A practical guide to non-parametric approximate Bayesian computation with improved implementation and error characterization

William R. Holmes

Center for Complex Biological Systems and Department of Mathematics, University of California Irvine, Department of Mathematics and Statistics, The University of Melbourne, Melbourne, Victoria 3010 Australia - (613) 8344 5550.

Abstract

A critical task in modelling is to determine how well the theoretical assumptions encoded in a model account for observations. Bayesian methods are an ideal framework for doing just this. Existing approximate Bayesian computation (ABC) methods however rely on often insufficient "summary statistics". Here, I present and analyze a highly efficient extension of a recently proposed (Turner et al., 2014) non-parametric approximate Bayesian computation (npABC) algorithm, which circumvents this insufficiency. This method combines Markov Chain Monte Carlo simulation with tools from non-parametric statistics to improve upon existing ABC methods. The primary contributions of this article: 1) A more efficient implementation of this method is described, that substantially improves computational performance and chain mixing. 2) Theoretical results describing the influence of methodological approximation errors on posterior estimation are discussed. In particular, while this method is highly accurate, even small errors have a strong influence on model comparisons when using standard statistical approaches (such as deviance information criterion). Thus care must be taken when using this (or any other ABC) method for model comparison. 3) An augmentation of the standard MCMC procedure, termed "Resampled MCMC", that reduces the negative influence of approximation errors on performance and accuracy, is presented. 4) In order to make this method accessible to a broader audience, a number of examples of varying complexity are presented along with supplementary code for their implementation.

Keywords: Non-parametric Approximate Bayesian Computation, Approximate Likelihood, Kernel Density Estimate, Markov Chain Monte Carlo, Linear Ballistic Accumulator Model

Email address: wrholmes@uci.edu (William R. Holmes)

1. Introduction

Numerous models of cognitive processes have been developed over the decades, varying in detail and complexity from neuro-physiological models (Brunel and Wang, 2001; Albantakis and Deco, 2009) to high level models (Ratcliff, 1978; Usher and McClelland, 2001; Brown and Heathcote, 2008). A central benefit of modelling over experimentation alone is the ability to translate formal theories into testable predictions. Subsequent testing can be performed in a number of ways. In more detailed models, qualitative comparisons between model and data are often made to determine if the theory can account for the qualitative trends observed in data. While such comparisons can quickly refute a theory, they can rarely provide more than weak support. Higher level models on the other hand are typically tractable enough for quantitative comparisons against data. This typically involves "fitting" a model to data and estimating the values of model parameters. Quantitative methods provide two distinct benefits. First, quantitative measures provide a more fine grained account of how well a theory accounts for observations. Second, access to parameter values and their uncertainty provides further inferences about underlying behaviour.

Historically, parameters have been estimated using linear programming methods (Dantzig and Thapa, 1997) designed to minimize some statistic such as sum squared error. This frequentist approach however provides no information about estimation uncertainty. For this reason, these methods are often supplemented with some form of sensitivity analysis (Saltelli et al., 2008). More recently however, following algorithmic improvements and computing advances, Bayesian methods have moved to the forefront. These have a number of powerful benefits (Abelson, 2008; Gallistel, 2009; Lee, 2008; Lee and Wagenmakers, 2013) which are beyond the scope of this article, but in short they provide a more principled way to account for uncertainty and incorporate prior knowledge.

Numerous methods for performing Bayesian analysis have been devised. Standard Markov Chain Monte Carlo (MCMC) techniques (Gelman et al., 2003; Robert and Casella, 2004; Cappé et al., 2004; Del Moral et al., 2006) have proven very powerful, but can only be applied to the simplest problems where a model can be analytically described by a closed form probability density function. Other approximate Bayesian computation (ABC) methods have been developed to circumvent this requirement (Csilléry et al., 2010; Turner and Van Zandt, 2012). Unfortunately they have a number of downsides, particularly that in most cases they do not actually estimate parameters of the desired model (see below for further discussion). Recently, a new method, referred to here as a non-parameteric ABC method (or npABC), was developed (Turner and Sederberg, 2014) that bridges the gap between these exact and approximate methods, alleviating some of these issues.

The goals of this article are three fold: 1) describe this new methodology in detail and determine its strengths, weaknesses, and limitations, 2) improve upon it, and 3) present it in an accessible way so it can be utilized by a broader audience of end users. In particular, toward this third goal, a number of examples of this method are presented along with documented MATLAB code for two of the examples. This is not intended as a "plug in" software package, but rather to aid implementation of this method by others. Also toward this goal, a detailed procedural overview of this method (with improvements presented here) is provided in Section 4.

1.1. Brief introduction to Bayesian methods

The canonical Bayesian model estimation problem is to determine the posterior probability distribution of a set of model parameters conditioned on observed data $\pi(\theta|X)$. This typically involves three critical steps. First, a *prior* distribution on the parameters $\pi(\theta)$, which encodes pre-existing knowledge of the system, must be supplied. This is rarely a problem as some form of subjective prior belief is typically available from which a prior can be constructed. Second, a probability density (aka. likelihood) function $L(X|\theta)$, which describes the likelihood that the model will give rise to the observed data given the parameters θ, must be computed. When observations are identical and independently distributed, this reduces to

$$L(X|\theta) = \prod_{i=1}^{I} L_i(\theta),$$

where $L_i(\theta) := L(x_i|\theta)$, so only the likelihood of each individual observation is required. The following exposition will be confined to this simplified setting. Third, with these model components provided, the models posterior distribution is computed via Bayes' theorem

$$\pi(\theta|X) = \frac{L(\theta|X)\pi(\theta)}{\int L(\theta|X)\pi(\theta)}. \tag{1.1}$$

In almost all practical situations, this third step is impossible to perform analytically since the required integral is usually not solvable. For this reason, numerous MCMC methods have been developed to circumvent this third step.

In many cases however, the second step is problematic as well since the model may either 1) not emit an analytic density function or 2) emit one that is too cumbersome to compute. Approximate Bayesian computation methods have been developed to deal with this difficulty, see (Csilléry et al., 2010; Turner and Van Zandt, 2012) for existing reviews. Generally speaking, ABC deals with the absence of a likelihood by prescribing a surrogate measure for how likely or plausible a particular parameter set (θ) is. To accomplish this, a large number of simulated data observations ($\tilde{X}$) are drawn from the model. The observed (X) and simulated ($\tilde{X}$) data are then compared in some way to determine how likely that parameter set is. Typically this comparison is accomplished by compressing both data sets into a set of summary statistics $S(X)$ and then defining a "distance" between them $\rho(S(X), S(\tilde{X}))$.

This method raises two distinct issues. First, a reasonable distance function $\rho(\cdot,\cdot)$ must be prescribed. A more serious issue however is that the summary statistics must adequately represent the models output, often referred to as a sufficiency condition. ABC methods do not approximate the models posterior $\pi(\theta|X)$, but rather a posterior augmented by the choice of S, $\pi(\theta|S(X))$. So ABC only estimates the posterior distribution of the intended model if $\pi(\theta|S(X)) = \pi(\theta|X)$, which is often not possible to verify. The essential problem here is that the use of summary statistics force assumptions on the structure of the underlying likelihood function, which if inaccurate lead to potentially serious errors (Robert et al., 2011) that no amount of computational effort will correct. As an extreme example, using mean and variance as summary statistics to describe a distribution implies a normality assumption, which could be very poor if the underlying model is multimodal or

heavily skewed. In a more cognitive context, choice response time distributions are often described by quantile summary statistics (Heathcote et al., 2002; Ratcliff and Tuerlinckx, 2002; Heathcote and Brown, 2004). This was however recently shown to be an insufficient summary of the data (Turner and Sederberg, 2014), leading to substantial posterior inaccuracies.

In the broader statistics field, such issues have been overcome through the development of "non-parametric statistical" methods, which free the user from having to make potentially erroneous assumptions (e.g. summary statistics) on the structure of their data / model. Recently, non-parametric methods have been incorporated into the ABC context (Turner and Sederberg, 2014) to improve Bayesian estimation methods (e.g. npABC). These methods begin the same way as canonical ABC by first simulating a large number of samples from the underlying distribution. Next however, they construct an approximation of the underlying likelihood $\hat{L}(X|\theta)$. In this case, no summary statistics are prescribed and much weaker assumptions on the structure of $L(X|\theta)$ are made. The approximate likelihood is then substituted ($L \rightarrow \hat{L}$) into the chosen MCMC framework and an approximate posterior is determined.

In the following sections, the implementation details required to apply this method will be discussed. First, the "kernel density estimate" (or KDE), which is the core of the likelihood estimation, is described. An improvement of this method that yields substantial efficiency gains over that in (Turner and Sederberg, 2014) is also provided. Second, the influence of likelihood estimation errors on posterior estimation and MCMC efficiency will be discussed from a theoretical perspective. Third, the full npABC will be demonstrated through three examples of increasing complexity. Using these examples, the strengths and weaknesses of the method will be described. Finally, a stand alone section (4) describing the core implementation steps of this method is provided for the user interested primarily in implementing this method.

2. Methods

2.1. The kernel density estimate

The critical step in npABC is the construction of the approximate likelihood $\hat{L}(X|\theta)$ that will replace L in canonical MCMC estimation. The kernel density estimate (KDE) is a powerful tool for doing just this (Silverman, 1982, 1986; Epanechnikov, 1969). The first step in this process is to simulate N_s draws ($\tilde{X}$) from $Model(\theta)$. This step is of course dependent on the model under consideration, which will dictate how these samples are produced. The second and final step is to use kernel density estimation (KDE) to extract likelihood estimates for the observed data X from the simulated $\tilde{X}$. For purposes of generality, the KDE process will first be discussed independent of posterior estimation.

