Journal of Mathematical Psychology 68–69 (2015) 13–24

Contents lists available at ScienceDirect

Journal of Mathematical Psychology

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

A practical guide to the Probability Density Approximation (PDA) with improved implementation and error characterization

William R. Holmes *

Department of Physics and Astronomy, Vanderbilt University, Nashville, TN 37212, United States
School of Mathematics and Statistics, University of Melbourne, United States

HIGHLIGHTS

- PDA couples non-parametric methods with MCMC to estimate a model's posterior.
- Kernel density estimation is used to construct an approximate likelihood.
- Signal processing methods are used to accelerate this process.
- A "resampled MCMC" that improves chain mixing for this method is presented.
- Approximation errors are characterized theoretically and through example.

ARTICLE INFO

Article history:
Received 16 September 2014
Received in revised form
7 August 2015

Keywords:
Non-parametric approximate Bayesian computation
Approximate likelihood
Kernel density estimate
Markov chain Monte Carlo
Linear Ballistic Accumulator Model

ABSTRACT

A critical task in modeling 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 the recently proposed (Turner and Sederberg 2014) Probability Density Approximation (PDA) method, 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 are: (1) A more efficient implementation of this method that substantially improves computational performance is described. (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). (3) An augmentation of the standard PDA procedure, termed "resampled PDA", that reduces the negative influence of approximation errors on performance and accuracy, is presented. (4) A number of examples of varying complexity are presented along with supplementary code for their implementation.

1. Introduction

Parameter estimation is a crucial step in determining how well a model, and by design the set of assumptions it encodes, account for observations. The canonical Bayesian model estimation problem is to determine the posterior probability distribution of a set of model parameters conditioned on observed data $\pi(\theta|X)$. Given a model likelihood $L(\theta|X)$ and a *prior* distribution $\pi(\theta)$, this accomplished via Bayes' theorem

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

In almost all practical situations however, computing the posterior is not possible since the required integral cannot be computed. For this reason, numerous MCMC methods have been developed to circumvent this third step.

In many cases however, it is not even possible to provide a model likelihood or it may be too cumbersome to compute. Approximate Bayesian Computation (ABC) methods have been developed to deal with this difficulty, see Csilléry, Blum, Gaggiotti, and François (2010) and Turner and Van Zandt (2012) for existing

* Correspondence to: Department of Physics and Astronomy, Vanderbilt University, Nashville, TN 37212, United States.
E-mail address: william.holmes@vanderbilt.edu.

http://dx.doi.org/10.1016/j.jmp.2015.08.006
0022-2496/

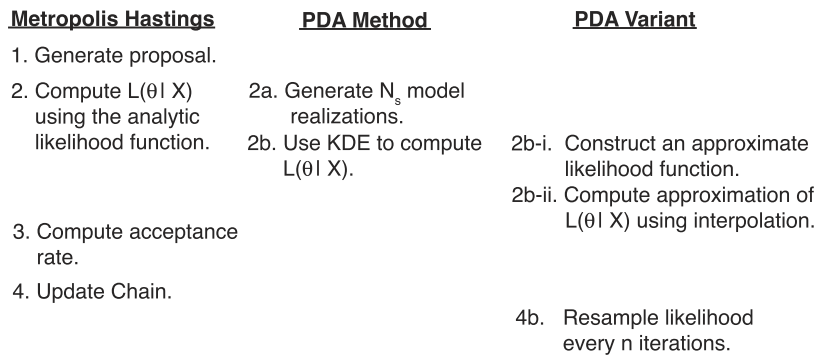

Fig. 1. Comparison of Metropolis–Hastings, PDA, and the PDA variant proposed here: This graphic explains the process of updating a single chain in a MCMC simulation. In MH, the model likelihood is used to compute an acceptance rate (step 2) for the current chain proposal. In the PDA method, step 2 is performed by numerically simulating the model many times (step 2a) and subsequently using kernel density estimation to approximate the likelihood (step 2b). All other components of the MH procedure remain the same. In the PDA variant discussed here, the likelihood approximation (step 2b from the PDA method) is instead performed by first constructing a full approximation to the likelihood function using highly efficient spectral methods (step 2b-i) and then using that to compute the approximation likelihood (step 2b-ii). The PDA variant also adds an additional step (4b) where the likelihood approximation of each chain is resampled (e.g. recomputed) every n iterations. Note that steps 1, 3, and 4, which form the core of the MH algorithm, are identical among these algorithms.

reviews. Generally speaking, ABC deals with the absence of a likelihood by prescribing a surrogate measure for how 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 $\rho(S(X), S(\tilde{X}))$.

