, Y)$ is less than ϵ_0, then 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 (e.g., the mean or variance). As one might imagine, the choice of $\rho(X, Y)$ can be tricky, in part because it will depend on the unknown likelihood. A distance function that is specified inappropriately can lead to bad estimates of the posterior of θ. As we will show in this paper, 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 rejection sampling (see Figure 2.1; e.g., Beaumont et al., 2002, Pritchard et al., 1999). Rejection samplers simply discard the candidate value θ^* if it does not meet the criterion $\rho(X, Y) < \epsilon_0$, as we described above. For very small values of ϵ_0, the rejection rate can be dramatically high. As a result, these algorithms can be very inefficient and we will not discuss them further. 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), Hickerson and Meyer (2008), Hickerson et al. (2006), Sousa et al. (2009) for additional options and more mathematical background.

2.3.1 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 et al., 2004, Robert and Casella, 2004). It has also been applied in ABC (Bortot et al., 2007, Marjoram et al., 2003), and we discuss this application here.

MCMC sampling is a process that filters proposed values for θ to arrive at a sample of values that follow the desired posterior distribution. We begin 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 value θ_0. For example, we could choose the proposal distribution q to be Gaussian. Formally,

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

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 τ^2. With some probability 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 MCMC algorithm, 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 MCMC algorithms 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.

The ABC approach can be easily embedded within the MCMC algorithm. After sampling θ^*, instead of computing the acceptance probability from the nonexistant likelihood, we compute it by producing simulated data X from the model. We evaluate θ^* by computing 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 $\theta_1 = \theta^*$. Otherwise we reject θ^*, and $\theta_1 = \theta_0$.

Formally, 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. After computing α for θ^*, we generate a uniform random [0,1] sample, and if this sample is less than α, we accept θ^*. As in

MCMC, if the proposal distribution q is symmetric, $q(\theta_i|\theta^*) = q(\theta^*|\theta_i)$, then α depends only on the prior distribution.

The chain $\{\theta_0, \theta_1, \ldots, \theta_m\}$ must be evaluated for convergence (see Gelman et al., 2004, Robert and Casella, 2004). Convergence diagnostics are important because MCMC algorithms suffer severely if the proposal distribution q is poorly chosen. For example, if τ^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 characteristic that can be remedied through thinning. Thinning refers to a procedure where only a subset of the chain, equally spaced, is retained as a sample from the posterior. For instance, we might decide to keep every 100^{th} value from $\{\theta_0, \theta_1, \ldots, \theta_m\}$, which will requiring 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 MCMC 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 ABC MCMC algorithms further.

2.3.2 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. This means only that 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 $\eta \sim \mathcal{N}(0, \tau^2)$ 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 τ^2.[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,

[2] This function serves the same purpose as the proposal distribution in the MCMC algorithm.

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 τ^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 et al., 2007). The optimal choice for the backward kernel can be difficult to determine (Del Moral et al., 2006). Symmetric kernels greatly simplify the algorithm, but may be a poor choice (see Toni et al., 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.

Partial Rejection Control

The partial rejection control (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 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. 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 we have recreated the pool with N new particles, each satisfying the requirement that $\rho(X, Y) < \epsilon_0$.

If we stop at after recreating the pool once, then PRC is equivalent to the rejection sampler described above (Figure 2.1). However, we continue to iterate over this process several (or many) 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 $\pi(\theta|\rho(X, Y) < \epsilon_0)$.

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 PRC produces biased estimates of the posterior (see Beaumont et al., 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.

Population Monte Carlo Sampling

ABC population Monte Carlo sampling (ABC PMC) has a different weighting scheme than PRC (Beaumont et al., 2009). While the 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 Figure 2.3, was inspired by the population Monte Carlo algorithm developed for standard Bayesian estimation by Cappé et al. (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

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

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

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

This weighting scheme allows for asymptotic improvements in the Kullback-Leibler divergence between the prior and the posterior (see Douc et al., 2007, for a proof). Used frequently in statistics, the Kullback-Leibler divergence measures the difference between two probability distributions. 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).

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 ABC PMC scheme is important because it automatically optimizes the acceptance rate regardless of the prior.

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 $\mathrm{Bin}(n, p)$ model.

Summary

This section summarized the most popular and efficient ABC algorithms. We have experimented with many of these, and ABC PMC has consistently provided good results for the psychological models to which we have applied it. Therefore, in the applications to follow we will focus primarily on the ABC PMC algorithm (see Figure 2.3).

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 the to demonstrate the accuracy of the posteriors estimated by ABC. We then show how ABC works with a more realistic model, but one for which the true posteriors are unknown (but see Footnote 3).

2.4 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 and 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.

2.4.1 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 $\{0, 1, 2, ..., n\}$. 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. 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, it is similar to having witnessed two outcomes, one a success and the other a failure. The uninformative $\text{Beta}(1, 1)$ prior places equal probability over 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 the beta-binomial model is (see Gelman et al., 2004, Wagenmakers, 2007)

$$p|\alpha,\beta,Y \sim \text{Beta}\left(\alpha = Y + \alpha_0, \beta = n - Y + \beta_0\right), \tag{2.3}$$

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 asses the accuracy of the estimated posteriors produced by ABC.

2.4.2 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 interval for p. However, our goal is to demonstrate the accuracy of the estimates of the posterior produced by the ABC approach, and so we suppose 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 Equation 2.3, 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

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

the absolute difference between the proportions of observed and simulated correct responses. The distance $\rho(X,Y)$ can be interpreted as 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 meant to relieve the computational burden (see Section 2.5).

2.4.3 Results

We simulated the model under three conditions reflecting the behavior of three different observers, each with $p = 0.7$. 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 Equation 2.3).

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 Figure 2.1 with tolerance threshold $\epsilon_0 = 0$. Figure 2.2 shows the distributions of values for p for each of the three observers. Overlaying each histogram is the true posterior given by Equation 2.3. Figure 2.2 shows that as the number of trials increases, the posterior becomes more narrow around the true value of p. 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 all three observers took only a few minutes using R (R Development Core Team, 2008). Furthermore, Algorithm 1 can be parallelized, and the computation time reduced considerably for more complex problems.

2.5 An Exponential Example

While the binomial example demonstrates that the ABC approach can accurately estimate the posterior of the beta-binomial model, 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 11.7815...) will be zero and perfect matches between X and Y will be impossible. In practice, we round continuous variables, so that 11.7815... becomes 11.78 (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 the 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 and Ludwig, 2008, Matzke and Wagenmakers, 2009, Rouder and Speckman, 2004), practice effects (e.g., Heathcote et al., 2000), relating stimulus similarity to psychological distance (e.g., Nosofsky, 1986), predicting change (Brown and Steyvers, 2009) and memory decay (e.g., Lee, 2004, Liu and Aitkin, 2008, Rubin and 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.

2.5.1 The Model

The exponential distribution Exp(λ) 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

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

where the hyperparameters $\alpha > 0$ and $\beta > 0$ are usually called the shape and scale parameters, respectively. The exponential distribution is a special case of the gamma distribution with $\alpha = 1$ and $\lambda = 1/\beta$. The gamma distribution is a conjugate prior for λ, 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 and β were fixed at 0.1.

2.5.2 Estimating the Posterior Using ABC PMC

We face three problems at this point. First, because Y is continuous, using Algorithm 1 we cannot hope to obtain samples that satisfy $\rho(X, Y) < \epsilon_0$ for some very small ϵ_0. For this reason, we will need to use the ABC PMC algorithm (Algorithm 2) described above and shown in Figure 2.3. Second, we must establish a reasonable set of monotonic decreasing tolerance thresholds ϵ. Very small values of ϵ like the $\epsilon_0 = 0$ used in Section 2.4 can greatly increase the computational burden associated with sampling from a high-dimensional parameter space. Third, we must consider the relationship between the distance function $\rho(X, Y)$ and the accuracy of the estimated posteriors. We will do this by exploring three different forms of $\rho(X, Y)$.

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

$$\begin{aligned}\rho_3(X, Y) &= \Big|\left[F^{-1}(.75, X) - F^{-1}(.25, X)\right] - \left[F^{-1}(.75, Y) - F^{-1}(.25, Y)\right]\Big| \\ &= \Big|\text{IQR}(X) - \text{IQR}(Y)\Big|,\end{aligned}$$

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. For example, the non-decisional component of a reaction time model, which is restricted to be less than the smallest observed RT, can be modeled as a uniform $[0, \theta]$, and the maximum likelihood estimate for θ is the smallest observed RT. 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 look to standard estimators of the parameters of the model (the likelihood) and the statistical properties of those estimators. For example, a statistic such as $\overline{X}$ (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, such as the minimum statistic for a lower limit, may also provide the basis for a good choice of $\rho(X,Y)$. In the case where a likelihood is not available, the situations 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 $\rho(X,Y)$.

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 monotonically decreasing set of values for ϵ that balances the number of iterations against the rejection rates within each iteration.

Currently, there are no good general guidelines for choosing such threshold criteria. The values for ϵ will depend on the scale of the data and the distance metric selected. 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 $\epsilon = \{1, 10^{-1}, 10^{-3}, 10^{-4}, 10^{-5}\}$. We could have selected other values for ϵ; ultimately, only the last (smallest) ϵ matters. When ϵ 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.

2.5.3 Results

The top panel of Figure 2.4 shows the estimated posteriors for three values 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 decreases, the accuracy of the estimated posterior increases. When ϵ is small enough (10^{-3}) the approximate posterior distribution will not change very much with further decreases in ϵ. This provides a check on whether or not the estimated posterior has been obtained: if reductions in ϵ do not produce changes in the estimated posterior, then the estimate has converged to its final target.

Each panel in Figure 2.4 shows, in the upper right corner, the Kullback-Leibler distance between the estimated and actual posteriors. The Kullback-Leibler distance is a statistic that measures the discrepancy between two density functions and is popular in a number of statistical applications (Beaumont et al., 2009, Kullback et al., 1987). Using the distance as a measure of accuracy of the estimated posterior, we can see that the accuracy under $\epsilon = 1$ is poorer than under $\epsilon = 10^{-3}$ or 10^{-5}, and that there is not much change in the accuracy for $\epsilon \leq 10^{-3}$. 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 distance function $\rho_1(X, Y)$ based on the mean difference produced more accurate posterior estimates than the other functions, but even the other functions, $\rho_2(X, Y)$ and $\rho_3(X, Y)$, produced estimates that were close to the true posterior. It must be noted, 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.

2.6 A Hierarchical Binomial Model

Both the binomial and the exponential examples demonstrated the accuracy of the ABC algorithm. These examples show that, if a model can be simulated, the algorithm can recover reasonable estimates of the posterior distributions even for continuous measurements which make evaluation of distance more difficult.

An important extension of Bayesian procedures is the construction of hierarchical models (Shiffrin et al., 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 more correct responses 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. The example in this section will extend the binomial model in Section 2.4 to a hierarchical design to further demonstrate the capabilities of the ABC algorithm.

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.

2.6.1 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.

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

$$\operatorname{logit}(p) = \log\left(p/(1-p)\right).$$

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

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

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

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

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

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

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

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 $\sigma_\mu^2 = 10,000$. This is a diffuse prior that gives approximately equal weight to values for μ ranging from -5000 to 5000. It is important to note that being more vague (e.g., setting $\sigma_\mu^2 = 10^{10}$) 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 $\text{logit}(.99) = -\text{logit}(.01) = 4.5951$. Thus, a still reasonable prior could be much more narrow (e.g., $\sigma_\mu^2 = 100$), without greatly affect the estimate of the posterior. We then set $\alpha_\tau = \beta_\tau = 0.1$. This also is an overly diffuse specification, following the same argument.

2.6.2 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 from the distribution of $\text{logit}(p_i)$, which was Gaussian with mean $\mu = -1.0$ and variance $\tau^2 = 0.5$. These values were then transformed back to the probability scale and used as the parameter for 100 $\text{Bernoulli}(p_i)$ random variables that simulated the detection responses Y_{ij} for Subject i.

We implemented Algorithm 3, shown in Figure 2.5, 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 τ. 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 distance for each subject computed as for the binomial example of Section 2.4.

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 efficient approach to this problem in the General Discussion.) For this algorithm we used 1,000 particles.

2.6.3 Results

Figure 2.6 shows the estimated posteriors for p_i, μ and $\log(\tau)$ for three selected levels of ϵ (rows; specifically, $\epsilon_1 = 0.1, \epsilon_2 = .05$, 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 $\epsilon \leq .03$. The left panel of Figure 2.6 shows the approximate marginal posterior distributions of $p_i|\mu, \tau^2, Y$ 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 gets smaller the approximate marginal posteriors for p_i approach the true marginal posteriors.

The right panels of Figure 2.6 show the approximate joint posterior distribution of the hyperparameters μ and $\log(\tau)$. We plot the parameter τ on a log scale to shrink 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 goes to 0 the variance of the joint distribution shrinks. The true sampled values of μ and $\log(\tau)$ 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 value of ϵ. Although the true values of the hyperparameters are contained within the marginal 95% credible intervals for μ and $\log(\tau)$, 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 crosshairs of 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 Figure 2.6 (left panel). For this model we have no expression for the true joint posterior of μ and $\log(\tau)$, but we can surmise that the estimated posteriors reflect the sampled values for the p_is and the choice of priors.

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.

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 and Steyvers, 1997). REM is a simulation model without an explicit likelihood.[3] This model serves as our final example.

2.7 Retrieving Effectively from Memory (REM)

The REM model can be applied to 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

[3] An unpublished manuscript by Montenegro et al. (2011) derives the likelihood for REM. The likelihood is very complex and we will not discuss it here.

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 and 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 then current memory models (see Glanzer et al., 1993, for a review). These regularities included the list strength effect, the mirror effect and the slope of the ROC curve. The list strength effect occurs when some items are strengthened relative to others (i.e., presented more often than others during study). Strength manipulations can have very small effects, or strong items can be better recognized than weak items. 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 et al., 1992, 1994).

REM is a global memory model, which means that recognition responses are based on a calculation of familiarity, which in turn is based on a representation of 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 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. 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.[4] 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

$$\lambda_j = (1-c)^{n_{jq}} \prod_{i=1}^{\infty} \left[\frac{c + (1-c)g(1-g)^{i-1}}{g(1-g)^{i-1}} \right]^{n_{ijm}} . \tag{2.4}$$

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

$$\Phi = \frac{1}{n} \sum_{j=1}^{n} \lambda_j, \tag{2.5}$$

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.

[4] 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 and McClelland, 2006).

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 – there is no likelihood (but see Footnote 3). 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 and 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, Nosofsky et al., 2011). 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.

2.7.1 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 et al. (2011a) for a hierarchical REM model fit to the data of Dennis et al. (2008).

We simulate REM in three stages. First, we generate a stimulus set using the parameter g. Next, we fill in the episodic matrix during the study phase using the parameters g, u and c. Finally, the test phase is completed by using the same parameters g, u and c and Equation 2.4. 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 weight 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.

2.7.2 Estimating the Posterior

The data we observe in the 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., 2011a). 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 3 and also Myung et al., 2007).

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

$$\rho(X, Y) = \frac{1}{2C}\left[\sum_{j=1}^{C}\left|(X_{j,\text{FA}} - Y_{j,\text{FA}})/N_{\text{NEW}}\right| + \sum_{j=1}^{C}\left|(X_{j,\text{HIT}} - Y_{j,\text{HIT}})/N_{\text{OLD}}\right|\right], \quad (2.6)$$

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 to the 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 1,000 particles to estimate the posteriors.

2.7.3 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 Figure 2.7 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.

Figure 2.7 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 of copying a feature u and the probability of copying it correctly c. 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 postive correlation between c and g is quite strong. 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 Equation 2.4). 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 being copied (high g and high u). However, there is a large degree of variability in the posteriors. 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 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 posteriors.

2.8 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 nonexistent 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, in contrast to MCMC techniques, are very easy to use. Once developed, the basic algorithm can be easily applied to new models.

Although the ABC approach provides a method to circumvent nonexistant or ill-behaved likelihood functions, this approach is certainly not without a cost. As we mentioned in the introduction, sometimes the ABC approach is computationally more expensive than MCMC. However, with modern multi-core computers, computation time is becoming less of an issue. One important feature of the ABC PMC algorithm is that, unlike MCMC methods, it 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 effective for most problems in cognitive modeling. Second, the choice of the distance function

$\rho(X,Y)$ will be determined 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. Most of the distance functions we presented for our examples were limited in their range, and so our choices for ϵ were not hard to come by. 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., 2011a,b).

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., 2011a). 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 neurologically-plausible leaky competing accumulator model (Usher and 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 something less 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 previously not have considered.

```
Given data and assumed simulation-based model $Y \sim \text{Model}(\theta)$, tolerance threshold
$\epsilon$, and prior distribution $\pi(\theta)$:
for $1 \leq i \leq N$ do
    while $\rho(X, Y) > \epsilon$ do
        Sample $\theta^*$ from the prior: $\theta^* \sim \pi(\theta)$
        Generate data $X$: $X \sim \text{Model}(\theta^*)$
        Calculate $\rho(X, Y)$
    end while
    Store $\theta_i \leftarrow \theta^*$
end for
```

Figure 2.1: A rejection sampling algorithm to estimate the posterior distribution of a parameter θ given data Y.

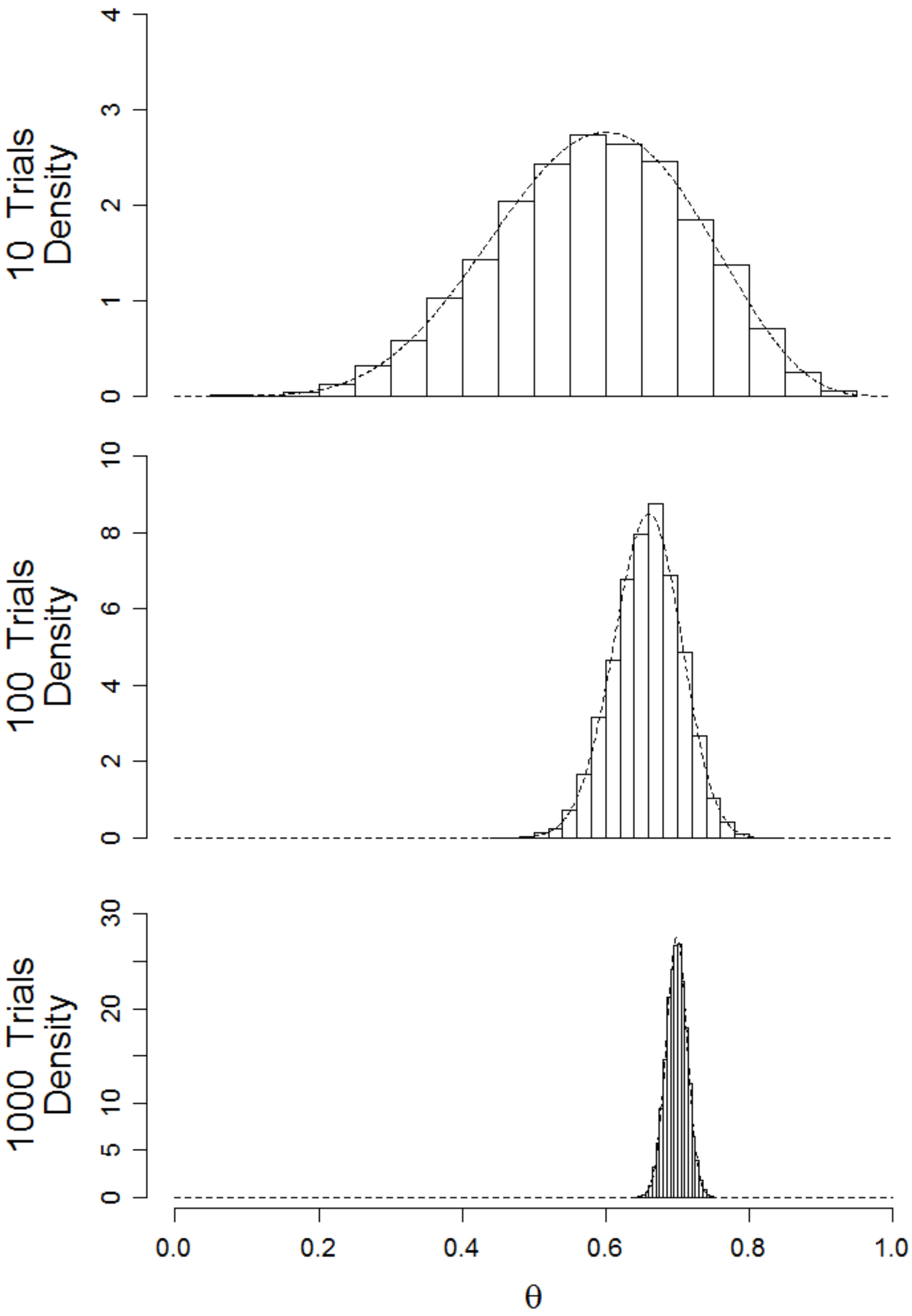

Figure 2.2: The posterior distributions for three different subjects performing $n = 10$ (top panel), $n = 50$ (middle panel) and $n = 100$ trials. The dashed curve shows the true posterior distribution.

```
Given data $Y$ and simulation-based model $Y \sim \text{Model}(\theta)$ :
At iteration $t = 1$,
for $1 \leq i \leq N$ do
    while $\rho(X, Y) > \epsilon_1$ do
        Sample $\theta^*$ from the prior: $\theta^* \sim \pi(\theta)$
        Generate data $X$: $X \sim \text{Model}(\theta^*)$
        Calculate $\rho(X, Y)$
    end while
    Set $\theta_{i,1} \leftarrow \theta^*$
    Set $w_{i,1} \leftarrow \dfrac{1}{N}$
end for
Set $\tau_1^2 \leftarrow 2\text{Var}(\theta_{1:N,1})$
for $2 \leq t \leq T$ do
    for $1 \leq i \leq N$ do
        while $\rho(X, Y) > \epsilon_t$ do
            Sample $\theta^*$ from the previous iteration: $\theta^* \sim \theta_{1:N,t-1}$ with probabilities $w_{1:N,t-1}$
            Perturb $\theta^*$ by sampling $\theta^{**} \sim N(\theta^*, \tau_{t-1}^2)$
            Generate data $X$: $X \sim \text{Model}(\theta^{**})$
            Calculate $\rho(X, Y)$
        end while
        Set $\theta_{i,t} \leftarrow \theta^{**}$ and $w_{i,t} \leftarrow \dfrac{\pi(\theta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q_f\left(\theta_{j,t-1} | \theta_{i,t}, \tau_{t-1}\right)}$
    end for
    Set $\tau_t^2 \leftarrow 2\text{Var}(\theta_{1:N,t})$
end for
```

Figure 2.3: The ABC PMC algorithm for estimating the posterior distribution of θ.

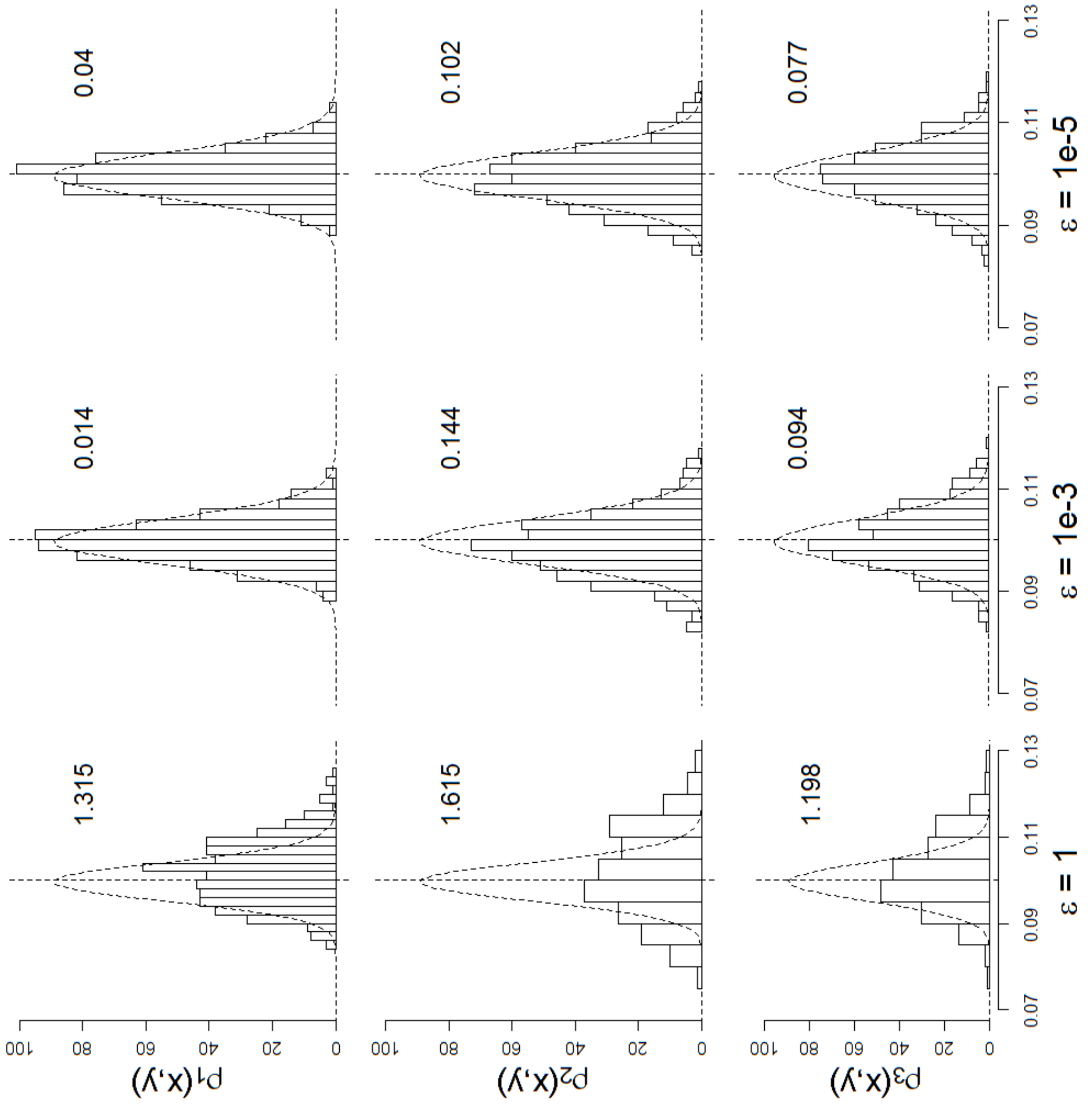

Figure 2.4: The posterior distribution of λ at three different tolerance thresholds (columns; $\epsilon = 1, 10^{-3}, 10^{-5}$) and three different $\rho(X, Y)$ functions (rows; see text for details). The dashed curve shows the true posterior distribution and the dashed vertical lines shows the true parameter value. The numbers in the upper right-hand corner of each panel are the Kullback-Leibler distances between the estimated and true posteriors.

```
Given data and model $Y \sim \text{Model}(\delta, \theta_{1:S})$:
for $1 \le i \le N$ do
    while $\rho(X, Y) > \epsilon_1$ do
        Sample $\delta^*$ from the hyper prior: $\delta^* \sim \pi_H(\delta)$
        Generate $\theta^*_{1:S}$ given the hyperparameter $\delta^*$: $\theta^*_{1:S} \sim \pi_L(\theta|\delta^*)$
        Generate data $X$: $X \sim \text{Model}(\delta^*, \theta^*_{1:S})$
        Calculate $\rho(X, Y)$
    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 $\tau_1^2 \leftarrow 2\text{Var}(\delta_{1:N,1})$
for $2 \le t \le T$ do
    for $1 \le i \le N$ do
        while $\rho(X, Y) > \epsilon_t$ do
            Sample $\delta^*$ from the previous iteration $\delta^* \sim \delta_{1:N,t-1}$ with probabilities $w_{1:N,t-1}$
            Perturb $\delta^*$ by sampling $\delta^{**} \sim N(\delta^*, \tau_{t-1}^2)$
            Generate $\theta^*_{1:S}$ given the hyperparameter $\delta^{**}$: $\theta^*_{1:S} \sim \pi_L(\theta|\delta^{**})$
            Generate data $X$: $X \sim \text{Model}(\delta^{**}, \theta^*_{1:S})$
            Calculate $\rho(X, Y)$
        end while
        Set $\delta_{i,t} \leftarrow \delta^{**}$, $\theta_{i,t,1:S} \leftarrow \theta^*_{1:S}$, and $w_{i,t} \leftarrow \frac{\pi_H(\delta_{i,t})}{\sum_{j=1}^{N} w_{j,t-1} q_f(\delta_{j,t-1}|\delta_{i,t}, \tau_{t-1})}$.
    end for
    Set $\tau_t^2 \leftarrow 2\text{Var}(\delta_{1:N,t})$
end for
```

Figure 2.5: A hierarchical ABC PMC algorithm.

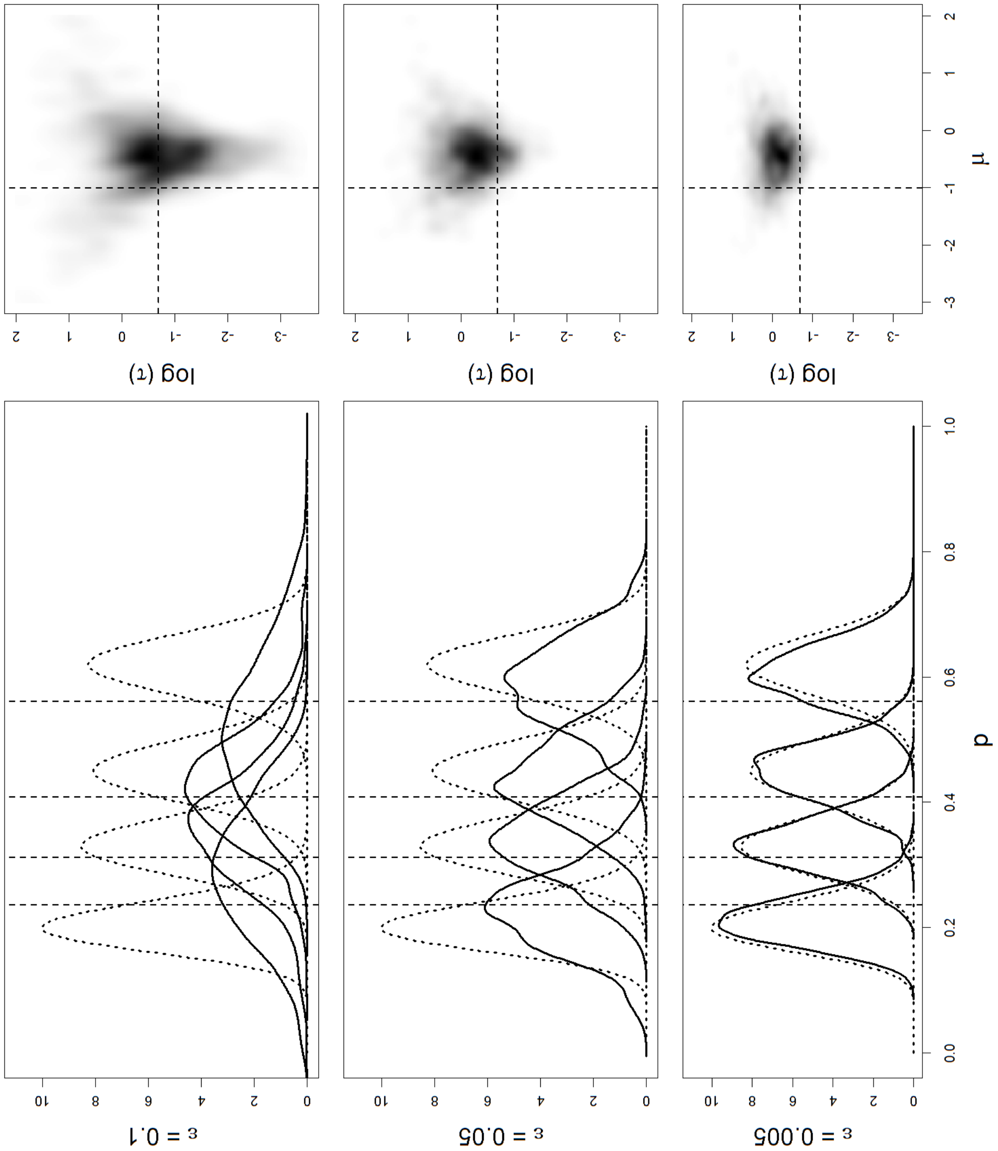

Figure 2.6: The posterior distributions for the probability of a correct response for four subjects (left panel; solid lines) at three levels of ϵ (rows). The true posterior distributions are shown by the dashed distributions and the true sampled values are shown by the vertical lines (left panel). The right panel shows contours of the approximate joint posterior distribution for the hyperparameters μ and $\log(\tau)$).