The basic problem is to estimate the probability density $f(x_i)$ (a placeholder for $L_i(\theta)$) of each individual observation x_i from the samples $\tilde{X} = \{\tilde{x}_j\}$, where $j = 1...N_s$. The KDE of this quantity is given by

$$f(x_i) \approx \hat{f}(x_i) := \frac{1}{N_s} \sum_{j=1}^{N_s} K_h(x_i - \tilde{x}_j). \tag{2.1}$$

From here on the $\hat{}$ will reference a kernel density estimate of the underlying likelihood, or a quantity

derived from it. Here K_h is a "smoothing kernel" defined by

$$K_h(z) = \frac{1}{h} K\left(\frac{z}{h}\right),$$

where K is a continuous function that is symmetric about $z = 0$ and integrates to 1. The parameter h, commonly referred to as a "bandwidth" size, determines the smoothing properties of the kernel: large h heavily smoothes the sampled data while small h provides less smoothing. To illustrate this, consider the uniform kernel $K(z) = \chi_{[-0.5,0.5]}(z)$ where χ is the standard indicator function that is one on the prescribed interval and zero elsewhere. This kernel produces a standard histogram estimator with h corresponding to the size of the histogram bins. Histograms with small bins (i.e. small h) of course produce noisy plots while those with large bins produce smoother but less refined plots. See (Silverman, 1986; Epanechnikov, 1969) for a full review of KDE theory.

There are a few critical properties of the KDE estimator relevant to this discussion. $\hat{f}$ is an approximation to f and as such can be thought of as an estimator with some underlying distribution. For the standard class of first order kernel functions (e.g. biweight, Gaussian, Epanechnikov, etc.), this distribution is approximately normal with intrinsic bias and variance

$$Bias(\hat{f}(x)) \approx \frac{h^2}{2} f''(x) M_2(K), \qquad Var(\hat{f}(x)) \approx \frac{1}{N_s h} f(x) \|K\|_2, \tag{2.2}$$

where $M_2(K)$ and $\|K\|_2$ denote the second moment and Euclidean (or L^2) norm of K respectively (Silverman, 1986). From these estimates we see that the bandwidth, number of samples, and specific choice of K all affect accuracy of this approximation. In practice, the specific choice of K has only marginal effects on accuracy, though the Epanechnikov kernel is known to minimize mean integrated square error (Epanechnikov, 1969). The bandwidth (h) and number of samples (N_s) however are of critical importance and will each have different effects on the posterior estimation process. For now, simply note that N_s plays a role in variance control while h modulates a classic bias-variance tradeoff.

2.1.1. An improved, more efficient KDE implementation

Before discussing the influences of this approximation procedure on posterior estimation, I will discuss a technical improvement on the classic implementation of KDE that substantially speeds computational implementation of npABC. Before continuing with this section however, note that it is technical in nature. This improved procedure has the same accuracy, the same bias / variance issues as the standard KDE procedure, and will lead to the same results as the standard KDE when embedded into the npABC procedure. This improvement does however substantially improve efficiency for reasons that will be discussed.

Direct computation of $\hat{f}$ from Equ. (2.1), while simple, is inefficient. Given a set of N_d observations, the kernel function must be evaluated $N_s \cdot N_d$ times. While a single evaluation of this size is reasonable, this becomes a computational bottle neck in MCMC applications where the likelihood must be evaluated at many chain iterations. The improvement presented here, first proposed in (Silverman, 1982), takes advantage of the observation that the KDE formula in Equ. (2.1) resembles

a convolution. The discrete model samples $\tilde{X}$ can be represented by the following function

$$d(x) = \frac{1}{N_s} \sum_{j=1}^{N_s} \delta_{\tilde{x}_j}(x), \tag{2.3}$$

where δ is the Dirac delta function. It is then direct to show that

$$d \star K_h(x) = \frac{1}{N_s} \sum_{j=1}^{N_s} K_h(x - \tilde{x}_j), \tag{2.4}$$

where $\star$ denotes the standard convolution. This is precisely the KDE formula in Equ. (2.1). The KDE thus resembles a canonical smoothing operation (with K as the smoother), proposed as early as 1944 in partial differential equations literature (Friedrichs, 1944).

While convolutions are well know to be intensive to compute directly, this burden can be greatly reduced by making use of techniques from signal processing theory, where convolutions are common. The "convolution theorem" states that $\mathcal{F}(f \star g) = \mathcal{F}(f) \cdot \mathcal{F}(g)$, where $\mathcal{F}$ is the continuous Fourier transform. Since multiplication is much more efficient than convolution, the basic idea of this method is to transform both d and K_h into the spectral domain, multiply in the spectral domain (which effectively is the convolution), then transform back. Given the high efficiency of Fast Fourier Transform methods (FFT), transferring to and from the spectral domain is fast relative to the convolution. This was originally proposed as an efficient method for generating a high resolution PDF on a regular grid, particularly for plotting purposes. Likelihood values of observations can however readily be interpolated from this regular grid.

There is a technical point that must be addressed before applying this method; the FFT is only efficient if the data being transformed is on an regular grid, which is not the case for the samples $\{\tilde{x}_j\}$. To circumvent this, the samples should first be binned to a very fine grid with 2^n points (a power of 2 is used for technical reasons related to FFT efficiency). This grid should be much more finely spaced ($n > 8$ typically) than a typical histogram grid for reasons discussed in a moment. The improved FFT based KDE procedure is then as follows:

1. Bin the simulated samples to a very fine grid, $d \to \tilde{d}$.
2. Transform the resulting data to the spectral domain ($\tilde{d}(x) \to \mathcal{F}[\tilde{d}](s)$) using a FFT (where $\mathcal{F}[\tilde{d}](s)$ is the contribution of wave number s, i.e. the frequency spectrum of $\tilde{d}$).
3. Carry out the convolution operation in the spectral domain

$$\mathcal{F}[\tilde{d} \star K_h](s) = \mathcal{F}[\tilde{d}](s) \cdot \mathcal{F}[K_h](s). \tag{2.5}$$

4. Using an inverse FFT, transform the resulting expression back to obtain the likelihood estimate on the same 2^n grid

$$\hat{f} = \mathcal{F}^{-1}\left(\mathcal{F}[\tilde{d}] \cdot \mathcal{F}[K_h]\right) \tag{2.6}$$

5. Interpolate the density from this grid to the observed data points to obtain $\hat{f}(x_i)$. Linear interpolation should be used here since higher order methods (such as cubic splines) can generate negative likelihood values in the tail of a distribution.

Since FFT, multiplication, and interpolation are each highly efficient and usually optimized within programming languages, this procedure is vastly more efficient than direct computation of the convolution.

A few notes about this procedure are in order. First, the FFT itself introduces some errors into the approximation, but these are orders of magnitude smaller than the primary error sources. Second, the binning and interpolation steps will introduce errors as well. However these will again be very small provided $n > 8$ ($n = 10$ is used in all following applications). Third, in principal, any kernel (K) can still be used in this process. However the canonical Gaussian kernel is particularly useful in this case since its Fourier transform is another Gaussian, $\mathcal{F}[K_h](s) \propto \exp(-0.5h^2s^2)$. Fourth, one must be careful when applying FFT's since various operations (scalings, shifts, etc.) must be performed to correctly prepare the data. Such details are not mentioned here as they are specific to programming language and the FFT implementation being called. Instead, supplementary files that demonstrate implementation in MATLAB are provided.

It also interesting to note that this view of the kernel density estimate is somewhat of a mathematical departure from the original view. In its original form, the KDE was essentially designed as an extension of the histogram to be a method of pooling information from nearby samples in a weighted manner to make a more accurate density estimate. This procedure however more closely resembles filtering of a noisy signal to determine the underlying "true" trend. In this way, the transformed function $\mathcal{F}[K_h]$ acts as a low pass filter in the spectral domain that attenuates high frequency noise. To illustrate this, and more generally how this procedure works, consider the following simple example.

2.1.2. Example 1: Reconstructing a Gaussian distribution

In this section, the KDE procedure is demonstrated through a simple example. We begin with a known normal distribution with known mean ($\mu = 5$) and variance ($\sigma = 1$). Next, this FFT based KDE procedure is used to reconstruct the underlying distribution from $N_s = 10,000$ simulated draws. The binning of these N_s observations to 2^{10} grid points yields a very noisy distribution (Figure 1a, grey). Application of the spectral filter (e.g. the smoothing step) attenuates the high frequency noise, revealing a smoothed normal distribution that agrees well with the exact distribution (Figure 1a, black).

To test the accuracy of log likelihood estimation, which is critical in MCMC applications, a fixed set of $N_d = 1,000$ "observations" from the known normal are drawn as a synthetic data set. Then, 100 independent reconstructions of this normal are performed, with the resulting approximate log likelihood computed. Results show the mean error in this example is 0.3% with a maximum error of 0.8%.

This example also demonstrates the critical effect of the bandwidth parameter (h) on the likelihood construction (Figure 1b). Recall that small bandwidths produce a less biased but higher variance estimate while larger bandwidths produce lower variance but higher bias. These results confirm this tradeoff and indicate the source of the increased bias at higher bandwidths. Specifically, with larger bandwidths, the peak of the distribution is attenuated and the tails are overestimated, due to over-smoothing. Essentially mass from the peak of the PDF is transferred to the tail in the smoothing process. For this reason, the bandwidth must be chosen carefully so that it is small enough to account for the most refined feature of the model / data but still large enough to pro-

duce a reliable estimate. When likelihood distributions are nearly normal, automated bandwidth determination methods can choose nearly optimal values (Silverman, 1986), however these automated methods can lead to poor results when distributions are more complicated (multi-modal for example, see the next example).