One issue with this process is that the summary statistics must adequately represent the models output, often referred to as a sufficiency condition (Dawid, 1979). ABC methods do not approximate the models posterior $\pi(\theta|X)$, but rather a posterior of a new model augmented by the choice of S, $\pi(\theta|S(X))$. So ABC only estimates the posterior distribution of the intended model if $\pi(\theta|S(X)) = \pi(\theta|X)$. The insidious issue with these summary statistics is that it is rarely possible to verify either sufficiency or insufficiency. Furthermore, if they are insufficient, it is usually not possible to determine how badly they have distorted results. Said another way, you know you are probably making errors, but you do not know how large they are.

The essential problem here is that the choice of summary statistics encodes assumptions on the structure of the likelihood function. Those assumptions may or may not be valid, potentially leading to serious errors (Robert, Cornuet, Marin, & Pillai, 2011). 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 likelihood is multimodal or heavily skewed. In a more cognitive context, choice response time distributions are often described by quantile summary statistics (Heathcote & Brown, 2004; Heathcote, Brown, & Mewhort, 2002; Ratcliff & Tuerlinckx, 2002). This was however recently shown to be an insufficient summary of the data (Turner & 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. These methods were incorporated into maximum likelihood estimation more than a decade ago (Fermanian & Salanie, 2004). Recently they have been incorporated into Bayesian Computation (BC) to improve estimation. Mengersen, Pudlo, and Robert (2013) proposed an algorithm (termed BC_{el}) where empirical likelihood methods are used as a replacement for the analytic likelihood function in an importance sampling algorithm. Zhu, Diazaraque, and Leisen (2014) proposed an alternative (BC_{bl}) where the likelihood is estimated using a bootstrap approximation. Turner and Sederberg (2014) proposed an alternative, the Probability Density Approximation (PDA) method, where a kernel density estimate (KDE) (Silverman, 1986) is used to approximate the likelihood in a Metropolis–Hastings (MH) framework. These methods all use likelihood approximations to replace the unknown likelihood in a BC framework (in this way, BC_{kde} would be an alternative name of the PDA method connecting it to existing methods).

The PDA method, which is built upon here, is a variation of the standard MH algorithm where the exact likelihood used to compute acceptance rates is replaced by an approximate likelihood value. Constructing this approximation begins the same way as ABC by first generating a synthetic data set ($\tilde{X}$) consisting of N_s model simulations. Next however, those model realizations are used to construct an approximation of the underlying likelihood $\hat{L}(\theta|X)$ (from here on, a hat will always indicate an approximation). The approximate likelihood is then substituted ($L \to \hat{L}$) into the MH framework and an approximate posterior is determined. Fig. 1 illustrates the similarities and differences between the MH and PDA algorithms as well as the PDA variant developed in this work. The critical step in PDA is the construction of the approximate likelihood $\hat{L}(\theta|X)$ that will replace L. The KDE is a powerful tool for doing just this (Silverman, 1982, 1986).

The basic density estimation problem is to determine the density $f(x)$ at a point x (the likelihood in the Bayesian context) from a collection of samples $\tilde{X} = \{\tilde{x}_j\}$ from f, where $j = 1 \ldots N_s$ and observations are assumed to be independently distributed. The density at x is then approximated by

$$f(x) \approx \hat{f}(x) := \frac{1}{N_s} \sum_{j=1}^{N_s} K_h(x - \tilde{x}_j). \tag{1.2}$$

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 zero and integrates to one. 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 $K(z) = I_{[-0.5,0.5]}(z)$ where I 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.

