
4.1 Introduction

In Chap. 3 we used the Minerva 2 model as a basis for a tutorial on the use of likelihood-free methods. We fit Minerva 2 to data, and compared the estimated posteriors using likelihood-free algorithms to those obtained using analytic expressions derived in Sheu [37]. Although these analyses presented interesting case studies about how we can use likelihood-free algorithms to estimate the model's parameters, as the true likelihood of the Minerva 2 model has not been derived for the general case, we have not yet demonstrated that the algorithms discussed in this book can provide estimates from the correct posterior distribution.

In this chapter, we show a few examples where likelihood-free algorithms have been used to recover posterior distributions correctly. First, we show how the parameters of the Bind Cue Decide Model of Episodic Memory (BCDMEM) [85] can be accurately recovered using the ABCDE [56] algorithm. This simulation study was carried out and reported in Turner et al. [92], and so we refer the reader to this work for more details on the model and simulation study reported below. Second, we show how parameters of a hierarchical signal detection theory [71] model can be recovered using the Gibbs ABC algorithm [70] and a kernel-based approach [55]. This simulation study was carried out in Turner and Van Zandt [70], and so we refer the reader to this work for details on the analyses we highlight below. Finally, we show how the parameters of the Linear Ballistic Accumulator [95] model can be accurately estimated using the PDA method [38], but not using the synthetic likelihood approach [43]. This simulation study was carried out in Turner and Sederberg [38], where the reader can find more details about the analysis.

4.2 Validation 1: The Bind Cue Decide Model of Episodic Memory

The BCDMEM postulates that when a probe item is presented for recognition, the contexts in which that item was previously experienced are retrieved and matched against a representation of the context of interest. BCDMEM consists of two layers of nodes. The input layer represents items in a local code: each node corresponds to one item. The output layer represents contexts in a distributed code: a pattern of activation over a set of nodes. When an item is studied, a random context pattern of length v for that study episode is constructed by turning on nodes in the output layer with probability s (the context sparsity parameter). The node in the input layer representing the studied item is connected to the nodes in the output layer through associative weights. These connections are established during study by connecting the active nodes on the input and the context layers with probability r (the learning rate).

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

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

When a person is asked whether he has seen an item before, he bases his “old” decision on a comparison between the activation patterns of the reinstated and retrieved context vectors. As in Dennis and Humphreys [85], the i th node in the reinstated context vector is denoted by c_i and the j th node in the retrieved context vector is denoted by m_j . Both c_i and m_j are binary, indicating that the nodes i and j are either inactive or active, so $c_i = 0$ or 1 and $m_j = 0$ or 1 , respectively.

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

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

$$\times \left[\frac{p(1-s) + ds(r+p-rp)}{p(1-s) + dsp} \right]^{n_{0,1}}, \quad (4.1)$$

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

When we use likelihood-free approaches to estimate the posteriors of the model parameters, we are always concerned about the accuracy of those posteriors. One way to evaluate accuracy is to try to recover the parameters that were used to simulate a data set by fitting the model and comparing the estimated posteriors to the known values of the parameters. In addition, because the likelihood function has been derived for BCDMEM [96], we can evaluate whether the estimates of the posterior distributions obtained using likelihood-free methods are similar to the estimates obtained using standard likelihood-based techniques. If the two estimates are similar, then we can declare that the likelihood-free method that was used provides an accurate posterior estimates for this model.

Equation 10 in [96] provides the explicit likelihood function for BCDMEM as a system of equations. We will refer to this likelihood as the “exact” equations. Unfortunately, the exact equations can be difficult to evaluate precisely for all values of θ . For this reason, Myung et al. [96] also derived asymptotic expressions (their Equations 15 and 16) that approximate the exact solution. We will refer to this second set of equations as the “asymptotic” equations. The exact and asymptotic expressions for the hit and false alarm rates allow us to use standard MCMC methods to estimate the posterior distribution for the parameter set θ so long as v is fixed to some positive integer (v must be fixed or the other parameters are not identifiable) [96].