2.2. npABC: Incorporating the KDE into ABC

The above sections describe a manner of approximating the likelihood $L(X|\theta)$. This however is only an approximation and it is important to ask the question, how do inevitable errors influence the MCMC procedure and posterior estimation. In this section, I will discuss critical points that must be considered when embedding KDE in a Bayesian MCMC framework. The primary results of this section are as follows. 1) Small likelihood estimation errors will inevitably propagate into small posterior errors. While this will in many cases have little effect on parameter estimation, these small errors can have dramatic effects on hypothesis testing and model comparison through the use of AIC, BIC, or DIC. 2) Variance in the KDE approximation influences Metropolis-Hastings acceptance probabilities in a manner that substantially degrades MCMC chain mixing. While a rigorous characterization of these points in a general setting is likely very difficult and beyond the scope of this article, I will first outline the theoretical reasoning behind each and subsequently demonstrate them through simple examples.

2.2.1. The influence of KDE on likelihood estimation

The likelihood $L(X|\theta)$ and choice of priors fully determine the posterior distribution for a model. In the context here however, we only have access to the approximate likelihood obtained as

$$\hat{L}(X|\theta) = \prod_{i=1}^{N_d} \hat{L}_i(\theta), \tag{2.7}$$

which is itself a stochastic quantity. Define the estimation error for the likelihood of observation i as

$$\epsilon_i = \hat{L}_i - L_i, \tag{2.8}$$

where for brevity, the dependence on θ has been omitted. The following relation then connects the approximate and true likelihood

$$\hat{L} := \prod_{i=1}^{N_d} \hat{L}_i = L \prod_{i=1}^{N_d} \left(1 + \frac{\epsilon_i}{L_i}\right). \tag{2.9}$$

From Equ. (2.2), we know that $1 + \epsilon_i/L_i \sim N(1 + \mu_i, \sigma_i)$ where

$$\mu_i = \frac{h^2}{2} M_2(K) \frac{L_i''}{L_i}, \qquad \sigma_i^2 = \frac{\|K\|_2}{N_s h}, \tag{2.10}$$

and $L_i^{''} = L^{''}(x_i|\theta)$. Using basic facts about normal distributions, we know that the product of independent normals is again normal so that

$$\frac{\hat{L}}{L} \sim N(\mu_{1...N_d}, \sigma_{1...N_d}), \tag{2.11a}$$

where

$$\mu_{1...N_d} = 1 + < \mu_i >, \qquad \sigma^2_{1...N_d} = \frac{\|K\|_2}{N_d N_s h}, \tag{2.11b}$$

and $< \mu_i >$ is the mean of the set $\{\mu_i\}$.

Unfortunately these quantities are not rigorous quantitative estimates of the mean and variance of the likelihood ratio and cannot be used to make post hoc error estimates. A critical assumption in this derivation was that $\{\epsilon_i\}$ are uncorrelated, which is not the case here since the density estimation of two nearby points will pool information from common samples. Nonetheless, these expressions provide valuable insights into the scaling behavior of the bulk mean and variance. In particular, the same bias variance tradeoff we saw in the estimate of an individual likelihood appears in the estimate of the full likelihood. So again, a tradeoff must be made between accuracy and precision of the estimate. Similarly, increasing the number of samples N_s drawn for estimation does not improve accuracy, but instead decreases the estimation variance and improves precision.

2.2.2. Influence of the KDE on model comparison statistics

Most common measures used for model comparison (except Bayes Factors) utilize values of the log likelihood $LL(X|\theta)$ (AIC, BIC, DIC). Thus, likelihood estimation errors will influence these model comparison statistics. To see this, note that

$$\hat{LL}(X|\theta) - LL(X|\theta) = \sum_{i=1}^{N_d} \hat{LL}(x_i|\theta) - LL(x_i|\theta) \approx \sum_{i=1}^{N_d} \frac{\epsilon_i}{L_i}, \tag{2.12}$$

where ϵ_i, L_i are as above and the approximation results from a Taylor expansion of the log function near 1 (assuming ϵ_i/L_i is small). We thus see that small relative errors in the approximate likelihood of each individual observation translate directly into small relative errors in the log likelihood.

While these errors are small in a relative sense, they raise a substantial problem from a model comparison standpoint. Model comparison tests will often use differences in these statistics of as little as $\Delta DIC = 10$ to conclude evidence for or against a model. However, log likelihoods and in turn DIC values are often on the order of $100 - 1000$. So these small relative errors in LL can easily be larger than the differential commonly taken as "significant". For this reason, care should be taken when using these measures for model comparison in this or any context where approximations are used. Examples in subsequent sections will further elucidate this issue.

Also note that, in contrast to estimates in the previous section, this estimate provides a quantitative approximation of the estimation bias being made. In particular, from Equ. (2.12) it is direct to show that

$$E\left(\hat{LL}(X|\theta) - LL(X|\theta)\right) \approx \sum_{i=1}^{N_d} \frac{\mu_i}{L_i} = \frac{h^2}{2} M_2(K) \sum_{i=1}^{N_d} \frac{L_i''}{L_i}, \tag{2.13}$$

which can be used for post hoc estimates of the expected error. Unfortunately an estimate of the underlying variance cannot be obtained, again because $\{\epsilon_i\}$ are not independent. However, if such an estimate is required, both the bias and variance of this estimate for any particular parameter set (the mean of the posterior for example) can easily be assessed through repeated simulation.

2.2.3. Influence of the KDE on rejection rates

A critical quantity in any MCMC procedure is the Metropolis-Hastings probability (α^n) of accepting a parameter set θ^n at the n^{th} chain iteration, defined as

$$\alpha^n = \frac{L(X|\theta^n)\pi(\theta^n)}{L(X|\theta^{n-1})\pi(\theta^{n-1})}. \tag{2.14}$$

Given this procedure provides only an estimate of $L(X|\theta^n)$, this quantity will be stochastic as well. While a simple description of the density for this quantity is not available, some intuition into the influence of the KDE on it is possible. Define $r^n = (\hat{L}^n - L^n)/L^n$ to be the relative sample error between the exact and approximated likelihood values. Note this quantity is distinct from ϵ_i, which is the absolute error in the approximate likelihood of observation i, where this is the relative error in the full approximate likelihood. Then it is direct to show that

$$\frac{\hat{\alpha}^n}{\alpha^n} = \frac{1 + r^n}{1 + r^{n-1}}, \tag{2.15}$$

where $\hat{\alpha}$ is obtained by substituting the likelihood estimate for the true value.

This raises the practical problem for MCMC efficiency. Suppose $\hat{L}^n$ is an overly optimistic estimate of the likelihood of L^n so that $r^n > 0$. This over estimate will increase the probability of accepting this particular chain iteration. However, since this over estimate enters into the acceptance probability of subsequent chain iterations, it will reduce the acceptance probability of every subsequent iteration of that chain until a new parameter set is accepted. If r^n is fairly near 0, this will likely only have a marginal effect on the time spent by that chain at the current state. However, if it departs significantly from 0, a significant number of chain iterations will be required to displace it, causing chains to stagnate and impairing exploration of parameter space.

This issue is exacerbated by the fact that the next acceptance in that chain is likely to result from yet another over estimation. Generally speaking, this process will lead to a net increase in r^n as n increases due to acceptance being influenced by the variance. Underestimates will rarely be accepted and quickly discarded, while overestimates are more likely to be accepted and rarely discarded. Performance will thus be further degraded and posterior estimates might become skewed. This issue will be demonstrated through practical examples in subsequent sections and an augmentation of the standard MCMC to correct this deficiency will be discussed.

3. Results

The capabilities of this method will next be demonstrated through three examples of increasing complexity. In the first, a bimodal posterior will be fit to demonstrate the substantial effects the bandwidth parameter h can have on accuracy. Second, a canonical response time model in decision

making will be fit using this method. Finally, a third example will be presented where standard ABC methods based on summary statistics are insufficient for posterior estimation.

3.1. Example 2: Fitting a mixture of Gaussians distribution

In this first example, the npABC algorithm is used to estimate the posterior of a mixture model

$$X \sim (1-p)N(\mu_1, \sigma) + pN(\mu_2, \sigma), \tag{3.1}$$

where N indicates the normal distribution and p is a weighting parameter indicating the probability that an observation is derived from the normal centred at μ_2. Mixture models are common in a number of applications where transitioning between discrete strategies might occur. For example participants might transition between actively participating in a task and simply guessing or may follow instructions in some instances while disregarding them in others (Cassey et al., 2014; Ollman, 1966; Yellott Jr, 1967, 1971; Yantis et al., 1991; Vandekerckhove and Tuerlinckx, 2007).

We begin by creating a data set with $N_d = 1,000$ simulated observations drawn from this distribution with $p = 0.6, \mu_1 = -6, \mu_2 = 4, \sigma = 1$. Next, the following prior distributions for the parameters are prescribed

$$p \sim U(0,1), \qquad \mu_1 \sim U(-10,0), \qquad \mu_2, \sigma \sim U(0,10), \tag{3.2}$$

where $U(a,b)$ indicates the uniform distribution on the interval $[a,b]$. For a simple model such as this, any standard MCMC procedure should be sufficient. Since subsequent examples require more sophisticated techniques though, a differential evolution MCMC procedure (DE-MCMC) (Ter Braak, 2006; Storn and Price, 1997; Turner et al., 2013) is used for consistency. For all simulations of this model, 15 chains are propagated for 500 burn in iterations followed by 2000 recorded iterations. All that is left now is to specify the density estimation parameters N_s and h. Rather than specify a single set of KDE parameters, different combinations of N_s and h are used to determine the influence of these parameters on results.