The most basic implementation of the PDA suffers from a number of inefficiencies: (1) computation of the KDE is time consuming and (2) approximation errors negatively impact proposal acceptance rates. Here, I present an improved implementation that alleviates these issues (to some extent). It can be applied in any setting where the original PDA method is applicable and will have essentially the same accuracy characteristics. As with any method, understanding the strengths, weaknesses, and sources of inaccuracy are required to implement this method and interpret its results. Thus, in addition to describing this variant, the influence of likelihood approximation errors on posterior estimation is discussed both theoretically and through example to illustrate its strengths and weaknesses.

The goals of this article are three fold: (1) describe improvements of the PDA method (Turner & Sederberg, 2014), (2) determine the strengths, weaknesses, and limitations of this method, and (3) present it in an accessible way so it can be utilized by a broad 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.

The remainder of this paper is structured as follows. Section 2 theoretically outlines some of error characteristics and some pitfalls that an interested user should be aware of. Section 3 details the methodological changes that improve performance of the PDA, namely an improved implementation of the KDE and the incorporation of likelihood resampling. Section 4 provides an algorithmic overview of this method. Section 5 demonstrates application of this method for a sequence of increasingly complex models, and the pitfalls outlined in Section 3 are revisited. Section 6 presents profiling and performance data for this improvement.

2. PDA estimation error analysis

While distinct from the standard class of ABC methods, the PDA is approximate in the sense that the stationary distribution of the resulting Markov chain is the posterior of a different, approximate model. With ABC methods, summary statistics are the primary source of posterior distortion. With the PDA, the KDE introduces approximation errors. To see this, note that $\hat{f}$ 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

$$\begin{aligned}\text{Bias}(\hat{f}(x)) &\approx \frac{h^2}{2} f''(x) M_2(K),\\ \text{Var}(\hat{f}(x)) &\approx \frac{1}{N_s h} f(x) \|K\|_2,\end{aligned} \tag{2.1}$$

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 (h) and number of samples (N_s) are of critical importance, which is discussed in more detail in following sections.

2.1. The influence of approximations on likelihood estimation

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

$$\hat{L}(\theta|X) = \prod_{i=1}^{N_d} \hat{L}(\theta|x_i) = \prod_{i=1}^{N_d} \hat{L}_i, \tag{2.2}$$

which is itself a stochastic quantity. In the future, L_i, and $\hat{L}_i$ will refer to the exact and approximate likelihood of observation x_i. Define the estimation error for L_i as

$$\epsilon_i = \hat{L}_i - L_i, \tag{2.3}$$

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

$$\begin{aligned}\hat{L} &:= \prod_{i=1}^{N_d} \hat{L}_i = \prod_{i=1}^{N_d} (L_i + \epsilon_i) = \prod_{i=1}^{N_d} L_i \left(1 + \frac{\epsilon_i}{L_i}\right)\\ &= L \prod_{i=1}^{N_d} \left(1 + \frac{\epsilon_i}{L_i}\right).\end{aligned} \tag{2.4}$$

From Eq. (2.1), we know that $1 + \epsilon_i/L_i \sim N(1 + \mu_i, \sigma_i)$ where

$$\mu_i = \frac{h^2}{2} M_2(K) \frac{L_i''}{L_i}, \qquad \sigma_i^2 = \frac{\|K\|_2 L_i}{N_s h}, \tag{2.5}$$

and $L_i'' = L''(\theta|x_i)$ is the derivative with respect to x_i. Further, the product of normal PDFs is again a normal PDF with known mean and variance so that

$$\frac{\hat{L}}{L} \sim N(\mu_{1\ldots N_d}, \sigma_{1\ldots N_d}), \tag{2.6a}$$

where

$$\mu_{1\ldots N_d} = 1 + \langle \mu_i \rangle, \qquad \sigma_{1\ldots N_d}^2 = \frac{\|K\|_2}{N_d N_s h}, \tag{2.6b}$$

and $\langle \mu_i \rangle$ 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, they provide valuable insights into the scaling behavior of the bulk mean and variance. In particular, reducing the smoothing parameter h decreases estimation bias (while increasing variance), while increasing the number of samples reduces variance (without affecting the bias).