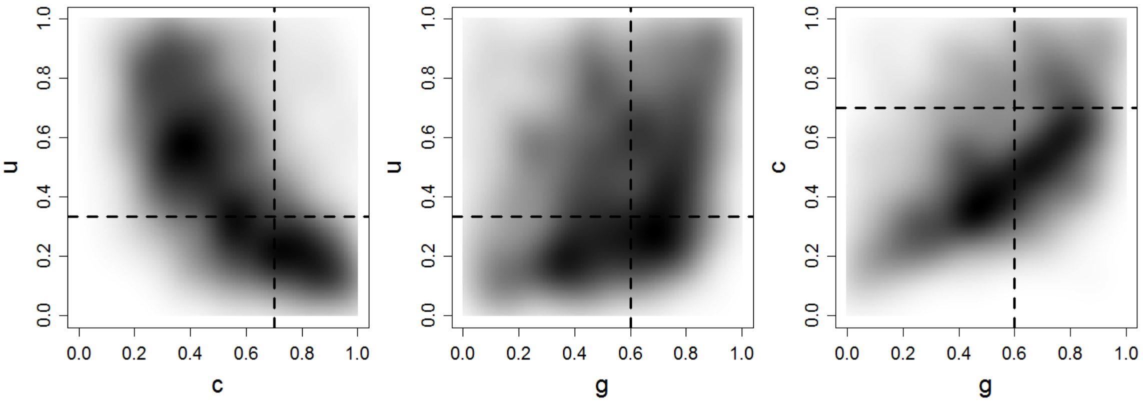

Figure 2.7: The estimated joint posterior distributions for each pair of the parameters in REM: c versus u (left panel), g versus u (middle panel) and g versus c (right panel). The dashed lines show the parameter values used to generate the data.

Chapter 3: Hierarchical Approximate Bayesian Computation

The previous chapter was useful in demonstrating the applicability of the ABC approach for a number of simple models. We argued that the application of the ABC PMC algorithm for the hierarchical binomial model, while sufficient, was inefficient. Given the simplicity of the hierarchical binomial model, more sophisticated algorithms must be developed before the ABC approach can be used in a more general setting.

In this chapter, I will demonstrate how to estimate hierarchical models using ABC. As mentioned in the introduction, this is a difficult task in ABC because of the accept/reject nature of the algorithms. With more than a few parameters, the estimation procedure can take a very long time, and the approximation obtained is likely to be less accurate. The first algorithm I present is a two-stage algorithm, developed by Bazin et al. (2010). This algorithm breaks the estimation into two components. This interesting innovation is considerably faster than the algorithm presented in Chapter 2, but it involves data generation at two stages, which can be inefficient. I then present a new mixture algorithm, which embeds the ABC technique into a Gibbs sampler. This algorithm allows for estimation of the hyperparameters based on standard Bayesian procedures, and localizes the ABC sampling. This procedure reduces the dimensionality of the model back to the non-hierarchical structures discussed in the previous chapter. Where possible, I compare the estimates obtained using the algorithms to their true Bayesian posteriors.

3.1 A Discussion of Algorithm 3

For the rest of the dissertation, I will denote the vector of hyperparameters as ϕ, and the matrix of individual-level parameters as θ. When referring to a specific vector of parameters for the jth subject, I use the notation θ_j. Similarly, I use Y_j (X_j) when referring to the partition of the observed (simulated) data specific to the jth subject. In the previous chapter, I presented a basic extension of the ABC PMC algorithm. This extension involved combining the hyper- and individual-level parameters into a single vector, and then proceeding as in Algorithm 2. This method, while correct is hopelessly inefficient. If the distribution of θ_js is highly variable, we will find that rejection rates can be overwhelming. The empirical Bayes method is one possible solution to this problem. In empirical Bayes, the hyperparameter ϕ is first estimated using classical techniques, and subsequently used to generate the θ_j^*s (Line 6 of Algorithm 3). Pritchard et al. (1999) used a moment-based estimator for the hyper-mean and hyper-variance in the mutation rates of the individual-level units (loci).

Another method is to allow the simulated data X_j to be arranged in any way possible so that it optimizes the acceptance rate. This is an acceptable procedure because the θ_js are *exchangeable* (see Gelman et al., 2004). This means that the likelihood is invariant to the ordering of the individual-level parameters. This feature of hierarchical designs carries over to the ABC setting, because we are estimating the likelihood function. Thus, in the comparison of simulated data X_j to observed data Y_j, j is actually an arbitrary index. This is the key idea behind Hickerson et al. (2006), Hickerson and Meyer (2008) and Sousa et al. (2009), who focus on ranking the simulated data X_j based upon some summary statistic $S(X_j)$ and comparing this statistic to the similarly-ranked observed data Y. If the simulated and observed data are both ranked using the same summary statistic, this will tend to lead to larger acceptance rates.

Although the ranking approach is significantly more efficient, it can still be extremely slow. For large data sets or multiple hierarchies (e.g., groups of subjects replicated among different conditions), this basic approach will not be practical for use. One reason for this inefficiency appears in Line 5 of Algorithm 3. If we sample a poor ϕ^*, then the subsequent θ_j^*s are likely to also be poor. When simulated, the θ_j^*s are likely to produce data X_j that will not pass inspection.

The previous chapter demonstrated that the basic ABC algorithm can be easily extended hierarchically. Although the procedure was straightforward, the computational complexity for the simple binomial model was great. Figure 2.6 shows that the smallest value for ϵ that I could obtain was 0.005. Although this is very near zero, it suggests that when the number of parameters increases, the smallest ϵ value we could obtain in a reasonable amount of time is likely to be larger than zero. Clearly, more sophisticated algorithms are required for practical implementation of the ABC approach for hierarchical models.

3.2 Two-Stage ABC

In this section, I describe the algorithm introduced in Bazin et al. (2010), which I will refer to as the "two-stage" algorithm. This algorithm first evaluates a proposed ϕ^* before using it in subsequent generations of θ_j^*. The key idea behind the algorithm is *conditional independence*, which means that if a probability distribution does not depend on some parameter it can safely be ignored because it will not influence the probability distribution. This allows us to factor the joint posterior distribution into a product of marginal posterior distributions,

$$\pi(\phi, \theta|Y) = \prod_{j=1}^{J} \pi(\theta_j|Y_j, \phi)\pi(\phi|Y). \tag{3.1}$$

This factorization shows that inference for the hyper- and individual-level parameters can be performed independently. To do so, we require the use of two discrepancy functions, one for the individual-level parameters $\rho(X_j, Y_j)$ and one for the hyperparameters $P(X, Y)$. These discrepancy functions also need tolerance thresholds, ϵ for $\rho(X_j, Y_j)$ and η for $P(X, Y)$, which will usually be different. Notice that this new discriminant function $P(X, Y)$ is concerned with the whole data set, not just the partition exclusive to the jth subject. This will allow us to evaluate ϕ^* at both levels of the model. We can now replace the equality in Equation 3.1 with the approximation

$$\pi(\phi, \theta|Y) \approx \prod_{j=1}^{J} \pi(\theta_j|\rho(X_j, Y_j) \leq \epsilon, \phi)\pi(\phi|P(X, Y) \leq \eta), \tag{3.2}$$

in accordance with the ABC approximation given in Chapter 2.

Bazin et al. (2010) provide two HABC algorithms, both of which use Equation 3.2. The first algorithm uses a procedure very similar to Algorithm 3. It samples from the joint prior $\pi(\phi, \theta)$, generates data, and then uses ABC to evaluate these proposals. However, if a hyperparameter does not satisfy $P(X, Y) \leq \eta$ then the hyperparameter and the resulting individual-level parameters are not considered further. However, this approach still suffers from unnecessary computation, because data are generated at each of the individual levels, prior to evaluating $P(X, Y) \leq \eta$. The second algorithm – shown in Algorithm 4 – presented in Bazin et al. (2010) breaks the computation into two stages. This algorithm also allows us to reject ϕ^* on the basis of a marginal comparison to the data. First, a pool of hyperparameters is created by evaluating $P(X, Y)$, as explained above. Once the pool is large enough, the algorithm moves to a second stage. In this stage, the proposal hyperparameter ϕ^* is sampled from the pool generated in the first stage and then it is conditioned upon in order to generate individual-level proposals θ_j^*. These θ_j^*s are then evaluated by the discrepancy function $\rho(X_j, Y_j)$. If the θ_j^*s satisfy $\rho(X_j, Y_j) \leq \epsilon$, then both ϕ^* and θ_j^* are retained to form the approximation in Equation 3.2. Thus, the final approximate marginal posterior distribution $\pi(\phi|\theta, Y)$ is obtained by rejecting proposals at two stages. This two-stage rejection scheme usually leads to a swelling and contracting of the estimated posterior distribution of ϕs, which can be inefficient.

Furthermore, Bazin et al. (2010) acknowledge that breaking the estimation into two stages introduces an approximation to the joint posterior distribution $\pi(\phi, \theta|Y)$ that is different from the approximation due to ABC alone. Beaumont (2010) argues that this difference is small, and that the effect is attenuated with an increasing number of subjects. This is concerning in psychology, when the number of subjects is far less than a typical study in biology to which he is referring (see Beaumont, 2010, for some applications). In the next section, I will show that this difference is indeed superfluous, and can be corrected by embedding the ABC approach into a standard Bayesian sampler.

Due to the conditioning on ϕ in the second stage of Algorithm 4, it is vital that the hyperparameters converge to the correct posterior distributions in the first stage. Otherwise, the individual-level parameters may encounter serious difficulty in convergence, costing valuable time and accuracy. To see this, consider the following situation where the individual-level parameters θ_j arise from a normal distribution with mean ϕ_1 and variance ϕ_2. Now, suppose the distribution of accepted ϕ_1s in Step 1 is approximately correct, but the distribution of accepted ϕ_2s is too diffuse. That is, the joint distribution of the hyperparameters is centered on the mean of the target joint posterior distribution $\pi(\phi_1, \phi_2|\theta, Y)$, but it possesses too much variance. This will lead to more extreme θ_j^*s being rejected at the second stage, costing valuable time. Poor accuracy might also result if the distribution of accepted ϕ_2s is too concentrated. This can also lead to high rejection rates, but more importantly, the individual-level distributions may suffer due to the sparse number of proposed θ_j^*s in the high density regions of $\pi(\theta_j|\phi, Y)$.

A final difficulty arises in developing the discriminant function $P(X, Y)$. Bazin et al. (2010) require that the discriminant function be based on symmetric summary statistics, that capture the distribution of the entire data set. Bazin et al. used principal component analysis in combination with the post-simulation correction presented later in this chapter. Bazin et al. also suggest the use of multivariate partial least squares in combination with their linear model. These multivariate $P(X, Y)$ functions are more difficult to specify than their univariate counterparts.

As an example, suppose the data for a group of subjects arise from a normal distribution with means μ_j and variances σ_j^2. To form a hierarchy, we could assume that the μ_js come from a common normal distribution with mean μ and variance σ. We could further assume that the variances σ_j^2 come from a gamma distribution with parameters α and β. When deciding upon a discriminant function $\rho(X, Y)$, we could simply compare the distributions of the simulated data X_j and the observed data Y_j through some statistics such as the difference in means or Kolmogorov-Smirnov test statistics. However, deciding upon a $P(X, Y)$ function is much harder because there are two variance components in this model. The first variance is on the distribution of the means μ_j, and the second is on the variance of the variances σ_j^2. For P to be a legitimate discriminant function, it would have to delineate between the hyperparameter σ governing the variance of the means μ_j and the hyperparameters α and β, which govern the distribution of σ_j^2.

In the next section, I will present a mixture algorithm which solves the above problems. First, it uses a Gibbs sampling approach to allow each parameter in the model to inform the other parameters, without data generation at multiple stages. Second, the algorithm combines standard Bayesian techniques and the ABC approach in order to avoid approximating the marginal posterior distributions of the hyperparameters. This approach removes the need for determining $P(X, Y)$ altogether.

3.3 A Mixture Algorithm

The algorithm presented in Bazin et al. (2010) is a significant improvement over the basic approach presented in Chapter 2. However, the two-stage algorithm is not without faults. First, the success of the algorithm hinges upon how well the algorithm recovers the target distribution of the hyperparameters. Misestimation in the first step of Algorithm 4 can be costly. Second, the algorithm only approximates the marginal posterior distribution of the hyperparameters (see Bazin et al., 2010, Beaumont, 2010). Third, the specification of the hyper discriminant function $P(X, Y)$ can be very difficult, especially for models with multiple hyperparameters. Finally, because the estimation of the posterior distribution of the hyperparameters is based on ABC techniques, it involves an approximation of the conditional distribution. In this section, I will show that this can be improved upon by sampling directly from the conditional posterior distribution using well-accepted techniques. The key insight to the algorithm is the fact that the conditional distribution of the hyperparameters does not depend on the likelihood function. This, in combination with a mixture of Gibbs sampling and ABC sampling provides an algorithm that offers a significant improvement in convergence. To implement the algorithm, we need to derive the conditional distribution of each of the parameters. Specifically, the conditional distribution of the individual-level parameters θ given all other parameters is given by

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

given the independence of the θ_js and Y_js. For a particular subject j, the conditional distribution then reduces to

$$\pi(\theta_j \mid Y, \phi) \propto L(\theta_j \mid Y_j)\pi(\theta_j \mid \phi).$$

Here, the conditional distribution of each of the θ_js depends on the likelihood function. However, the likelihood function only depends on the partition of the data corresponding to the jth subject. Thus, the problem simplifies to performing ABC for each subject, and we can approximate each conditional distribution by

$$\pi(\theta_j \mid Y, \phi) \approx \pi(\theta_j|\rho(X_j, Y_j) \leq \epsilon, \phi). \tag{3.3}$$

Noting that $\pi(\phi \mid \theta) \propto \pi(\theta \mid \phi)\pi(\phi)$, the conditional distribution of the hyperparameters is given by

$$\begin{aligned} \pi(\phi \mid Y, \theta) &\propto L(\theta \mid Y)\pi(\theta \mid \phi)\pi(\phi) \\ &\propto \pi(\theta \mid \phi)\pi(\phi) \\ &\propto \left[\prod_{j=1}^{J} \pi(\theta_j \mid \phi)\right] \pi(\phi). \end{aligned} \tag{3.4}$$

Because ϕ influences the likelihood only through the parameter θ, the conditional distribution of ϕ does not depend on it. This insight allows us to sample from the conditional distribution using standard techniques. If this distribution has a convenient form, we can sample directly from it. Otherwise, any numerical technique can work here, such as discretized sampling (e.g., Gelman et al., 2004), adaptive rejection sampling (Gilks and Wild, 1992), or MCMC (e.g., Robert and Casella, 2004).

Algorithm 5 uses the above ideas to estimate the posterior distributions of the parameters ϕ and θ. I use $\theta_{j,k}$ to denote the kth individual-level parameter for Subject j and $\theta_{j,k,i}$ to denote the ith sample of $\theta_{j,k}$ obtained on iteration i. For the hyperparameters, $\phi_{m,i}$ is the value of the mth hyperparameter ϕ_m on iteration i.

In the Gibbs sampling framework, we draw samples for θ_m conditional on all other parameters in the model, including the other variables contained in the vector θ. To denote this conditioning, it is common to use a negative subscript; that is, on iteration i, the $M-1$ other components of the vector θ not including the mth component θ_m is written as

$$\theta_{i,-m} = \{\theta_{i,1}, \ldots, \theta_{i,m-1}, \theta_{i-1,m+1}, \ldots, \theta_{i-1,M}\}.$$

Note that the vector $\theta_{i,-m}$ contains components sampled on the current iteration i as well as components that have yet to be sampled. Defining $\theta_{i,-m}$ in this way, we are specifying the use of the current value for each component in the vector θ. In addition, I use the notation $\theta_{1:M} = \theta$ to explicitly specify using all M components in the vector θ.

Algorithm 5 is much more flexible than other HABC algorithms. We are able to use any sampling methods available to sample from the conditional distribution of ϕ. In addition, the algorithm allows for subject-specific discriminant functions and tolerance thresholds. This can be useful when the data do not arise from their assumed models, and allowing for closer fits to some subjects improves convergence speed. For example, if a model simply does not predict a certain performance (e.g., an inverse list length effect in REM; Shiffrin and Steyvers, 1997), then the algorithm will have a difficult time producing data that are similar enough to the observed data to satisfy the tolerance threshold.

Algorithm 5 specifies that each of the M components be drawn sequentially, but this is not necessary. One could choose to “block” certain components together and sample from their joint conditional distribution. This can be useful when parameters are systematically related to one another, and are not independent. For example, many models of RT involve a drift rate and threshold parameter, which are often highly related to one another (Wagenmakers et al., 2007). We could choose to sample from the joint distribution of these two parameters, specifying a degree of relatedness (such as a correlation in a multivariate normal distribution) that would attenuate rejection rates.

One may also find it convenient to remove certain parameters from blocks which have no effect on the distribution of simulated data. For example, most models of RT have a nondecision component t_0, which is assumed to model processes of the decision that are not of immediate interest (e.g., Brown and Heathcote, 2005, 2008, Ratcliff, 1978, Usher and McClelland, 2001). Typically, this parameter simply shifts the RTs, and otherwise does not affect the distribution. If simulating data for the model is time-consuming, it can be much more efficient to perform ABC on all of the parameters excluding t_0 first, because it does not affect the shape of the distribution, only the location. Then, conditional on the other parameters in the model, we can sample a t_0^* to evaluate. This is helpful because the simulated data X_j do not need to be simulated again, the data can simply be shifted by the proposed t_0^* in the subsequent step so that $X' = X + t_0^*$. This can improve the speed of the algorithm by offsetting the computational cost of rejection.

3.4 A Hierarchical Poisson Model

Although Algorithm 5 is anchored in standard Bayesian methods, an example may introduce the algorithm more easily. A common problem in genetics is inferring mutation rates at different locations on genes, known as loci. Although this problem is not common in psychology, inferring rates for say, sequential-sampling models certainly is (e.g. Ratcliff and Tuerlinckx, 2002, Tuerlinckx, 2004). For this illustrative example, we consider data arising from a Poisson distribution and our goal will be to estimate the posterior distributions of the rate parameters in a hierarchical design.

The data could come from a variety of experimental paradigms, but suppose we were interested in modeling the rate at which $J = 4$ teenagers, who frequently play violent video games, engage in negative behaviors (e.g. behaving aggressively; see Carnagey and Anderson, 2004, for a literature review). The data we collect might be the number of "incidents" or scenarios in which the subject behaves aggressively per week. Further suppose we monitor these subjects for $n = 100$ weeks. For this example, we are interested in not only the incident rate for each of the four subjects, but also the incident rate for the group.

Model

To begin building a hierarchical Bayesian model, we first assume that the number of incidents Subject j has during the tth week can be modeled with a Poisson distribution

$$Y_{j,t} \mid \theta_j \sim \text{Poisson}(\theta_j),$$

where θ_j is the incident rate of the jth subject. The Poisson distribution is a single-parameter distribution often used in statistics to model low probability events. To extend the model hierarchically, we will assume that the incident rates $\theta =\{\theta_1, \theta_2, \ldots, \theta_J\}$ come from an exponential distribution with rate λ given by

$$\theta_j \mid \lambda \sim \text{Exp}(\lambda).$$

The parameter λ represents the incident rate for the group of subjects. If one were interested in comparing this group of active gamers to a control group, in which video games were not played, a comparison of the parameter λ for the two groups would be appropriate. I will assume that we know nothing about the distribution of λ, so we might specify a noninformative prior for λ which follows the gamma distribution

$$\lambda \sim \Gamma(0.1, 0.1).$$

This model has only one hyperparameter, and so we need only derive the conditional distribution for λ, given by

$$\begin{aligned}
\pi(\lambda \mid \theta, Y) &\propto \left[\prod_{j=1}^{J} \pi(\theta_j \mid \lambda)\right] \pi(\lambda) \\
&= (0.1\lambda)^{0.1} \lambda^{-1} \Gamma(0.1)^{-1} \exp(-0.1\lambda) \left[\prod_{j=1}^{J} \lambda \exp(-\lambda\theta_j)\right] \\
&= 0.1^{0.1} \lambda^{J+0.1-1} \Gamma(0.1)^{-1} \exp\left[-\lambda \left(\sum_{j=1}^{J} \theta_j + 0.1\right)\right],
\end{aligned}$$

from which we can determine that

$$\lambda \mid \theta, Y \sim \Gamma\left(\lambda \,\middle|\, J + 0.1,\ \sum_{j=1}^{J} \theta_j + 0.1\right), \tag{3.5}$$

a gamma distribution with shape $J + 0.1$ and rate $\sum_{j=1}^{J} \theta_j + 0.1$. Thus, the conditional distribution has a convenient form from which we can sample directly. Again notice that this conditional distribution does not depend on the data Y, but the data remain in the notation for consistency. For the individual-level parameters θ, the conditional posterior distribution is a gamma distribution with shape $1 + \sum_{t=1}^{n} Y_{j,t}$ and rate $\lambda + n$, or

$$\theta_j \mid \lambda \sim \Gamma\left(1 + \sum_{t=1}^{n} Y_{j,t},\ \lambda + n\right). \tag{3.6}$$

Results

Having both of these equations implies that we could perform standard Bayesian analysis using Gibbs or rejection sampling. However, to demonstrate the mixture algorithm, we will remove our mathematical sophistication and assume that the likelihood function can not be easily calculated. To use Algorithm 5, we first select a $\rho(X, Y)$ function. For this example, I chose to compare the absolute differences in the means of the data, given by

$$\rho(X_j, Y_j) = \frac{1}{n} \left| \sum_{t=1}^{n} X_{j,t} - \sum_{t=1}^{n} Y_{j,t} \right| . \tag{3.7}$$

Next, we need to choose an ϵ that is small enough to satisfy the approximation in Equation 3.3. For this example, I chose $\epsilon = 10^{-20}$.

In addition to using Algorithm 5 to fit this data, I also used the two-stage algorithm (see Algorithm 4). For Algorithm 4, in addition to specifying $\rho(X, Y)$ and ϵ, we must specify $P(X, Y)$ and η. In this example, it is convenient to again use the absolute differences in the means given by

$$P(X, Y) = \frac{1}{Jn} \left| \sum_{j=1}^{J} \sum_{t=1}^{n} X_{j,t} - \sum_{j=1}^{J} \sum_{t=1}^{n} Y_{j,t} \right|$$

and set $\eta = \epsilon = 10^{-20}$.

Comparing the mixture algorithm to Algorithm 4 may not be very useful in this situation because the hyperparameter discriminant function is easily specified. As discussed, one of the disadvantages of using the two-stage algorithm is that it can be hard to find a convenient discriminant function for more complicated models (e.g., multiple hyperparameters).

Finally, because each of the conditional distributions have convenient forms, we can easily compare the two algorithms to their fully-Bayesian counterparts. To do so, I will compare the estimates to the true posterior distributions by using the Kullback-Leibler divergence statistic (see Chapter 2).

For both of the algorithms, 5,000 samples were drawn with a burn-in period of 100 samples. With only five parameters, estimation was easy for both algorithms, taking only a few minutes to complete. Standard techniques were used to asses convergence (see Gelman et al., 2004).

To generate the data, I sampled four values for θ from an exponential distribution with $\lambda = 1$. These four values were then used to generate 100 samples of Y_j from a Poisson distribution with rate θ_j for each subject j. The estimated posterior distributions obtained using Algorithms 4 and 5 were compared to the true posterior distributions by calculating the Kullback-Leibler divergence statistics.

Figure 3.3 shows the estimated posterior distributions using Algorithm 5 (left panel) and Algorithm 4 (right panel). The dashed vertical line represents the true value of the parameter used to generate the data. Overlaying each histogram is the true posterior distribution. Figure 3.3 shows that the two approaches are very similar. However, there is a slight misestimation when using Algorithm 4. This slight difference is also reflected in the Kullback-Leibler statistics, presented in each of their respective panels. Although both distances are very small, the mixture algorithm outperforms the two-stage algorithm by a factor of about three. This result should not be surprising because the mixture algorithm's estimated posterior is obtained by sampling directly from the conditional distribution. By contrast, the two-stage algorithm approximates both this distribution *and* the individual-level posteriors.

Although the close agreement of the marginal posterior distributions for the hyperparameter is comforting, the real test of the mixture algorithm is in the recovery of the individual-level parameters. Figure 3.4 shows these estimated posteriors. The vertical dashed line represents the true sampled value, which was used to generate the data for each subject. The histograms represent the estimated posterior obtained using the mixture algorithm. Overlaying each histogram is the estimated posterior obtained by using the two-stage algorithm (dashed gray lines) and the true posterior distribution (solid black lines). For each of the four subjects, we see very close agreement between both of the estimates and the true posterior. This result validates the use of ABC for this model.

The results of this example suggest that the mixture algorithm is able to recover the posterior distributions of both the hyper and individual-level parameters well. For the hyperparameters, the mixture algorithm outperformed the two-stage algorithm, but for the individual-level parameters this discrepancy is negligible. However, for more complicated models, the differences between these two approaches will likely be more dramatic.

3.5 Improving the Sampler

The example above was successful in demonstrating that the mixture algorithm is capable of recovering the true posterior distributions. In addition, even for this simple example, it showed that the mixture algorithm outperforms the two-stage algorithm when estimating the posterior distribution for the hyperparameters. While the algorithm works well for the simple example above, there are many improvements that can be made to decrease the computational time and increase the accuracy of the estimated posterior. In this section, I present several methods for improving the algorithm.

At the hyper level, Algorithm 5 is capable of using any technique to sample from the conditional distribution. At the individual level, Algorithm 5 specifies the use of rejection sampling as in Algorithm 1 in Chapter 2. As previously noted, rejection sampling can be inefficient. This is because the search for a suitable θ_j^* is a global one, guided only by the hyperparameters sampled in the previous step. For example, consider the case when the distribution of θ_js is very disperse. Even with adequate recovery of the hyperparameters, it may take many attempts to propose a successful θ_j^*. These many unsuccessful attempts result in a large rejection rate and can result in long computational times. In the next section, I show that one can easily improve upon this approach by using MCMC within steps to guide the search for the individual-level parameters.

3.5.1 MCMC Within Steps

Line 9 of Algorithm 5 specifies that the proposal $\theta_{j,k}^*$ is to be sampled from the conditional distribution $\pi(\theta|\phi)$. However, this conditional distribution does not provide specific information about which values of $\theta_{j,k}^*$ are more likely to be accepted. Because the search for a suitable $\theta_{j,k}^*$ is global, it is the same for each individual-level parameter. This can be problematic if the conditional distribution $\pi(\theta|\phi)$ is diffuse, because it could take many samples to acquire a suitable $\theta_{j,k}^*$. One way to minimize the rejection rate in this situation is to localize the search for a suitable $\theta_{j,k}^*$. To do this, we will rely upon the current position of the chain $\theta_{j,k,i-1}$ and a transition kernel $q(\theta)$ to confine the proposal space. One method is to embed a Markov chain in the ABC framework (Bortot et al., 2007, Marjoram et al., 2003). The details of the MCMC approach were presented in Chapter 2. The only difference is that now, the prior distribution is specified by the current values of the hyperparameter ϕ in the chain.

As mentioned in Chapter 2, the MCMC approach suffers when the starting values are poorly specified. For example, if τ^2 is too small, and we initialize $\theta_{j,k,1}$ to a value far outside the posterior distribution, it is conceivable that no proposed $\theta^*_{j,j}$ will allow the chain to move out of the region.

The idea of selecting a set of monotonically decreasing thresholds was discussed in Chapter 2 and is regularly used when sequential Monte Carlo techniques are employed in the ABC framework (Beaumont et al., 2009, Sisson et al., 2007, Toni et al., 2009). However, this idea can also be used in the mixture algorithm by defining ϵ to be a function of decreasing values dependent on the iteration.

Specifying tolerance conditions this way allows the algorithm to gain a "foothold." For example, one could define

$$\epsilon_t = \exp\{-at + b\} + c. \tag{3.8}$$

Here, three tuning parameters are introduced to allow for better control of the ϵ function. We still need to specify a final ϵ_f value so that

$$\pi(\theta|Y) \approx \pi(\theta|\rho(X,Y) \leq \epsilon_f) \tag{3.9}$$

is satisfied. We will then burn-in or discard any samples drawn with an ϵ_t greater than ϵ_f. Currently, there are no available methods to guide the selection of the tuning parameters that define ϵ_t, nor are there methods to determine the choice of ϵ_f. For these choices, often one must experiment with several values.

Choosing a monotonically decreasing set of ϵ values removes deficiencies the algorithm may face due to a high dependence upon the starting values $\theta_{j,k,1}$. In the next section, I will show that this method, in combination with MCMC within steps, improves Algorithm 5 to an impressive degree.