3.1.1. Example 2: Results

As discussed previously, h mediates a bias variance tradeoff. To determine how this parameter effects posteriors, the model is fit to data for different values of h. Figure 2a shows the posterior distribution for (μ_1, μ_2) for two values of h, where Silv indicates the value derived by "Silverman's rule of thumb" (Silverman, 1986)

$$h = 1.06\bar{\sigma}N_s^{-0.2}. \tag{3.3}$$

Here, $\bar{\sigma}$ is the sample variance of the data, and for $N_s = 10,000$ is $h = 0.67$. These results show that posterior estimates are visually identical for the two values. This however is misleading. To investigate the influence of h further, 1) the quality of posterior model fit to the data and 2) the computed log likelihood were determined by comparing to the analytic solution, Figure 2b.

In this figure, the mean parameter values from the posterior for each value of h were used to construct the likelihood. These values are effectively identical, differing by $< 0.1\%$ between the two simulations. Yet, when the model's PDF is constructed from these parameter sets, we see significant deviation of the $h = Silv$ case from the analytic solution while $h = 0.2$ faithfully

captures the analytic solution. This results from over-smoothing of the density in the $h = Silv$ case, since the underlying distribution is bimodal. Essentially, the smoothing kernel is too broad, so when it is applied to smooth the simulated PDF, the peak of the distribution is attenuated leading to fatter tails.

These results also show the choice of h influences log likelihood estimates. While the difference between the computed and actual log likelihood is small in a relative sense ($\sim 2\%$ for $h = Silv$), it is still relatively large in an absolute sense (~ 38). This raises a substantial problem for hypothesis testing and model comparison based on AIC, BIC, and DIC, all of which rely on log likelihoods. It is important to remember however that this is a problem with ABC in general since the posterior being estimated is always only an approximation of the actual posterior.

To further determine the extent to which small posterior errors influence these statistics, DIC was computed for different pairings of (N_s, h). For each, 100 independent posterior estimation simulations were performed (all on the same synthetic data set), and the mean and standard deviation of the DIC computed from those simulations is shown, Figure 2c. While even the worst model fit ($N_s = 5,000, h = 0.8$) leads to a relative DIC error of $< 2\%$, the resulting absolute DIC error is > 200. Given that DIC differences between different models of as little as $\Delta DIC = 10$ is often taken as strong evidence for a particular model, clearly these small relative errors can overwhelm standard hypothesis tests.

3.1.2. The influence of the bias-variance tradeoff

These results (Figure 2c) also further illustrate the influence of h (and the bias variance tradeoff it mediates) on results. Broadly speaking, as h increases, error in the DIC increases as well, largely independent of N_s which has no influence on estimation bias. Further, as h decreases, the DIC error decreases while the DIC variance increases. This is consistent with the fact that increasing h reduces estimation variance but increases estimation bias. Thus the MCMC procedure does not abrogate this tradeoff and h should be chosen carefully. Unfortunately there is no universal way of choosing this value and for example, the Silverman value is usually too large for multimodal distributions and too small for heavy tailed distributions. Thus some trial and error is required for choosing this bandwidth.

There is one last point to consider here. Recall from the previous section that variability in the likelihood estimation is hypothesized to impair MCMC performance by causing chains to get stuck when a likelihood is significantly overestimated. To determine the extent of this problem, proposal acceptance rates as a function of N_s and h are computed, Figure 2d. Again, 100 independent simulations of the posterior are used for each KDE parameter set. Results show a clear decrease in the acceptance rate as the number of samples N_s decreases, consistent with the supposition that likelihood variability leads to poor MCMC performance. One way to ameliorate this issue is to simply increase the number of samples used for estimation. In many cases however this will not be possible for performance reasons. In the next example, an alternative correction that ameliorates this performance issue is presented.

3.2. Example 3: Fitting the Linear Ballistic Accumulator (LBA)

In this example, the canonical Linear Ballistic Accumulator (LBA) model (Brown and Heathcote, 2008) is considered as an example of a large of class of evidence accumulation models in

decision making literature. A number of accumulator models, including Ratcliff's drift diffusion model (Ratcliff, 1978; Ratcliff and Rouder, 1998), the leaky competing accumulator (Usher and McClelland, 2001), the ballistic accumulator (Brown and Heathcote, 2005), and decision field theory (Busemeyer and Townsend, 1993), have been developed over the years to account for different aspects of decision making. What differentiates the LBA from the other models, is that evidence accumulation for different choice alternatives are independent, linear, and deterministic. This simplicity allows for a closed form solution for the simplest settings. However as will be discussed in the next example, even simple variations of this model make it impossible to obtain a tractable likelihood function. This example will thus be used to demonstrate the potential power of these methods in response time modelling. A brief description of this model will be provided and the interested reader can find further details in (Brown and Heathcote, 2008).

The basic assumptions of the LBA are that following the presentation of information, evidence for each of a set of choice alternatives accumulates linearly and deterministically until an evidence threshold b is reached. The rate of evidence accumulation for choice alternative i, given by v_i, is assumed to be fixed within a trial (this is the deterministic assumption) but to vary among trials. This rate is sampled from an underlying normal distribution $v_i \sim N(\mu_i, \sigma)$, while the start point $x_{0,i}$ for the i^{th} accumulator, which is also assumed to vary across trials, is uniformly distributed $x_{0,i} \sim U(0, A)$. Additionally, a non-decision time τ_{er} is included to account for encoding and motor response delays. This simplest LBA variant is thus fully parameterized by the parameters b, A, σ, τ_{er} and the collection of mean drift rates $\{\mu_i\}$. The likelihood $L(c_i, \tau_i|\theta)$ of a option c_i being chosen at time τ_i can then be described by an analytic function (Brown and Heathcote, 2005).

This method will be used to perform parameter recovery, as was done previously in (Turner and Sederberg, 2014), and assess the properties of this method in a cognitive modelling context. To begin, a synthetic data set for a two choice experiment, consisting of $N_d = 1,000$ observations, is created by simulating N_d trials with $A = 1.6, b = 2.7, \mu_1 = 3.4, \mu_2 = 2.1, \tau_{er} = 0.1$. The canonical assumption $\sigma = 1$ is further made to identify the model. To place the model in a Bayesian framework, the following priors on the parameters are further prescribed

$$b, A \sim U(0, 10), \qquad \mu_1, \mu_2 \sim U(-10, 10), \qquad \tau_{er} \sim U(0, 1). \tag{3.4}$$

The same differential evolution MCMC procedure used previously is used here as well, again with 15 chains, a 500 iteration burn in, and 2000 recorded chain iterations.

3.2.1. Example 3: Results

The posterior of this model is fit both analytically and using this FFT based npABC procedure. Again, for each combination of (N_s, h), 100 independent fits are performed to determine how estimation variability influences various quantities, Figure 3. For all but the largest value of h, the posterior estimated by the two methods was visually indistinguishable, and so they are not shown. Panel a shows the quality of fit for the two methods, analytic MCMC and npABC (using $N_s = 10,000$ and $h = Silv$ where $Silv$ again indicates $h = 0.028$ was chosen according to Silverman's rule of thumb). In each case, the mean value of the parameters from the associated posterior were determined and the PDF was constructed from those values. The resulting PDF's are virtually indistinguishable and in this case, the log likelihoods are very close.

Again however, we see that small errors propagate into the DIC measure, Figure 3b. For all but the worst case fits ($h = 0.07$), the relative DIC error is $\sim 1 - 2\%$. This translates into absolute errors of $\Delta DIC \sim 10 - 20$, which will again have a strong influence on model comparison. We again see that N_s has effectively no influence on the DIC error. This along with results from the previous example confirms that errors cannot be reduced by increasing the number of samples used in the likelihood reconstruction. Only reductions in h can improve posterior estimates.

We also again see a problem with acceptance rates, Figure 3c, which generally decrease when either N_s or h decreases (black dots). Further recall that both of these parameter changes lead to increased likelihood estimation variance. These results are thus again consistent with the fact that increased estimation variance reduces acceptance rates and hence MCMC efficiency. In particular, the acceptance rate for this procedure with $N_s = 10,000, h = Silv$ (which are the same estimation parameters used in (Turner and Sederberg, 2014)) is only $\sim 6\%$. Fortunately, this can be ameliorated to a significant extent with a minor augmentation of the MCMC procedure and a little more computation.

3.2.2. Resampled MCMC for npABC

The central problem that leads to chain stagnation and poor performance is that the likelihood of a particular parameter set θ^n can, on a rare occasion, be significantly overestimated. While this will be rare, it will substantially degrade performance. This overestimation will increase the chance of that parameter set being accepted. Subsequently, acceptance probabilities will significantly favor keeping that state on further MCMC chain iterations. A simple way to "unstick" chains that become stagnant for this methodological reason is to simply resample that likelihood value frequently. This will of course increase computational cost, but it will ensure that no chain becomes stuck due to mis-estimation of the likelihood. This will have the additional benefit of reducing contamination of the posterior by oversampling less likely parameters. From here on, this augmented MCMC will be referred to as a "resampled MCMC".

This adjustment was added so that the likelihood of every chain is resampled every three chain iterations, Figure 3c (grey dots). That is for each of the N_c chains, the likelihood of the current state of that chain is resampled every third MCMC iteration, independent of the history of the chain. Why was this frequency chosen? It is well established that the theoretical acceptance rate for this form of MCMC is $\sim 25\%$ for five or more parameters. The resampling rate was chosen to be faster than the theoretical frequency of chain movement. Results show this augmentation substantially improves acceptance rates, increasing them to $\sim 17 - 18\%$. Furthermore, the resulting acceptance rate is only weakly dependent on N_s and h, suggesting the effects of variance on performance have been removed. The exception to this is that for large $h = 0.07$, there is a substantial drop in performance. It is unclear what is causing this, but this value is well above any reasonably choice for h.