2.2. Influence of the errors on model comparison statistics

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

$$\begin{aligned}\hat{LL}(\theta|X) - LL(\theta|X) &= \sum_{i=1}^{N_d} \left(\hat{LL}(\theta|x_i) - LL(\theta|x_i)\right)\\ &= \sum_{i=1}^{N_d} \log\left(1 + \frac{\epsilon_i}{L_i}\right) \approx \sum_{i=1}^{N_d} \frac{\epsilon_i}{L_i},\end{aligned} \tag{2.7}$$

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

While the individual errors are small in a relative sense, they raise a substantial problem model comparison, where differences in these log likelihood based statistics of as little as $\Delta DIC = 10$ (where ΔDIC indicates the difference of DIC between two models) are construed as evidence for or against a model. Log-likelihoods

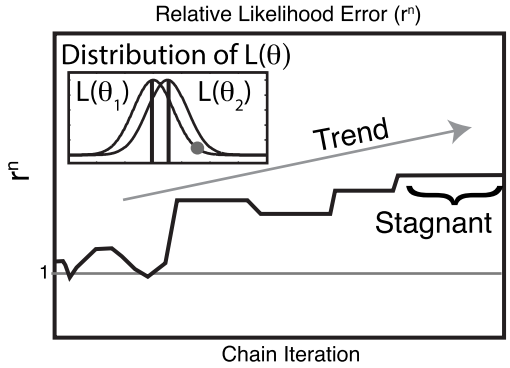

Fig. 2. Biased chain acceptance impairs MCMC performance: *Inset:* The computed value of the likelihood of a given parameter set ($\hat{L}(X|\theta)$) is only an estimate drawn from a distribution. The two distributions schematically illustrate the distribution for two parameter sets where $L(X|\theta_1) < L(X|\theta_2)$ (on the figure $L(X|\theta)$ is replaced with $L(\theta)$ for brevity). The exact likelihood is indicated as the mean of each distribution, since the mean of the estimator approximates the exact likelihood. The grey dot indicates a single estimate of $L(X|\theta_1)$, which in this case is an overestimate. If this chain proposal is accepted on the basis of this overestimate and θ_2 is the subsequent chain proposal, even though θ_2 is more likely, it will be difficult to accept due to the overestimate of $L(X|\theta_1)$. *Full figure:* Schematic illustration of the effect of biased acceptance on chain dynamics in PDA. Since overestimated likelihoods are more likely to be accepted and more difficult to replace on subsequent iterations, the natural tendency is for the error in the likelihood r^n to increase as chains progress. Thus as chains proceed, it will become increasingly difficult to accept a new proposal, resulting in stagnation.

and in turn DIC values are often on the order of 1000–10 000. So these small relative errors can easily be larger than the differential commonly taken as "significant". For this reason, care should be taken when using log-likelihood based model comparison statistics in conjunction with the PDA method or any other approximate method. Examples in subsequent sections will further elucidate this issue.

Also note that this estimate provides a quantitative approximation of the estimation bias. From Eq. (2.7) it is direct to show that

$$E\left(\hat{LL}(\theta|X) - LL(\theta|X)\right) \approx \sum_{i=1}^{N_d} \frac{\mu_i}{L_i} = \frac{h^2}{2} M_2(K) \sum_{i=1}^{N_d} \frac{L_i''}{L_i}, \tag{2.8}$$

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.

2.3. Influence of the errors on rejection rates

A critical quantity in any MCMC procedure is the Metropolis–Hastings ratio (α^n) of accepting a proposal θ^n (where n indicates the chain iteration). Assuming symmetry of the proposal distribution (which is true for all examples to follow), this becomes

$$\alpha^n = \frac{L(\theta^n|X)\pi(\theta^n)}{L(\theta^{n-1}|X)\pi(\theta^{n-1})}. \tag{2.9}$$

Given the approximation procedure provides only an estimate of $L(\theta^n|X)$, the acceptance ratio will be stochastic as well. While we do not have a description of the distribution of this quantity, we can gain some intuition into the influence of errors. Define $r^n = (\hat{L}^n - L^n)/L^n$ to be the relative sample error between the exact and approximated likelihood values. It is direct to show that

$$\frac{\hat{\alpha}^n}{\alpha^n} = \frac{\hat{L}^n}{L^n} \cdot \frac{L^{n-1}}{\hat{L}^{n-1}} = \frac{1 + r^n}{1 + r^{n-1}}, \tag{2.10}$$

where $\hat{\alpha}$ is an approximation of the true acceptance ratio obtained by substituting the likelihood estimate for the true value.