3.5.2 A Simulation Study

To demonstrate the benefits of adding MCMC within steps, I will complete another illustrative example. First, the claim is that rejection sampling is inefficient when the prior distribution is different from the posterior distribution (Beaumont, 2010, Sisson et al., 2007). In Algorithm 5, the hyperparameters are used at each iteration to specify a new prior for the individual-level parameters. Although the prior for the individual-level parameters is constantly changing, the problems residing in standard rejection sampling algorithms carry over into the hierarchical context. As previously mentioned, this is because the search for a successful $\theta^*_{j,k}$ is global, meaning any parameter value in the prior is likely to be chosen for evaluation.
A simple remedy to this problem is to localize the search for a successful $\theta^*_{j,k}$ by specifying a transition kernel, as described in the previous chapter. In this case, the prior distribution only influences the search by the Metropolis-Hastings probability. Thus, optimizing our algorithm now only involves optimizing our transition kernel.
For this example, suppose we are interested in estimating the means μ_j of 48 individual-level normal distributions, while simultaneously inferring the group-level mean μ. To simplify this problem, suppose the standard deviation of the individual-level normals is fixed at one and that the data Y_j come from the

distribution $Y_j \sim \mathcal{N}(\mu_j, 1)$. To build a hierarchy, I assume that these μ_js also come from a normal distribution, $\mu_j \sim \mathcal{N}(\mu, \sigma^2)$.

To generate the data, I considered three different conditions for $\sigma = \{10, 25, 50\}$ and used the same hyper-mean, $\mu = 0$ for each simulated data set. I then sampled 48 values from the hyper-prior $\mu_j \sim \mathcal{N}(\mu, \sigma)$. These μ_js were then used to generate 100 observations from the individual-level distributions, $Y_j \sim \mathcal{N}(\mu_j, 1)$. The standard deviation of these individual-level parameters was assumed to be equal; although this assumption is unreasonable for real data, it allows us to focus on the methods and not the inference.

The increasing standard deviation suggests that, when recovering the posterior distribution for σ, the prior for the individual-level parameters will be increasingly more variable, which results in more difficult global searches for the individual-level parameters. Thus, we should predict that as σ increases, the rejection rates should also increase when a mixture algorithm is used without MCMC within steps. However, if MCMC within steps are used, we can localize the search for each individual-level parameter and the rejection rates will stabilize across the various σ conditions.

I fit the model to the data using Algorithm 5 both with and without MCMC within steps as described in the previous section. For both models, the prior on μ was $\mu \sim \mathcal{N}(0, 100^2)$ and the prior on σ was $\sigma \sim \Gamma(0.1, 0.1)$. $\rho(X, Y)$ was selected as in Equation 3.7 and ϵ was set to $\epsilon_t = \exp(-.003t) + .01$. For both algorithms, 5,000 samples were drawn and a burn-in period was not used in order to illustrate the effect of a decaying ϵ. The only difference between the two algorithms is the specification of the transition kernel, which was set to $q(\mu_t|\mu_{t-1}) = \mathcal{N}(\mu_{t-1}, 0.1^2)$.

Figure 3.5 shows the mean (across subjects) number of proposals required before accepting a single $\theta^*_{j,k}$ when using rejection sampling (top panel) and when using MCMC within steps (bottom panel). The columns of Figure 3.5 correspond to the three different conditions for $\sigma = \{10, 25, 50\}$. The black lines overlaying each plot is a moving average of the data, with a window width of 100. Note that the scales for the two rows are very different. As σ increases, Figure 3.5 shows that the mean number of proposals rapidly increases. Specifically, when $\sigma = 50$, the mean number of proposals reaches an overwhelming 5,000. However, this is not the case for the improved sampler (with MCMC within steps), which remains stable for each of the three conditions. This is due to a localized parameter search, guided by the transition kernel specified above. Because the transition kernel was not changed from one condition to another, we would not expect the number of required proposals to change.

Figure 3.5 also shows that as ϵ decreases (the number of iterations increases), the mean number of required proposals increases very quickly, and then stabilizes. This quick increase in proposals is directly related to a quickly decreasing ϵ function for the tolerance thresholds. For this example, I used the very simple method for constructing initial values by calculating the means of Y_j for each of the 48 individual-level parameters μ_j. This method worked well enough that a decreasing set of ϵ was not really needed, but was still used for illustrative purposes.

Finally, the estimated posterior distributions of each of the parameters were compared. There were not any notable differences between the estimates provided by the two algorithms. These results were not shown because they were not of direct interest for this simulation study. This section has demonstrated that using

MCMC within steps in Algorithm 5 can provide a drastic improvement over simple rejection sampling. In the example, I have also shown how to incorporate a set of decreasing tolerance thresholds in an MCMC framework.

Although the previous chapter provided a general guideline for determining the discriminant function $\rho(X, Y)$, I have exclusively relied upon a single comparison of the simulated and observed data. In addition, the reason for comparing summary statistics (such as the mean) was not fully explained. In this next section, a more detailed explanation is given for why summary statistics are often convenient and a more general approach to the selection of $\rho(X, Y)$ is introduced.

3.6 Summary Statistics and Sufficiency

For computational ease, it is often convenient to define $\rho(X, Y)$ as a distance between summary statistics $S(Y)$ (e.g. the mean or variance). However, some statistics are better at informing the estimation of a parameter than others. Ideally, the summary statistic should be sufficient. Sufficient statistics are functions of the data, and knowing the value of a sufficient statistic provides just as much information about the parameter θ as the whole data set itself. Thus, if $S(Y)$ is a sufficient statistic for the parameter θ, then the posterior distribution can be written as

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

To determine if a statistic $S(Y)$ is sufficient, we must be able to reexpress the likelihood as a function of the sufficient statistic and the data. Formally, by the

Factorization Theorem, if the probability distribution $f(y|\theta)$ can be factored into

$$f(y|\theta) = g\left(S(Y), \theta\right) h(Y), \tag{3.10}$$

then $S(Y)$ is sufficient for the parameter θ. As an example, consider a series of n Bernoulli trials, where the probability of a single observation Y_i is $P(Y_i = y) = \theta^y (1-\theta)^{1-y}$. Then the joint probability function can be written as

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

As the last line shows, the function $f(y|\theta)$ can be written as a function of the unknown parameter θ and the statistic $S(Y) = \sum_{i=1}^{n} Y_i$. By Equation 3.10, we can let $g = (S(Y) = \sum_{i=1}^{n} y_i, \theta)$ and $h(Y) = 1$, demonstrating that the statistic $S(Y) = \sum_{i=1}^{n} Y_i$ is a sufficient statistic for the parameter θ. It is this result that motivated the selection of $\rho(X, Y)$ in the binomial examples presented in Chapter 2. In the ABC context, when comparing the data X and Y, if we can instead compare the sufficient statistics $S(X)$ and $S(Y)$, Equation 3.9 becomes

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

Performing inference on the sufficient statistic $S(Y)$ may be useful when the data are very large. However, sufficient statistics are not universal. That is, different parameters from different models may require different types of statistics in order to be sufficient. Fortunately, we do not require sufficient statistics for the ABC approach to work (see the section on the exponential distribution in the previous chapter).

3.6.1 Multiple Summary Statistics

We will often find it useful to compare multiple summary statistics at once. The idea is that while one summary statistic may not be sufficient for the unknown parameter θ, several summary statistics together may be jointly sufficient for θ. To demonstrate this idea, consider the normal distribution with both the mean μ and the standard deviation σ unknown. The joint probability function can be written and factorized as

$$\begin{aligned}
f(y|\mu,\sigma) &= \prod_{i=1}^{n} \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(y_i-\mu)^2}{2\sigma^2}\right] \\
&= \frac{1}{\sigma^n(2\pi)^{n/2}} \exp\left[\frac{-1}{2\sigma^2}\sum_{i=1}^{n}(y_i-\mu)^2\right] \\
&= \frac{1}{\sigma^n(2\pi)^{n/2}} \exp\left[\frac{-1}{2\sigma^2}\left(\sum_{i=1}^{n} y_i^2 - 2\mu\sum_{i=1}^{n} y_i + n\mu^2\right)\right].
\end{aligned}$$

This last line again shows that the joint probability function can be factored into a function of the data $h(Y) = 1$ and a joint function of two statistics and the two parameters $g(S_1(Y) = \sum_{i=1}^{n} Y_i, S_2(Y) = \sum_{i=1}^{n} Y_i^2, \mu, \theta)$. Thus, the two statistics $S_1(Y) = \sum_{i=1}^{n} Y_i$ and $S_2(Y) = \sum_{i=1}^{n} Y_i^2$ are jointly sufficient for the parameters μ and σ.

The use of multiple summary statistics will be essential when the probability distribution $f(y|\theta)$ cannot be specified. This situation is actually the main reason for performing likelihood-free inference, because once $f(y|\theta)$ is fully specified along with an independence assumption on the set $\{Y_1, Y_2, \ldots, Y_n\}$, then the likelihood function is defined by the product of the probability distributions.

The above sections showed how to enhance the algorithm by helping it to converge to the true posterior distribution sequentially. While these methods are effective, in practice it is sometimes difficult to minimize the discriminant function to a suitable value. This is mostly due to the assumption of a correct model, which is never true for real data. The algorithm's task is to simulate data from this incorrect model with the goal of minimizing $\rho(X, Y)$. However, the minimum value of $\rho(X, Y)$ may not be obtainable with a certain model. As a consequence, we either have to settle for a poor approximation to the posterior by increasing our tolerance threshold ϵ, or we must spend inordinate (perhaps infinite) amounts of time minimizing $\rho(X, Y)$. Additionally, all previous algorithms have suggested that when a proposed parameter value is rejected, that proposed parameter value is discarded, regardless of how close it may have been to the tolerance threshold. This is extremely wasteful because each proposed parameter value and the $\rho(X, Y)$ associated with it carry valuable information regarding the quality of these proposed parameter values. The nature of the posterior distribution alone guarantees that some parameter values are more likely to have generated the data than others.

In an effort to simultaneously solve these problems, Beaumont et al. (2002) proposed an algorithm that applies a post-simulation correction by weighting parameter values based on their corresponding summary statistic $S(X)$, which are then followed by a regression adjustment. In this section, I will present this method and demonstrate how it can be used to improve the estimates obtained in an illustrative example.

3.6.2 Regression Adjustment

The method presented in Beaumont et al. (2002) is based on the idea that there is a systematic relationship between the parameter values θ and the resulting summary statistics $S(X)$ calculated from the simulated data X. It should be clear that some parameter values produce better-fitting simulated data than others. If we consider the joint distribution of θ and $S(X)$, we should expect that values of θ with higher posterior densities should produce values for $S(X)$ that are closer to the target value (i.e. $S(X) = 0$).

Figure 3.6 illustrates a realization of this situation. After a simulation, the proposed parameter values and their corresponding summary statistics have been plotted together (gray dots). There is a clear linear relationship between these two values. Specifically, as the proposed parameter values approach 0.30, their corresponding summary statistics approach 0.0. If zero happens to be the optimal value for $\rho(X, Y)$ (i.e. as in Equation 3.7), then the proposed parameter values resulting in $S(X) = 0$ are much better than say, the values resulting in $S(X) = -0.05$. For this figure, any proposed values resulting in $S(X) \geq \epsilon$ or $S(X) \leq -\epsilon$ were discarded. Clearly, there is much more information available than a simple reject/accept decision rule conveys.

If the relationship between θ and $S(X)$ is approximately linear, we can use linear regression to obtain an estimate for the optimal value of θ. This estimate can then be used to adjust the remaining samples from the approximate posterior distribution. For this section, I denote the set of posterior samples with Θ and the target value for the vector of summary statistics $\mathbf{S}(x)$ with $\mathbf{S}_0$. Individual components of the summary statistics are represented with a double subscript reflecting the calculation for the ith sample and mth statistic, $S_{i,m}(x)$.

A model for linearly regressing $\mathbf{S}(x)$ onto Θ is given by

$$\Theta_i = \alpha + \left[\mathbf{S}_i(x) - \mathbf{S}_0\right]^T \beta + \zeta_i, \tag{3.11}$$

where the set of ζs are uncorrelated and have common variance for $i \in \{1, 2, \ldots, N\}$. When $\mathbf{S}_i(x) = \mathbf{S}_0$, we are drawing samples directly from our desired posterior distribution, which has a mean of α. Because this is a standard regression model, the least squares estimate for α and β is

$$\{\hat{\alpha}, \hat{\beta}\} = (X^T X)^{-1} X^T \Theta,$$

where X is the matrix of summary statistics given by

$$X = \begin{bmatrix} 1 & S_{1,1}(x) & S_{1,2}(x) & \cdots & S_{1,M}(x) \\ 1 & S_{2,1}(x) & S_{2,2}(x) & \cdots & S_{2,M}(x) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & S_{N,1}(x) & S_{N,2}(x) & \cdots & S_{N,M}(x) \end{bmatrix}.$$

Then, after the adjustment

$$\Theta_i^* = \Theta_i - \left[\mathbf{S}_i(x) - \mathbf{S}_0\right]^T \hat{\beta}, \qquad (3.12)$$

Θ^* will form an approximate random sample from the posterior distribution $\pi(\theta|y, \rho(\mathbf{S}(x), \mathbf{S}_0) = 0)$.

3.6.3 Localized Weighting

While Equation 3.11 does not make distributional assumptions about ζ, it does assume that there is a linear relationship between Θ and $\mathbf{S}(x)$. This is rarely true in practice, but Beaumont et al. (2002) argue that it may be true in a localized region around $\mathbf{S}_0$. Thus, we can perform localized linear regression by applying a weighting function to the parameter values based on their corresponding $\mathbf{S}(x)$ values.

To localize the regression problem, we define a kernel function $K(d)$ that provides weights to the $\mathbf{S}(x)$ values as a function of their distance d away from the desired

$\mathbf{S}_0$. This kernel function can take many forms, such as a Gaussian, exponential or Epanechnikov. We now define the weight matrix W such that

$$W = \begin{bmatrix} K(||\mathbf{S}_1(x) - \mathbf{S}_0||) & 0 & \cdots & 0 \\ 0 & K(||\mathbf{S}_2(x) - \mathbf{S}_0||) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K(||\mathbf{S}_N(x) - \mathbf{S}_0||) \end{bmatrix}$$

where $||\mathbf{S}_i(x) - \mathbf{S}_0)|| = \sqrt{\sum_{j=1}^{M} [(\, S_{i,j}(x) - S_{0,j}\,)^2]}$ is the standardized performance on each of the $\mathbf{S}(x)$, and the new estimates for α and β from Equation 3.11 become

$$\{\hat{\alpha}, \hat{\beta}\} = (X^T W X)^{-1} X^T W \Theta.$$

The kernel function K may be specified so that $\mathbf{S}(x)$ values outside the region of $\mathbf{S}_0$ are given a weight of zero, excluding them from influencing the estimates for α and β. However, other methods incorporate these samples by instead changing the model specification in Equation 3.11. For example, Blum and François (2010) use nonlinear regression techniques and correct for heteroscedascity. Other applications more focused on model selection problems perform logistic regression on the binary model selection parameter (Fagundes et al., 2007). These methods, while important, are unnecessary for the purposes of this dissertation, because we will be able to obtain samples when $\epsilon \approx 0$.

3.7 The Poisson Example Revisited

To illustrate the extreme impact the post-simulation corrections can have on the estimated posterior distribution, I will briefly revisit the Poisson example presented earlier. In that example, I presented an inference problem on the rate parameters θ_j. We were able to set $\epsilon = 0$ and sample directly from the posterior distributions.

However, suppose that the model assumptions were wrong, making the target $S_0 = 0$ unobtainable for the discriminant function in Equation 3.7. Suppose we were instead only able to achieve an ϵ value of 0.25. Our estimates of the posterior would suffer severely, but the correction method presented in the previous section can ameliorate this degeneracy.

Despite this new constraint, all model and algorithm details presented in the earlier treatment of this problem – except where noted – were maintained. The same data were also used for this example, to allow for easy comparison. Notice that in Equation 4.8, we can write $\rho(X, Y)$ by using the summary statistics $S(X) = \frac{1}{n}\sum_{i=1}^{n} X_{j,i} = \bar{X}$. We will also drop the absolute value so that the distribution of $\rho(X, Y)$ is symmetric around zero. Thus, the new $\rho(X, Y)$ function can be written as $\rho(X, Y) = \bar{X} - \bar{Y}$. We will now impose our constraint in the rejection algorithm with $\epsilon = 0.25$ being the smallest value we can acquire.

Figure 3.7 shows the estimated posteriors (solid gray lines) under this new restriction. The solid black line still shows the true posterior distribution and the dashed vertical line shows the true sampled value of the parameters used to generate the data. The newly estimated posteriors are far too variable, and are not representative of the true posteriors.

Now we can apply the correction methods outlined in the previous section. To make the correction, we set the kernel K to be an exponential, or

$$K(d) = \lambda \exp^{-\lambda d}, \tag{3.13}$$

and the rate parameter for the kernel to be $\lambda = 10$. Figure 3.6 shows that a clear linear relationship between $S(X)$ and Θ making localized weighting unnecessary. Regardless, I still performed the localized weighting for illustrative purposes.

Estimates for α and β were obtained for each subject separately. Figure 3.6 shows the joint distribution of Θ and $S(X)$ for Subject 3 (gray dots). For Subject 3, the OLS estimates were $\hat{\alpha} = 0.608$ and $\hat{\beta} = 0.996$. This line of best fit is plotted in Figure 3.6 (black solid line). Figure 3.6 shows that the estimate $\hat{\alpha}$ is reasonably close to the true value for θ_3 (shown by the "X" symbol).

One the correction was applied to each of the four subjects, the corrected estimates were plotted in Figure 3.7 (histograms). The corrected estimates form nearly identical distributions as those obtained in the previous section when $\epsilon = 0$, and are remarkably similar to the true posterior distributions (black lines).

This section has shown that even with limited summary statistics or computation power, we can obtain valid estimates of the posterior distributions using the post-simulation correction method described in Chapter 2. This post-simulation correction is often necessary for real data. In the next section, we investigate a more complicated model, requiring several summary statistics in combination with a post-simulation correction.

Now that several methods have been discussed for improving the ABC approach, and specifically the mixture algorithm, we can now begin estimating more complicated models, where sufficient statistics may not be available (although in the previous chapter's estimation of REM, it was not clear that the statistic was sufficient for the parameters g, u, and c). We begin by estimating the parameters for the Wald model, and in the next chapter we turn to models of episodic memory.

3.8 A Wald Model for Response Times

Bayesian modeling of RTs has proven very useful in the recent decade. RT distributions provide a complex structure of data, rich with modeling opportunities. The analysis of RT has a long and interesting history in psychology. The treatment of RT distributions as a tool for underlying cognitive processes has evolved from the simple comparison of mean RTs, to mathematical models accounting for the full shape of the distributions (e.g., Brown and Heathcote, 2005, 2008, Ratcliff, 1978, Usher and McClelland, 2001), to models accounting for sequential structures of the parameters from these mathematical models (Craigmile et al., 2010, Peruggia et al., 2002). In the Bayesian context, RT distributions have been modeled with a variety of statistical distributions such as the Weibull, Wald, ex-Gaussian, or log Gaussian (e.g., Craigmile et al., 2010, Lee and Wagenmakers, 2010, Peruggia et al., 2002, Rouder et al., 2005, 2003) and recent interest has turned to mathematical models (Lee et al., 2006, Oravecz et al., 2009, Vandekerckhove et al., 2011).

In this section, I show how Algorithm 5, along with the improvements suggested in the previous sections, may be used to fit hierarchical RT data using the Wald distribution (e.g., Chhikara and Folks, 1989, Wald, 1947). The Wald or inverse Gaussian distribution is a single boundary diffusion process which can be used to fit RTs in simple tasks (Luce, 1986) or in go/no-go experiments (e.g., Heathcote, 2004, Schwarz, 2001).

3.8.1 Model

Suppose we gathered 500 RTs for 4 subjects in a simple task in which response choice was not a feature of the design. The ith RT for Subject j will be denoted as

$Y_{i,j}$. To fit the data, we will use the three-parameter Wald distribution, given by

$$\text{Wald}(Y_{i,j}|\alpha_j, \nu_j, \tau_j) = \frac{1}{\sqrt{2\pi(Y_{i,j}-\tau_j)^3}} \exp\left(-\frac{\left[\alpha_j - \nu_j(Y_{i,j}-\tau_j)\right]^2}{2(y_{i,j}-\tau_j)}\right). \tag{3.14}$$

This parameterization uses parameters which have psychological interpretations. The parameter α corresponds to a threshold for a response, ν corresponds to a drift rate or rate of accumulation of evidence, and τ represents the nondecision time (Heathcote, 2004, Matzke and Wagenmakers, 2009). As discussed in the introduction, estimating these parameters will provide some insight into individual and group differences. For example, if the estimate for α for one subject in an experiment is much higher than the rest of the group, this may reflect an overly cautious responder. Perhaps this subject was very concerned with answering correctly, which has a very different meaning than inferring slow cognitive processing due to a simple comparison of the subject's mean RT (e.g., Ratcliff et al., 2003, 2004).

To build a hierarchical model, we assume that the four subject-specific parameters arise from some overarching distribution. Each of the three individual-level parameters have positive infinite support, so a natural prior distribution is the gamma distribution. The priors for α_j, ν_j and τ_j are

$$\begin{aligned}
\alpha_j &\sim \Gamma(\alpha_\alpha, \beta_\alpha), \\
\nu_j &\sim \Gamma(\alpha_\nu, \beta_\nu), \text{ and} \\
\tau_j &\sim \Gamma(\alpha_\tau, \beta_\tau),
\end{aligned}$$

with noninformative priors for the set of hyperparameters $\phi = \{\alpha_\alpha, \alpha_\nu, \alpha_\tau, \beta_\alpha, \beta_\nu, \beta_\tau\}$, so

$$\phi_m \sim \Gamma(0.01, 0.01),$$

for each element $m \in \{1, \ldots, 6\}$ of ϕ.

3.8.2 Results

The above model was used to generate the data. Each of the four α_js were drawn from a gamma distribution with $\alpha_\alpha = 40$ and $\alpha_\beta = 30$. Similarly, $\nu_j \sim \Gamma(90, 25)$ and $\tau_j \sim \Gamma(10, 50)$. Once each individual-level parameter was obtained, 500 RTs were randomly generated using Equation 3.14.

To fit the data, both standard rejection sampling and Algorithm 5 were used. When using the mixture algorithm, rejection sampling was also performed for the hyperparameters, and MCMC within steps were used for the individual-level parameters. Truncated normal distributions were used as the transition kernels for each of the individual-level parameters. For α_j and ν_j, these normals were truncated so that they could not go below zero and with standard deviations equal to 0.1 and 0.5, respectively. However, for τ, the normal was truncated informatively, by restricting the Markov chain to be greater than zero and less than the minimum RT observed for each subject. The mean of the truncated normal was set to the current state of the chain with standard deviation of 0.05.

For the three-parameter Wald distribution, jointly sufficient statistics are difficult to find (but see Chhikara and Folks, 1989). Regardless, the purpose of this example is to show that even without the density function, ABC – specifically the mixture algorithm – is able to recover the “true” posterior distributions for the parameters of a psychological model. Six summary statistics were chosen to reflect the features of the Wald distribution such as the shape, skew and shift (see Rouder et al., 2003, for motivation). The summary statistics chosen to fit the data were the standard deviation, the minimum, the sum of the Y_j, the sum of $1/Y_j$, the skew and the kurtosis. Using Equation 3.8, the as were all equal to 0.02, the bs were all equal to 0, and the cs were equal to 0.05, 0.05, 30, 30, 0.5, and 0.5, respectively. To perform the localized regression, an exponential kernel was used (see Equation 3.13) with $\lambda = 2$. I also used a Gaussian kernel with standard deviation of one and an Epanechnikov kernel with a window width of two (Silverman, 1986, see) were used, but the posterior estimates were unaffected by these various kernels.

Figure 3.8 shows the estimated posteriors for each of the individual-level parameters. The "true" estimates (using standard rejection sampling) are shown by the black solid lines, and the results of using the mixture algorithm are shown with the gray lines. The histograms show the estimated posteriors after the regression correction was used. We can see that although the regression correction was very helpful, it did not perfectly match the standard Bayesian results. The reasons for this are due to the nature of the approximation in ABC. Had we selected truly sufficient statistics, we would have recovered the true posteriors. However, even without sufficient statistics, we have obtained a very close estimate of the posterior. The estimated joint posterior distributions for the hyperparameters corresponding to the three individual-level parameters are shown in Figure 3.9. In this figure, the 90%, 95% and 98% credible intervals are shown as contours. The bottom panel shows the posteriors using the mixture algorithm prior to (gray lines) and after (black lines) the post-simulation correction, and the top panel shows the estimates obtained using standard Bayesian techniques. The estimates prior to the correction are lower than the true estimates, but after the correction, the estimates closely resemble the true posteriors. However, the posteriors for the hyperparameters are much less affected by the post simulation correction when compared to the individual-level posterior distributions.

3.9 Summary

This chapter focused on using the ABC approach to estimate posterior distributions for the parameters of hierarchical models. Efficiently estimating the parameters of these hierarchical models is essential to the understanding of subject-, condition- and group-specific characteristics. To begin this section, I first discussed the disadvantages of using the extension of the ABC PMC algorithm to hierarchical models presented in Chapter 2. I then summarized the second algorithm of Bazin et al. (2010). This algorithm is much more efficient because it evaluates the contribution of hyperparameters by a marginal inspection of the simulated and observed data. Despite its utility, it can only approximate the distribution of the hyperparameters (Bazin et al., 2010, Beaumont, 2010). In addition, it involves two data generation steps, which can produce long computation times.

Finally, I introduced a new mixture algorithm that embeds the ABC technique into a Gibbs sampling framework. This algorithm is highly flexible, and is amenable to decreasing tolerance thresholds, MCMC within steps, post-simulation corrections, and multiple summary statistics. Instead of approximating the distribution of the hyperparameters, the mixture algorithm allows us to sample directly from it. The strength of this algorithm was demonstrated in two examples. First, the algorithm was used to infer incident rates in a hierarchical Poisson model. Finally, after several improvements were suggested, the algorithm and its extensions were used to fit RT data from a hierarchical Wald model.

```
for $1 \leq i \leq N$ do
    while $P(X, Y) > \eta$ do
        Sample $\phi^*$ from the prior: $\phi^* \sim \pi(\phi)$
        for $1 \leq j \leq J$ do
            Sample $\theta_j^*$ from the prior: $\theta_j^* \sim \pi(\theta \mid \phi^*)$
            Generate data $X_j$ using the model: $X_j \sim \text{Model}(\theta_j^*)$
        end for
        Calculate $P(X, Y)$
    end while
    Store $\phi_i \leftarrow \phi^*$
end for
for $1 \leq i \leq N$ do
    for $1 \leq j \leq J$ do
        while $\rho(X_j, Y_j) > \epsilon$ do
            Sample $\phi^*$ from previous pool: $\phi^* \sim \pi(\phi)$
            Sample $\theta_j^*$ from the prior: $\theta_j^* \sim \pi(\theta \mid \phi^*)$
            Generate data $X_j$ using the model: $X_j \sim \text{Model}(\theta_j^*)$
            Calculate $\rho(X_j, Y_j)$
        end while
        Store $\theta_{i,j} \leftarrow \theta_j^*$
    end for
end for
```

Figure 3.1: Bazin et al.'s (2010; Algorithm 2) two-stage HABC to estimate the posterior distributions of ϕ and θ.

```
Initialize $\phi_{m,1}$ and each $\theta_{j,m,1}$ for all $m \in \{1, 2, \ldots, M\}$ and $j \in \{1, 2, \ldots, J\}$
for $2 \leq i \leq N$ do
    for $1 \leq m \leq M$ do
        Sample $\phi_{m,i}$: $\phi_{m,i} \sim \pi(\phi_m \mid Y, \theta_{1:J,1:M,i-1}, \phi_{-m,i})$
    end for
    for $1 \leq j \leq J$ do
        for $1 \leq k \leq K$ do
            while $\rho(X_j, Y_j) > \epsilon_i$ do
                Sample a value $\theta^*_{j,k}$ from a proposal distribution: $\theta^*_{j,k} \sim q(\theta)$
                Generate data $X_j$ using the model: $X_j \sim \text{Model}(\theta_{j,-k,i}, \theta^*_{j,k})$
                Calculate $\rho(X_j, Y_j)$
            end while
            Store $\theta_{j,k,i} \leftarrow \theta^*_{j,k}$
        end for
    end for
end for
```

Figure 3.2: A HABC mixture algorithm to estimate the posterior distributions for ϕ and θ.

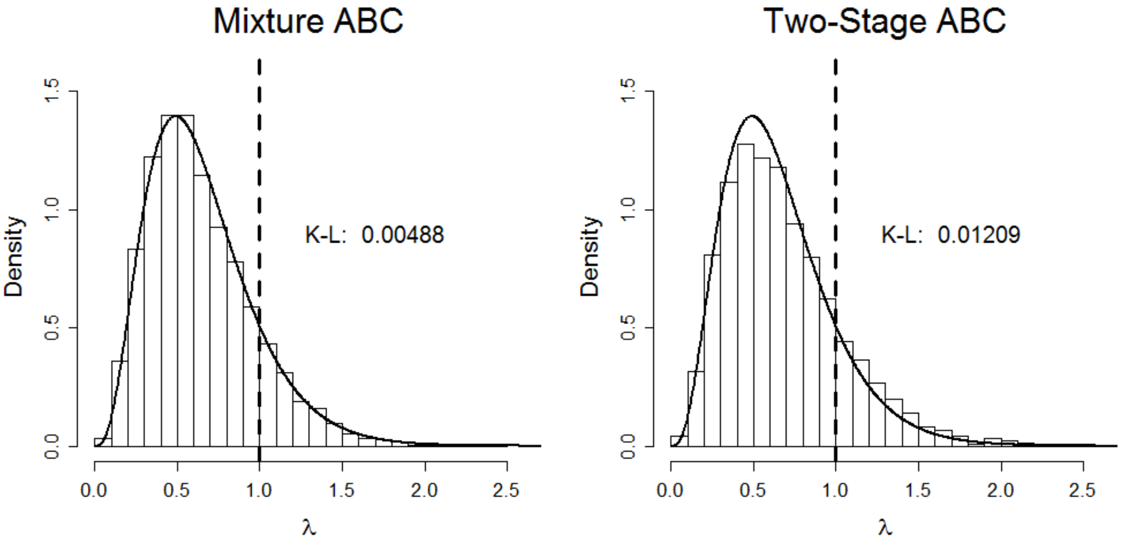

Figure 3.3: The estimated marginal posterior distributions using the mixture algorithm (left panel, see Algorithm 5) and the two-stage algorithm (right panel, see Algorithm 4). The vertical dashed line represents the true sampled value, which was used to generate the data. Overlaying each histogram is the true posterior density. The Kullback-Leibler statistic is reported in the upper right portion of each respective panel.