3.2.3. A note on performance

A brief note is in order regarding practical performance of this algorithm. To assess performance, both the standard MCMC with the analytic LBA likelihood and the npABC algorithm with the resampled MCMC were timed for a single posterior fit. To obtain consistent timings, all but one active computational core on a Mac Pro computer were turned off. Results show it takes ~ 137

seconds for the npABC algorithm to complete while it takes ~ 315 seconds for the standard MCMC to complete. Thus the npABC with resampled MCMC and FFT based density estimation is more than twice as fast as the standard MCMC using the analytic likelihood.

Profiling of the codes shows the primary reason for this is that computation of the cumulative density function (CDF) of the normal distribution, which is required to evaluate the LBA likelihood, is cumbersome. While we have a tendency consider functions such as the normal CDF to be "analytic", the term analytic has little meaning in computational settings. In fact, a complex numerical procedure is required to approximate the normal CDF, which in this application is slower than Monte Carlo sampling of the full likelihood. To be clear, this is not meant to advocate for abandoning the standard methods, since accuracy should always be favored over efficiency when reasonable. Rather, this is intended to demonstrate that for many types of models, with proper coding techniques, computational cost can be very reasonable.

3.3. Example 4: Fitting the piecewise Linear Ballistic Accumulator (pLBA)

In this final example, a piecewise LBA type model will be considered. The canonical LBA describes decision process that might be described as stationary in the sense that the information available to the decision maker remains the same over time. In many cases however, information may change during the course of the decision process. In (Huk and Shadlen, 2005; Kiani et al., 2008; Thura et al., 2012; Tsetsos et al., 2012; Winkel et al., 2014) for example, a random dot motion paradigm where the direction motion of dots change at discrete times during the course of individual trials was utilized. In these cases, the information itself is non-stationary and one would expect the decision process to change in response to the new information. To account for this, a piecewise variant of the standard LBA was first presented in (Holmes et al., 2014). This non-stationary model, which has no tractable closed form likelihood function, will be used to demonstrate this method in a context where existing methods are insufficient.

Briefly, to account for the changes in information, this model makes two assumptions on top of those of the standard LBA. First, that changes in information influence the rate of evidence accumulation so that the rates prior to the change are $v_i \sim N(\mu_{vi}, \sigma)$ while those after the change are $w_i \sim N(\mu_{wi}, \sigma)$. This model is referred to as "piecewise LBA" since evidence accumulation is linear and deterministic on each of two segments corresponding to the two separate pieces of information. Second, there is some delay (t_{delay}) between onset of new information and its incorporation into the decision process, which is assumed fixed across trials. In the context of a two choice decision, after setting $\sigma = 1$, the model is fully described by the eight parameters $A, b, \mu_{v1}, \mu_{v2}, \mu_{w1}, \mu_{w2}, t_{er}$, and t_{delay}. See Holmes et al. (2014) for further details. To begin, a data set consisting of $N_d = 1,000$ observations is created, assuming $A = 1.6, b = 2.7, \mu_{v1} = 3.4, \mu_{v2} = 2.5, \mu_{w1} = 1.5, \mu_{w2} = 3.6, t_{er} = 0.1, t_{delay} = 0.3$.

While this model is simple to describe, it does not have an analytic description. Nonetheless, the methods described here can be applied to this model without too much augmentation. In fact, the density estimation procedure itself is identical to that used in the previous examples. The only changes that are required for this application are entirely in the resampled MCMC procedure itself. In this example, KDE parameters $h = 0.02, N_s = 10,000$ will be used with no other changes to the likelihood approximation. A slightly more complex MCMC procedure must however be used. The DE-MCMC procedure will again be used, this time with 24 chains. However, since the size of the

parameter space has increased (8 parameters), a blocked variant must be used to improve sampling performance. Here, as in Holmes et al. (2014), the parameters will be grouped into two blocks: $(A, b, \mu_{v1}, \mu_{v2}, t_{er})$ which describe the accumulation process prior to the change of information, and $(\mu_{w1}, \mu_{w2}, t_{delay})$ which describe the process after the change.

3.3.1. Example 4: Results

Estimated posteriors for this piecewise LBA model are shown in Figure 4a for all model parameters. At first glance, it may appear the method has performed poorly since the posteriors are quite broad. This is not however the case. First, the mean parameter set from these posteriors provides a good fit to the data, Figure 4c. Second, it is well known that the LBA model exhibits significant parameter correlations, which commonly lead to poorly localized posteriors. In biological and physics literature, this is commonly referred to as a "sloppy model" (Gutenkunst et al., 2007; Apgar et al., 2010) since the likelihood is nearly unchanged over a wide range of parameters, Figure 5. To confirm parameter correlations are the source of this posterior spread, principal component analysis (PCA) was performed on the saved MCMC chain data. This reveals that the first and second principal components account for $\sim 92\%$ and 5% of the variability in the posterior respectively. Furthermore, the eigenvector of the principal component shows this correlated direction involves only the pre-switch model parameter A, b, μ_{v1}, μ_{v2}.

To determine how the log likelihood varies along this principal component, the mean parameter set $\vec{\mu}$ and eigenvector for the principal component $\vec{v_1}$ were extracted and the log likelihood was computed at values $\vec{p}$ along the affine linear subspace

$$\vec{p} = \mu + k\vec{v_1}, \tag{3.5}$$

see Figure 5 for a schematic depiction. For even a relatively large displacement ($k = 4$, Figure 4d), the log likelihood and quality of fit change only marginally. Furthermore, it is simple to check that the exact parameter set used to construct the data lies nearly on this subspace, and that it provides only a marginally better fit, Figure 4e. This supports the supposition that there is a single, strong correlation within the model and that the poor localization of the posterior is intrinsic to the standard LBA model.

Since the model degeneracy (i.e. correlation) involves only a one dimensional subspace of the 8 dimensional parameter space, fixing a single parameter in that subspace should in theory fully localize the posterior. To test this, the threshold parameter was fixed to the value $b = 2.7$, which was originally chosen to produce the data. The same procedure was carried out to sample the posterior (Figure 4c), and results indeed show it becomes substantially more localized. Furthermore, the mean parameter set from the constrained model is almost identical to the exact parameters used to construct the data. This confirms the spread in the posterior is a result of the strong correlation.

3.3.2. npABC and sloppy models

These observations do however raise an important issue. Recall from the previous two examples that this approximation procedure yields small errors in the log likelihood of on the order of $\sim 1 - 2\%$. While this might not seem too large, it can have a substantial effect on estimation of sloppy models such as this. The issue is that along this correlated parameter dimension, the

variation of the log likelihood is the same size or slightly larger than the log likelihood estimation variance. The npABC will thus explore the corridor along this correlated dimension more so than an MCMC with an analytic likelihood would. This variance can of course be reduced with extra computational power, but in practice this will not be practical.

If the goal is to understand the behavior of the model and its capacity to account for observations, this may not be an issue. However, if the context being considered requires one to extract a single parameter set, more must be done. Strategies for dealing with sloppiness in models have been discussed extensively in other literature (Gutenkunst et al., 2007; Apgar et al., 2010), but such exposition is beyond the scope of this article. It is important to reiterate though that the sloppiness of posterior estimates here is more a reflection of an underlying model property that prevents accurate estimation.

4. A procedural overview of npABC for the practitioner

The previous sections outline many points that must be considered when using this method. Here, a procedural overview building on these results is provided for the interested practitioner. Familiarity with MCMC methods is assumed and only the details that relate to non-parametric component of this method is provided. It is impossible to list all details, however I again note that MATLAB codes have been supplied to aid the interested practitioner fill in the technical details.

0) Choose the number of samples to be used in the estimation process (N_s) and the kernel bandwidth (h). A minimum of $N_s = 10,000$ should generally be used. Choosing h will require trial and error, but Silverman's rule of thumb (Silverman, 1986) provides a good starting point. As a general rule however, err on the side of smaller h since this will reduce estimation bias (at the expense of performance).

1) Loop over chains.

   a) Generate a proposal θ^n. In the applications here this was done using DE-MCMC (Ter Braak, 2006; Turner et al., 2013), but any MCMC procedure can be used.

   b) Compute $\hat{LL}(X|\theta^n)$ using the KDE.

      i) Generate N_s samples from the model.

      ii) Create a discrete representation of the likelihood by binning those samples into 2^n ($n > 8$) equally spaced bins with centers $z_0, ..., z_l$. Set these bin centers so that $z_0 < \min(X) - 3h$ and $z_l > \max(X) + 3h$. This pads both sides of the histogram with zeros so the FFT is more accurate.

      iii) Apply a FFT to map the data into the spectral domain.

      iv) Apply the Gaussian smoothing filter. This is essentially the convolution step in the spectral domain.

      v) Map the filtered signal back to the data space, producing a likelihood function $\hat{L}(z_i|\theta^n)$ on the regularly spaced grid.

      vi) Interpolate this likelihood on the grid to the observation values, $\hat{L}(z_i|\theta^n) \rightarrow \hat{L}(x_i|\theta^n)$, using linear interpolation. Do not use cubic splines or anything higher order than linear as they can induce negative values in the tail of the distribution.

vii) Replace any zero values of $\hat{L}(x_i|\theta^n)$ with a minimum value, say $L_{min} = 1/(10 * N_s)$.

viii) Compute the approximate log likelihood as

$$\hat{LL}(X|\theta^n) = \sum_{i=1}^{N_d} \log\left(\hat{L}(x_i|\theta^n)\right). \tag{4.1}$$

c) Compute the acceptance probability $\hat{\alpha}^n$ and accept or reject the proposal.

2) Resample the log likelihood of any previous chain. The algorithms here resample each chain every third MCMC iteration, though more efficient schemes are certainly possible. For example, the length of time a chain remains stuck can be recorded and used to determine when to resample.