4.2.1 Generating the Data

To perform our simulation study, we first generated data from the BCDMEM for a single person in a recognition memory experiment with four conditions. In each of the four conditions, the simulated person was given a 10-item list during the study phase. At test, the person responded “old” or “new” to presented probes according to whether it was more likely that the probe was a target or distractor. The test lists consisted of 10 targets and 10 distractors.

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

4.2.2 Recovering the Posterior

To assess the accuracy of our likelihood-free approach, we fit the model to the simulated data in three ways. First, we used the ABCDE [56] algorithm to approximate the likelihood function. As discussed in Chap. 2, the ABCDE algorithm is a kernel-based approach, and so we specified a normal kernel function where the summary statistics of the data were the hit and false alarm rates. The spread of the kernel function was fixed to $\delta_{\text{ABC}} = 0.1$ through the estimation process. Second, we used the standard Bayesian approach by using the exact expressions of the likelihood function from Myung et al. [96]. Third, we used the asymptotic expressions of the likelihood function from Myung et al. [96]. As we did in the simulation study from Chap. 3, the details of the sampling algorithm were fixed across the three likelihood approaches, and only the code corresponding to the evaluation (or approximation) of the likelihood function was changed across the three methods. We used DE-MCMC [57,60] as the sampling algorithm and obtained 10,000 samples in total. After discarding a burn-in period of 1000 samples, we were left with 9000 samples collapsed across 12 chains. We used standard techniques to assess convergence of the chains (using the coda package in R) [97,98].

4.2.3 Results

Figure 4.1 shows the estimated marginal posterior distributions for d , p , and r using ABCDE (the gray lines), exact (solid black lines), and the asymptotic (dashed black lines) expressions. Each panel includes the distributions obtained for a single parameter, and the dashed vertical lines indicate the true value of that parameter that generated the data.

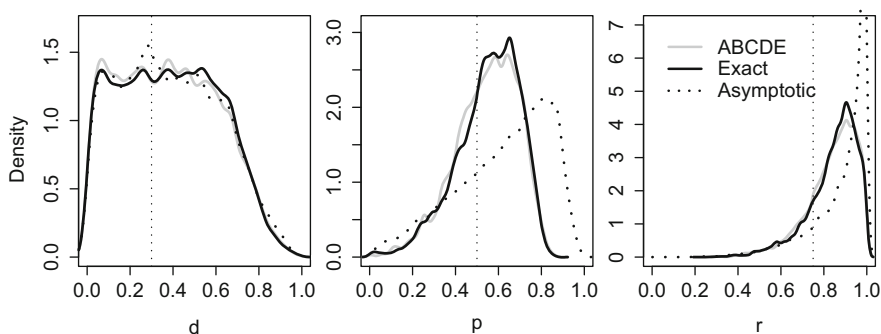


Fig. 4.1 The approximate marginal posterior distributions for the parameters d (left panel), p (right panel), and r (right panel) in BCDMEM using ABCDE (gray lines), the exact likelihood expressions (solid black lines), and the asymptotic likelihood equations (dashed black lines). The dashed vertical lines are located at the values of the parameters used to generate the data

There are two main results of our simulation study. First, the estimated posteriors we obtained using ABC are very similar to those obtained using the exact expressions for the likelihood. This suggests that the combination of $\rho(X, Y)$, \mathcal{H} , and δ we selected produced accurate ABC posterior estimates. Second, the posterior estimates we obtained using the asymptotic expressions are different from those we obtained with the true likelihood, especially the posterior estimates for the parameters p and r . This inaccuracy suggests that the asymptotic expressions will not be very useful for Bayesian analyses of BCDMEM.