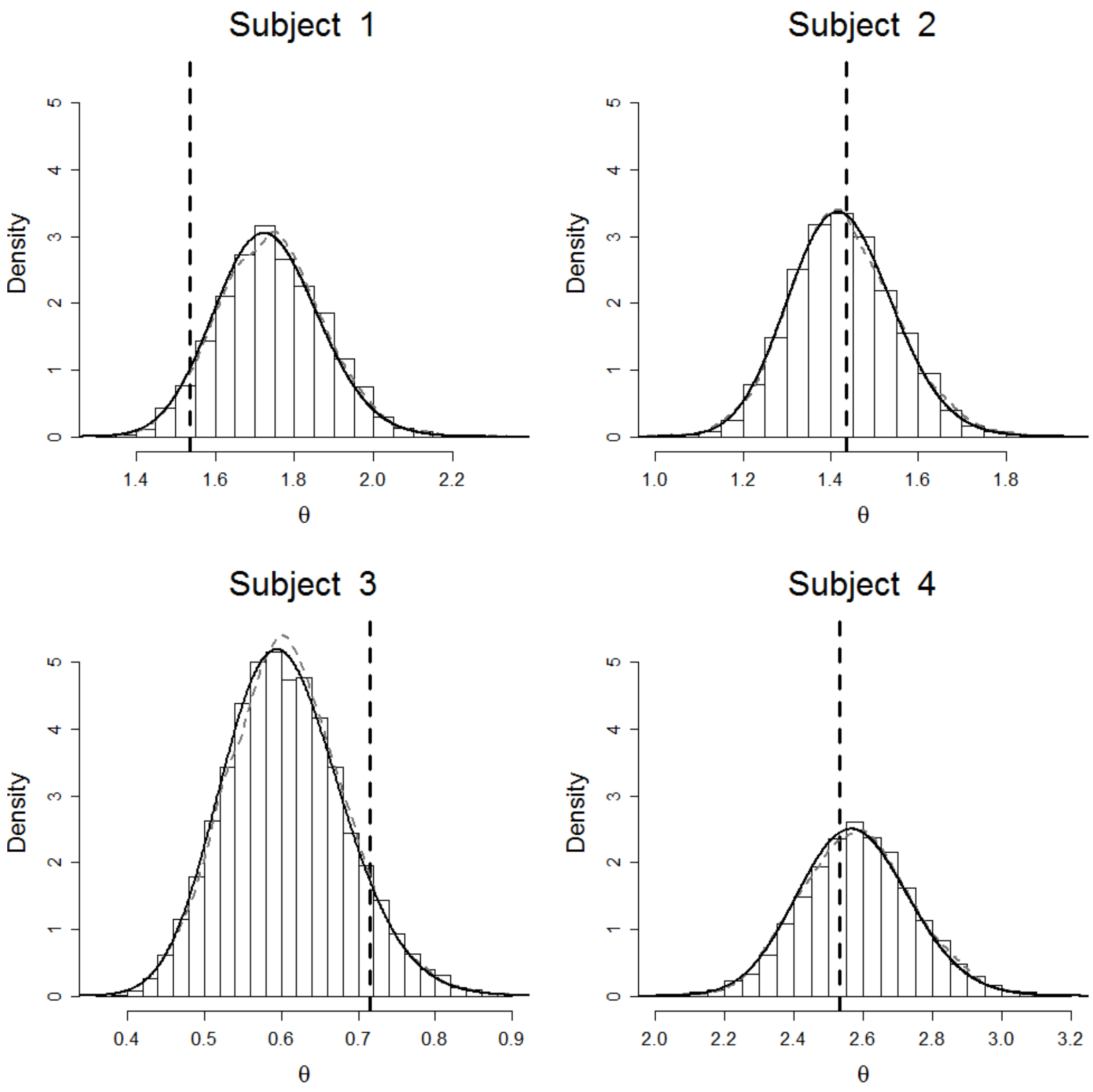

Figure 3.4: Estimated marginal posterior distributions using the mixture algorithm (histograms, see Algorithm 5). Overlaying each histogram is the estimated posterior distributions using Algorithm 4 (dashed gray lines) and the true posterior distributions (solid black lines). The vertical dashed lines represent the true sampled values used to generate the data for each subject.

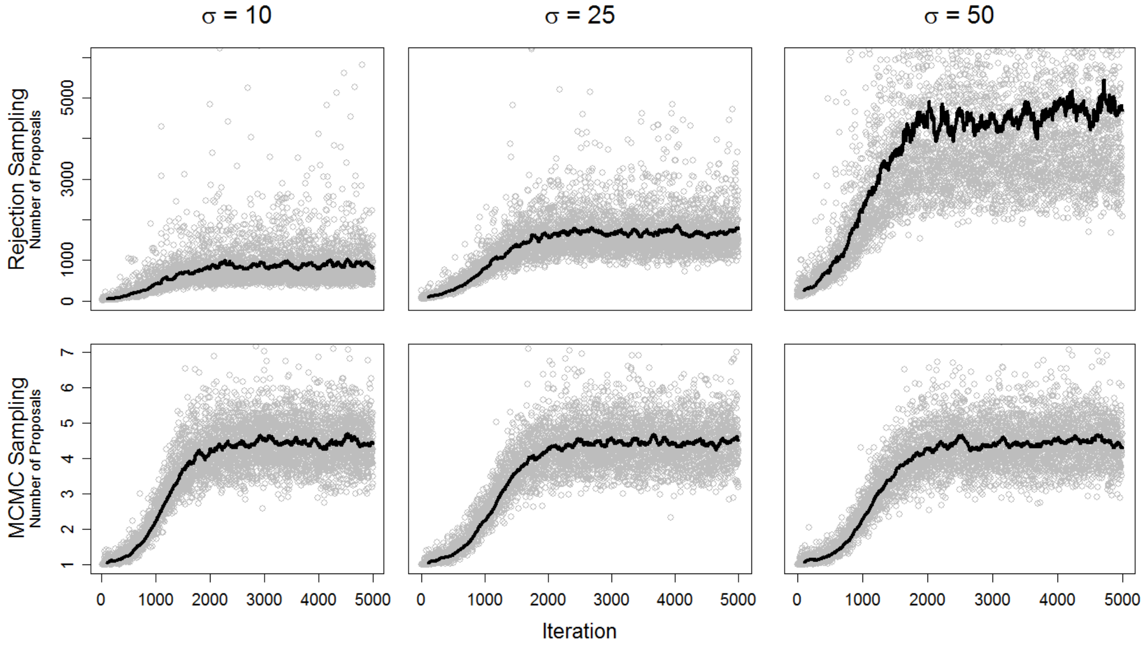

Figure 3.5: The mean number of proposals required before one acceptance for each iteration (gray dots). Overlaying each figure is a moving average of the data, with window width equal to 100 (black lines). The top panel is the result of this simulation when only rejection sampling is used (see Algorithm 5), and the bottom panel is the result when MCMC within steps are used. The left, middle, and right panels display three different conditions, where the standard deviation was varied from 10 to 25 to 50, respectively.

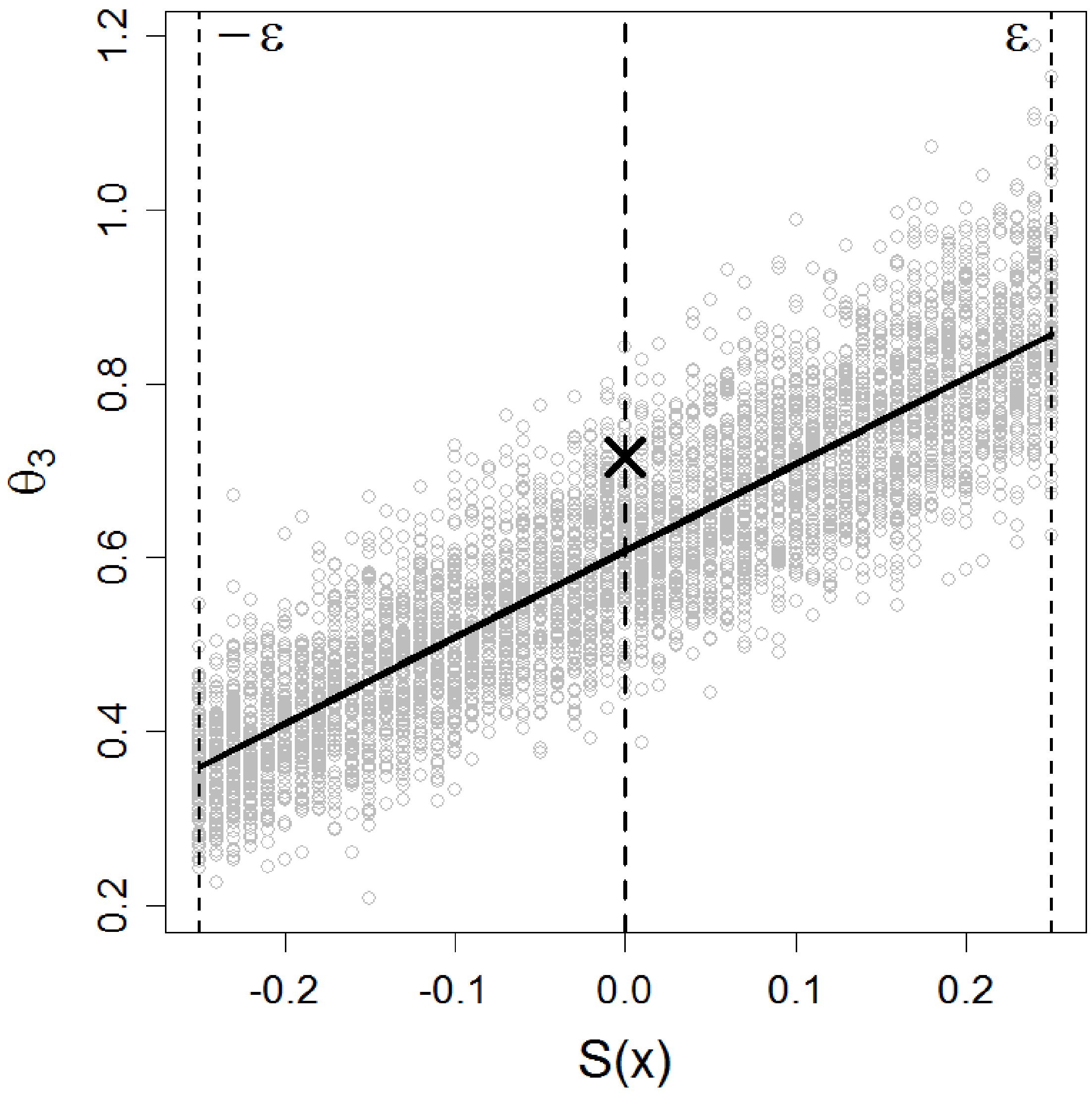

Figure 3.6: Joint distribution of $S(X)$ and θ for Subject 3 in Section 3.4. The left and right vertical dashed lines represent the tolerance thresholds, the middle dashed line represents S_0 and the solid diagonal line represents the line of best fit using localized linear regression. The “X” symbol represents the sampled value that generated the data for Subject 3.

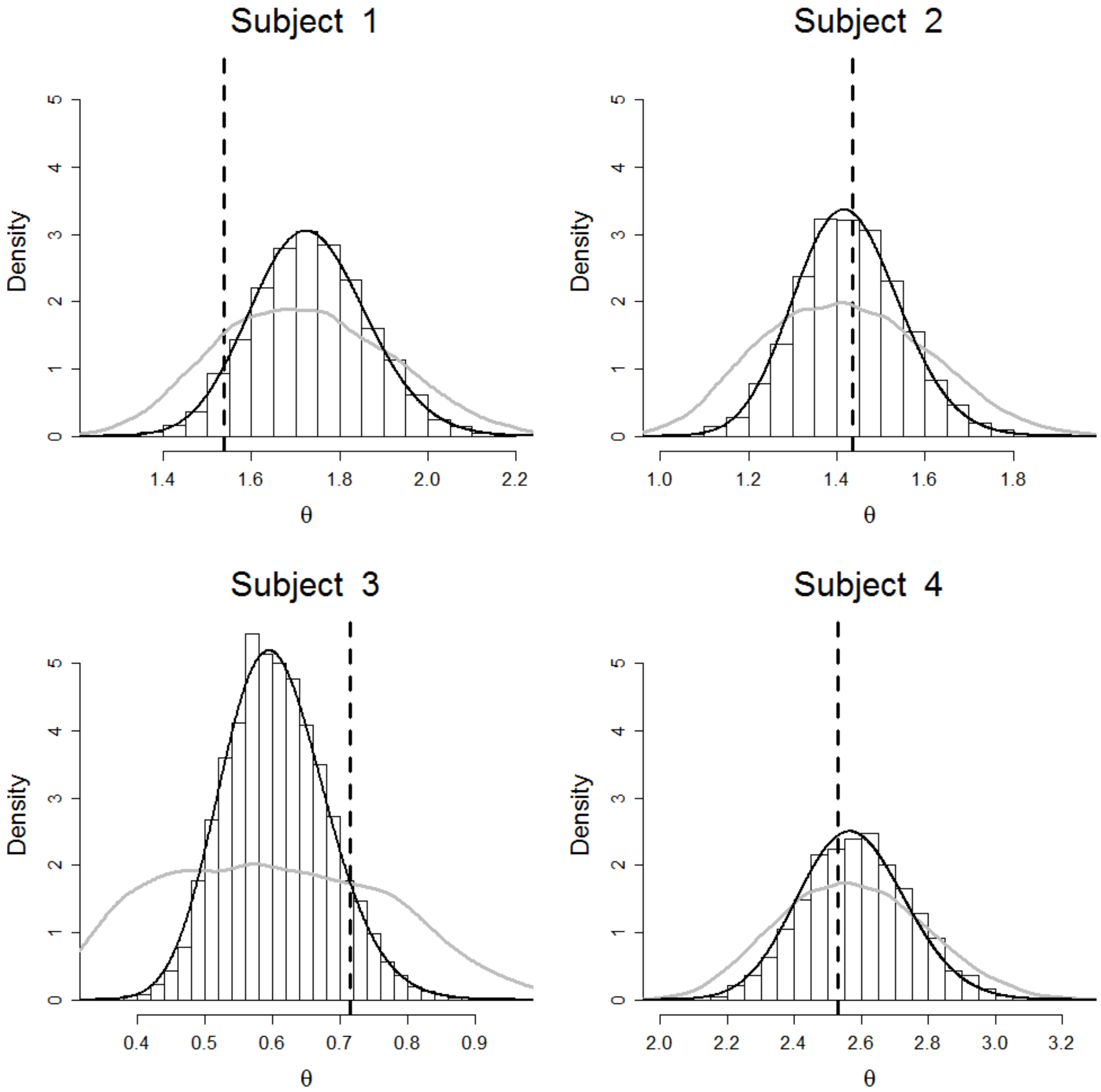

Figure 3.7: Estimated posterior distributions using $\epsilon = 0.25$ (solid gray lines) and the corrected estimates of this posterior distribution (histogram). The true posterior is represented by the solid black line and the true sampled value is represented by the dashed line.

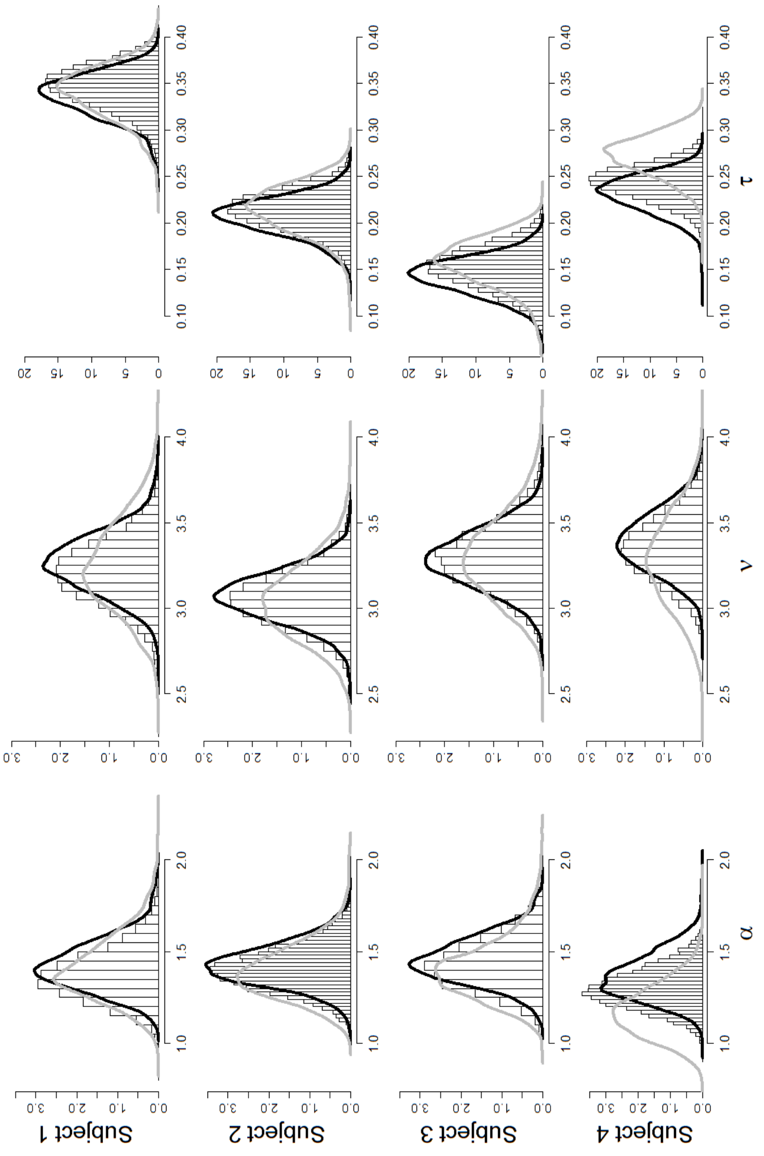

Figure 3.8: The estimated posteriors using the mixture algorithm prior to the post-simulation correction (gray lines), after the correction (histograms), and the result of the standard Bayesian fits (black lines). The columns represent the different parameters α, μ, and τ, respectively, and the rows represent the four different subjects.

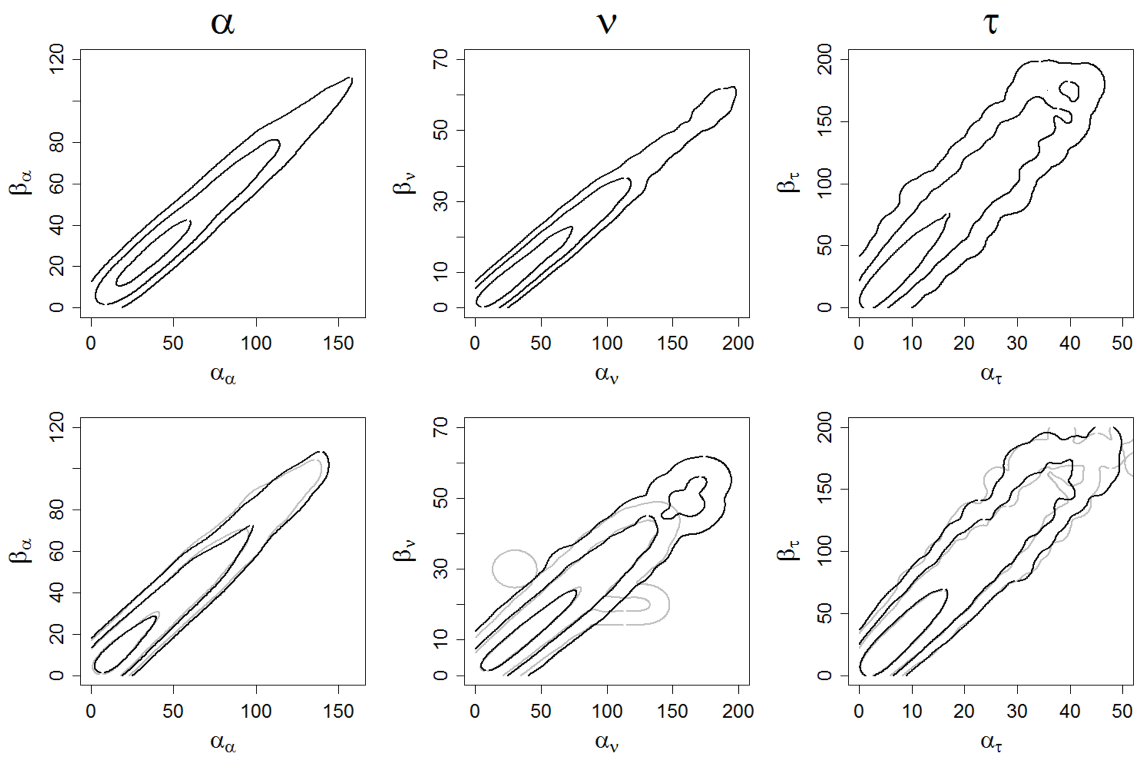

Figure 3.9: The estimated joint posteriors corresponding to α (left), ν (middle), and τ (right) using the mixture algorithm prior to (gray contours, bottom panel) and after the post-simulation correction (black contours, bottom panel). The joint posterior estimates using standard Bayesian techniques (rejection sampling) are shown on the top panel. The 90%, 95% and 99% credible intervals are shown as contours.

Chapter 4: Applications to Memory Models

This chapter is devoted to estimating the posterior distributions of the parameters of memory models using ABC. While the focus of this chapter is very specific, the utility is quite general. Many popular memory models are simulation based, such as REM (Shiffrin and Steyvers, 1997), BCDMEM (Dennis and Humphreys, 2001) , SLiM (McClelland and Chappell, 1998), and SAM (Gillund and Shiffrin, 1984). However, for this chapter we will focus on REM and BCDMEM.

The two models take very different approaches to modeling episodic memory. For our purposes, episodic memory will entail the recognition memory task exclusively, although the two models can be applied to a variety of experimental data. In a recognition memory experiment, a subject is given a list of study items (e.g., words or pictures) during the study phase and is instructed to commit them to memory. After the study phase, the subject might participate in some filler task, such as completing a puzzle. Following these two phases is the 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 (called a target) or it could be a new word (called a distractor).

Given the two possible types of probes and the two possible types of responses, there are four possible outcomes for every presented stimulus. However, the classic treatment of the data is to focus on "hits" and "false alarms." A hit occurs when a target is presented and the subject responds "old". A false alarm occurs when a distractor is presented, but the subject incorrectly responds "old". A hit rate can be computed by dividing the number of hits by the number of targets presented in the test phase and a false alarm rate can be computed by dividing the number of false alarms by the number of distractors presented in the test phase. These two rates are bounded by zero and one, and they can be plotted in the unit square (the space ranging from zero to one in both x and y directions), which is known as the receiver operating characteristic (ROC) space (e.g., Egan, 1958, Green and Swets, 1966).

BCDMEM and REM account for the data very differently. When a probe word is presented, there are two ways in which the retrieval process may occur. One way assumes that the item recognition process is based on the contexts in which that item has been associated. Under this assumption, performance is determined by the degree of overlap between the contexts associated with an item at study and the all contexts previously associated with that item. The alternative is that the item recognition process is based on the features (e.g., an image of a face might have the feature "big nose") of that item and how well they were encoded at study. Performance is determined by the degree to which features of items are correctly stored in memory.

This chapter is divided into two sections. The first section introduces BCDMEM and demonstrates how BCDMEM can be fit using ABC as well as standard Bayesian techniques, because the likelihood function for BCDMEM has been derived. Once convinced that the ABC approach provides an adequate approximation to the "true" posterior distributions, I fit a hierarchical version of BCDMEM to the data presented in Dennis et al. (2008). The second section proceeds in a similar fashion by first introducing REM and exploring the relationship between the model predictions and the model parameters. I demonstrate that the ABC approach can be used to answer a few interesting questions. Finally, a hierarchical version of REM is fit to the data from Dennis et al.

4.1 BCDMEM

For BCDMEM, item recognition is a context noise process; that is, when a probe is presented, the context in which that word was experienced are reinstated. A decision is made based on the overlap between the context reinstated from study and all other contexts in which the item is associated. For example, if a probe word is presented that is very familiar to an observer, he/she will be more likely to respond "old" due to the interference of previous contexts in which that word is recalled in everyday situations. However, this interference can be overcome if the study phase context is learned and reinstated accurately enough.

Formulating the recognition process in this way makes very different predictions for the data. For example, because the study phase context is a single event, it does not differentiate between short and long lists. That is, there is currently no mechanism in the BCDMEM model that can account for a decline in performance based solely on a list length increase. Early in the study of recognition memory, the list length effect was considered a regularity (e.g., Bowles and Glanzer, 1983, Gillund and Shiffrin, 1984, Glanzer et al., 1993). However, Dennis et al. (2008) showed that if the retention interval (study time) is equated between short and long lists, a null list length effect is seen where performance neither decreases nor increases. Regardless of whether the list length effect exists, BCDMEM predicts a null list length effect and REM predicts the list length effect.

4.1.1 The Model

Although specific details of the model can be found in Dennis and Humphreys (2001), I will outline the basic structure of the BCDMEM model here. BCDMEM assumes that at study, items are represented by active nodes on an input layer. On the output layer, nodes are active with probability s (the study context sparsity parameter), and the pattern of active nodes represents the current study context. These two layers are each represented by a vector of length v. Active nodes on the input layer are connected to each node on the output layer through associative weights. During study, learning occurs by connecting the active nodes on the input and the output layers, each with probability r (the learning rate).

During the test phase, the input layer again consists of items represented by active nodes. However, the output layer now consists of two vectors: a reinstated context vector and a retrieved context vector. The model possesses a contextual reinstatement parameter d so that some nodes which were active at study become inactive with probability d once the vector is reinstated. The retrieved context vector consists of contextual patterns representing the prior context with which the item has been associated. Nodes become active in this vector with probability p. For targets, this retrieved context vector also consists of the context induced at study. Thus, for targets the patterns of the two vectors on the output layer will be more similar.

A decision is made by comparing the activation patterns of the reinstated context vector and the retrieved context vectors. As in Dennis and Humphreys (2001), we let the ith node in the reinstated context vector be denoted by c_i and the ith node in the retrieved context vector be denoted by m_i. If we let the first component be the reinstated context and the second component be the retrieved context, then we can use the notation n_{ij} to denote the number of times the reinstated context vector is equal to i while the retrieved context vector is equal to j. For example, n_{11} denotes the number of times both nodes are active in the same position for the reinstated and retrieved context vectors. For convenience, we group the parameters of the model into the parameter vector θ so that $\theta = \{d, p, r, s, v\}$. With this notation in hand, the BCDMEM model compares the two context vectors through the likelihood ratio

$$\begin{aligned} L(\mathbf{n}|\theta) = &\left[\frac{1-s+ds(1-r)}{1-s+ds}\right]^{n_{00}} \left[\frac{r+p-rp}{p}\right]^{n_{11}} (1-r)^{n_{10}} \\ &\times \left[\frac{p(1-s)+ds(r+p-rp)}{p(1-s)+dsp}\right]^{n_{01}}, \end{aligned} \tag{4.1}$$

where $\mathbf{n}$ represents the vector of node pattern matches and mismatches $\mathbf{n} = \{n_{00}, n_{01}, n_{10}, n_{11}\}$.

Equation 4.1 represents the probability that an item is a target divided by the probability that the item is a distractor. For a given item, a response of “old” will be given if the probability that the item is a target is greater than the probability that it is a distractor. Thus, when $L(\mathbf{n}|\theta) > 1$, an “old” response will be given, otherwise a “new” response will be given.

4.1.2 Deriving the Likelihood Function

Although the above likelihood ratio governs the response, it unfortunately does not provide the probability of a hit or false alarm. As I will discuss later, these probabilities are essential for specifying the likelihood function. Myung et al. (2007) derived analytical expressions for these probabilities. First, with the assumption that nodes are activated independently, the vector of matching counts $\mathbf{n}$ follows a multinomial distribution. For targets, if we define $p_{i,j}(\theta)$ as

$$\begin{aligned} p_{0,0}(\theta) &= (1-s)(1-p) + sd(1-r)(1-p) \\ p_{0,1}(\theta) &= (1-s)p + sd(r+p-rp) \\ p_{1,0}(\theta) &= s(1-d)(1-r)(1-p) \\ p_{1,1}(\theta) &= s(1-d)(r+p-rp) \end{aligned}$$

where $\theta = \{d, p, r, s\}$, and for distractors we define $q_{i,j}(\theta)$ as

$$\begin{aligned} q_{0,0}(\theta) &= [1-s(1-d)]\,(1-p) \\ q_{0,1}(\theta) &= [1-s(1-d)]\,p \\ q_{1,0}(\theta) &= s(1-d)(1-p) \\ q_{1,1}(\theta) &= sp(1-d), \end{aligned}$$

then the likelihood ratio for the decision rule can be reexpressed as the log likelihood ratio

$$\log\left[L(\mathbf{n}|\theta)\right] = \sum_{i=0}^{1}\sum_{j=0}^{1} \log\left[\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right] n_{i,j}.$$

Myung et al. (2007) then used this decision rule to express the hit and false alarm rates as expectations. These resulting expectations can be broken up into an infinite sum by using the Fourier transformation, resulting in the following expressions for the probability of a hit (denoted HIT) and of a false alarm (denoted FA), given by

$$\begin{aligned} P(\text{HIT}|\theta) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\sum_{i=0}^{1}\sum_{j=0}^{1}\exp\left[ik\log\left(\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right)\right]p_{i,j}(\theta)\right)^{v}\frac{dk}{ik} \quad (4.2)\\ P(\text{FA}|\theta) &= \frac{1}{2\pi}\int_{-\infty}^{\infty}\left(\sum_{i=0}^{1}\sum_{j=0}^{1}\exp\left[ik\log\left(\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right)\right]q_{i,j}(\theta)\right)^{v}\frac{dk}{ik}. \quad (4.3)\end{aligned}$$

While Equations 4.2 and 4.3 are invaluable for specifying the likelihood for BCDMEM, they can be difficult to evaluate precisely for all values of θ. Myung et al. (2007) also derived asymptotic expressions that approximate Equations 4.2 and 4.3. To simplify the equations, they defined the mean $\mu(\theta)$ and variance $\sigma^2(\theta)$ of the single-trial “payoff” as

$$\begin{aligned} \mu_p(\theta) &= \sum_{i=0}^{1}\sum_{j=0}^{1}\log\left(\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right)p_{i,j}(\theta)\\ \mu_q(\theta) &= \sum_{i=0}^{1}\sum_{j=0}^{1}\log\left(\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right)q_{i,j}(\theta)\\ \sigma_p^2(\theta) &= \sum_{i=0}^{1}\sum_{j=0}^{1}\log\left(\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right)^2 p_{i,j}(\theta)-\mu_p^2(\theta)\\ \sigma_q^2(\theta) &= \sum_{i=0}^{1}\sum_{j=0}^{1}\log\left(\frac{p_{i,j}(\theta)}{q_{i,j}(\theta)}\right)^2 q_{i,j}(\theta)-\mu_q^2(\theta).\end{aligned}$$

From this, the asymptotic expressions for the hit and false alarm rates can be approximated using

$$\begin{aligned} P(\text{HIT}|\theta) &\approx \Phi\left(\frac{\sqrt(v)\mu_p(\theta)}{\sigma_p(\theta)}\right) \quad (4.4)\\ P(\text{FA}|\theta) &\approx \Phi\left(\frac{\sqrt(v)\mu_q(\theta)}{\sigma_q(\theta)}\right). \quad (4.5)\end{aligned}$$

These equations are very easy to evaluate in R, and will also be used to estimate the posterior.