Steps 1,2 define a single update of every chain in the re-sampled MCMC procedure. Simply iterate these steps the desired number of times and apply burn-in rules. Note that steps such as computation of the prior, its incorporation into the acceptance probability, and specifically how to call the FFT have been neglected for brevity. Details can be found in the codes associated with these examples.

5. Discussion

This article presents a non-parametric approximate Bayesian computation (npABC) algorithm. This method, which combines non-parametric statistical methods with Bayesian inference techniques, is an extensible methodology for performing Bayesian posterior estimation. The purpose of this article is to elaborate this methodology in detail, discuss its pitfalls, improve its efficiency, and make it accessible a broader audience of end users.

A great many algorithms, ranging from Markov Chain Monte Carlo (MCMC) (Gelman et al., 2003; Robert and Casella, 2004) to particle filtering methods (Cappé et al., 2004; Del Moral et al., 2006), have been developed for the purpose of posterior estimation in contexts where Bayes' formula cannot be computed directly. These methods however typically require a closed form description of the model's likelihood. More recently, numerous approximate Bayesian computation (ABC) methods have extended these to likelihood free contexts. These methods however require the user to prescribe a set of summery statistics that describe the model / data. Unfortunately, these summary statistics are rarely sufficient to describe the model, and so the model that is fit is different from the one intended, by a substantial margin in some cases. More recently, non-parametric methods have been incorporated into ABC to circumvent this requirement (Turner and Sederberg, 2014).

Both npABC and ABC are similar in that they seek to determine the likelihood or plausibility of a given parameter set by first simulating a large number of model realizations, and second comparing those model realizations to the data. The central feature that differentiates npABC from other ABC methods however is that for each parameter set under consideration, a non-parametric approximation of the underlying likelihood is constructed, as opposed to some surrogate based on summary statistics. This provides two distinct benefits. First, the user does not have to make a possibly erroneous assumption about the form of the underlying model distribution. Second, this

method more fully utilizes the data since it does not compress it into a small number of summary statistics.

The key step in this method of course is to construct an approximation of the underlying models density function $L(x|\theta)$. This is accomplished using a kernel density estimation (KDE) technique (Silverman, 1986), which is a method of directly computing an approximate of the likelihood of any particular observation $L(x_i|\theta)$ from a collection of simulated model observations. The KDE can thus be used to directly compute an approximation ($\hat{LL}$) to the models log likelihood $LL(X|\theta)$, which is the key piece of information needed for MCMC sampling. Thus a third practical benefit of this method, in addition to theoretical benefits mentioned above, is that this KDE can be directly integrated into standard MCMC techniques, since the likelihood itself is being assessed rather than some surrogate.

Results here and elsewhere (Holmes et al., 2014; Turner and Sederberg, 2014) show this methodology is highly efficient and performs well. There are however a number of implementation details that must be considered. First, the standard KDE procedure is highly inefficient and can itself become a computational bottleneck. For this reason, a highly efficient implementation of KDE, which utilizes only standard and highly optimized fast Fourier transform and linear interpolation subroutines, is presented. In the applications discussed here, this implementation improved computation times by a factor of 10 or more. Second, while this method can be directly plugged into standard MCMC procedures, doing so can lead to inefficiencies. This stems from the fact that the KDE is a statistical estimator of the underlying likelihood and as a result has an intrinsic variance. To overcome this issue, a "resampled MCMC" procedure is proposed, which accounts for the variability in this estimator and substantially improves performance.

While these investigations demonstrate the efficacy and efficiency of this methodology, like any approximate method, it does come with drawbacks that must be kept in mind. First, the KDE likelihood estimator is inherently biased. In applications discussed here, this bias is quite small, being on the order of 1% or less. Unfortunately, these very small errors can have a profound effect on model comparison and hypothesis testing. The essential problem is that standard quantities such as AIC, BIC, or DIC are inherently flawed as they are absolute measures of model comparison rather than relative measures. It is commonly accepted that ΔDIC of 10 is interpreted as "significant" evidence for or against a model. However, if DIC measures are on the order of $1,000$ (which is common), a difference between two models of 1% would be considered "significant", which is rarely sensible. Since KDE approximation errors are on the order of $1-2\%$, those errors will often overwhelm these model comparison statistics since the distinguishing difference is within the methods margin of error.

A second issue is that this method can have difficulties with models containing very strong parameter correlations, which in other fields are commonly referred to as "sloppy models" (Gutenkunst et al., 2007). The essential issue here is that the models with strong correlations are under-determined in the sense that large parameter variations along the correlated dimension can lead to very small changes in log likelihoods. In the final example presented here, varying parameters by a factor of 10 along the correlated dimension leads to a $\sim 1\%$ variation in log likelihood and nearly indistinguishable fits to the models data. Given these model fit variations are within the small margin of error of the KDE approximation, posteriors become broadened. Thus, care must be taken in interpreting the results of this method when such under-determined, highly correlated

models are being considered.

Despite these issues, this method has distinct benefits over existing ABC methods. With standard methods, it is rarely possible to know how good or bad the summary statistics being used are. Using npABC however, the quality of an approximation can be controlled in a predictable way by varying kernel density estimation parameters. Furthermore, since the types of errors being made with npABC are somewhat predictable and quantifiable, their influence on results is also reasonably predictable. Additionally, the efficiency of this methodology is comparable to existing methods, especially with the more efficient KDE implementation presented here. Thus it can be applied in almost any context where ABC methods are currently being or might be used. For these reasons, this method should be added to the toolbox of any researcher performing Bayesian analysis of complex models beyond the reach of existing toolboxes such as JAGS (Plummer, 2003) or WinBUGS (Lunn et al., 2000). The hope is that this article (along with the supporting MATLAB codes) will make this method more accessible to those who could benefit from its use.

Acknowledgements

I would like to thank Joachim Vandekerckhove and Jennifer S. Trueblood for a critical reading of this manuscript.

References

Abelson, R. P., 2008. Statistics as principled argument. Psychology Press.

Albantakis, L., Deco, G., 2009. The encoding of alternatives in multiple-choice decision-making. BMC Neuroscience 10 (Suppl 1), P166.

Apgar, J. F., Witmer, D. K., White, F. M., Tidor, B., 2010. Sloppy models, parameter uncertainty, and the role of experimental design. Mol. BioSyst. 6, 1890–1900.

Brown, S., Heathcote, A., 2005. A ballistic model of choice response time. Psychological review 112 (1), 117.

Brown, S. D., Heathcote, A., 2008. The simplest complete model of choice response time: Linear ballistic accumulation. Cognitive psychology 57 (3), 153–178.

Brunel, N., Wang, X.-J., 2001. Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition. Journal of computational neuroscience 11 (1), 63–85.

Busemeyer, J. R., Townsend, J. T., 1993. Decision field theory: a dynamic-cognitive approach to decision making in an uncertain environment. Psychological review 100 (3), 432–459.

Cappé, O., Guillin, A., Marin, J.-M., Robert, C. P., 2004. Population monte carlo. Journal of Computational and Graphical Statistics 13 (4).

Cassey, P., Heathcote, A., Brown, S. D., 07 2014. Brain and behavior in decision-making. PLoS Comput Biol 10 (7), e1003700.

Csilléry, K., Blum, M. G., Gaggiotti, O. E., François, O., 2010. Approximate bayesian computation (abc) in practice. Trends in ecology & evolution 25 (7), 410–418.

Dantzig, G. B., Thapa, M. N., 1997. Linear Programming 1: 1: Introduction. Vol. 1. Springer.

Del Moral, P., Doucet, A., Jasra, A., 2006. Sequential monte carlo samplers. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68 (3), 411–436.

Epanechnikov, V., 1969. Non-parametric estimation of a multivariate probability density. Theory of Probability and Its Applications 14 (1), 153–158.

Friedrichs, K. O., 1944. The identity of weak and strong extensions of differential operators. Transactions of the American Mathematical Society 55 (1), 132–151.

Gallistel, C., 2009. The importance of proving the null. Psychological review 116 (2), 439.

Gelman, A., Carlin, J. B., Stern, H. S., Rubin, D. B., 2003. Bayesian Data Analysis, 2nd Edition. Chapman and Hall/CRC.

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., Sethna, J. P., 10 2007. Universally sloppy parameter sensitivities in systems biology models. PLoS Comput Biol 3 (10), e189.

Heathcote, A., Brown, S., 2004. Reply to speckman and rouder: A theoretical basis for qml. Psychonomic Bulletin & Review 11 (3), 577–578.

Heathcote, A., Brown, S., Mewhort, D., 2002. Quantile maximum likelihood estimation of response time distributions. Psychonomic Bulletin & Review 9 (2), 394–401.

Holmes, W. R., Trueblood, J. S., Heathcoat, A., 2014. Asymmetric updating and hysteresis in perceptual decision-making with changing information. In review.

Huk, A. C., Shadlen, M. N., 2005. Neural activity in macaque parietal cortex reflects temporal integration of visual motion signals during perceptual decision making. The Journal of neuroscience 25 (45), 10420–10436.

Kiani, R., Hanks, T. D., Shadlen, M. N., 2008. Bounded integration in parietal cortex underlies decisions even when viewing duration is dictated by the environment. The Journal of Neuroscience 28 (12), 3017–3029.

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

Lee, M. D., Wagenmakers, E.-J., 2013. Bayesian cognitive modeling: A practical course. Cambridge University Press.

Lunn, D. J., Thomas, A., Best, N., Spiegelhalter, D., 2000. Winbugs a bayesian modelling framework: concepts, structure, and extensibility. Statistics and computing 10 (4), 325–337.

Ollman, R., 1966. Fast guesses in choice reaction time. Psychonomic Science 6 (4), 155–156.