In addition to the differences in posterior estimates that we obtained with each method, the computation times required to obtain these estimates also varied considerably with different methods. The method of estimation using exact likelihoods required 2 h and 20 min of computation. The method using the asymptotic expressions took only 36 s. The ABCDE approach took 2 min and 33 s. While the asymptotic expressions did provide the fastest results, they were considerably less accurate compared to the ABCDE approach. Perhaps more interesting is that the ABCDE approach was 55 times faster than when using the exact expressions, which we take as a testament to the usefulness of these kernel-based approaches for fitting simulation models such as BCDMEM.

4.2.4 Summary

In this section, we illustrated the utility of the ABC approach by fitting the model BCDMEM to simulated data. The derivations in Myung et al. [96] provided expressions for the model's likelihood, which allowed us to compare estimates of the posterior distribution obtained with standard Bayesian techniques to the estimates obtained with likelihood-free techniques. We showed that the estimates obtained using ABCDE were very close to the estimates obtained using the exact expressions, but the estimates obtained using the asymptotic expressions did not closely match either the ABCDE or exact expression estimates.

4.3 Validation 2: Signal Detection Theory

Signal detection theory (SDT) is one of the most widely applied theories in all of cognitive psychology for explaining performance in two-choice task. In these tasks, someone is presented with a series of stimuli and asked to classify each one as either signal (a “yes” response) or noise (a “no” response). What constitutes noise and signal can be flexible. For example, a person may be asked to indicate whether they have observed a flashing light by responding “yes” if they’ve detected it or “no” if they have not. The variability in the sensory effect of the stimulus, due either to noise in the person’s perceptual system or to variations in the intensity of the signal itself, is represented by two random variables: the first is the sensory effect of noise when no light is presented, and the second is the sensory effect of signal when the light is presented. Typically, a presentation of a signal (a flashing light) will result in larger sensory effects than the presentation of noise alone.

The psychological representations of the effects of signals and noise are frequently modeled with two random variables. These variables are assumed to be normally distributed with equal variance, although neither of these assumptions is necessary. The equal-variance, normal version of the SDT model has only two parameters. The first parameter d represents the discriminability of signals and is the standardized distance between the means of the signal and noise distributions. Higher values of d result in greater separation and less overlap between the two distributions, meaning that signals are easier to discriminate from noise. The second parameter is a criterion c , which is along the axis of sensory effect. A person makes a decision by comparing the perceived sensation to c . If the perceived magnitude of the effect is above this criterion, the person responds “yes.” If not, the person responds “no” [99].

When the two representations (signal and noise) have equal variance and the payoffs and penalties for correct and incorrect responses are the same, an “optimal” observer should place his or her criterion c at $d/2$. This is the point where the two distributions cross, or equivalently the point at which the likelihoods that the stimulus is either signal or noise are equal. We can then write the “non-optimal” observer’s criterion c as $d/2 + b$, where b represents bias, or the extent to which the person prefers to respond “yes” or “no.” Negative bias shifts the criterion toward the noise distribution, increasing the proportion of “yes” responses, while positive bias shifts the criterion toward the signal distribution, increasing the proportion of “no” responses.

SDT has been influential because it separates effects of response bias from changes in signal intensity. The parameter d , the distance between the means of the two representations, increases with increasing stimulus intensity. The parameter b , the person’s bias, is an individual-level parameter that can be influenced by instructions to be cautious, payoffs or penalties that reward one kind of response more than another, or changes in the frequency of each type of stimuli.

SDT is meant to be used as a tool to measure discriminability and response bias. The likelihood function for the SDT model is easy to compute, which makes it yet another model we can use as a case study to examine other sampling algorithms. In the BCDMEM example above, we examined a single-level model with the ABCDE algorithm. For this example, we will investigate the ability of the Gibbs ABC algorithm to recover both subject-level and hyper-level parameters.

4.3.1 Generating the Data

The parameters for an individual j are that person’s discriminability d_j and bias b_j . We built a hierarchy by assuming that each discriminability parameter follows a normal distribution with mean d_μ and standard deviation d_σ , and that each bias parameter follows a normal distribution with mean b_μ and standard deviation b_σ . To generate data to which the model could be fit, we first set $d_\mu = 1$, $b_\mu = 0$, $d_\sigma = 0.20$, and $b_\sigma = 0.05$. We then drew nine d_j and b_j parameters from the normal hyperdistributions defined by the hyperparameters for nine hypothetical subjects.