I will refer to Equations 4.2 and 4.3 as the “exact” equations and Equations 4.4 and 4.5 as the “asymptotic” equations. Having expressions for the hit and false alarm rates, we can now fully specify the likelihood function and obtain the posterior distribution for the parameter set θ when v is fixed to some positive integer (Myung et al., 2007). Thus, BCDMEM is a perfect model with which to validate the ABC approach. In the next section, I compare the estimates of the posterior obtained using both ABC and standard Bayesian techniques. If the posteriors closely match, then this will justify the ABC approach in fitting models of recognition memory.

4.1.3 Validating the ABC Approach

Equations 4.2 and 4.3 provide the probability of a hit and a false alarm, respectively. These probabilities enable the specification of the likelihood function, which allows us to perform inference on the model BCDMEM using standard Bayesian procedures. We can use this result to validate the ABC approach for BCDMEM, and for memory models in general. To do so, we will compare estimates of the posterior distribution obtained using ABC to estimates obtained using standard Bayesian techniques, such as MCMC or rejection sampling. If the two estimates obtained are similar, then the use of ABC for the estimation of memory models could be considered a legitimate alternative.

The next few sections are critical in showing that ABC provides acceptable approximations to the posterior distributions for memory models. Once we are satisfied with this approximation, the acceptance of ABC need not be limited to BCDMEM. This is because we will be using the entire data set in the specification of $\rho(X, Y)$, so the problem of determining sufficiency is not an issue because the data will be sufficient to themselves.

The Model

For this example, I generated data from the BCDMEM model for a single subject performing a recognition memory experiment. This hypothetical experiment consists of two conditions. In one condition, the subject is given a 10-item list during the study phase. In the second condition, the subject is given a 15-item list. At test, the subject is asked to respond "old" when a previously-studied word is presented and "new" when a new word is presented. During the test phase, the subject is presented with a mixed list consisting of 10 targets and 10 distractors in the first condition, and 15 targets and 15 distractors in the second condition. Myung et al. (2007) concluded that the vector length parameter v must be fixed to identify the model. Accordingly, to both generate and fit the data, v was fixed at 20. With v fixed, we wish to obtain the joint posterior distribution for the parameters d, p, r, and s.

As mentioned, the categorical nature of the responses given imply that there are only four possible outcomes for each observation. For example, if a target is presented, the subject may either respond "new" (a miss) or they may respond "old" (a hit). Similarly, if a distractor is presented, the subject may either respond "new" (a correct rejection) or "old" (a false alarm). Because there are only two possible outcomes for each presented stimulus, we only need to be concerned with two of the possible four outcomes: a hit and a false alarm. Having the number of hits and false alarms, along with the number of each stimulus type, we can express a likelihood function and build a Bayesian model for the data.

Standard Bayesian Approach Myung et al. (2007) expressed the hit and false alarm rates as expectations in two ways. First, they provided integral expressions for these rates, which serve as the true expectation (see Equations 4.2 and 4.3). These integrals are difficult to evaluate precisely, and when using R, if proper precautions are not taken, errors will occur. Myung et al. also derived asymptotic expressions for the integrals equations (see Equations 4.4 and 4.5). These asymptotic expressions are much simpler equations and only require the calculation of the cumulative density function of the normal distribution. To investigate these two sets of equations, both forms will be used in the estimation of the posteriors. The responses to stimuli of one particular type (i.e. a target or distractor) arise from a binomial distribution. The binomial distribution specifies the probability of observing a certain number of "successes" out of a possible n attempts. For each independent, identically distributed response, the probability of a single success is the only unknown parameter. The total number of hits arises from a binomial distribution with n equal to the number of possible targets presented in the jth condition, denoted N_{OLD_j}, and probability parameter equal to $P(HIT_j|\theta)$, the hit rate for the jth condition. Similarly, the total number of false alarms arises from a different binomial distribution with n equal to the number of possible false alarms in the jth condition, denoted N_{NEW_j} and probability parameter equal to $P(FA_j|\theta)$, the false alarm rate in the jth condition. Thus, the likelihood of the data $Y_j = \{Y_{j,HIT}, Y_{j,FA}\}$ consisting of the number of hits $Y_{j,HIT}$ and false alarms $Y_{j,FA}$ in the jth condition is the product of these two binomial distributions, or

$$L(\theta|Y_j) = \text{Bin}(N_{OLD_j}, P(HIT_j|\theta))\text{Bin}(N_{NEW_j}, P(FA_j|\theta)). \tag{4.6}$$

To build our Bayesian models, we need to specify a prior distribution for each parameter. Each of the four parameters in the model are bounded between $[0, 1]$. There is some research available that we could (and should) use to guide our decision about the priors for each of the four parameters. However, the data were simulated, so we can not rely on previous experiments to inform our prior. That is, each of the parameters may fall anywhere in the interval $[0, 1]$ with equal probability. Thus, we establish priors reflecting this situation by choosing continuous uniform priors for each parameter

$$d, p, r, s \sim \text{Beta}(1, 1), \tag{4.7}$$

where the beta distribution places uniform density on the interval $[0, 1]$ as discussed in Chapter 2.

Given the priors and the likelihood function in Equation 4.6, we can now write the joint posterior distribution of the parameters as

$$\begin{aligned}\pi(d, p, r, s|Y) &\propto \prod_{j=1}^{C} L(\theta|Y_j)\pi(d)\pi(p)\pi(r)\pi(s)\\ &\propto \prod_{j=1}^{C} L(\theta|Y_j),\end{aligned}$$

where C is the number of conditions ($C = 2$ here). The last line in the equation above holds because the density for each of the priors on d, p, r and s is always 1 on the interval $[0, 1]$.

To estimate the posterior distribution, I used Markov chain Monte Carlo (MCMC; see Gelman et al., 2004, Robert and Casella, 2004). I took 20,000 samples with a burn-in period of 1,000 samples, and collapsed across 8 chains. This same procedure was used for both the exact and the approximate evaluations of the hit and false alarm rates. Standard techniques were used to assess convergence using the coda package in R (Plummer et al., 2006).

ABC Approach To use the ABC approach, we will pretend that the expressions for the likelihood function are unavailable. For ABC, all that is required is the specification of a discriminant function $\rho(X, Y)$ and a tolerance threshold ϵ.

The choice for ϵ is dependent upon the selection of $\rho(X, Y)$. Thus, we begin by first choosing a function that compares the simulated data X to the observed data Y. An obvious choice, given the reliance upon the hit and false alarm rates in the likelihood function (see Equation 4.6) is to simply compare the hit and false alarm rates of the two data sets. Accordingly, I compared the mean absolute differences in the two rates, to equally weight each response type. Formally, we set

$$\rho(X, Y) = \frac{1}{2C} \left(\sum_{j=1}^{C} \left| \frac{X_{j,FA} - Y_{j,FA}}{N_{NEW}} \right| + \sum_{j=1}^{C} \left| \frac{X_{j,HIT} - Y_{j,HIT}}{N_{OLD}} \right| \right) \tag{4.8}$$

where the simulated data $X_j = \{X_{j,HIT}, X_{j,FA}\}$ are defined similar to the observed data above.

Ideally, we would like our simulated data to perfectly match our observed data. For $\rho(X, Y)$ in this example, this situation arises only if $\rho(X, Y) = 0$. This provides us with a final tolerance threshold to converge to, and the other values of ϵ may be chosen somewhat arbitrarily. The selection of these intermediate values for ϵ is one of computational consideration. Selecting ϵ values that are spaced far apart may make it difficult to find a proposed θ^* that will lead to acceptance, resulting in more rejection and longer computational times. By contrast, selecting ϵ values that are very close together will make it very easy to find a proposed θ^* that will lead to acceptance, but more iterations will need to be performed, resulting in longer computational times. In Chapter 2, I suggested that ϵ values should monotonically decrease, and that each step size is small enough to result in reasonably high acceptance rates. I set $\epsilon = \{0.2, 0.1, 0.08, 0.04, 0\}$.

For this simulation, I implemented the ABC PMC algorithm developed by Beaumont et al. (2009), discussed in Chapter 2. I used 1,000 particles to approximate the posterior distribution. The priors for the parameters d, p, r, and s were identical to the standard Bayesian specifications above.

Results

To generate the data, I simulated 20 responses in one condition and 30 responses for the second condition using the parameters $d = 0.4$, $p = 0.5$, $r = 0.75$ and $s = 0.2$ for each condition. Figure 4.1 shows the estimated posterior distributions using ABC (the histograms) and MCMC (densities). The black solid lines show the estimated posteriors using the exact expressions for the likelihood, whereas the dashed black lines show the estimated posteriors using the asymptotic expressions for the likelihood. The dashed vertical lines show the true parameter values used to generate the data. The estimated posterior distributions obtained by using ABC closely resemble the posterior distributions obtained using MCMC and the exact expressions for the likelihood. However, when using the asymptotic expressions for the likelihood, we obtain slightly different estimates of the posteriors. The close alignment of the ABC estimates with the exact posteriors suggests that the selected combination of $\rho(X, Y)$ and ϵ worked well for this model.

To evaluate the exact expressions consistently, it was necessary to specify a very low tolerance value for the numerical integration function in R to produce the fully Bayesian estimates of the posterior. This increase in precision drastically slowed the MCMC sampler. In fact, obtaining the exact Bayesian estimates took several days. By contrast, when using the asymptotic expressions, the simulation took around half an hour, but the estimated posteriors were systematically different. When using the ABC PMC algorithm, fitting the data took about two hours, but the estimated posterior distribution closely matched the true posterior. It is interesting that using ABC one can obtain approximately the same posterior distribution, but more than an order of magnitude faster than the preferred method.

For the ABC method, our selected values for ϵ resulted in very high rejection rates. Even for the largest value for $\epsilon = 2$, the acceptance rate was 8.41%. This number dropped to 0.22% for $\epsilon = 0.04$ and an astonishing 0.024% for the final $\epsilon = 0$. Although the simulation was successful, I recommend using more values for ϵ in future applications.

This section has demonstrated the effectiveness of the ABC approach in estimating the parameters of BCDMEM. Although this example was a simple modeling problem, it was critical in demonstrating that ABC can be used when no likelihood function is available for memory models. Specifically, the combination of $\rho(X, Y)$ and ϵ in this example led to a suitable approximation of the posterior. This means that the selections made in this example can be applied more generally, and can be used in fitting other models to data of this type.

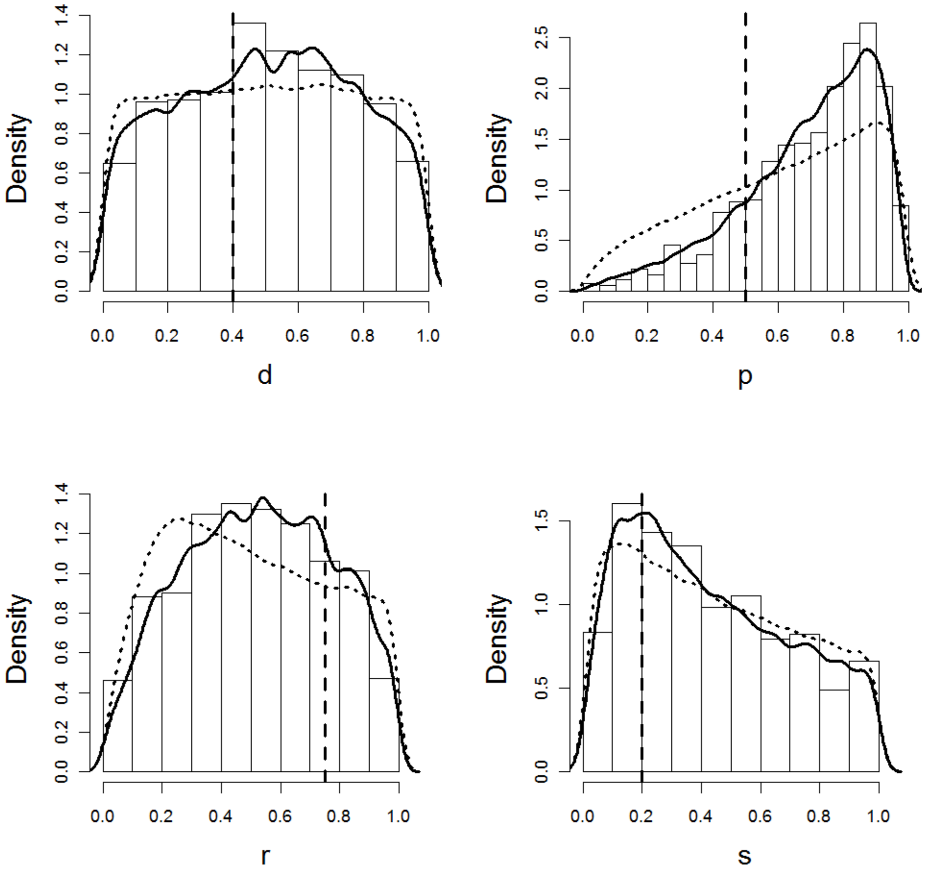

Figure 4.1: The approximate marginal posterior distributions for each of the free parameters in the BCDMEM model using ABC (histograms) and MCMC (densities). The solid black densities represent the estimated posteriors using the exact expressions and the dashed densities represent the estimate posteriors using the asymptotic expressions. The dashed vertical lines represent the true value of the parameters used to generate the data.

4.1.4 Fitting Experimental Data

The section above demonstrated that ABC is a legitimate alternative to fully Bayesian techniques in the absence of a likelihood function. However, the above example made the correct assumption that the data arose from BCDMEM. None would argue that the psychological models used to explain behavior are the "true" models. We are all well aware that the proposed model is wrong, but it may be useful in providing insight into the underlying mechanisms that trigger the overt response. This begs the question, "How does ABC compare when the model is wrong?" In this section, I intend to answer that question by fitting the data presented in Dennis et al. (2008) using ABC and the asymptotic equations.

The key feature that distinguishes BCDMEM from other models of episodic recognition is that it assumes that noise is generated from the other contexts in which an item has appeared rather than from the other items that appeared in the study context. This assumption predicts that as the number of items in the list increases there should be no decline in performance due to interference. However, other factors may produce a list length effect that are not related to interference. For example, one might predict that as the retention interval increases there would be a decline in performance. A retention interval is the amount of time an observer spends committing words to memory. Without appropriate controls, if the test phase immediately follows the study phase, then the longer list will have a longer retention interval. If a longer retention interval results in a decline in performance, then the reason for a decline in performance as a function of longer list lengths will be confounded. That is, the reason for the decline might be due to the longer retention interval, or it might be due to the extra noise encountered from additional items.

Another factor that could contribute to a decrease in performance for long lists is the nature of the contextual reinstatement. One way to examine this is the retroactive design. In this design, the retention interval is equated for the two list lengths by introducing filler activity after the short list and by only testing items from the start of the short list. In this situation, subjects will need to recall the items of interest from the start of the study phase. However, in the long list condition testing begins shortly after the end of the study phase and subjects are not informed that only items from the start of the list will be tested. Thus, the subjects will have no reason to reinstate context, which implies that the subjects may use the initial context from the beginning of the study phase. If context drifts from the beginning to the end of the study phase, the context at the end of the list will systematically mismatch the context at the beginning, where the subjects studied the items of interest. One way to test this possibility is to introduce additional filler activity between the study and test phases in both the short and long list conditions. This additional filler activity will require subjects to reinstate context in both conditions. This manipulation was employed in the Dennis et al. (2008). They found that when context was not reinstated (the no filler condition), there was a significant list length effect (at least using standard significance testing techniques). However, when context was reinstated (the filler condition), there was no list length effect.

The Model

To model the data presented in Dennis et al. (2008) using BCDMEM, I assume that the d parameter is fixed across length conditions when the additional filler activity is present. Under these assumptions there is no mechanism by which BCDMEM can predict a list length effect. However, when filler is only present in the short condition, the d parameter is allowed to vary between the short and long conditions. This assumption will allow us to demonstrate that reinstatement is more likely occurring in the short condition rather than the long condition. To do so, we let δ represent the d parameter in the short, no filler conditions, but let d represent the d parameter for every other condition. For notational convenience, I introduce a binary indicator variable, F_j to indicate when d ($F_j = 0$) or δ ($F_j = 1$) should be used for the forgetting parameter in condition j.

The data from Dennis et al. (2008) also included a word frequency manipulation. Word frequency is expected to influence the amount of context noise that an item will generate, but is not allowed to change across length conditions. Word frequency is modeled by the retrieved context activation parameter p. When words are more frequent, they will be associated with more contexts, and more nodes in the retrieved context vector will become active. To model this, we let p represent the retrieved context activation parameter in low frequency word conditions and let τ represent p in high frequency word conditions. We then introduce the binary indicator variable W_j to indicate high- ($W_j = 1$) or low- ($W_j = 0$) frequency words on the jth condition. For the ith subject in the jth condition, we let the number of hits be denoted by $Y_{i,j,HIT}$ and the number of false alarms by $Y_{i,j,FA}$. To fit the model to the data, v was set to be 200 and s was fixed at 0.02, to be consistent with current literature.

Standard Bayesian Approach Equations 4.2 and 4.3 can be very difficult to calculate precisely. It was mentioned in the previous section that a much simpler, four-parameter model took several days to complete. With many more parameters and calculations, as in this hierarchical model, it is no longer reasonable to continuously evaluate the exact expressions in the MCMC chain. However, for the sake of comparison, I will use Equations 4.4 and 4.5 to estimate the posterior distributions for the data.

Using the new parameters and indicator variables defined above, and given that s is now fixed, the parameter vector θ for the ith subject and jth condition will be redefined as

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

With this short-hand, we continue to write the hit and false alarm probabilities $P(HIT_{i,j}|\theta_{i,j})$ and $P(FA_{i,j}|\theta_{i,j})$ as in Equations 4.2 and 4.3.

As in the previous example, the number of hits $Y_{i,j,HIT}$ and false alarms $Y_{i,j,FA}$ arise from a binomial distribution with the number of trials equaling the number of targets and distractors, respectively, with the hit $(P(HIT_{i,j}|\theta_{i,j}))$ and false alarm $(P(FA_{i,j}|\theta_{i,j}))$ rates defined by Equations 4.2 and 4.3. We can then write the likelihood function as

$$L(\theta_{i,j}|Y_{i,j}) = \prod_{i=1}^{S}\prod_{j=1}^{C} \text{Bin}(N_{OLD}, P(HIT_{i,j}|\theta_{i,j}))\text{Bin}(N_{NEW}, P(FA_{i,j}|\theta_{i,j})), \tag{4.9}$$

where $S = 48$ is the number of subjects and $C = 8$ is the number of conditions.

Each of the parameters in this model have the common restriction of $d, p, r, \delta, \tau \in [0, 1]$. As such, we can again use the flexible beta distribution. However, because some of the parameters in BCDMEM are often very near the boundaries, we use the reparameterized beta distribution to facilitate easy estimation of the hyperparameters. This beta distribution is given by

$$\text{Beta}(p|\omega, \xi) = \begin{cases} \frac{\Gamma(\xi)}{\Gamma(\omega\xi)\Gamma(\xi(1-\omega))} p^{\omega\xi-1}(1-p)^{\xi(1-\omega)-1} & \text{if } 0 < p < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{4.10}$$

where $\omega \in (0, 1)$ is a mean parameter and $\xi \in (0, \infty)$ is a scale or dispersion parameter. To build a hierarchy, we assume that each of the subject-specific parameters come from a common distribution, or

$$\begin{aligned} d_i &\sim \text{Beta}(\omega_d, \xi_d), \\ p_i &\sim \text{Beta}(\omega_p, \xi_p), \\ r_i &\sim \text{Beta}(\omega_r, \xi_r), \\ \delta_i &\sim \text{Beta}(\omega_\delta, \xi_\delta), \text{ and} \\ \tau_i &\sim \text{Beta}(\omega_\tau, \xi_\tau). \end{aligned}$$

We can now specify priors for each of the group-level parameters. Because BCDMEM has never been fit hierarchically to data (or in a Bayesian framework), we have no information about the support of the hyperparameter distribution. We use noninformative priors to reflect this situation. The hyper-priors are the same for each of the five parameters, so 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 specify the following priors:

$$\begin{aligned} \omega &\sim \text{Beta}(1,1) \text{ and} \\ \xi &\sim \Gamma(.1,.1) \end{aligned}$$

for each element in the vectors ω and ξ.

To fit the model to the data, I used adaptive rejection sampling (Gilks and Wild, 1992) in a Gibb's sampling framework and Equations 4.4 and 4.5 to evaluate the probabilities $P(HIT_{i,j}|\theta_{i,j})$ and $P(FA_{i,j}|\theta_{i,j})$, respectively. I took 10,000 samples from the posterior distribution with a burn-in period of 1,000 samples and collapsed across 10 chains. Standard techniques were again used to assess convergence using the coda package in R (Plummer et al., 2006).

ABC Approach When extending ABC algorithms to hierarchical models, it becomes much more difficult to obtain a complete set of proposed parameter values that will lead to acceptance. Even the simple four-parameter model presented earlier in this section had incredibly low acceptance rates. There has been some interesting work to correct for this problem, which was discussed in Chapters 2 and 3. Although these methods are helpful in attenuating the overwhelming rejection rates, they are still quite slow. In addition, the stability and accuracy of the ABC approach decreases in high-dimensional spaces, attributable to the infamous curse of dimensionality (Beaumont, 2010).

To fit BCDMEM to the data without using the likelihood function, we will use the mixture algorithm presented in Chapter 3. In doing so, we can put the two approaches on the same grounds for the hyperparameters by again using adaptive rejection sampling (Gilks and Wild, 1992) for the hyperparameters, and standard rejection sampling for the lower-level parameters. I chose to use rejection sampling for the lower-level parameters because the interval for four of the parameters is bounded between zero and one, and after a preliminary investigation the posteriors for these parameters were reflective of the prior distribution. That is, they were quite variable, covering nearly all of the area in the interval $[0, 1]$. Thus, the MCMC within steps would not be very helpful in this situation.

For this model, we will use all of the same priors outlined in the previous section. ρ was set to Equation 4.8. However, the tolerance thresholds must be specified a little differently. To aid in convergence of the parameters, we specify an ϵ for each iteration t. I set

$$\epsilon(t) = \exp^{-.01t} + 0.10.$$

Notice that this function asymptotes at 0.10. It was necessary to raise the final tolerance threshold to allow a simultaneous fit for the many subjects in the many conditions of the experiment. I drew 10,000 samples with a burn-in period of 1,000 samples.

Results

A common procedure for evaluating the fit of a model to data in Bayesian statistics is to examine the posterior predictive density. The posterior predictive density is created by sampling parameter values from the posterior distributions, and using these parameter values to generate a new set of data. If we generate enough data sets, we will arrive at a Monte Carlo estimate for this predictive density. Formally, for a given parameter θ, assumed model $f(\cdot|\theta)$ and observed data Y, the posterior predictive distribution of new, future data set $\tilde{Y}$ is given by

$$f(\tilde{Y}|Y) = \int f(\tilde{Y}|\theta)\pi(\theta|Y)d\theta.$$

The posterior predictive density provides us with model predictions for what the data should look like, given the best-fitting model parameters obtained using the data Y. If the posterior predictive distribution exhibits any systematic deviations from the data that were observed, then this suggests that the model fit is poor. The degree of model fit can be assessed by the amount of data that falls outside some specified credible set for the posterior predictive distribution.

Figure 4.2 plots the data along with the posterior predictive distribution when using the ABC approach, and Figure 4.3 shows this distribution when using the asymptotic equations. The data are shown by the black contour lines and the posterior predictive density is shown by the gray dots. The gray dots have had random noise added to them to create the illusion of a density. Notice that the gray dots only take on values in certain regions of the receiver operating characteristic (ROC) space. This is because the data for each condition consisted of only 20 recognition judgments, 10 of which were targets and 10 of which were distractors. Thus, the hit rate and the false alarm rates for each subject in each condition can only be one of the eleven values in the set $\{0, 1/10, 2/10, ..., 10/10\}$. To evaluate the density at each square, we look at the proportion of gray dots in each square. A larger number of gray dots suggest a higher density at that particular value in the ROC space.

When comparing Figures 4.2 and 4.3, we see a misfit of the asymptotic equations due to the prediction of lower hit rates when the false alarm rate is equal to 0. Other than this region, the two densities are very similar.

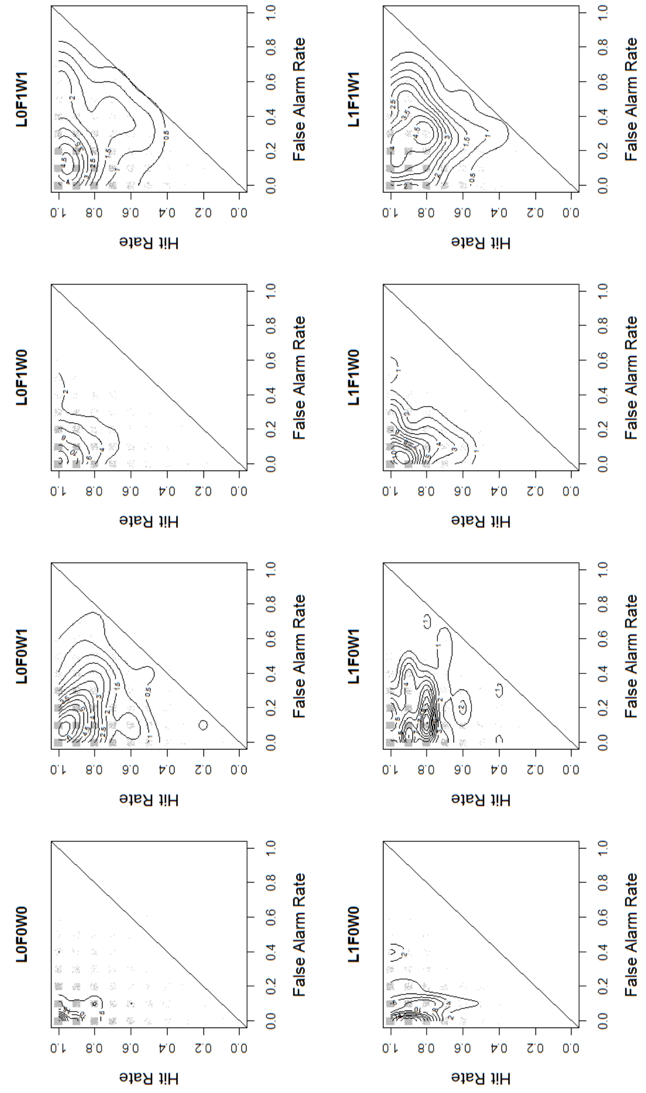

Figure 4.2: The posterior predictive distributions obtained when using ABC are represented by gray dots in the squares of possible values for the hit and false alarm rates in the experiment presented in Dennis et al. (2008). The observed data are shown by the black contours. Each condition is plotted with the condition code on the top of each panel: L corresponds to short (0) or long (1) conditions, F corresponds to filler (1) or no filler (0) conditions, and W corresponds to high (1) or low (0) word frequency conditions.

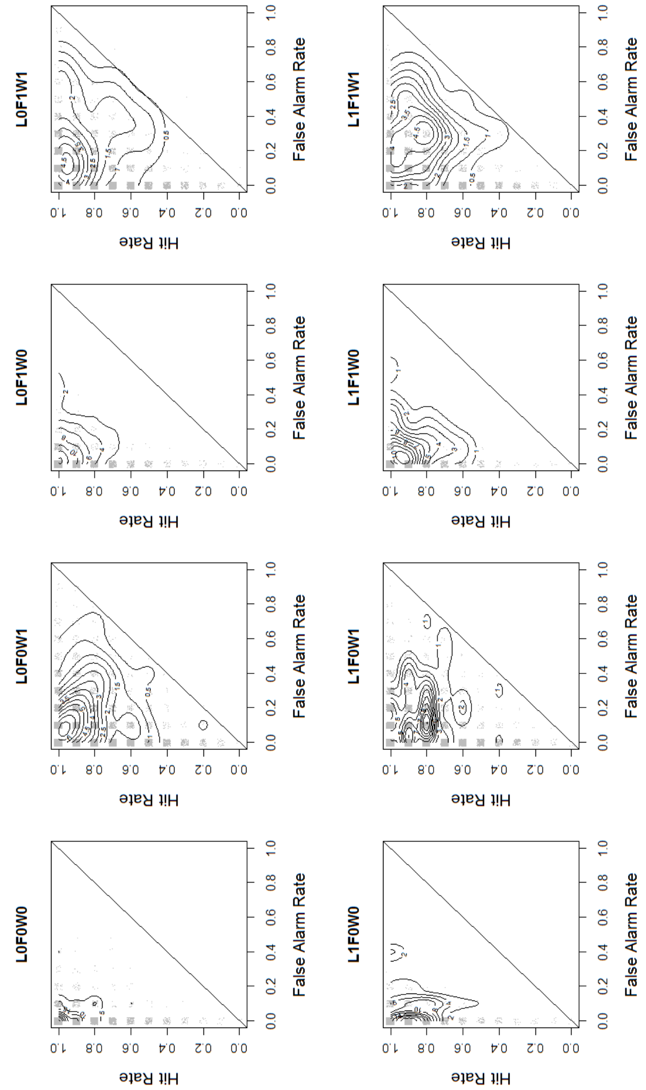

Figure 4.3: The posterior predictive distributions when using the asymptotic equations are represented by gray dots in the squares of possible values for the hit and false alarm rates in the experiment presented in Dennis et al. (2008). The observed data are shown by the black contours. Each condition is plotted with the condition code on the top of each panel: L corresponds to short (0) or long (1) conditions, F corresponds to filler (1) or no filler (0) conditions, and W corresponds to high (1) or low (0) word frequency conditions.

Typical fits of BCDMEM to data collapse across subjects to estimate the parameters. In a Bayesian context, we gain information about the subjects and the group simultaneously. A standard frequentist fit would result in a single value for each parameter. However, as the previous simulation suggested, the posterior distributions for each of these parameters are quite variable, spanning the entire range of the parameter space. It would be interesting to examine the range of possible parameter values for any subject, predicted by the model. This can be accomplished by generating a posterior predictive distribution for the lower-level parameters, conditional on the hyperparameters. These densities are similar to densities generated above (see Figures 4.2 and 4.3); however, we are predicting lower-level parameters for hypothetical subjects, rather than predicting hypothetical data. If these posterior predictive densities are highly variable, then that suggests that summarizing a group's performance with a single point value superfluously condenses the data.