Plummer, M., 2003. Jags: A program for analysis of Bayesian graphical models using gibbs sampling. In: Proceedings of the 3rd International Workshop on Distributed Statistical Computing.

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

Ratcliff, R., Rouder, J. N., 1998. Modeling response times for two-choice decisions. Psychological Science 9 (5), 347–356.

Ratcliff, R., Tuerlinckx, F., 2002. Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. Psychonomic bulletin & review 9 (3), 438–481.

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

Robert, C. P., Cornuet, J.-M., Marin, J.-M., Pillai, N. S., 2011. Lack of confidence in approximate bayesian computation model choice. Proceedings of the National Academy of Sciences 108 (37), 15112–15117.

Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., Tarantola, S., 2008. Global sensitivity analysis: the primer. John Wiley & Sons.

Silverman, B. W., 1982. Algorithm as 176: Kernel density estimation using the fast fourier transform. Journal of the Royal Statistical Society. Series C (Applied Statistics) 31 (1), pp. 93–99.

Silverman, B. W., 1986. Density estimation for statistics and data analysis. Vol. 26. CRC press.

Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11 (4), 341–359.

Ter Braak, C. J., 2006. A markov chain monte carlo version of the genetic algorithm differential evolution: easy bayesian computing for real parameter spaces. Statistics and Computing 16 (3), 239–249.

Thura, D., Beauregard-Racine, J., Fradet, C.-W., Cisek, P., 2012. Decision making by urgency gating: theory and experimental support. Journal of Neurophysiology 108 (11), 2912–2930.

Tsetsos, K., Gao, J., McClelland, J. L., Usher, M., 2012. Using time-varying evidence to test models of decision dynamics: Bounded diffusion vs. the leaky competing accumulator model. Frontiers in Neuroscience 6.

Turner, B., Sederberg, P., 2014. A generalized, likelihood-free method for posterior estimation. Psychonomic Bulletin and Review 21 (2), 227–250.

Turner, B. M., Sederberg, P. B., Brown, S. D., Steyvers, M., 2013. A method for efficiently sampling from distributions with correlated dimensions. Psychological methods 18 (3), 368–384.

Turner, B. M., Van Zandt, T., 2012. A tutorial on approximate bayesian computation. Journal of Mathematical Psychology 56 (2), 69–85.

Usher, M., McClelland, J. L., 2001. The time course of perceptual choice: the leaky, competing accumulator model. Psychological Review 108 (3), 550–592.

Vandekerckhove, J., Tuerlinckx, F., 2007. Fitting the ratcliff diffusion model to experimental data. Psychonomic Bulletin & Review 14 (6), 1011–1026.

Winkel, J., Keuken, M. C., Van Maanen, L., Wagenmakers, E.-J., Forstmann, B. U., 2014. Early evidence affects later decisions: Why evidence accumulation is required to explain response time data. Psychon Bull Rev 21, 777–784.

Yantis, S., Meyer, D. E., Smith, J. K., 1991. Analyses of multinomial mixture distributions: new tests for stochastic models of cognition and action. Psychological bulletin 110 (2), 350.

Yellott Jr, J. I., 1967. Correction for guessing in choice reaction time. Psychonomic Science 8 (8), 321–322.

Yellott Jr, J. I., 1971. Correction for fast guessing and the speed-accuracy tradeoff in choice reaction time. Journal of Mathematical Psychology 8 (2), 159–199.

Figures

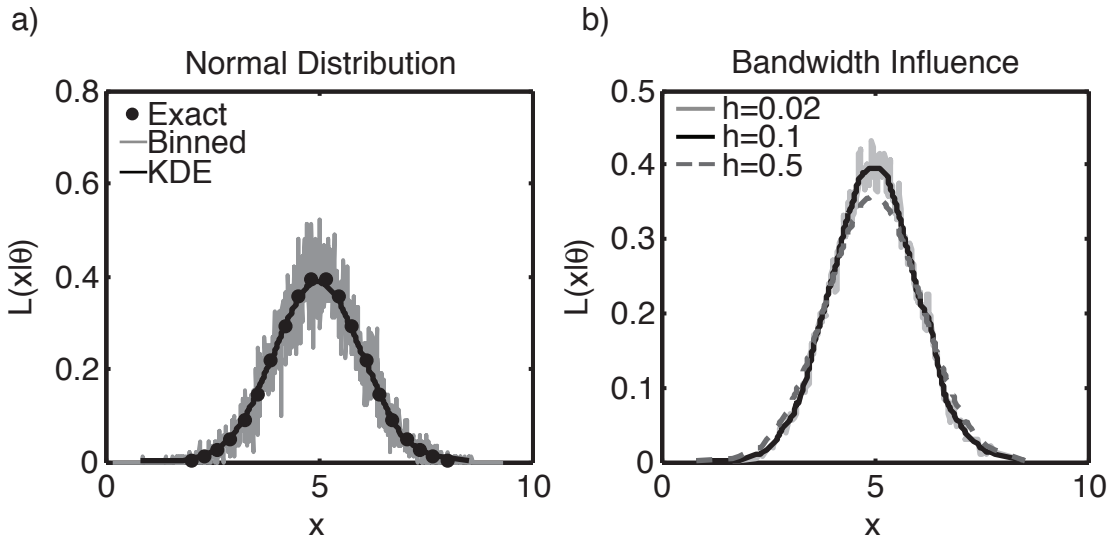

Figure 1: **Reconstructing a Gaussian: Panel a)** Reconstruction of a normal distribution using the FFT based KDE method. Gray lines indicate the noisy density estimate derived from binning $N_s = 10,000$ sampled points into 2^{10} bins and normalizing to produce a density. Black line indicates the smoothed version after convolving in the spectral domain. Circles show the exact likelihood at a few values, indicating agreement between the constructed and analytic likelihood. The bandwidth $h = 0.1$ is used here. **Panel b)** Reconstructed likelihood for three choices of the bandwidth parameter h. The mean and standard deviation parameters used are $\mu = 5, \sigma = 1$.

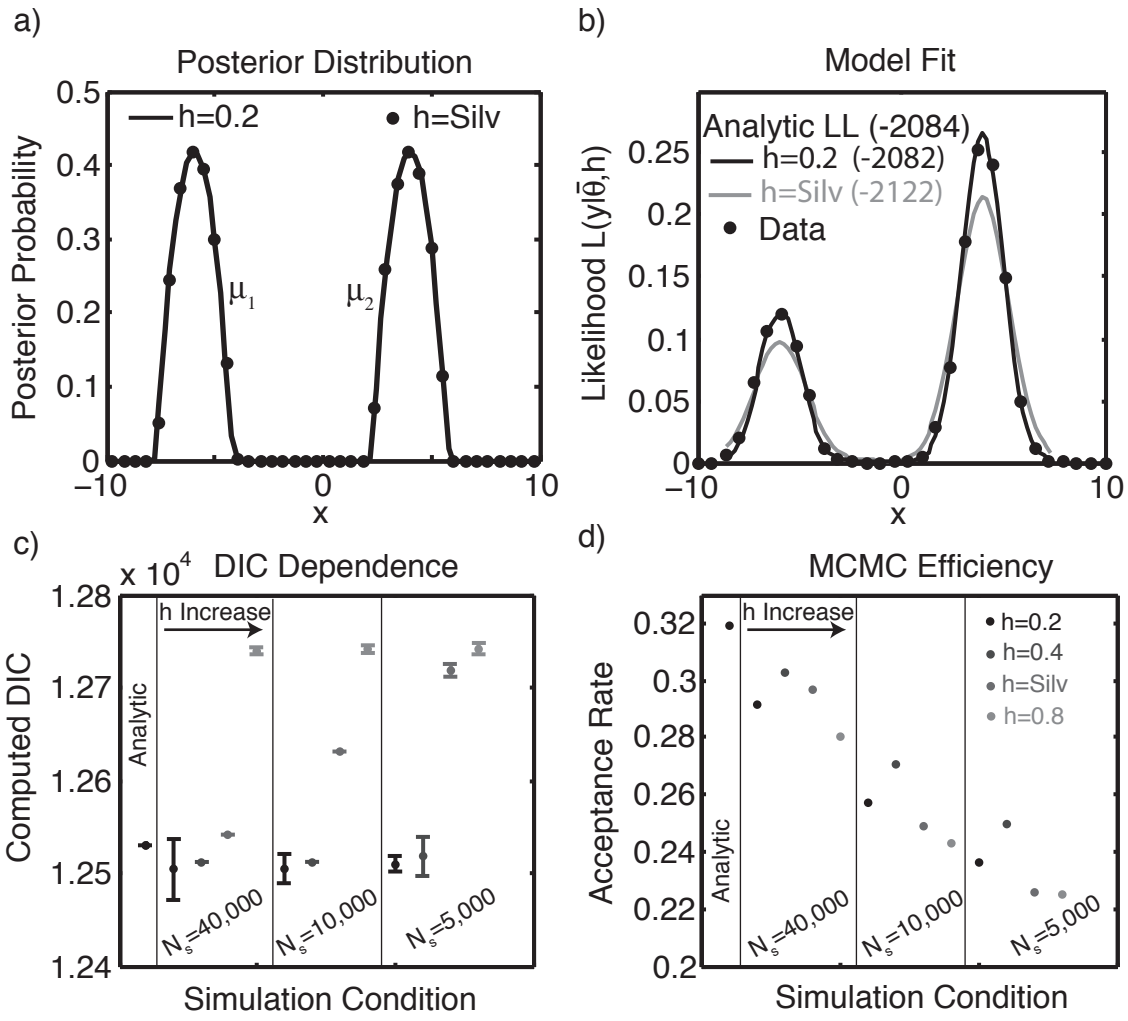

Figure 2: **Fitting a bimodal distribution:** *Panel a)* Posterior distribution obtained using $N_s = 10,000$ and two separate values of $h = 0.2, 0.68$, the latter is computed from Silverman's rule of thumb. Posterior for the two means μ_1, μ_2 are shown. *Panel b)* Quality of model fit. For each value of h (and $N_s = 10,000$). The mean of the posterior for each parameter was computed, which was used to reconstruct the likelihood. In both cases, the approximate likelihood method with the associated h was used to construct the likelihood, but with $N_s = 1,000,000$ to reduce variance. In both cases the log likelihood along with that computed analytically are reported. *Panel c)* Dependence of DIC on N_s and h. DIC was computed by fitting the posterior using the analytic likelihood. Then, 100 fits of the posterior were obtained for each of different combinations of N_s and h. Within each value of N_s, the h values increase from left to right $h = 0.2, 0.4, Silv, 0.8$ where $Silv$ indicates the bandwidth computed from Silverman's rule of thumb. *Panel d)* Acceptance rates as a function of N_s, h. Data from the 100 posterior fits in (c) were used, though variance was so small it is not shown. The reported values of h increase from left to write, as in (c). Note the reduced efficiency with decreased N_s. A correction to the MCMC that alleviates this performance reduction is discussed in Section 3.2.2.