We used these person-level parameters to generate “yes” responses for $N = 500$ noise and signal trials by sampling from binomial distributions with probabilities equal to the areas under the normal curves to the right of the criterion (i.e., $d_j/2 + b_j$).

4.3.2 Recovering the Posteriors

We fit the hierarchical SDT model in two ways. The first way uses the true likelihood function [6, 9]. The second approach used the Gibbs ABC algorithm and a kernel-based ABC algorithm to approximate the likelihood function [55], both of which are described in Chap. 2. We set $\rho(X, Y)$ equal to the Euclidean distance between the observed hit and false alarm rates (i.e., the simulated data to which the model is being fitted) and the hit and false alarm rates arising from simulating the model with a set of proposed parameters θ^* . This distance was weighted with a Gaussian kernel using a turning parameter $\delta = 0.01$. For both the likelihood-free and likelihood-informed estimation procedures, we generated 24 independent chains of 10,000 draws of each parameter, from which we discarded the first 1000 iterations. This left 216,000 samples from each method with which we estimated the joint posterior distributions of each parameter.

4.3.3 Results

Figure 4.2 shows the estimated posterior distributions for the model’s hyperparameters, d_μ , b_μ , d_σ , and b_σ , as histograms. Overlaid on these histograms are the posterior density estimates (solid curves) obtained from the likelihood-informed method (i.e., MCMC), and the vertical lines represent the values of the hyperparameters with which the fitted data were simulated. The left panels of Fig. 4.2 show the estimated posteriors for the hypermeans b_μ (top) and d_μ (bottom). The right panels show the estimated posteriors for the log hyper standard deviations b_σ (top) and d_σ (bottom). The estimates obtained from the Gibbs ABC algorithm closely match the estimates obtained using conventional MCMC.

At the individual level, Fig. 4.3 shows the estimated posterior distributions for the discriminability (d_j) parameters. As for the group-level parameters, the two methods produced posterior estimates that do not differ greatly. Although we can also examine the posterior distributions for the subject-level bias parameters (b_j), these estimates were similarly accurate as the discriminability parameters shown in Fig. 4.3.

4.3.4 Summary

We used a combination of Gibbs ABC and kernel-based ABC to estimate the parameters of the SDT model. The likelihood function for this model is well known and simple so the true posterior distribution can be estimated using standard MCMC techniques. We showed that the estimated posteriors using both likelihood-informed