To examine this, I generated posterior predictive densities for every possible outcome for each of the five parameters d, p, r, δ, τ. The procedure for generating these distributions entails sampling from the joint distribution of the hyperparameters, and drawing a single lower-level parameter from the resulting beta distribution specified by the sampled hyperparameters. This procedure was replicated 1,000 times. The resulting densities are shown in Figure 4.4 for the estimates obtained using ABC (black lines) and using the asymptotic equations (histograms). The left two panels show the densities corresponding to filler (bottom) and no filler (top) conditions, the middle two panels correspond to high (bottom) and low (top) word frequency conditions, and the right panel shows the learning rate which was assumed to be the same for each subject in each condition. Comparing the densities obtained using ABC and the asymptotic equations, we can see some large discrepancies in all of the parameters, with the possible exception of d. The reason for the discrepancies is not yet clear. Both posteriors were obtained through approximate methods, so deciding which estimate is best is difficult. However, in the previous example, the posterior estimates obtained using the approximations were unsatisfactory. It is possible that using the asymptotic equations results in unreliable posterior estimates for more complex models. It is also possible that the estimates obtained using ABC are unreliable, because we were unable to obtain a perfect match to the data (the smallest value for ϵ was 0.10). Additionally, comparing Figures 4.2 and 4.3 we see that the estimates obtained using ABC did produce a better fit to the data. Regardless of the reason for the discrepancies, I will only consider the estimates obtained using ABC further.

By comparing the two left panels in Figure 4.4, we see that there is a slight difference between the filler and no filler conditions. This relationship was explored further by computing the proportion of samples such that $f(\tilde{\delta}|Y) - f(\tilde{d}|Y) > 0$ for each subject. For all of the subjects, this proportion was greater than 0.5 (the mean proportion was 0.70). This means that all of the subjects experience more forgetting in the short, filler and no filler conditions than all of the other conditions combined. Similarly, the effects of the word frequency manipulation were examined by computing the proportion of samples such that $f(\tilde{\tau}|Y) - f(\tilde{p}|Y) > 0$ for each subject. In this comparison, each of the 48 subjects had proportions greater than 0.5, which means that the context sparsity parameter was higher for high frequency words than for low frequency words (the mean proportion was 0.86). BCDMEM predicts that as words become more frequent, the context sparsity parameter p should increase, which increases the amount of interference a subject experiences during test (Dennis and Humphreys, 2001).

Fitting such a complex model is reassuring for the utility of the ABC approach. However, the accuracy of the posteriors is questionable for this data. Even with an advanced algorithm, I was unable to obtain perfect estimates of the posterior distribution. In fact, I was forced to converge to an ϵ of 0.10. While this may seem like a large number, we emphasize that an average discrepancy of 0.10 in Equation 4.8 indicates only a single mismatch for each of the 16 data types (eight conditions by two possible response outcomes). In addition, $\rho(X, Y)$ is bounded by zero and one and reflects the percentage of mismatching data, which implies that we obtained a 90% match to the observed data. Thus, I argue that an ϵ of 0.10 does provide a suitable approximation to the data.

It is also important to point out that a large final value for ϵ may not necessarily indicate poor accuracy of the estimated posterior. When fitting experimental data, we are guaranteed that the model assumption is wrong. The best we can hope for is to estimate the posterior distribution of the parameters for the assumed model to hopefully gain insight into the behavior of interest. However, if the model we have assumed is very wrong, it will have a very difficult time producing data X that are sufficiently close to the observed data Y. In a standard Bayesian analysis, this type of misfit would be reflected by smaller values of the likelihood function. Because $\rho(X, Y)$ is the ABC counterpart to the likelihood, large values are more reflective of a misspecified model than an indicator of accuracy of the estimated posterior.

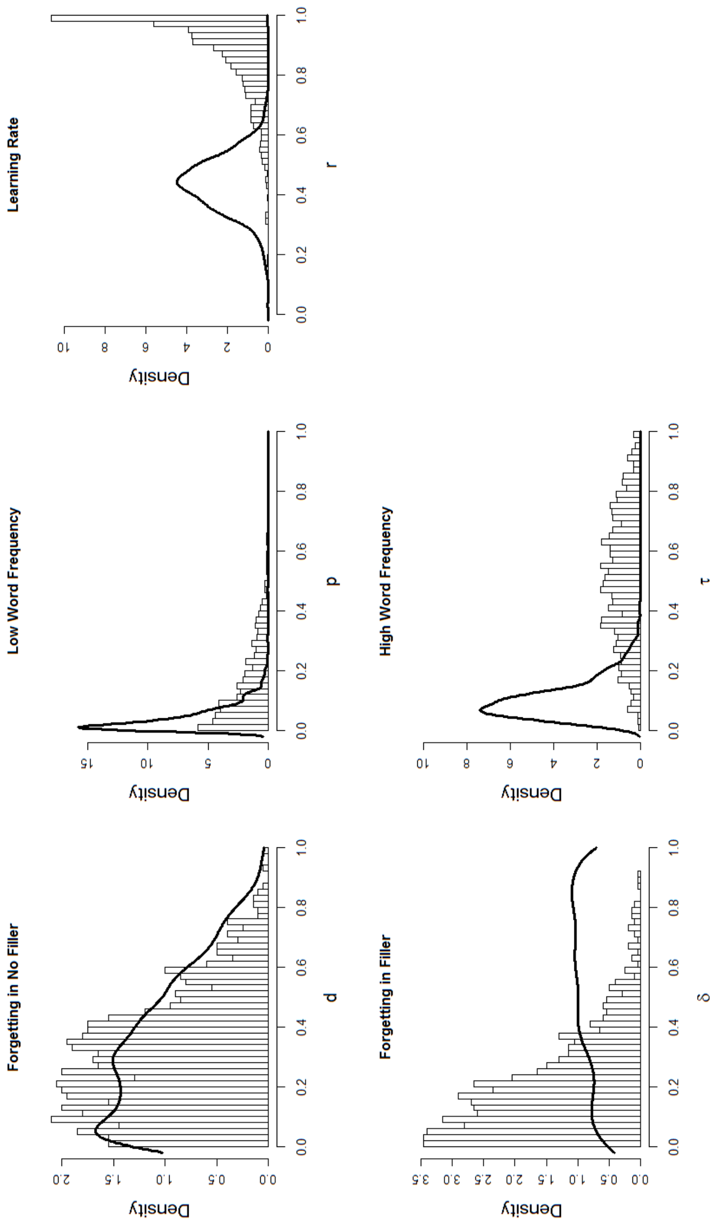

Figure 4.4: The posterior predictive distributions for each of the five lower-level parameters. These distributions represent the probability for each parameter value for any given subject. The left panels show the effects of filler (bottom) and no filler (top) conditions, the middle panels show the effects of high (bottom) and low (top) word frequency condition, and the right panel shows the learning rate parameter.

4.1.5 Summary

In this section, I first introduced the model BCDMEM and summarized how Myung et al. (2007) derived expressions for the hit and false alarm rates. These expressions allowed us to compare estimates of the posterior distribution obtained by standard Bayesian techniques to the estimates obtained using ABC. I showed that the ABC estimates closely resembled the estimates obtained using the exact expressions (Equations 4.2 and 4.3), but that the estimates obtained using the asymptotic expressions (Equations 4.4 and 4.5) did not closely match either the ABC or exact expression estimates. After demonstrating that ABC could recover the true posterior distribution, I used it to fit the data presented in Dennis et al. (2008). To do so, I built a hierarchical BCDMEM model that incorporated both subject and condition specific parameters. This model is fairly complex, consisting of 15 parameters and three levels, and provided us with a great modeling challenge. I then used Algorithm 5 to obtain estimates of the posterior distribution for the parameters in BCDMEM. I showed that the model fit the data well through inspection of the posterior predictive distributions. I also showed that summarizing the model parameters with a single point estimate may not be the best choice, given the large variances observed in the posterior predictive densities.

In this section, I demonstrated the utility of the ABC approach for the model BCDMEM. We now turn our attention to another model of episodic memory with a very different view of how noise in the decision process may arise. In the next section, I will apply ABC techniques to the Retrieving Effectively from Memory model (REM; Shiffrin and Steyvers, 1997).

4.2 REM

Another popular model of recognition memory is the retrieving effectively from memory (REM) model (Shiffrin and Steyvers, 1997). I chose this model for its simplicity, popularity and because, at the time I began fitting this model, I was unaware of work deriving the likelihood function. As mentioned in Chapter 2, Montenegro et al. (2011) have derived expressions for the hit and false alarm rates for REM as well. These derivations are formidable, and will not be discussed in detail. While REM was introduced formally in Chapter 2, the predictions of REM were not formalized. I will begin by investigating the predictions of REM by using ABC to fit synthetic data.

4.2.1 Predicting The List Length Effect

Unlike the model BCDMEM, global memory models such as REM make different predictions as the number of items presented in the study phase increases. In particular, REM predicts that as the number of items presented increases, performance will decrease. This intuitive prediction has been found in several studies. However, as the number of experimental controls for the recognition memory task have increased, the magnitude of this effect has been attenuated. For example, Dennis et al. (2008) show that if you control for the length of the list by equating the lengths of the retention intervals, the list length effect diminishes (in their study they conclude that there was no length effect).

Although the list length effect is a well-known prediction of REM, there has not been much work dedicated to quantifying the relationship between the number of study items and the model predictions. Montenegro et al. (2011) provide expressions for the hit and false alarm rates for REM. To accomplish this, it was necessary to treat the number of items at study as a parameter in the model. However, the number of items at study is usually known and is not modeled. Despite this, the number of items at study could be used as a parameter meant, for one reason or another, to decrease performance in the recognition memory task. This technique will be used in the final section to model the effects of filler activity. In the next section, I investigate the predictions of REM under varying list length assumptions.

Model Predictions for List Length

The goal of this simulation study is to understand the complexity of the model under different study list lengths. One way to examine this relationship is to simulate the model under an exhaustive set of parameter values and examine the distribution of the resulting statistics, such as the hit and false alarm rates. For this simulation, the exhaustive set of parameter values will consist of the prior distribution. The distribution of possible data generated under the prior is known as the prior predictive density (Gelman et al., 2004). In the case of noninformative priors, this density is similar to the normalized likelihood function. In other words, to see the raw predictions of the model, we will select priors that have constant density across the parameter space.

For this simulation, we will consider the three parameter version of REM as it was presented above, with w set equal to 20. Because these parameters are probabilities bounded by zero and one, a convenient noninformative prior is given by

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

As a check against the predictions of BCDMEM, we will also consider a three parameter version of BCDMEM, containing only d, p, and r, with $v = 20$ and $s = 0.02$. These models have the same prior dimensions, and so they can easily be compared. We place the same prior on these parameters, or

$$d, p, r \sim \text{Beta}(1, 1).$$

These priors make each of the parameter combinations of the two models equally likely to be chosen. Thus, the only differences in model predictions we observe will be due entirely to the models themselves, and their interactions with the experimental design.

To examine the prior predictive density for different list lengths, I chose four different list lengths: 10, 20, 80, and 2,000 words. Of course, this final list length is impractical, but it serves to demonstrate the limiting behavior of REM. At test, the test list was comprised of 10 targets and 10 distractors under each study list length. For each list length, we first sample a parameter combination from the prior distribution. Next, we generate predictions for 20 observers using the sampled parameter combination, and record the mean hit and false alarm rates. We perform this simulation 10,000 times under each model by list length combination.

Figure 4.5 shows the results of this simulation. The top panel shows the predictions for REM and the bottom panel shows the predictions for BCDMEM. The columns of Figure 4.5 correspond to the four different list lengths. For a guide, the line representing unbiased responding (going from top left to bottom right) and a line of chance performance (going from bottom left to top right) are plotted. As expected, for BCDMEM we see that no systematic changes occur in the predictions of the model under the different list lengths. By contrast, REM makes very different predictions for these different list lengths. In particular, as the list length increases, the hit rate drops and the false alarm rate increases. These predictions also show that REM predicts a very narrow range of biases. That is, it tends to predict that the model makes slightly more “new” responses than “old” responses.

One final note is that the variance of the predictions made by REM decreases with increasing list length. At first, it nearly covers the line representing unbiasedness responses, but as the list length increases, it seems to be converging to the point $(0.45, 0.50)$. When a model makes very specific predictions, it will be heavily favored when observed behaviors fall in that region. However, if the data do not fall in that region, it will be difficult for the model to account for these data, clearing the way for more flexible models.

This section has demonstrated that the experimental design alone can alter the predictions of REM. Thus, not only must the classic model characteristics outlined in Myung and Pitt (1997) influence the model selection procedure, but so must the design of the experiment. This simulation was useful in demonstrating the effects of the design absent the use of the model parameters. However, it would be interesting to see if REM can still make the list length prediction when the number of words at study is held constant. While this section did not highlight the use of ABC, it motivates the following section, which can be answered easily with ABC techniques.

Effects of List Length on Model Parameters

Although the previous simulation showed that the number of items presented at study is crucial to the prediction of a list length effect in REM, the question remains as to whether or not other parameters can produce the same effect. To

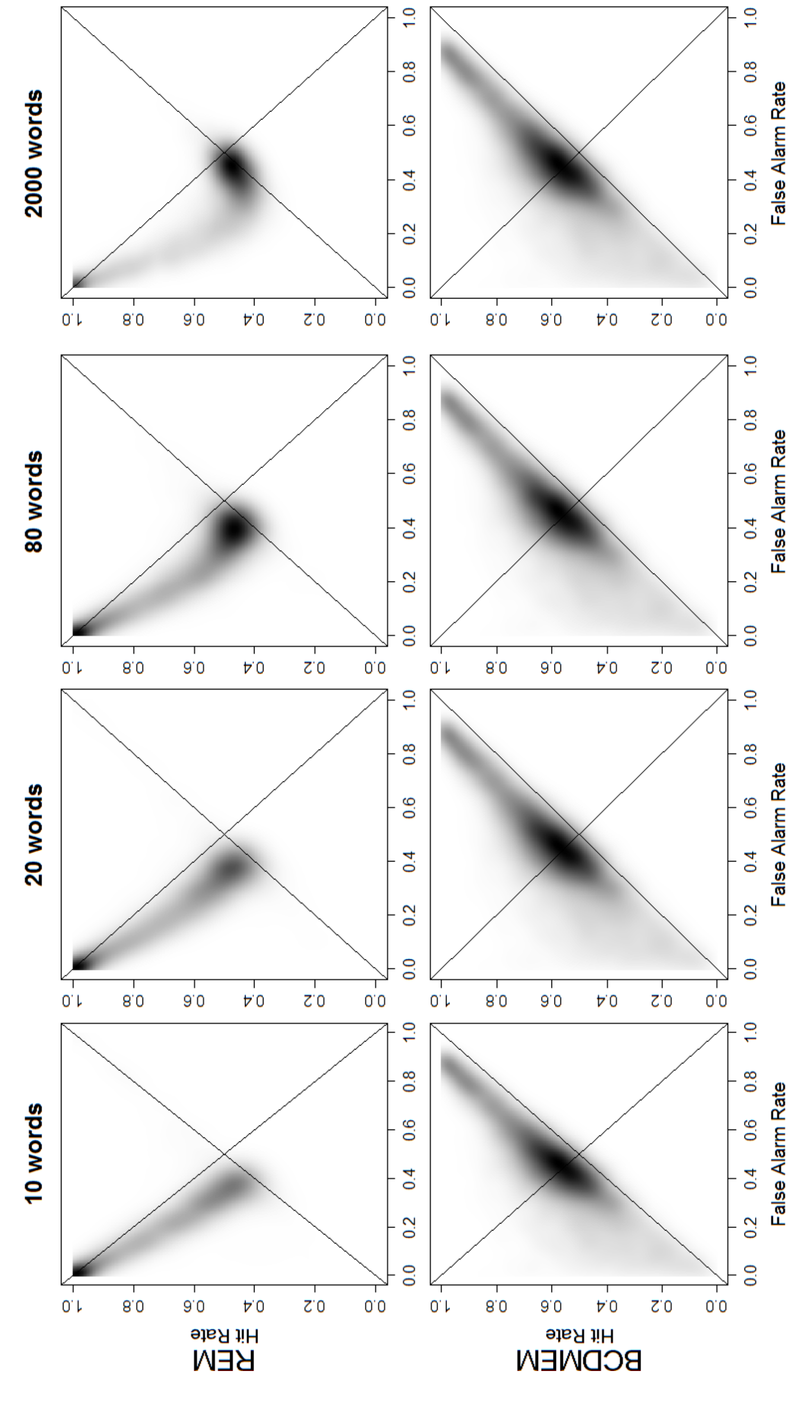

Figure 4.5: The prior predictive density under four different list lengths for REM (top panel) and BCDMEM (bottom panel). Darker regions indicate higher density.

examine this, we will need to hold the number of items at study constant. Otherwise, we will be unable to dissociate the effects of the parameters from the effects of the experimental design.

For this simulation, I generated data to mimic the three types of list length effects. Three simulated subjects participated in a recognition memory experiment with five conditions. The five conditions varied only in the number of items presented at study, which were 10, 20, 40, 60, and 80. At test, the number of targets and distractors were 10, composing a 20-item test list.

The subjects' responses were constructed to produce a list length effect (performance decreases as the number of study words increase), a null list length effect (performance remains constant regardless of the number of words at study), and an inverse list length effect (performance increases with increasing study words). To generate data showing the list length effect, I manipulated the number of hits and false alarms so that the hit rates were 0.9, 0.8, 0.7, 0.6, and 0.5, and the false alarm rates were 0.1, 0.2, 0.3, 0.4, and 0.5 for the five conditions. Constructing the data in this way resulted in d' values of 2.56, 1.68, 1.05, 0.51, and 0. To generate data showing a null list length effect, the hit and false alarm rates were held constant at 0.7 and 0.3, respectively. This resulted in d' values of 1.05 for each of the five conditions. For the inverse list length effect, the hit and false alarm rates used for the list length effect were simply reversed. Generating the data in this way keeps the overall d' constant for each of the three subjects, so that any systematic changes in the estimated parameters will indicate that a list length effect can be produced by manipulating those parameter values. The left panels of Figure 4.6 plot the d' values for each of the three effects or subjects (rows) as a function of the number of words at study for each of the three simulated subjects.

To fit each of these subjects, I again used the ABC PMC algorithm with $\rho(X, Y)$ again defined to be the average difference in the hit and false alarm rates for each of the five conditions (see Equation 4.8. I chose the set of $\epsilon = \{0.3, 0.2, 0.17, 0.12, 0.10, 0.08, 0.06, 0.05\}$. It should be noted that I am not requiring that the simulated data equal the observed data, as I have done in previous simulations. This is due to the difficulties encountered in fitting the third subject (the inverse list length effect). As mentioned, REM does not predict this type of behavior; instead, it predicts the opposite type of effect. Thus, to fit the inverse list length data, I had to settle for slightly larger values of ϵ for these data. To remain consistent, the estimates obtained using the same restrictions across subjects are reported here. [5] For this simulation, I used 1,000 particles.

Figure 4.6 shows the estimated joint posterior distributions for c versus u (middle-left panels), g versus u (middle-right panels), and g versus c (right panels). In comparing the three rows, no obvious systematic differences in the estimated posteriors appear. However, there may be a slight decrease in the marginal distribution for c. I argue that the reason for this decrease is not due to the effects of list length, but rather the low performance on the first two conditions of the last subject (the inverse list length effect). To see this, first note that the number of words is not equally spaced. Secondly, as demonstrated in Shiffrin and Steyvers (1997), REM predicts a nonlinear, monotonic (i.e. a decaying power function)

[5] For the standard and null list length effects, I was able to converge to a perfect fit of $\epsilon = 0$. However, after careful inspection of the joint posteriors, no systematic changes were present, justifying the fitting procedure used.

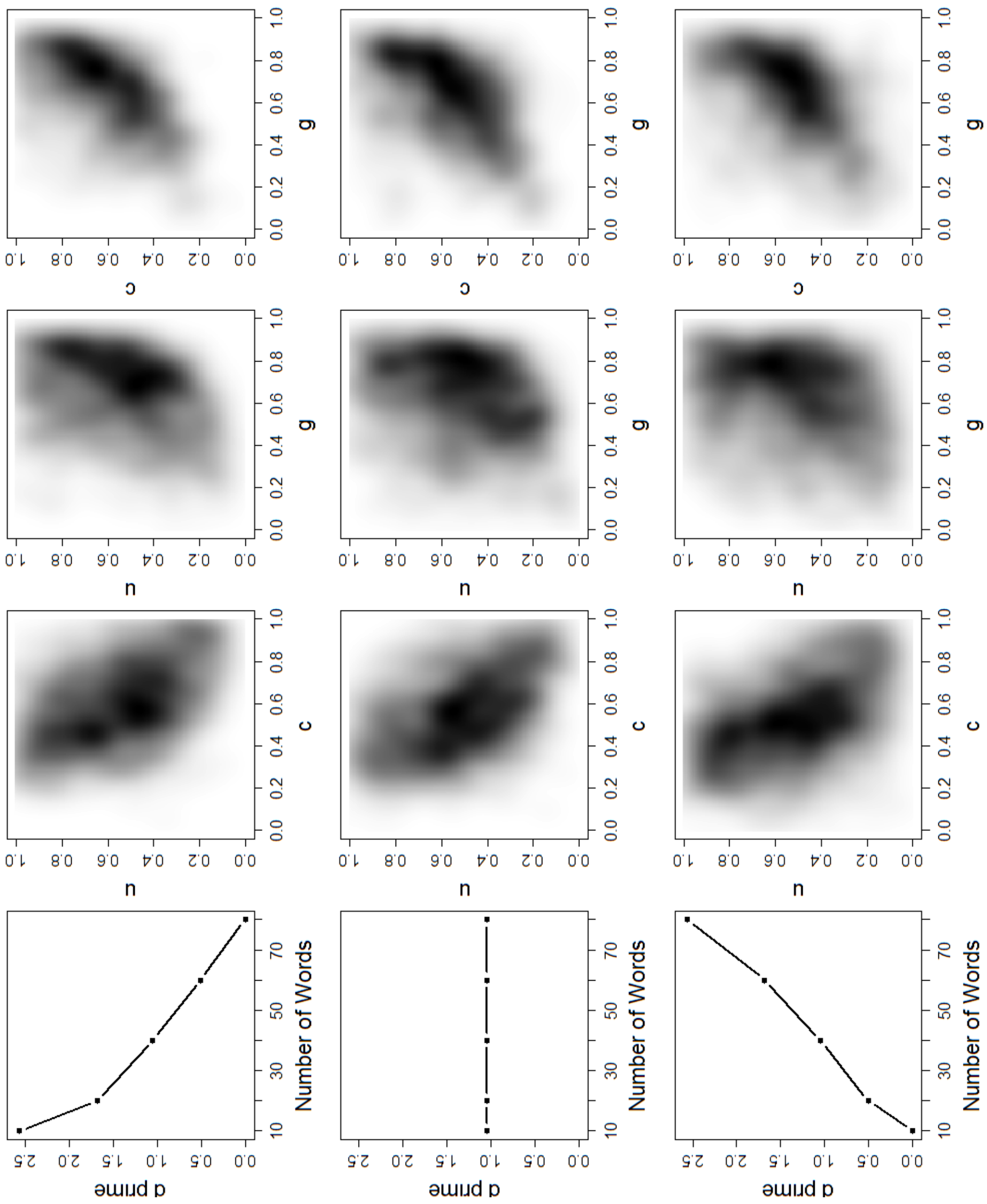

Figure 4.6: The relationships between c and u (second column), c and g (third column), and u and g (last column) for three different types of list length effects (first column). The first row shows the standard list length effect, the middle row shows no difference across list lengths, and the last row show the inverse list length effect.

decrease in d' as the number of words at study increase. To maximize the fit of REM to the data, c must be slightly lower to minimize the average differences in the hit and false alarm rates.

This section has shown that no other combination of the parameter values in REM can produce the list length effect. This finding emphasizes the consideration of the experimental design when comparing competing models. That is, under different experimental conditions, REM or BCDMEM may fit the data better due to an implicit constraint on global memory models. This constraint occurs because the number of traces in the episodic matrix increases with the number of study items. On average, this increase in the number of traces will produce a decrease in the λ_js in Equation 2.4 for targets. This average decrease in λ_j carries over into the decision rule, because Φ in Equation 2.5 will also be slightly smaller, which results in more "new" responses. This reasoning explains the slightly biased patterns observed in Figure 4.5.

4.2.2 Predicting Improvements in Performance

From the description of the parameters in the above section, predicting how REM might account for varying degrees of discriminability may be intuitive. Because u is the probability of copying a feature, and c is the probability of copying a feature correctly, then one might predict that for high discriminability, u would need to be high enough to allow for some features to be copied, and c would also need to be high to copy these features correctly. However, to explain how REM would account for low discriminability, one would now need to adjust the above explanation by restricting u and c from becoming too high. This restriction will lead to fewer features being copied and fewer features copied correctly. Thus, the performance would be worse.

While these explanations are correct, they are qualitative and not quantitative. Researchers who are very familiar with REM might be able to provide a more detailed account on how these parameters might interact, but the nature of the joint relationships has never been examined. Understanding how parameters are jointly distributed is imperative to understanding how model parameters affect model predictions. For example, consider the random variables X and Y, who are jointly distributed from a multivariate normal distribution. Knowing this joint distribution implies that both X and Y are marginally normally distributed. However, the converse is not true. That is, just knowing that X and Y are both normally distributed does not imply that the joint distribution of (X, Y) is multivariate normally distributed. Considering this, the above explanation of how REM might account for low values of discrimination is insufficient. Additionally, it does not address how small c and u must each be restricted to predict the low discrimination. For example, perhaps c has a more dramatic effect on performance than u. This implies that c may need to be restricted to smaller values than the restriction placed on u.

This section is devoted to providing a more systematic explanation of how REM accounts for improvements in performance. To do so, I generated data for three simulated subjects to represent three different levels of discrimination (high, medium and low) in a recognition memory experiment. In this experiment, the study list consisted of 20 words and the test list consisted of 40 words, 20 of which were targets (the study list) and 20 were distractors. The responses for the three subjects were then constructed to reflect the three different levels of discriminability. The first subject responded correctly for each item, resulting in a hit rate of 1.0, a false alarm rate of 0, and after the standard edge correction (Kadlec, 1999), a d' of 3.92. The second subject was constructed to have a hit rate of 0.85 and a false alarm rate of 0.25, resulting in a d' of 1.53. The last subject had a hit rate of 0.70 and a false alarm rate of 0.30, resulting in a d' of 0.91.

I then used the ABC PMC algorithm to fit the data for each of the three subjects. $\rho(X, Y)$ was again chosen as in Equation 4.8, where $C = 1$. The values of ϵ were set to $\{0.2, 0.17, 0.12, 0.08, 0.04, 0\}$. I again used 1,000 particles to estimate the joint posterior distribution of the parameters.

Figure 4.7 shows the estimated joint posterior distributions for c versus u (left panels), g versus u (middle panels), and g versus c (right panels). Clearly, the joint posteriors are very different for the three subjects. First, we see the relationship predicted above between c and u; to fit data with low discriminability, both c and u must be reduced. However, this reduction is much more pronounced for c than it is for u. This means that although the copying parameter is important, it only needs to be high enough to allow *some* copying. In contrast, c must be lowered substantially to reduce the probability of copying correctly.

In addition to the relationship between c and u, the feature distribution parameter g also plays an important role in predicting discriminability. Specifically, g must increase to help predict decreasing d's. This can be explained through the properties of the geometric distribution. The mean of the geometric distribution is $1/g$, so when g is increased, the geometric distribution decays more rapidly, sampling fewer unique values which are toward the lower support of the distribution (i.e. many ones and a few twos). When a match occurs, the calculation of the density of the geometric ($g(1-g)^{i-1}$) in Equation 2.4 will be smaller. Thus, the familiarity values will be smaller, resulting in fewer "old" responses for target items. For distractors, the features will have more in common with the traces in the episodic matrix resulting in more "new" responses and increasing the false alarm rates. These two changes (called the mirror effect) together produce a decrease in d'.

This section has stressed the importance of the joint posterior distributions in relating model parameters to model predictions. Specifically, I have demonstrated that all three parameters for REM play a pivotal role in predicting changes in discriminability. Equipped with a better understanding of the parameters in the REM model, I will fit the model to the data presented in Dennis et al. (2008) in the next section.

4.2.3 A Hierarchical REM Model

The Model

In the original formulation of REM (Shiffrin and Steyvers, 1997), features of items were assumed to have been generated by a word frequency parameter (i.e. g_H for high frequency word lists). However, a different "base" g was used during the comparison of a probe to the traces stored in memory. A common alternative to modeling these parameters separately is to instead set them to be equal (e.g. Criss and McClelland, 2006). Criss and McClelland (2006) call a model with the g stimuli generation parameter and the g used during the study phase as "fully informed." To fit REM to the data presented in Dennis et al. (2008), we will also make this assumption.

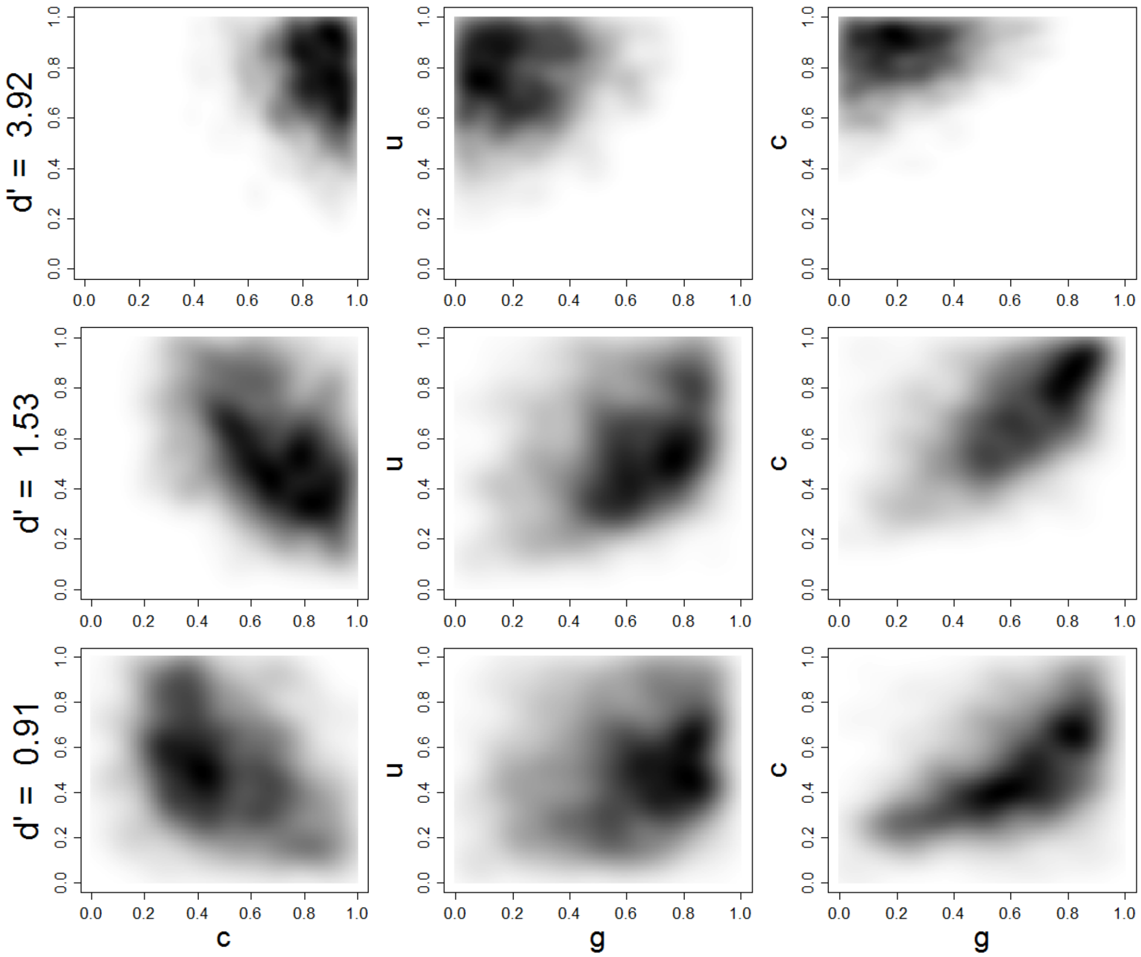

Figure 4.7: Approximate joint posterior distributions for u and c (left panel), u and g (middle panel), and c and g (right panel) for three different values of discriminability.