Suppose $\hat{L}^n$ is an over estimate of the likelihood of L^n so that $r^n > 0$ (Fig. 2). This will increase the chance of accepting this proposal. If this proposal is accepted, it will further reduce the chance that subsequent proposals are accepted. If r^n is significantly larger than 0, a significant number of chain iterations will be required to displace it, which will lead to chain stagnation.

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. Underestimates will rarely be accepted and quickly discarded, while overestimates are more likely to be accepted and rarely discarded. This will have two practical implications. First, it will negatively impact efficiency as demonstrated through examples in subsequent sections. Second, it can potentially skew posterior estimation. A chain that gets stuck at a particular parameter set will lead to over representation of that region of the posterior (and necessarily under representation of other regions). Since this a methodological artifact, it will lead to errors in the posterior. In Section 3.1, an augmentation of the PDA to correct this is discussed.

3. Improved PDA implementation

One issue with the PDA is that computing the kernel density estimate can be computationally expensive. Here, an improvement that ameliorates this is presented. Before continuing, it is important to note that this improvement has the same accuracy, the same bias/variance issues as the standard KDE procedure, and will lead to essentially the same results as the standard KDE when embedded into the PDA procedure. It does however substantially improve efficiency. In what follows, I will assume knowledge of foundational mathematical and computational concepts such as convolutions, Fourier transforms, and interpolation. For further information on these topics, see de Boor (1972) or any standard numerical analysis textbook.

The likelihood approximation that is central to the PDA method has two critical steps: (i) generating a large number of model simulations and (ii) synthesizing those samples into an approximate likelihood function. The former will always be "embarrassingly parallel" (e.g. fully parallelizable and able to take advantage of an arbitrary number of computational cores) and can be made highly efficient using increasingly standard desktop computing resources (i.e. GPU's or Xeon Phi co-processors). Thus, step (ii) will in many cases be a computational bottleneck.

The goal here is to integrate an improved implementation of step (ii) into the PDA. The essential problem with this step is that direct computation of the KDE in Eq. (1.2), while simple, is inefficient. Given a set of N_d observations, the kernel function must be evaluated $N_s \cdot N_d$ times. Here, we will take advantage of an old idea (Silverman, 1982) to utilize tools from signal processing to improve performance of this step.

We take advantage of the observation that the KDE formula in Eq. (1.2) resembles a convolution. The discrete model samples $\tilde{X}$ can be represented by the following function

$$d(x) = \frac{1}{N_s} \sum_{j=1}^{N_s} \delta_{\tilde{x}_j}(x), \tag{3.1}$$

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

$$d \star K_h(x) = \frac{1}{N_s} \sum_{j=1}^{N_s} K_h(x - \tilde{x}_j), \tag{3.2}$$

where $\star$ denotes the standard convolution. This is precisely the KDE formula in Eq. (1.2). 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. 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. This theorem essentially says convolution can be performed by transforming f and g into the spectral domain and then multiplying. This is highly beneficial since multiplying two vectors of length n requires n operations while convolving them requires n^2 operations. The basic idea of this method is thus to transform both d and K_h into the spectral domain, multiply in the spectral domain, 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 a regular grid. To generate data on a regular grid, the samples $\{\tilde{x}_j\}$ must first be binned into a histogram with 2^n points (a power of 2 greatly improves FFT efficiency). The improved FFT based likelihood estimation procedure is then as follows:

1. Generate a synthetic data set $\tilde{X}$.
2. Bin the simulated samples to a very fine grid, $d \to \tilde{d}$.
3. 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 $\tilde{d}$).
4. Carry out the convolution operation in the spectral domain

$$\mathcal{F}[\tilde{d} \star K_h](s) = \mathcal{F}[\tilde{d}](s) \cdot \mathcal{F}[K_h](s). \tag{3.3}$$

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

$$\hat{L}(x) = \mathcal{F}^{-1}\left(\mathcal{F}[\tilde{d}] \cdot \mathcal{F}[K_h]\right). \tag{3.4}$$

6. Interpolate the density from this grid to the observed data points to obtain $\hat{L}_i$. 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 more efficient than direct computation of the convolution.