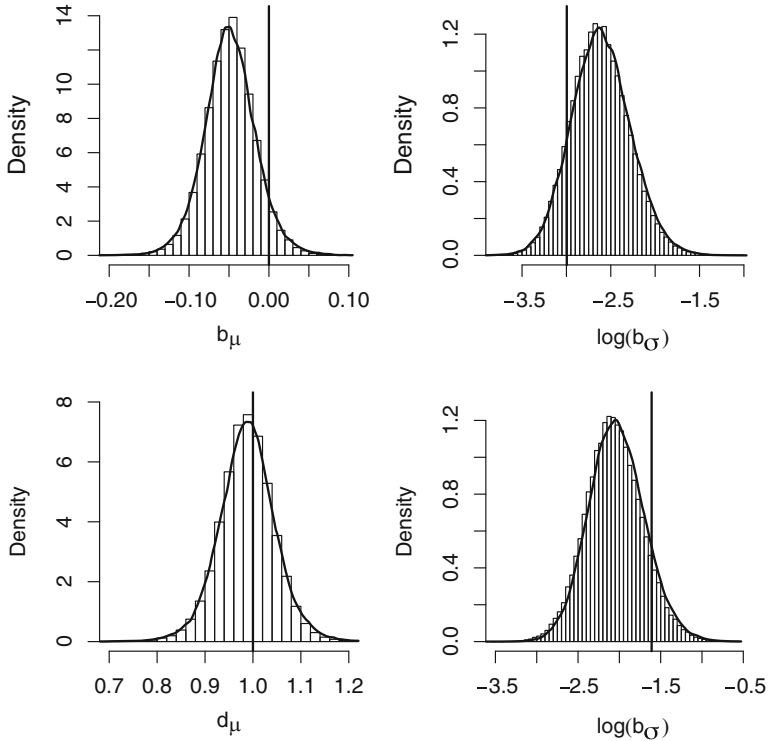


Fig. 4.2 The estimated posterior distributions obtained using likelihood-informed methods (black densities) and the Gibbs ABC algorithm (histograms) for the hyperparameters of the classic SDT model. Vertical lines are placed at the values used to generate the data. The rows correspond to group-level parameters for the bias parameter b (top) and the discriminability parameter d (bottom). The columns correspond to the hypermeans (left) and the hyper standard deviations on the log scale (right)

and likelihood-free methods were similar for both the individual-level and the group-level parameters. These results demonstrate that Gibbs ABC fused with a kernel-based approach can recover the true posterior distributions of the hierarchical SDT model accurately.

4.4 Validation 3: The Linear Ballistic Accumulator Model

The Linear Ballistic Accumulator model (LBA; [95]) is a stochastic accumulator used to explain choice response time data. In a typical choice response paradigm, people are asked to make a decision with two or more response alternatives. For example, in a numerosity task, a person might be asked to select which of two boxes contains more of a certain type of object (e.g., stars, dots, etc.). The time between the onset of the stimulus and the execution of the response is the response time (RT) and the box that is chosen (left or right) determines the choice response. The