To model the effects of word frequency in these data using REM, we assume that encoding for high frequency words is a different parameter than for low frequency words. To do this, we use the parameter g to be used for low word frequency conditions, and γ to be used in the high word frequency conditions. We will again make use of the indicator variable W for high ($W = 1$) and low ($W = 0$) word frequency conditions.

To model the effects of filler, we assume that spurious traces are added to memory prior to the test phase. These extra traces distort global memory models by introducing interference. This happens because at test the observer compares a probe to all other traces stored in memory. If the probe has similar features to any of the other traces in memory, then this manifests in the form of increased familiarity. Here, we redefine the indicator variable F so that it is one in the filler conditions but zero in the no filler conditions. I introduce the new variable η to represent the number of spurious features that are added to memory.

For these data, we will use the same notation for the number of hits $Y_{i,j,HIT}$ and false alarms $Y_{i,j,FA}$. However this time, we will assume we are unable to perform any likelihood calculations (but see Montenegro et al., 2011). To identify the model, Montenegro et al. show that w must be fixed to some positive integer. By convention, we will fix $w = 20$ (e.g., Criss and McClelland, 2006, Shiffrin and Steyvers, 1997). 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 ith subject and jth condition, so

$$\theta_{i,j} = \{N_{STUDY} + \eta_i F_j, g_i(1 - W_j) + \gamma_i W_j, u_i, c_i, w\}.$$

Four of the parameters in this model have the common restriction of $g, \gamma, u, c \in [0, 1]$. For these parameters, we can again use the reparameterized beta distribution given in Equation 4.10. For these parameters, we assume common distributions for each of the subjects, given by

$$\begin{aligned} g_i &\sim \text{Beta}(\omega_g, \xi_g), \\ u_i &\sim \text{Beta}(\omega_u, \xi_u), \\ c_i &\sim \text{Beta}(\omega_c, \xi_c), \text{ and} \\ \gamma_i &\sim \text{Beta}(\omega_\gamma, \xi_\gamma). \end{aligned}$$

The parameter η has only positive infinite support, but may only take on discrete values. To model η, we chose the negative binomial distribution with hyperparameters denoted ω_η (the size parameter) and ξ_η (the probability parameter), or

$$\eta_i \sim \text{NegBin}(\omega_\eta, \xi_\eta).$$

We now specify a prior for each of the group-level parameters. Because REM has never been fit in a Bayesian framework, and empirical fits are either by hand or by using least-squares and Monte Carlo simulations (Criss and McClelland, 2006, Malmberg et al., 2003), we will assign noninformative priors to each of the group-level parameters. We will again group the hyper-means $\omega = \{\omega_g, \omega_u, \omega_c, \omega_\gamma\}$ and hyper scales $\xi = \{\xi_g, \xi_u, \xi_c, \xi_\gamma\}$ and use the same priors as before:

$$\begin{aligned} \omega &\sim \text{Beta}(1, 1), \text{ and} \\ \xi &\sim \Gamma(.1, .1). \end{aligned}$$

For ω_η and ξ_η, we assume

$$\begin{aligned}\omega_\eta &\sim \Gamma(.1, .1), \text{ and}\\ \xi_\eta &\sim \text{Beta}(1, 1).\end{aligned}$$

To estimate the posteriors, we again set $\rho(X, Y)$ equal to Equation 4.8 and ϵ equal to

$$\epsilon_t = \exp^{-.01t} + 0.10,$$

both of which were used in fitting BCDMEM to these data. I drew 10,000 samples with a burn-in period of 1,000 samples.

Results

As in the BCDMEM section, we will once again rely on the posterior predictive distributions to evaluate our model fits. The posterior predictive distributions were generated in the same way as in the BCDMEM section. Figure 4.9 shows the posterior predictive distribution for the data in the ROC space. We see that again the contours (representing the data) are covered by the predictive density, and there are few areas where the predictive density lies that the data do not. This indicates a close model fit.

The second posterior predictive density corresponds to the lower level parameters. This distribution is generated by the hyperparameters and indicates the range of possible future values for the five parameters. Highly variable distributions will indicate that summarizing subject performance with a single value may not be the best choice. Figure 4.9 shows that these densities are quite variable. The parameters g, γ, u and c cover a large portion of their parameter spaces. This suggests that

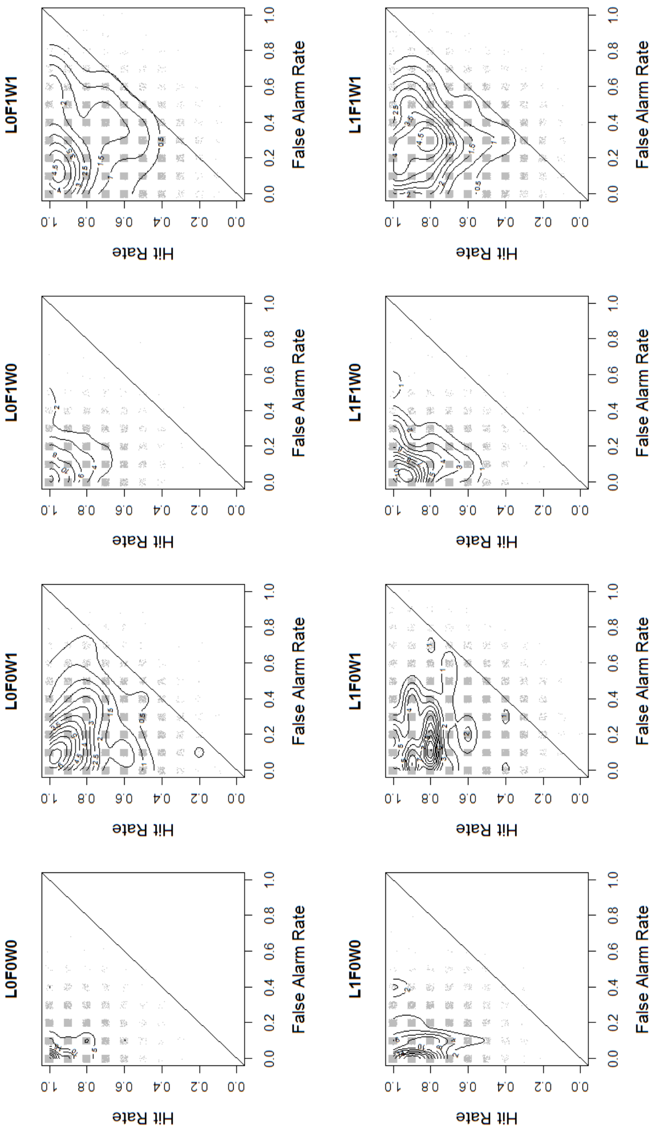

Figure 4.8: The posterior predictive distributions are represented by gray dots in the squares of possible values for the hit and false alarm rates in the experiment presented in Dennis et al. (2008). The observed data are shown by the black contours. Each condition is plotted with the condition code on the top of each panel: L corresponds to short (0) or long (1) conditions, F corresponds to filler (1) or no filler (0) conditions, and W corresponds to high (1) or low (0) word frequency conditions.

trying to determine a single “best” parameter value neglects information about the parameter values that is now available in the form of the posterior.

The distributions for the word frequency parameters g and γ are very different. Recall that for the geometric distribution, larger values for the rate parameter will cause the distribution to have fewer unique values. In REM, this translates to meaning that traces will have more features in common with increasing g. When there are more features in common, it becomes more difficult to discriminate targets from distractors, as was shown in the previous section. For g, we see that the density is covering lower probabilities, but for γ, the density is covering higher probabilities. This suggests that the subjects are sensitive to the word frequency manipulation, and that they are more discriminable for low frequency words than they are for high frequency words (e.g., Glanzer et al., 1993).

The effects of the filler task were modeled by adding spurious traces to the episodic memory matrix. The bottom middle panel shows the number of traces that need to be added in order for REM to account for future data. The mode of this distribution is zero, indicating that most subjects are not sensitive to the effects of filler conditions. However, notice that this density has long tails, extending out to around 60. This implies that some subjects are and will be very sensitive to the effects of filler tasks. Specifically, performance will decrease in the presence of filler tasks.

The densities of c and u are shown in the top middle and top right panels of Figure 4.9. These densities are centered toward high probabilities indicating high discriminability (see the above section). The joint distribution for c and u is plotted in the bottom right panel. This type of joint distribution was replicated in the above simulation study, and it implies that subjects are copying many features correctly, which ultimately leads to better discrimination.

Notice that for g, u, and c, there are large peaks toward the ends of the parameter space (zero or one). In the case of c and u, this implies perfect feature copying for a list of words, which will lead to perfect performance. Several subjects had perfect performance (the hit rates were equal to 1.0 and the false alarm rates were equal to 0), so the densities are correct in placing high probabilities in these regions. For g, the high density for low levels is also consistent with the data, because small values of g produce features that are uncommon. As shown in the previous simulation study, this in combination with medium-to-high values for c and u will imply very large "familiarity" scores, leading to high discriminability.

4.2.4 Summary

This section has provided clear interpretations for the parameters and their effects on model predictions. Specifically, I have shown that the list length effect in REM is generated exclusively through the number of items presented at study. That is, the list length effect is predicted entirely on the basis of the experimental design. This is an important feature in the model because it makes such a specific prediction for how performance should be affected by the number of items at study. Such a narrow prediction may lead to the demise or continued longevity of the REM model.

In addition to list length, I examined the effects of data with low, medium and high discriminability on the model parameters. I emphasized the importance of the joint posteriors in explaining this relationship. It was shown that all three parameters jointly affect the model predictions for discriminability.

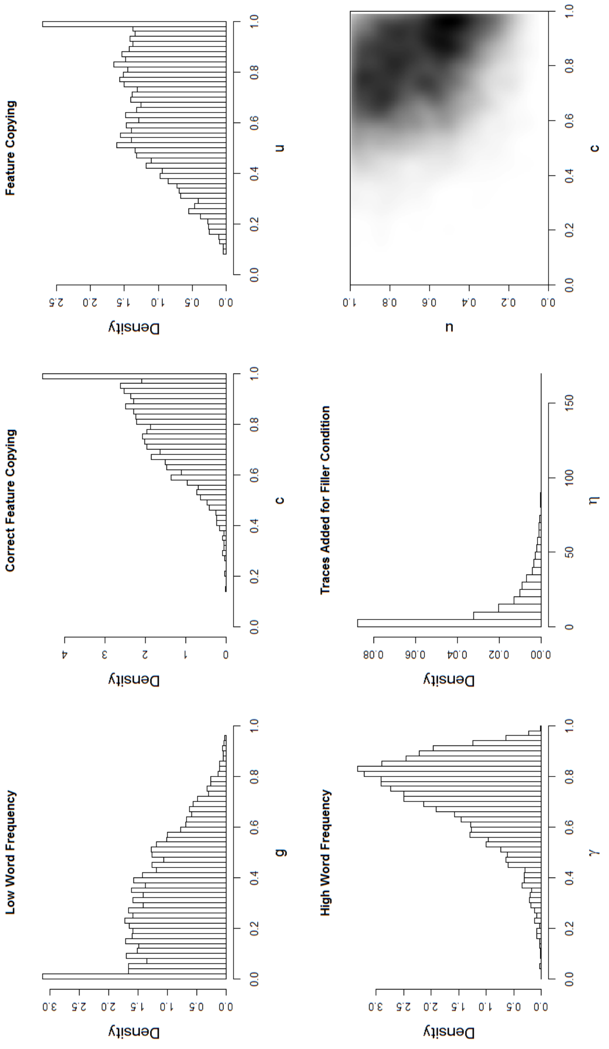

Figure 4.9: The posterior predictive distributions for each of the five lower-level parameters. These distributions represent the probability for each parameter value for any given subject. The left panels show the effects of high (bottom) and low (top) word frequency conditions, the middle panels show the effects of filler activity (bottom) and the probability of copying a feature correctly (top), and the right panel shows the probability of feature copying (top) and the joint distribution of c and u (bottom).

I then fit REM to the data presented in Dennis et al. (2008). Using ABC, I was able to provide good fits to the data, which were evaluated by inspection of the posterior predictive distributions. Furthermore, the distributions of the parameters were consistent with both the data and the new insight to the model parameters gained through the previous simulation studies.

4.3 Concluding Remarks

The importance of this section is two-fold. First, neither BCDMEM or REM have been fit to data in a Bayesian framework. In this chapter, both BCDMEM and REM were fit to data from a real recognition memory experiment, with a hierarchical structure. The second point is one of practicality. Although the work of Myung et al. (2007) provides us with a method for fitting BCDMEM to data, accurate estimates are only obtained through the exact integral expressions (see Equations 4.2 and 4.3), which I found difficult to evaluate precisely and consistently. By contrast, I showed that ABC can be used to obtain a suitable approximation for the posteriors in an illustrative example. I then used the mixture algorithm presented in Chapter 3 to fit both models to the data from Dennis et al. (2008). The models used to explain these data were somewhat complicated. They contained five free parameters with two hyperparameters for each of these lower-level parameters. In addition, the data contained 48 subjects in 8 conditions with 20 responses in each condition. The mixture algorithm reduces this complexity by localizing the parameter space by subject. This localization method was effective in combating the “curse of dimensionality” which drastically limits the applicability of the ABC approach.

Chapter 5: Mixture Modeling

When comparing competing models, there are many attributes we may consider. One model may be computationally simpler to implement than another, or perhaps one model is founded upon neurological principles and provides a presumably more plausible model. However, the traditional methods of model comparison involve two things: how well a model fits experimental data and the complexity of each model Myung and Pitt (1997).

Models assume that behavioral data are manifestations of a set of underlying mechanisms in the cognitive system. These mechanisms can be mimicked by a set of statistical or mathematical processes governed by a set of parameters, which comprise the model. It is crucial then that a model be capable of fitting experimental data, otherwise the processes assumed by that model are not representative of the underlying cognitive system.

A model's complexity can be measured by the interaction of a model's parameters and the functional form. Myung and Pitt (1997) show that one can increase the complexity of the model and account for larger variations in data. However, a delicate trade off exists between increasing a model's complexity and maintaining a parsimonious explanation of the data. Complex models are often not very generalizable; that is, they tend to over-fit the data. Thus, we seek mathematical models that are both simple and can account for the data.

There are several methods for model comparison. One such method, termed the Bayes factor, is the ratio of the posterior odds to the prior odds (e.g., Kass and Raftery, 1995, Liu and Aitkin, 2008). However, Dennis et al. (2008) criticize the Bayes factor because it assumes that one model is correct and the other is incorrect. They argue that when analyzing data sets with large numbers of subjects, if most of the subjects favor a simple model and only a few extreme subjects favor a more complex model, then the Bayes factor will choose the more complex model. This is because if one model is assumed to be true, it must explain the behavior of all subjects. By this logic, the more complex model is more correct than the simple model.

Another approach, used by Dennis et al. (2008) and Lee (2008) is mixture modeling. In contrast to the Bayes factor, the mixture modeling approach assumes that both models are useful, but one is more likely to be correct when explaining the behavior of many subjects. This approach penalizes complex models more heavily for their lack of generalizability.

Algorithms facilitating the mixture modeling approach can be difficult to implement. For example, one technique called reversible jump MCMC (Green, 1995, RJMCMC;) requires that one specifies a system of mapping functions connecting the parameters from one model to another. One then computes a Jacobian of these mapping functions which is then used to specify the probability of "jumping" from one model space to the other. Another technique called the product space method (Carlin and Chib, 1995, Lodewyckx et al.) is much simpler but relies on the specification of “pseudo” priors. These pseudo priors keep a model – which is not currently active in the Markov chain – available so that a probability of transitioning back to this model can be computed. As noted in Lodewyckx et al., misspecification of these pseudo priors can have drastic consequences on the sampling behavior of the chain. They suggest fitting each model in turn prior to performing the mixture modeling, and then using the obtained posteriors as the pseudo priors. While this a principled solution to the problem, it is obviously much more time consuming than fitting a single mixture model.

In contrast to many other transdimensional samplers, the ABC approach to mixture modeling is only a slight modification of the previously presented algorithms. The reason for this simplicity is because ABC does not require the likelihood function, which for mixture models can be difficult to specify. In this chapter, I will show how the ABC approach can be used to perform mixture modeling on the previously presented episodic memory models BCDMEM and REM. First, I will use a mixture model to distinguish between BCDMEM and REM in a simulation study. I then apply a mixture model to each subject’s data in Dennis et al. (2008) to determine the probability of data generation for each of the models.

5.1 A Mixture Model

Toni et al. (2009) provide an algorithm for performing model selection using ABC SMC (see Algorithm 6). The ABC SMC algorithm – presented in Chapter 2 – is similar to the ABC PMC algorithm, but is more general because it does not require a normal transition kernel to perturb particles from one iteration to the next.

Algorithm 6 extends the ABC SMC algorithm by parameterizing the model selection process and incorporating M models into a single analysis.

The basic idea of a mixture model is to introduce a new indicator variable z, which allows our sampler to switch between generating data from REM and generating data from BCDMEM. One may think of the proposal z^* as the result of a coin flip. If z^* lands on heads, then we simulate from REM, but if it lands on tails, we simulate from BCDMEM. Once we have decided which model to simulate from, we continue by sampling a proposal parameter $\theta(z^*)^*$ from the z^*th model using SMC techniques. We then generate data X from the z^*th model using the proposed $\theta(z^*)^*$ and compare the resulting data X to the observed data Y. A graphical diagram for this type of model is shown in Figure 5.1, where the observed data Y are separated and shown as "HIT" and "FA."

During the sampling, when, say, REM is chosen and it passes inspection, we record the proposed values for c, g, u and z. However, if say, BCDMEM is chosen and it passes inspection, then we record the proposed values of d, p, r and z. Because it is unlikely that the two models will fit equally well, recording the parameter values in this way usually results in distributions for the two sets of model parameters that are not of equal size. This can present a problem, especially when one model is heavily favored over another. Toni et al. (2009) suggest that one supplement each iteration by adding additional samples from the poorly-performing model. In contrast to model-specific parameters, the number of samples for z will always be the same. At the end of the simulation, we calculate the proportion of times REM fit the data successfully in the sequence. This proportion is an estimate of the probability that the observed data were generated by REM.

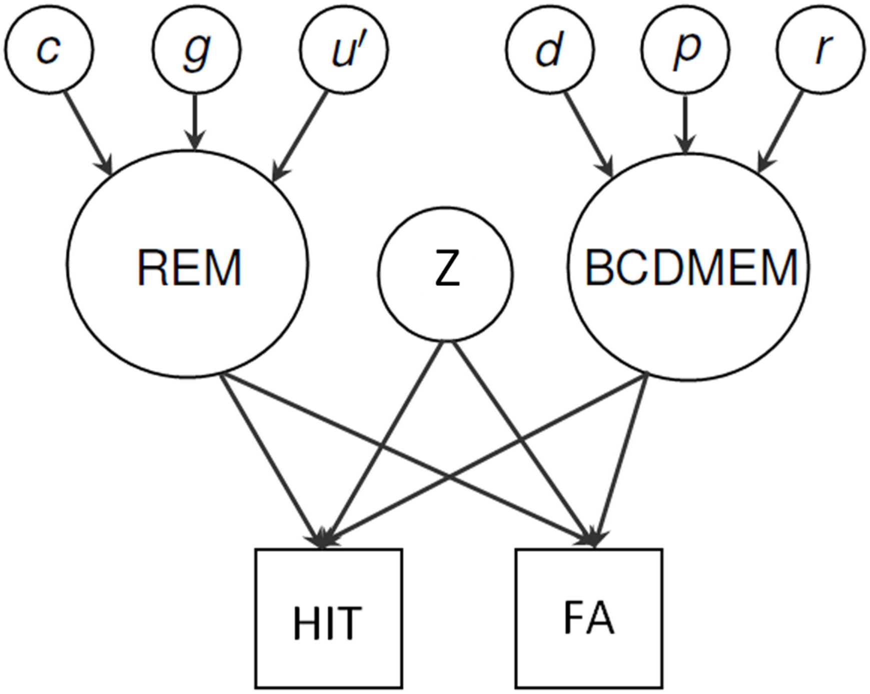

Figure 5.1: A graphical diagram for the mixture model used in the simulation study.

5.2 Simulation Study

In this section, I will generalize the model selection problem for the two models REM and BCDMEM. For any given data set from a recognition memory experiment, we could use Algorithm 6 to estimate the probability that REM (or BCDMEM) generated the data. This would answer our question for a particular data set (e.g., for a particular subject), but would not inform us about this probability for future data sets (e.g., different subjects). Each data set corresponds to a particular location in the ROC space, as discussed in Chapters 2 and 3. Thus, we would obtain the probability that REM generated the data at a particular location in the ROC space. This information is only important, however, for this subject and any other subject having the exact same hit and false alarm rates. To generalize the comparison of REM with BCDMEM, we could map out the entire area in the ROC space.

For continuous measures such as response time distributions, this approach would certainly be impractical. However, for recognition memory data, performance is summarized by a hit and false alarm rate. Hit rates are calculated by taking the total number of hits and dividing by the number of targets presented. Similarly, false alarms are calculated by taking the total number of false alarms and dividing by the number of distractors presented. These calculations result in a set of possible rates for hits and false alarms. For example, when the test list consists of 10 targets and 10 distractors, the set of all possible rates is $\{0, 1/10, 2/10, \ldots, 9/10, 1\}$. Thus, there are 11 possible hit and false alarm rates, producing 121 possible performances or locations in the ROC space.

When the false alarm rate is equal to the hit rate, a subject is performing at chance. When the hit rate is lower than the false alarm rate, the data are called "perverse" and are typically discarded. As a consequence, we are typically only interested in data with the hit rates greater than or equal to the false alarm rates, producing a triangle in the upper-left portion of the ROC space. Thus, in the example above, the number of interesting possible performances is reduced to 66.

Given the discrete nature of the performance data in the ROC space, it is possible to map out the entire region. To do so, I first generate a data set to correspond to each location in the space. I then fit the mixture model and estimate the probability that REM generated the data. Doing this for each location will produce a grid of estimates that might inform us as to whether there are systematic patterns in the model selection problem. To further generalize the results, I varied the length of the study list in four conditions: 10, 20, 40 and 80 items presented at study. Because the two models are in stark opposition to one another, this simulation study might provide insight about the differences between context and item noise models.

5.2.1 The Model

Figure 5.1 shows a graphical diagram for the model used in this simulation study. The model is very similar to the examples presented in Chapters 2 and 4. For REM, I will use the simple three-parameter version consisting of $\theta(1) = \{g, u, c\}$. For BCDMEM, I will also use the simple three-parameter version consisting of $\theta(2) = \{d, p, r\}$. Although these models have the same dimensionality, we are not restricted to this constraint when using Algorithm 6. More complex models having higher dimensionality are penalized by the ABC SMC algorithm through rejection sampling. That is, it becomes increasingly more difficult to sample a candidate parameter vector from the joint posterior distribution with increases in the number of dimensions.

For this model, I will again negate the influence of the prior distribution on the posteriors by specifying a noninformative prior for each of the three parameters in REM and BCDMEM:

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

For each simulation of REM, I first generated a new set of items each consisting of $w = 20$ features sampled from a geometric distribution with rate parameter equal to the proposed value g^* for g. The new set consisted of 500 new items, from which I sampled the study list set consisting of 10, 20, 40 or 80 items. At test, the same proposed parameter value g^* was used in the calculation of the overall familiarity given by Equation 2.4. For BCDMEM, the vector length parameter v was set to 200 and the study context sparsity parameter s was set to 0.02.

For the model selection parameter z, I specified equal probabilities for each model with the discrete uniform distribution,

$$z \sim \text{DU}[1, 2].$$

Other priors may be more appropriate, such as priors geared toward specific predictions. For example, if a list length effect is evident in the data, increasing the prior probability for REM might be more appropriate because REM naturally predicts the list length effect whereas BCDMEM predicts a null list length effect (see Chapter 4). However, in this simulation study, there was no reason to favor one model over the other.

5.2.2 Results

For each of the study list length conditions, I generated 496 data sets, each of which represented a unique location in the ROC space as described above. Each data set consisted of 20 responses from a single subject to a test list of 10 targets and 10 distractors in three conditions. I assumed that the parameters used by the subject were the same across conditions, and the responses were pooled across the three conditions to obtain a single hit and false alarm rate. This method stabilized the variance of the predictions for the two models and provided a fine grid across the ROC space, ranging from 0 to 1 in increments of 1/30 in both directions. The 10 targets were the first 10 items presented at study, followed by the 10 distractors which were randomly generated in the same way study items were for each model. Algorithm 6 was used to fit each data set. The discriminant function $\rho(X, Y)$ was set equal to Equation 4.8 and ϵ was set to $\{0.20, 0.15, 0.13, 0.10, 0.08, 0.04, 0.02, 0\}$. To estimate the posteriors, I used 1,000 particles. I specified a truncated normal transition kernel bounded by 0 and 1 with a mean of the previously accepted value for each parameter, and standard deviation of 0.01.

When one model is heavily favored over another, the vector of accepted particles for z can sometimes consist of all ones or zeros, effectively eliminating one of the models from consideration. This happens because there are too few particles to estimate the posterior distribution for the poorly-performing model suitably. Toni et al. (2009) suggest generating samples to estimate the posterior for the poorly-performing model so that a larger pool can be used in subsequent iterations. I used this strategy by monitoring the number of particles in the pool for each model. If the number of particles went below 100 for either model, I added 200 particles to the pool and calculated the weights as in Algorithm 6. Thus, when sampling particles in the next iteration, the particles would be representative of the joint posterior distribution, giving the losing model a chance to continue competing. In addition to this precaution, Algorithm 6 specifies that the z^*s be sampled from the prior and not the previous pool. This means that each model will be simulated with the same probability as specified by the prior regardless of how well (or poorly) the model is doing.

Figure 5.3 shows the results of the simulation. Each plot shows the probability that REM generated the data for each study list length condition: 10 (top left panel), 20 (top right panel), 40 (bottom left panel) and 80 (bottom right panel) items. Regions between possible values of the hit and false alarm rates have been interpolated to provide a smooth overlay. The key, shown in the far right margin shows that as the region becomes more white, REM is more heavily favored. By contrast, as the region becomes more black, BCDMEM is more heavily favored. There is a general tendency for BCDMEM to cover the top right and bottom left portions of the ROC space and for REM to cover the middle portion. This indicates that BCDMEM is better able to capture biased responding than REM.

As the number of items presented at study increases, the figure becomes more black. This indicates that BCDMEM becomes more capable of fitting larger ranges of data. In Chapter 4, the predictions of REM and BCDMEM as a function of the number of items presented at study was examined. In this simulation, I showed that as the number of study items presented increased, the predictions of REM changed but the predictions of BCDMEM remained the same. Specifically, REM predicted worse performance (i.e. d' decreased), creating the list length effect. Given that

REM's predictions change as a function of the experimental design, REM is implicitly constrained by the number of items presented at study. As an example, consider the final study list length condition ($N = 2000$) in the simulation study presented in Chapter 4. Figure 4.5 shows that the predictions become more concentrated on unbiased chance behavior, near the point (0.5, 0.5).

In the context of this model selection simulation, the behavior of REM can be characterized as increased model complexity. As the number of items presented at study increases, REM is less capable of predicting variations in the data. By contrast, BCDMEM's complexity remains constant, and can predict variations in the data consistently. Thus, as the number of study items are increased, BCDMEM gains a "foothold" and becomes more heavily favored for more data.

Although this argument seems to favor BCDMEM, it only explains the behavior of the two models, not the data. If the data behave, say, in the same manner as REM, then REM provides the better explanation of the data. In the next section, I will apply the mixture model to the data from Dennis et al. (2008).

5.3 Fitting Experimental Data

The data presented in Dennis et al. (2008) were described in Chapter 4, where I also fit the data using hierarchical versions of BCDMEM and REM. The data will again be examined here, but I will now fit the data for each subject in turn. Hierarchical versions of the mixture model presented below could be used to answer general questions of which model fit an entire set of data better. However, I was more concerned with answering the model selection question for each particular subject.

5.3.1 The Model

The model will be an amalgam of the hierarchical models presented in Chapter 4. To apply Algorithm 6 to this data, I define two $\theta_{i,j}$s to distinguish a simulation of BCDMEM ($z = 1$) from a simulation of REM ($z = 2$), so

$$\begin{aligned} \theta(1)_{i,j} &:= \{d_i(1-F_j) + \delta_i F_j, p_i(1-W_j) + \tau_i W_j, r_i, s, v\} \text{ and} \\ \theta(2)_{i,j} &:= \{N_{STUDY} + \eta_i F_j, g_i(1-W_j) + \gamma_i W_j, u_i, c_i, w\}. \end{aligned}$$

I use priors similar to Chapter 4, but fix the hyperparameters since I am only interested in performing inference on each subject in turn. For BCDMEM, I set

$$d_i, \delta_i, p_i, r_i, \tau_i \sim \text{Beta}(1,1),$$

and for REM I set

$$\begin{aligned} g_i, u_i, c_i, \gamma_i &\sim \text{Beta}(1,1), \text{ and} \\ \eta_i &\sim \text{NegBin}(10, 0.1). \end{aligned}$$

For BCDMEM, I again set $v = 200$ and $s = 0.02$ and for REM I set $w = 20$. The only new parameter in this model is the selection parameter z, and I again place a noninformative prior on it, so

$$z \sim DU[1,2].$$

From the above equations, we can see that while the two models have the same number of parameters, their dimensionality is different. Excluding η, all of the parameters are bounded by $[0,1]$. However, for REM the parameter η is allowed to vary between $[0, \infty)$. Taking only this into consideration, REM is the more flexible model for these data. As mentioned in the previous section, the complexity of the parameter space is easily dealt with by the ABC approach, because it does not require the calculation of a likelihood, as in standard Bayesian approaches to mixture modeling such as transdimensional MCMC (e.g., Carlin and Chib, 1995, Green, 1995, Lodewyckx et al.).

5.3.2 Results

To fit the data, Algorithm 6 was used to estimate the posterior distributions for each subject. I used 1,000 particles to estimate the posteriors. For each of the above parameters excluding η_i, I used a truncated normal transition kernel bounded by 0 and 1 with mean equal to the previously accepted value for each respective parameter and standard deviation of 0.05. For η_i, I used a negative binomial transition kernel with mean equal to the previously accepted value and size or dispersion parameter equal to 1,000.