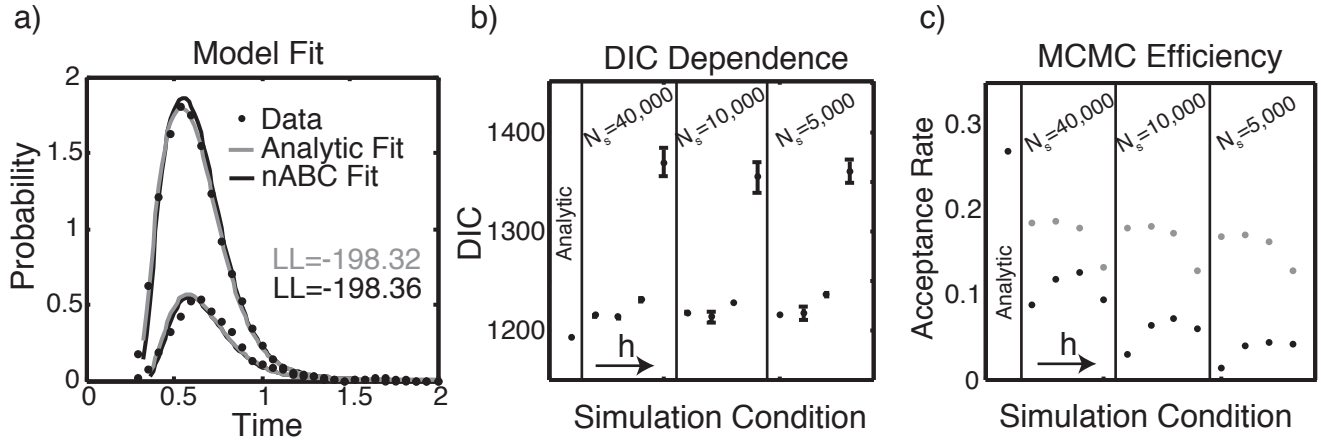

Figure 3: **Fitting the LBA:** *Panel a)* Quality of model fit using the analytic likelihood and approximate likelihood methods (with resampled MCMC). For the npABC fit, the approximate likelihood function was used to reconstruct the likelihood using the same value of h, but $N_s = 1,000,000$ to reduce variance. The analytic LBA likelihood was used for the analytic case. The high and low peaked curves correspond to correct and incorrect choice options respectively. In both cases, the log likelihood is reported. *Panel b)* Dependence of DIC on N_s and h. DIC was computed by fitting the posterior using the analytic likelihood. Then, 100 fits of the posterior were obtained for each of different combinations of N_s and h. Within each value of N_s, the h values increase from left to right $h = 0.01, Silv, 0.04, 0.07$ where $Silv$ indicates the bandwidth computed from Silverman's rule of thumb. Mean and variance of the DIC samples is shown for each. The resampled MCMC is again used here. *Panel c)* Acceptance rates as a function of N_s, h for the standard (black) and resampled MCMC (gray). The same values of h as in panel b, again increasing from left to right. Resampling occurred every third chain iteration for the resampled MCMC.

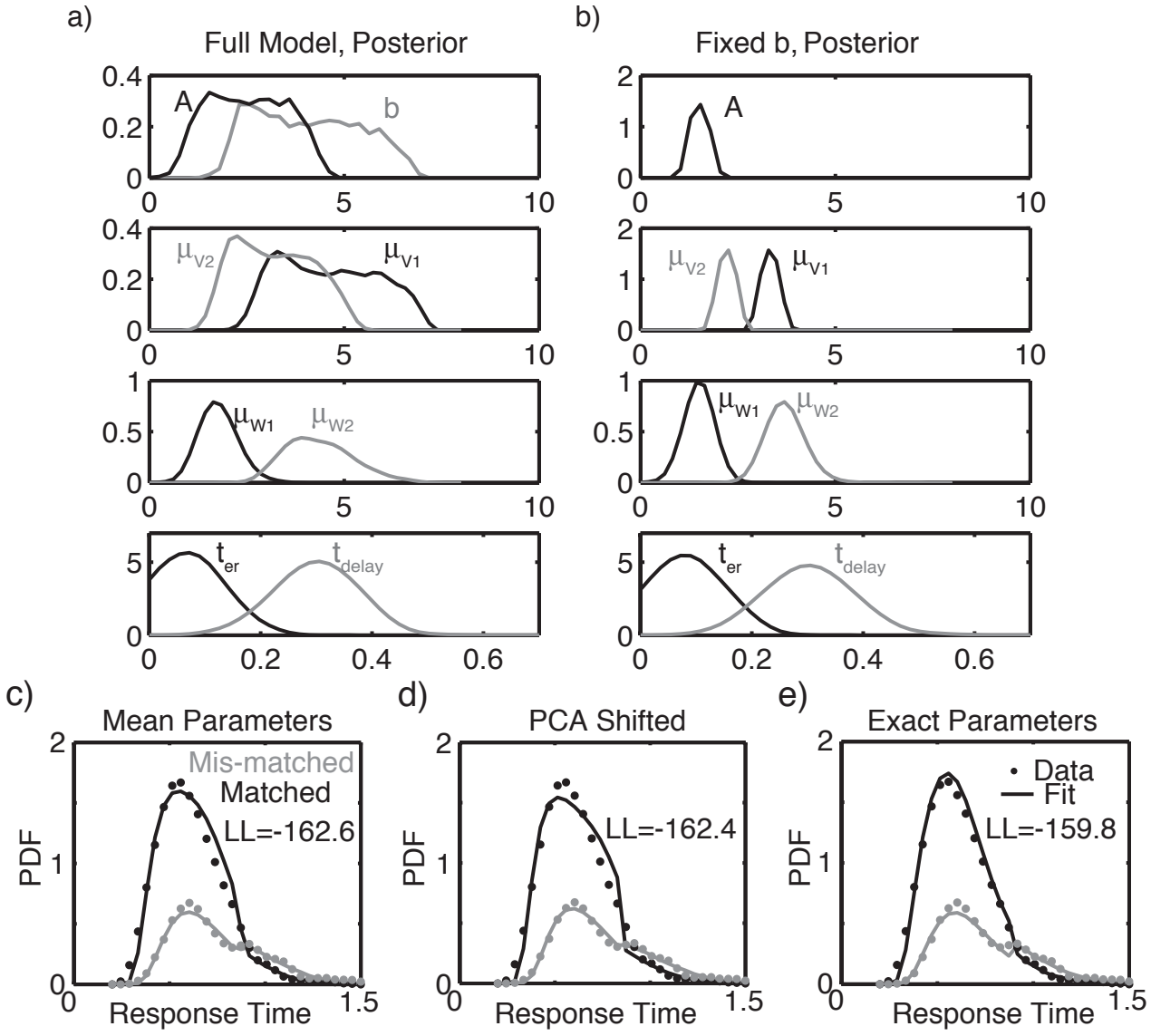

Figure 4: **Fitting the piecewise LBA:** *Panels a,b)* Posteriors for each parameter in the full piecewise LBA model and the restricted model with $b = 2.7$ fixed. *Panels c,d,e)* Fit to data for three parameter sets: c) the mean parameter set taken from (a), d) a translation from the mean parameter set along the first principal component, and e) the parameter set used to produce the data. Matched (dark) and mis-matched (gray) refer to response times for correct / incorrect prior to the information change. Note that in computing the quoted log likelihoods, h was taken very small and N_s very large to effectively remove any bias / variance in the estimates.

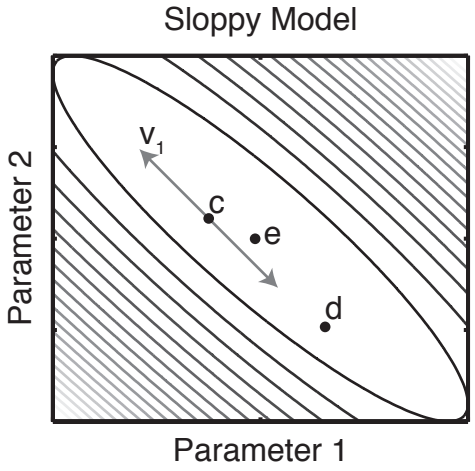

Figure 5: **Sloppy models:** Schematic depiction of a sloppy model showing a strong one dimensional correlation in the likelihood space. v_1 indicates the first principal component while the point c would be akin to the mean parameter set determined from MCMC chain samples. Point d indicates a point along the same principal component subspace while point e indicates the "best fit" parameter (e.g. maximum likelihood). These points schematically indicate the parameters used to produce Figures 4c-e respectively.