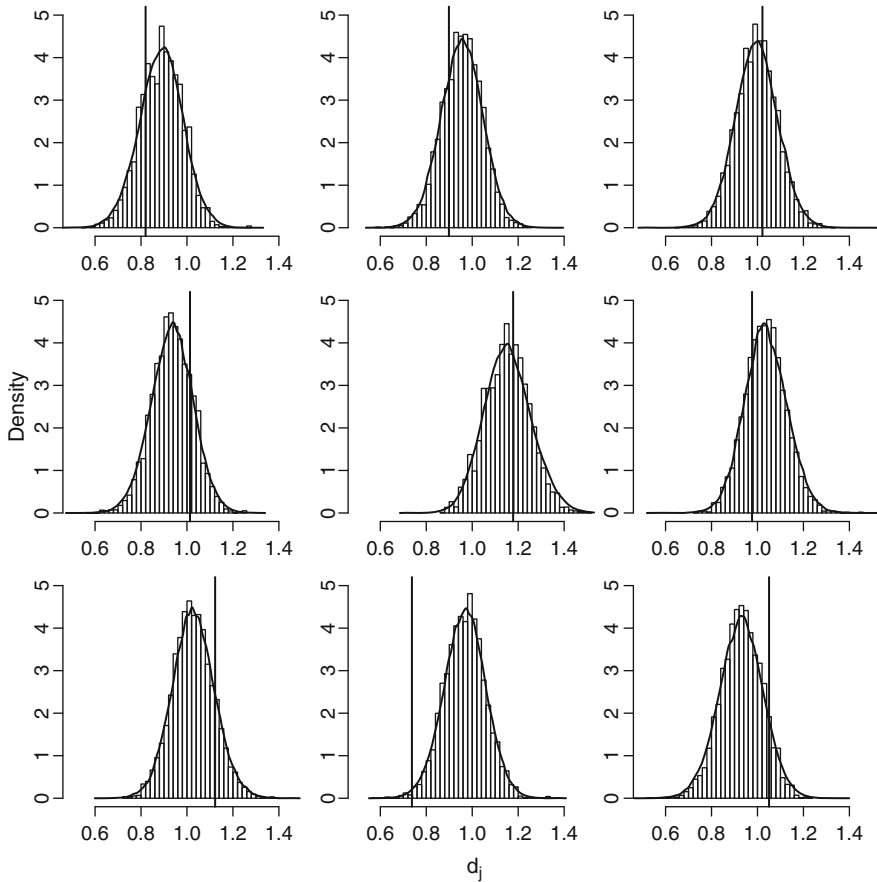


Fig. 4.3 The estimated posterior distributions obtained using likelihood-informed methods (black densities) and the Gibbs ABC algorithm (histograms) for the person-level discriminability parameters d_j of the classic SDT model. Vertical lines are located at the values used to generate the data. Each panel shows the posteriors for a different person

data from this task are therefore mixed, including both continuous RT measures and discrete response measures.

The LBA model postulates the existence of separate accumulators for each possible choice alternative. Each accumulator stores evidence for a particular response, and this evidence increases over time following the presentation of a stimulus. In particular, on accumulator c , the function describing the amount of evidence as a function of time is linear with slope d_c and starting point (y -intercept) k_c . The evidence on the accumulator grows until it reaches a threshold b . The starting point k_c is uniformly distributed in the interval $[0, A]$, and the rate of evidence accumulation d_c is normally distributed with mean $v^{(c)}$ and standard deviation s . After the fastest accumulator reaches the threshold b , a decision is made

corresponding to the winning accumulator, and the RT is the sum of the finishing time of the winning accumulator and some nondecision time τ (encompassing initial perceptual processing time and the motor response time).

The LBA is one of a large class of information accumulation models that explain choice and response time data. It differs from other models in this class by eliminating certain complexities such as competition between alternatives [100,101], passive decay of evidence (“leakage”) [100], and even within-trial variability [102, 103]. The LBA is therefore mathematically much simpler than other models of this type of decision, and has closed-form expressions for the joint RT and response likelihoods. This simplicity is a big reason for this model’s wide acceptance [104–109]. Like the earlier applications in this chapter, we will simulate data from the LBA model using known parameters, and then attempt to recover those parameters by fitting the model back to the simulated data using both the known likelihoods and likelihood-free algorithms. Although in Chap. 3 we showed how to use the PDA method [38] to estimate the parameters of the Minerva 2 model, we have not yet shown how the PDA method can be used to fit a model to data with both continuous and discrete measures. Further, to stress the importance of selecting good statistics in kernel-based approaches, we will investigate whether or not quantiles provide sufficient information for the parameters of the LBA model.

4.4.1 Generating the Data

We simulated data from a hypothetical 2-choice task. For two stimulus types (say, “more on the left” and “more on the right” for the task described above), there are two possible responses (“left” and “right”). The LBA model for this task would therefore require two accumulators, one for the “left” decision and one for the “right” decision. On each trial, accumulator c will have a starting point k_c and an accumulation rate d_c , where c indexes the “left” or “right” accumulator. If a “more on the left” stimulus is presented, the “left” accumulator’s rate will be greater on average than the “right” accumulator’s rate, and if a “more on the right” stimulus is presented, the “right” accumulator’s rate will be greater on average than the “left” accumulator’s rate. For simplicity, we assumed no asymmetry between the “left” and “right” accumulators, so we can write the effects of stimulus more compactly in terms of the means of d_{left} and d_{right} , letting $v^{(C)}$ represent the mean accumulation rate for correct responses (“left” to “more on the left” and “right” to “more on the right”) and $v^{(I)}$ represent the mean accumulation rate for incorrect responses (“right” to “more on the left” and “left” to “more on the right”) and setting $v^{(C)} > v^{(I)}$.

We generated 500 responses from the LBA model using a threshold $b = 1.0$, setting the upper bound of the uniform starting point distribution $A = 0.75$, setting the mean accumulation rate for correct responses $v^{(C)} = 2.5$, and the mean accumulation rate for incorrect responses $v^{(I)} = 1.5$. We also added to each simulated RT a nondecision time $\tau = 0.2$. We set the standard deviation of the accumulation rates to $s = 1$ to satisfy the scaling properties of the model. All of these parameter values are consistent with previously published fits of the LBA model to experimental data.