The same rule was used as in the simulation study to avoid depletion of the pools for the particles for each model. I again used the discriminant function given by Equation 4.8. To fit the data for each subject, I required different tolerance conditions ϵ. This is because some subjects are easily fit by one model or the other, but some subjects were very difficult to fit by either model.

The degree of closeness becomes an important issue when using the ABC SMC algorithm. Ideally, we would like to converge to a perfect fit to the data ($\epsilon = 0$). However, these data have several responses across eight conditions, making a perfect match to the data very difficult. Thus, it is important to develop a stopping rule for data of this type.

To develop such a rule, I examine the distribution of the statistic $\rho(X, Y)$ for all *accepted* particles at iteration t. These distributions eventually converge to exponential distributions and will have long tails when the data are easy to fit, or short tails when the data are difficult to fit. The rule is to decrease ϵ at a consist rate until eventually arriving at some specified variance for the distribution of $\rho(X, Y)$ for the accepted particles. For the fits of these data, I chose a final variance value of 0.0065. The final distribution of $\rho(X, Y)$ for each subject is shown in Figure 5.4. The figure shows that the tails of the distribution of $\rho(X, Y)$ are quite short. These short tails suggest that I have arrived at the best-fitting posterior distributions for the models.

In summary, there are several precautions in place to ensure the accuracy of the estimated posterior. First, each model is subjected to identical tolerance thresholds ϵ. Thus, each model undergoes the same rigorous path to convergence until it becomes too difficult to fit the observed data. Second, convergence is not acquired until the distribution of $\rho(X, Y)$ for accepted particles reaches some small degree of variance. Requiring convergence on this basis ensures that the best possible fits to the data have been determined. Third, at each iteration, I sample from the prior distribution for z. In this particular case, each model is sampled with equal probability, meaning that even when a model is performing poorly, it is not removed from consideration. Finally, the estimated posterior distribution of z is monitored after each iteration. If the estimate for the probability that a model generated the observed data goes below 0.10 for either model, a separate simulation is performed to ensure accurate estimates of the joint posterior distribution for the parameters of the poorly-performing model. To do so, I sample an additional 200 particles and evaluate their weights. On the next iteration, I sample from a new pool consisting of the particles from the additional simulation and any accepted particles from the previous iteration. With these precautions in place, we can be confident in the accuracy of the estimated posterior distribution.

Figure 5.5 shows the mean of the estimated posterior distribution for the model selection parameter z for each subject. The dashed horizontal line represents the cutoff point for determining which model fit best. That is, when the mean of the posterior distribution is greater than 0.5, the data are more likely to have been generated by REM. By contrast, when the mean of the posterior distribution is less than 0.5, the data are more likely to have been generated by BCDMEM. For these data, 17 of the 48 subjects were fit better by BCDMEM (35%). Although this seems like a large victory for REM, Figure 5.5 shows that when subjects were fit better by BCDMEM, the win was much more compelling than when the subjects were fit better by REM. Despite this, the mean of all 48 estimated posterior distributions across subjects was 0.55, indicating a win for REM.

5.4 Conclusions

This chapter has shown that the ABC approach can be applied to the model selection problem. First, I used Algorithm 6 to map out the model selection surface for four study list length conditions. For each study list length condition, I fit the mixture model to 496 points in the ROC space to determine which model would fit the data best at that location. Comparing these surfaces across list length conditions, Figure 5.3 shows that as the length of the study list increases, BCDMEM becomes the favored model. I concluded that these surfaces change because the complexity of REM is implicitly constrained by the experimental design (list length in this chapter). Specifically, as the number of items presented at study increase, REM's predictions become more concentrated at a particular area, making it less capable of fitting biased and high-accuracy data.

While the simulation study provided good insight to the model selection problem for the two models, it is inconclusive without experimental data. Thus, I applied a mixture model to the data of Dennis et al. (2008). To fit this data, I fit each subject independently. For this data, I found that 31 of the 48 subjects were fit better by REM, suggesting that although REM's specific predictions about how performance is affected by the number of items presented at study, are consistent with the data.

```
At iteration $t = 1$,
for $1 \leq i \leq N$ do
    while $\rho(X, Y) > \epsilon_1$ do
        Sample $z^*$ from $\pi(z)$
        Sample $\theta^*$ from the prior, $\theta^* \sim \pi(\theta(z^*))$
        Generate data $X$ using the model, $X \sim \text{Model}(\theta^*(z^*))$
        Calculate $\rho(X, Y)$
    end while
    Set $z_{i,1} \leftarrow z^*$ ,$\theta_{i,1} \leftarrow \theta^*$, and $w_{i,1} \leftarrow 1/N$
end for
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 $z^*$ from $\pi(z)$
            Sample $\theta^* \sim \theta(z^*)_{1:N,t-1}$ with probabilities $w_{1:N,t-1}$
            Perturb $\theta^*$ by creating $\theta^{**} \sim K(\theta|\theta^*)$
            Generate data $X$ using the model, $X \sim \text{Model}(\theta^{**}(z^*))$
            Calculate $\rho(X, Y)$
        end while
        Set $z_{i,t} \leftarrow z^*$, $\theta_{i,t} \leftarrow \theta^{**}$ and $w_{i,t} \leftarrow \frac{\pi(\theta^{**})}{\sum_{j=1}^{N} w_{j,t-1} q\left(\theta_{j,t-1}|\theta^{**}\right)}$
    end for
end for
```

Figure 5.2: ABC SMC algorithm for performing model selection.

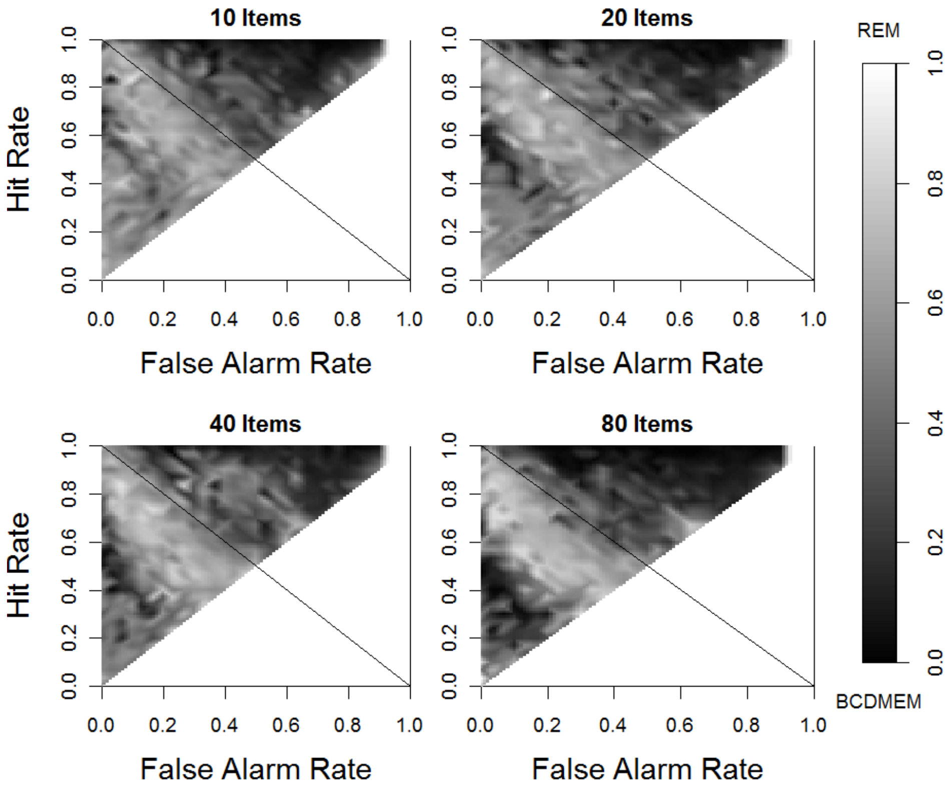

Figure 5.3: The probability of data generation by REM across the ROC space for four study list length conditions: 10 (top left panel), 20 (top right panel), 40 (bottom left panel) and 80 (bottom right panel) items.

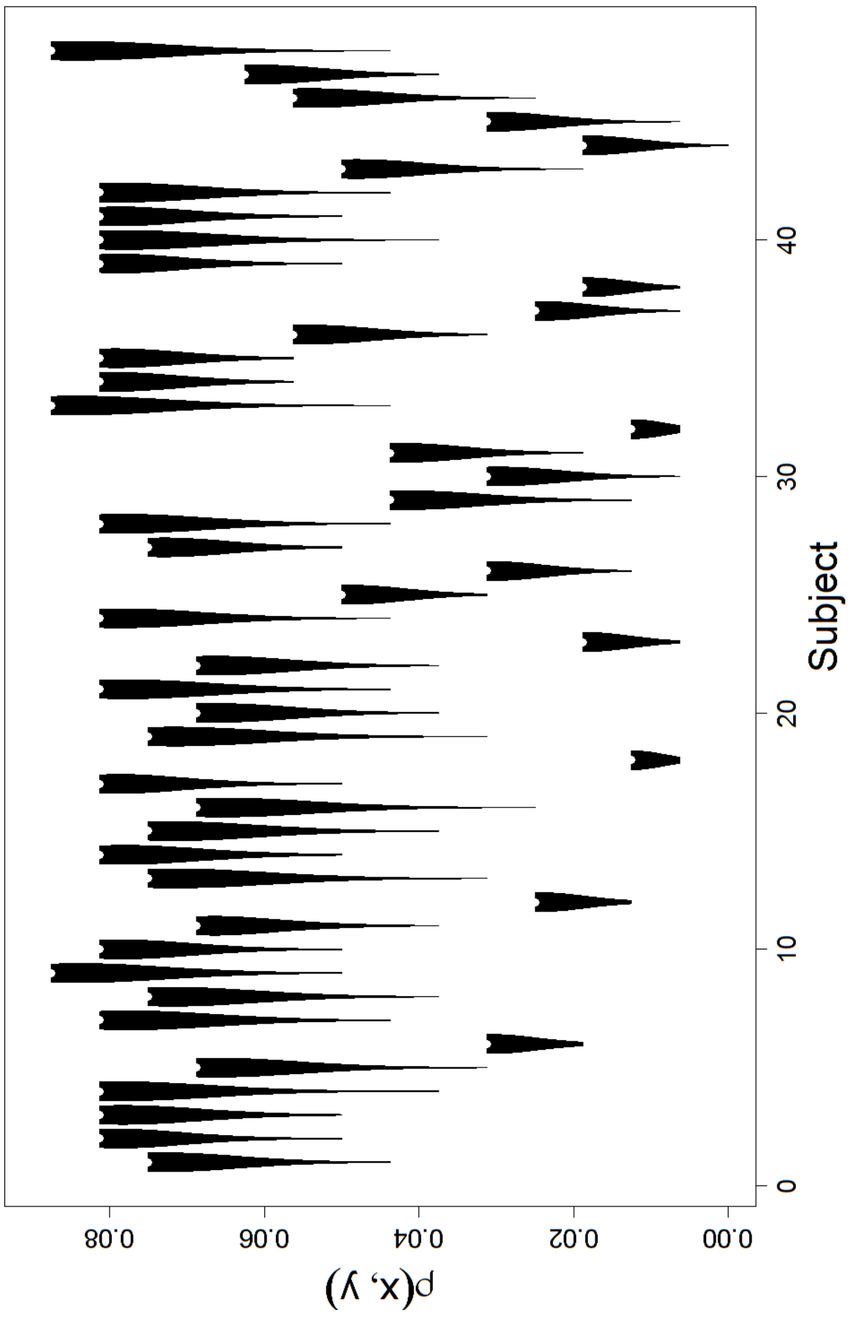

Figure 5.4: The distribution of $\rho(X, Y)$ for accepted particles on the final iteration, for each subject.

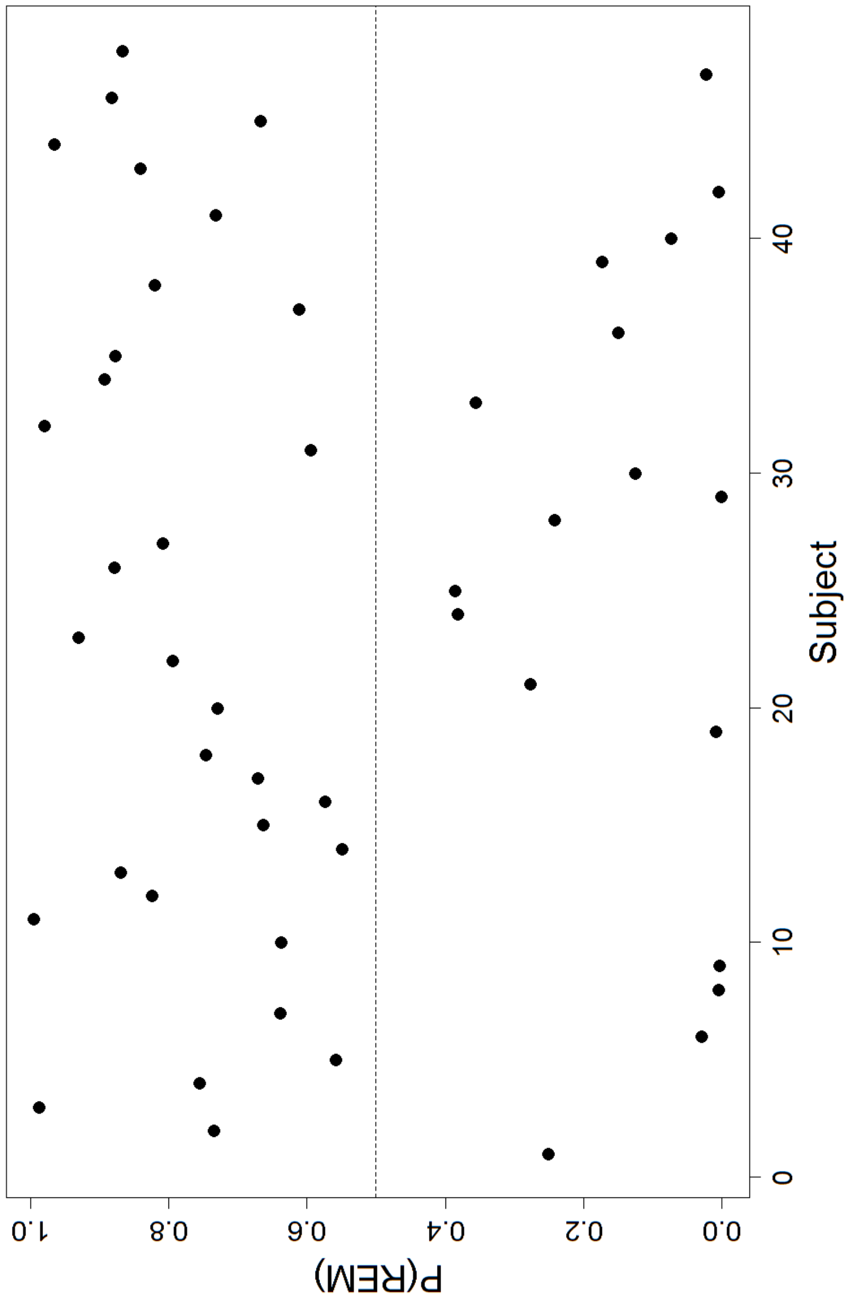

Figure 5.5: The mean of the estimated posterior distribution for z for each subject. Subjects with estimates greater than 0.5 favor REM whereas estimates less than 0.5 favor BCDMEM.

Chapter 6: Conclusions

In this dissertation, I have investigated the use of a new method for posterior estimation, called approximate Bayesian computation (ABC), that does not require the evaluation of the likelihood function. This technique has the potential for widespread use in psychology, where sophisticated, simulation-based models are often required to explain complex behavior.

In Chapter 2, I presented a manuscript, submitted to the Journal of Mathematical Psychology, that serves as a tutorial for ABC methods. In this tutorial, I contrasted several popular algorithms for performing ABC estimation. I then applied the ABC PMC algorithm to several toy problems. This chapter was useful in introducing the ABC technique in an accessible way, and in demonstrating that ABC can recover the true posterior distribution for a few examples.

In the next Chapter, I extended the ABC technique for estimation of the posterior distribution of the parameters for hierarchical designs. I first summarize the most popular algorithms and explained where these algorithms may suffer. I then introduced a new mixture algorithm that combines standard Bayesian sampling techniques with ABC. This algorithm is accurate, fast, and highly flexible. First, because we do not require ABC for the hyperparameters, we can use well-accepted Bayesian techniques to accurately sample from the posterior distribution, which is not possible in the two-stage algorithm. Second, because we are only using ABC for the individual-level parameters, the accept/reject nature of the ABC approach does not imped the speed of the sampler as it does in the naïve approach, presented in Algorithm 3. I then suggested three methods for optimizing the mixture algorithm and applied these methods to a hierarchical Wald model for response times. In Chapter 4, I turned from simple problems to more complicated psychological models of episodic memory. First, I demonstrated that ABC could accurately recover the posterior distributions of the model BCDMEM by comparing estimates obtained using ABC to estimates obtained using the "true" posterior distributions (based on equations presented in Myung et al., 2007). Once convinced that ABC was suitable for BCDMEM, I then applied the mixture algorithm to a hierarchical version of BCDMEM, which was fit to the data presented in Dennis et al. (2008). I then turned to the model REM, which does not have a likelihood function, and showed that ABC can be used to answer several questions about how the parameters in REM influence its predictions. I then fit a hierarchical version of REM to the data of Dennis et al. (2008).

In the final chapter, I showed how ABC can be used to estimate the posterior distribution of the parameters in a mixture model. The standard Bayesian approaches to posterior estimation of mixture models are quite difficult to implement. However, using ABC, we require only a slight modification of the algorithms presented in Chapter 2. I applied a mixture model to a simulation study to compare the models REM and BCDMEM. In this study, I varied the number of items presented at study from 10, 20, 40 and 80. The results of the simulation showed systematic patterns in the model selection space, which I attributed to the implicit constraint on global memory models. I argued that in addition to the classic model selection considerations presented in Myung and Pitt (1997), one must also consider the experimental design. I then fit a mixture model to the data of Dennis et al. (2008) and concluded that most of the subjects were fit better by REM than BCDMEM.

The wide array of ABC techniques presented in this dissertation is encouraging for Bayesian modelers. Now, equipped with a technique that does not require the full derivation of the likelihood function, we can estimate the posterior distribution for the parameters of *any* model that can be simulated. While the ABC approach is currently time consuming, I feel that with the development of more sophisticated algorithms and more powerful computers, this computational burden will become negligible in the near future.

Bibliography

Robert P. Abelson. *Statistics as Principled Argument.* Lawrence Erlbaum Associates, Hillsdale, NJ, 1995.

Eric Bazin, Kevin J. Dawson, and Mark A. Beaumont. Likelihood-free inference of population structure and local adaptation in a Bayesian hierarchical model. *Genetics*, 185:587–602, 2010.

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

Mark A. Beaumont, W. Zhang, and D. J. Balding. Approximate Bayesian computation in population genetics. *Genetics*, 162:2025–2035, 2002.

Mark A. Beaumont, Jean-Marie Cornuet, Jean-Michel Marin, and Christian P. Robert. Adaptive approximate Bayesian computation. *Biometrika*, asp052:1–8, 2009.

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

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

N. L. Bowles and M. Glanzer. An analysis of interference in recognition memory. *Memory and Cognition*, 11:307–315, 1983.

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

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

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

O. Cappé, A. Guillin, J. M. Marin, and C. P. Robert. Population monte carlo. *Journal of Computational and Graphical Statistics*, 13:907–929, 2004.

B. P. Carlin and S. Chib. Bayesian model choice via Markov chain Monte Carlo. *Journal of the Royal Statistical Society*, 57:473–484, 1995.

Nicholas L. Carnagey and Craig A. Anderson. Violent video game exposure and aggression: A literature review. *Minerva Psichiatrica*, 45:1–18, 2004.

Raj S. Chhikara and Leroy Folks. *The inverse Gaussian distribution: Theory methodology and applications*. Marcel Dekker, Inc., New York, NY, 1989.

Ronald Christensen, Wesley Johnson, Adam Branscum, and Timothy E. Hanson. *Bayesian Ideas and Data Analysis: An Introduction for Scientists and Statisticians*. CRC Press, Taylor and Francis Group, Boca Ranton, FL, 2011.

Peter Craigmile, Mario Peruggia, and Trisha Van Zandt. Hierarchical Bayes models for response time data. *Psychometrika*, 75:613–632, 2010.

Amy H. Criss and James L. McClelland. 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, 2006.

Pierre Del Moral, Arnaud Doucet, and Ajay Jasra. Sequential monte carlo samplers. *Journal of the Royal Statistical Society B*, 68:411–432, 2006.

Simon Dennis and Michael S. Humphreys. A context noise model of episodic word recognition. *Psychological Review*, 108:452–478, 2001.

Simon Dennis, Michael Lee, and A. Kinnell. Bayesian analysis of recognition memory: The case of the list-length effect. *Journal of Mathematical Psychology*, 59:361–376, 2008.

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

Bradley Efron. Why isn't everyone a Bayesian? *The American Statistician*, 40:1–11, 1986.

James P. Egan. Recognition memory and the operating characteristic. Technical Report AFCRC-TN-58-51, Hearing and Communication Laboratory, Indiana University, Bloomington, Indiana, 1958.

L. Excoffer, A. Estoup, and J.-M. Cornuet. Bayesian analysis of an admixture model with mutations and arbitrarily linked markers. *Genetics*, 169:1727–1738, 2005.

N. J. R. Fagundes, N. Ray, M. Beaumont, S. Neuenschwander, F. M. Salzano, S. L. Bonatto, and L. Excoffier. Statistical evaluation of alternative models of human evolution. *Proceedings of the National Academy of Science*, 104:17614–17619, 2007.

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

Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall, New York, NY, 2004.

W. R. Gilks and P. Wild. Adaptive rejection sampling for Gibbs sampling. *Applied Statistics*, 41:337–348, 1992.

G. Gillund and R. M. Shiffrin. A retrieval model for both recognition and recall. *Psychological Review*, 91:1–67, 1984.

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

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

P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika*, 82:711–732, 1995.

Andrew Heathcote. Fitting Wald and ex-Wald distributions to response time data: An example using functions for the S-PLUS package. *Behavioral Research Methods, Instruments, & Computers*, 36:678–694, 2004.

Andrew Heathcote, Scott D. Brown, and D. J. K. Mewhort. The power law repealed: The case for an exponential law of practice. *Psychonomic Bulletin and Review*, 7:185–207, 2000.

M. J. Hickerson and C. Meyer. Testing comparative phylogeographic models of marine vicariance and dispersal using a hierarchical Bayesian approach. *BMC Evolutionary Biology*, 8:322, 2008.

M. J. Hickerson, E. A. Stahl, and H. A. Lessios. Test for simultaneous divergence using approximate Bayesian computation. *Evolution*, 60:2435–2453, 2006.

Helena Kadlec. Statistical properties of d' and beta estimates of signal detection theory. *Psychological Methods*, 4:22–43, 1999.

R. E. Kass and A. E. Raftery. Bayes factors. *Journal of the American Statistical Society*, 90:773–795, 1995.

S. Kullback, J. C. Keegel, and J. H. Kullback. *Topics in statistical information theory (Lecture Notes in Statistics, Vol. 42)*. Springer-Verlag, New York, 1987.

Michael D. Lee. A Bayesian analysis of retention functions. *Journal of Mathematical Psychology*, 48:310–321, 2004.

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

Michael D. Lee and Eric-Jan Wagenmakers. A course in Bayesian graphical modeling for cognitive science. Available from http://users.fmg.uva.nl/ewagenmakers/BayesCourse/BayesBook.pdf; last downloaded February 26, 2010., 2010.

Michael D. Lee, I. G. Fuss, and D. J. Navarro. A Bayesian approach to diffusion models of decision-making and response time. In B. Scholkopf, J.C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing*, pages 809–815. MIT Press, Cambridge, MA, 19 edition, 2006.

Charles C. Liu and Murray Aitkin. Bayes factors: Prior sensitivity and model generalizability. *Journal of Mathematical Psychology*, 52:362–375, 2008.

T. Lodewyckx, W. Kim, F. Tuerlinckx, P. Kuppens, M. D. Lee, and E.-J. Wagenmakers.

R. D. Luce. *Response times: Their role in inferring elementary mental organization.* New York: Oxford University Press, 1986.

K. J. Malmberg, R. Zeelenberg, and R.M. Shiffrin. Modeling Midazolam's effect on the hippocampus and recognition memory. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15, pages 75–82. MIT Press, Cambridge, MA, 2003.

K. J. Malmberg, R. Zeelenberg, and R.M. Shiffrin. 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, 2004.

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

Dora Matzke and Eric-Jan Wagenmakers. Psychological interpretation of the ex-gaussian and shifted wald parameters: A diffusion model analysis. *Psychonomic Bulletin and Review*, 16:798–817, 2009.

J. McClelland and M. Chappell. Familiarity breeds differentiation: A subjective-likelihood approach to the effects of experience in recognition memory. *Psychological Review*, 105:724–760, 1998.

Maximiliano Montenegro, Jay I. Myung, and Mark A. Pitt. REM integral expressions. Unpublished manuscript., 2011.

I. J. Myung and M. A. Pitt. Applying Occam's razor in modeling cognition: A Bayesian approach. *Psychonomic Bulletin and Review*, 4:79–95, 1997.

In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of Mathematical Psychology*, 47:90–100, 2003.

Jay I. Myung, Maximiliano Montenegro, and Mark A. Pitt. Analytic expressions for the BCDMEM model of recognition memory. *Journal of Mathematical Psychology*, 51:198–204, 2007.

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

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

Robert M. Nosofsky, Daniel R. Little, Christopher Donkin, and Mario Fific. Short-term memory scanning viewed as exemplar-based categorization. *Psychological Review*, 118:280–315, 2011.

Zita Oravecz, Francis Tuerlinckx, and Joachim Vandekerckhove. A hierarchical Ornstein-Uhlenbeck model for continuous repeated measurement data. *Psychometrika*, 74:395–418, 2009.

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

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

R.C. O'Reilly and Y. Munakata, editors. *Computational Explorations in Cognitive Neuroscience: Understanding the Mind by Simulating the Brain*. MIT Press, Cambridge, MA, 2000.

Mario Peruggia, T. Van Zandt, and Meng Chen. Was it a car or a cat i saw? An analysis of response times for word recognition. *Case Studies in Bayesian Statistics*, VI:319–334, 2002.

Martyn Plummer, Nicky Best, Kate Cowles, and Karen Vines. CODA: Convergence diagnosis and output analysis for MCMC. *R News*, 6(1):7–11, March 2006. URL `http://CRAN.R-project.org/doc/Rnews/`.

Jonathan K. Pritchard, Mark T. Seielstad, Anna Perez-Lezaun, and Marcus W. Feldman. Population growth of human Y chromosomes: A study of Y chromosome microsatellites. *Molecular Biology and Evolution*, 16:1791–1798, 1999.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL `http://www.R-project.org`. ISBN 3-900051-07-0.

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

R. Ratcliff and F. Tuerlinckx. Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction time and parameter variability. *Psychonomic Bulletin and Review*, 9:438–481, 2002.

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

R. Ratcliff, G. McKoon, and M. H. Tindall. 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, 1994.

R. Ratcliff, A. Thapar, and G. McKoon. A diffusion model analysis of the effects of aging on brightness discrimination. *Perception and Psychophysics*, 65:523–535, 2003.

R. Ratcliff, A. Thapar, P. Gomez, and G. McKoon. A diffusion model analysis of the effects of aging in the lexical-decision task. *Psychology and Aging*, 19:278–289, 2004.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer, New York, NY, 2004.

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

Jeffrey N. Rouder and Paul L. Speckman. An evaluation of the Vincentizing method of forming group-level response time distributions. *Psychonomic Bulletin and Review*, 11:419–427, 2004.

Jeffrey N. Rouder, Jun Lu, Paul Speckman, Dongchu Sun, and Yi Jiang. A hierarchical model for estimating response time distributions. *Psychonomic Bulletin and Review*, 12:195–223, 2005.

J.N. Rouder, D. Sun, P.L. Speckman, J. Lu, and D. Zhou. A hierarchical Bayesian statistical framework for response time distributions. *Psychometrika*, 68:589–606, 2003.

David C. Rubin and Amy E. Wenzel. One hundred years of forgetting: A quantitative description of retention. *Psychological Review*, 4:734–760, 1996.

W. Schwarz. The ex-Wald distribution as a descriptive model of response times. *Behavioral Research Methods, Instruments, & Computers*, 33:457–469, 2001.

R. M. Shiffrin, M. D. Lee, W. Kim, and E.-J. Wagenmakers. A survey of model evaluation approaches with a tutorial on hierarchical Bayesian methods. *Cognitive Science*, 32:1248–1284, 2008.

Richard M. Shiffrin and Mark Steyvers. A model for recognition memory: REM – retrieving effectively from memory. *Psychonomic Bulletin and Review*, 4:145–166, 1997.

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

S.A. Sisson, Y. Fan, and Mark M. Tanaka. Sequential Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences of the United States*, 104:1760–1765, 2007.

V. C. Sousa, M. Fritz, M. A. Beaumont, and L. Chikhi. Approximate Bayeisian computation without summary statistics: the case of admixture. *Genetics*, 181: 1507–1519, 2009.

Tina Toni, David Welch, Natalja Strelkowa, Andreas Ipsen, and Michael P.H. Stumpf. Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems. *Journal of the Royal Society Interface*, 6: 187–202, 2009.

Francis Tuerlinckx. The efficient computation of the cumulative distribution and probability density functions in the diffusion model. *Behavior Research Methods, Instruments, & Computers*, 36:702–716, 2004.

Brandon M. Turner and Trisha Van Zandt. A tutorial on approximate Bayesian computation. In progress, a.

Brandon M. Turner and Trisha Van Zandt. Hierarchical approximate Bayesian computation. In progress, b.

Brandon M. Turner, Simon Dennis, and Trisha Van Zandt. Bayesian analysis of memory models. Manuscript in preparation., 2011a.

Brandon M. Turner, Trisha Van Zandt, and Mario Peruggia. An application of hierarchical approximate bayesian computation to the diffusion model. Manuscript in preparation., 2011b.

M. Usher and J. L. McClelland. On the time course of pereptual choice: The leaky competing accumulator model. *Psychological Review*, 108:550–592, 2001.

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

J. Vandekerckhove, F. Tuerlinckx, and Michael D. Lee. Hierarchical diffusion models for two-choice response time. *Psychological Methods*, 16:44–62, 2011.

E.-J. Wagenmakers. A practical solution to the pervasive problems of p values. *Psychonomic Bulletin and Review*, 14:779–804, 2007.

E.-J. Wagenmakers, H. J. L. van der Maas, and R. P. P. P. Grasman. An EZ-diffusion model for response time and accuracy. *Psychonomic Bulletin and Review*, 14:3–22, 2007.

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

J. T. Wixted. Analyzing the empirical course of forgetting. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 16:927–935, 1990.