4.4.2 Recovering the Posterior

We used three different approaches to estimate the posterior distribution of the model parameters (i.e., $b, A, v^{(I)}, v^{(C)}, \tau$). The first approach makes use of the likelihood function (see [57, 95, 108, 109], for applications). The second is the PDA method for mixed data types as described in Chaps. 2 and 3. When using this method, we simulated the model 10,000 times for each parameter proposal. The third is the synthetic likelihood algorithm [43], which requires the specification of a set of summary statistics $S(\cdot)$. To implement the algorithm, we decided to use the sample quantiles (corresponding to the cumulative proportions $\{0.1, 0.3, 0.5, 0.7, 0.9\}$) for both the correct and incorrect RT distributions.¹ Thus, for a given set of RTs Y and choices Z , we summarized the data by computing the vector $S(Y, Z)$ comprising 11 statistics: 5 quantiles for each of the samples of correct and incorrect RTs, plus the proportion of correct responses. When using the synthetic likelihood method, we generated 50,000 model simulations per parameter proposal.

It has been noted that the parameters of the LBA model are highly correlated by examining the correlation of samples from the joint distribution of model parameters (see [57]). The correlation in the posteriors makes it difficult to propose sets of parameters that will be accepted at the same time. As a result, conventional sampling algorithms such as Markov chain Monte Carlo (MCMC) [46] can be inefficient, requiring very long chains, and are therefore impractical. For this reason we used the DE-MCMC algorithm to draw samples from the posterior distribution for each of the three methods. For each of the three different likelihood evaluation methods, we implemented a DE-MCMC sampler with 24 chains for 5000 sampling iterations, discarding the first 100 observations in each chain.

4.4.3 Results

Figure 4.4 shows the estimated posterior distributions obtained using the PDA method (top row) and the synthetic likelihood method (bottom row). The columns of Fig. 4.4 correspond to the threshold parameter b , the starting point upper bound parameter A , the accumulation rate for correct responses $v^{(C)}$, the accumulation rate for incorrect responses $v^{(I)}$, and the nondecision time τ . In each panel, the true estimated posterior distribution (i.e., the posterior obtained using the true likelihood) is shown as the black curve plotted over the histogram, and the vertical dashed line marks the value of the parameter used to generate the data.

The figure shows two important things. First, the histograms from the PDA method align closely with the true density (black curve) and are centered at the values of the parameters that generated the data. Therefore, we can state that the PDA method produces posterior estimates that are close to the true posterior

¹We treated the correct and incorrect RT distributions as the accumulators themselves, rather than the response alternatives.

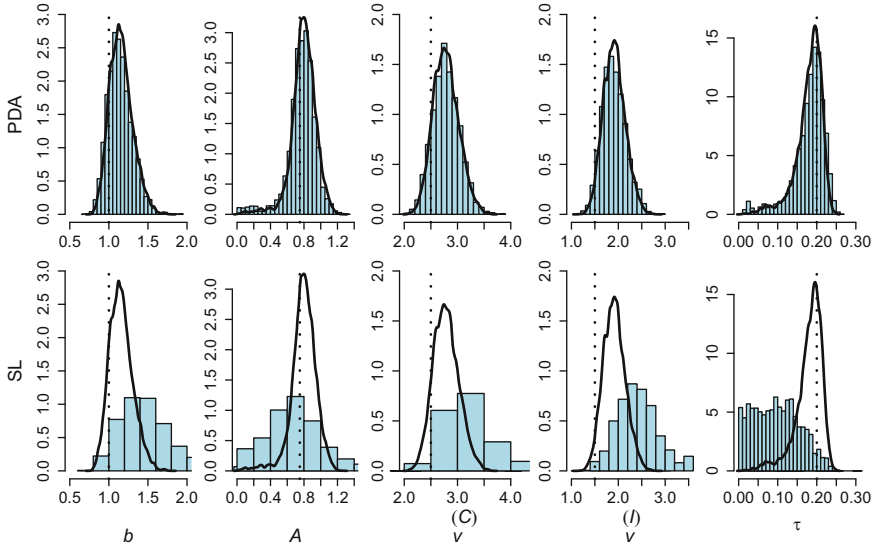


Fig. 4.4 Estimated marginal posterior distributions obtained using the PDA method (top row), and the synthetic likelihood algorithm (SL; bottom row). In each panel, the true estimate of the posterior distribution (i.e., the likelihood-informed estimate) is shown as the black density, and the vertical dashed lines are placed at the values of the parameters used to simulate the data

estimates. Because the PDA method is a general technique that makes use of all the observations in the data set, we can be sure the accuracy of the posterior estimates only depends on the kernel density estimate. Second, the histograms generated using the synthetic likelihood methods vary widely around the true posterior densities and are not centered on the values of the parameters that generated the data. Therefore, we can state that the posterior estimates obtained using the synthetic likelihood method are probably inaccurate. There may be several reasons for this, but the most likely is that the summary statistics (i.e., the quantiles) used for the parameters of the LBA model are not sufficient. The use of quantiles seems to have resulted in high proposal rejection rates even in the high-probability regions of the posteriors.

4.4.4 Summary

In this section we showed that the PDA algorithm can produce accurate estimates of the posterior distributions of the parameters of the LBA model. These results are reassuring, because they imply that the problem of generating sufficient statistics for a model with no implicit likelihood can be safely bypassed with the PDA method. The PDA method does not require the use of sufficient statistics. Using this method we demonstrated accurate recovery of the posterior distribution with only minimal assumptions and specifications of how to approximate the likelihood function. By

contrast, the synthetic likelihood algorithm did not produce accurate estimates of the posterior. While there are several reasons why this can happen, we suspect the main reason is that the quantiles we used are not sufficient for the parameters of interest.

This study provides a cautionary tale about the use of likelihood-free algorithms in inappropriate circumstances. Rejection-based and kernel-based algorithms are likely to produce errors in the estimated posterior distribution when sufficient statistics are not known. In these situations, we recommend using the PDA method if computational resources are available to make extensive model simulation feasible.

4.5 Conclusions

In this chapter, we have illustrated the effectiveness of different likelihood-free techniques for three popular models in cognitive science. The models we chose all have a likelihood function, which enabled us to make comparisons between the estimates obtained using likelihood-free methods and the estimates obtained when using the true likelihood function.

In the first application, we showed how the ABCDE algorithm could be used to estimate the posterior distribution of BCDMEM's parameters. We compared the estimates obtained using ABCDE to those of the exact and asymptotic equations put forth by Myung et al. [96]. In the end, we concluded that the estimates obtained using likelihood-free algorithms were not only as accurate as those obtained when using the true likelihood function, but they were achieved at a much faster computation time. We find this promising for the practical implementation of ABCDE, at least for BCDMEM.

In the second application, we showed how a combination of Gibbs ABC and a kernel-based approach could be used to accurately estimate the parameters of a hierarchical SDT model. We found that at both the subject- and group-levels, the estimates obtained using our algorithms were accurate, given that they closely resembled the shape of estimates obtained using a standard, likelihood-informed Bayesian approach.

In the third application, we showed that the PDA method could be used to estimate the parameters of the LBA model. The application of likelihood-free algorithms is tricky when the data consist of choice and response time measures because it is unclear what statistics should be used to convey sufficient information about the full response time distribution to the model parameters. For this reason, we used the PDA method to reconstruct the entire choice RT distribution for a given parameter proposal θ^* , which we then compared to the observed data. We concluded that even in the context where sufficient statistics are not known, if we choose the appropriate likelihood-free algorithm (see Chap. 2), we can still arrive at accurate estimates of the posterior distribution.