A few notes about this procedure are in order. First, the interpolation step will introduce estimation errors. However, interpolation error will be very small provided $n > 8$ ($n = 10$ is used in all following applications) since a very fine grid minimizes errors. Second, in principle, any kernel (K) can be used in this process. However the canonical Gaussian kernel is particularly useful in this case since its Fourier transform is another Gaussian, $\mathcal{F}[K_h](s) \propto \exp(-0.5h^2s^2)$.

3.1. Resampled PDA

In Section 2.3 (and subsequent examples) it was shown that errors in density estimation can lead to chain stagnation. A simple way to "unstick" chains that become stagnant for this methodological reason is to simply resample that likelihood value frequently. The reasoning is as follows. While a significant overestimation of the likelihood will be rare, that rare event will have a outsized impact on the posterior. By periodically throwing away the existing likelihood estimate for each chain and recomputing it completely, the impact of any individual approximation is limited. This will thus limit the influence of these rare events. Note we are not changing any of the methodological parameters (e.g. N_s or h), we are simply re-computing the likelihood using the same process. This will of course increase computational cost, but it will substantially reduce the chance that a chain becomes stuck due to mis-estimation of the likelihood. This will have the additional benefit of reducing contamination of the posterior caused by randomly oversampling parameters.

4. A procedural overview of the enhanced PDA method for the practitioner

Here, a procedural overview of the resampled PDA with an FFT based KDE implementation is outlined. Familiarity with MCMC methods is assumed and only the details that relate to non-parametric component of this method is provided.

(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. The reduced bias will lead to an increased variance. This is partially ameliorated by the resampling procedure in step (2) however.

(1) Loop over chains.
  (a) Generate a proposal θ^n. In the applications here this was done using DE–MCMC (Ter Braak, 2006; Turner, Sederberg, Brown, & Steyvers, 2013), but any MCMC procedure can be used.
  (b) Compute $\hat{LL}(\theta^n|X)$ 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 $z_0, \ldots, z_l$. 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, $\hat{L}(\theta^n|z_i) \to \hat{L}(\theta^n|x_i)$, 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 $\hat{L}(\theta^n|x_i)$ with a minimum value, say $L_{\min} = 1/(10 * N_s)$.
    (viii) Compute the approximate log-likelihood as

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

  (c) Compute the acceptance ratio $\hat{\alpha}^n$ 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.

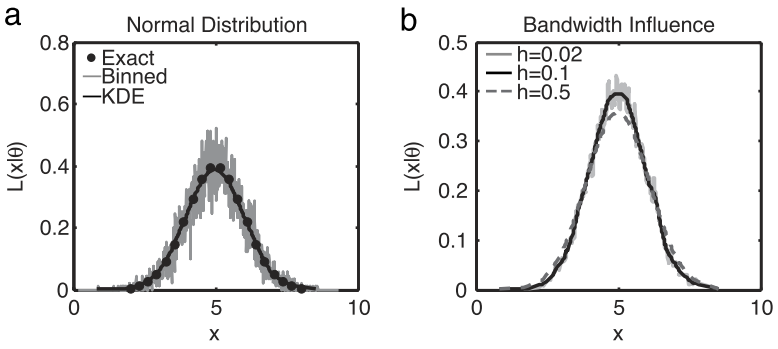

Fig. 3. Reconstructing a Gaussian: *Panel* (a) Reconstruction of a normal distribution using the FFT based KDE method. Gray lines indicate the noisy density estimate derived from binning $N_s = 10{,}000$ sampled points into 2^{10} bins and normalizing to produce a density. Black line indicates the smoothed 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 $\mu = 5, \sigma = 1$.

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 ratio, and specifically how to call the FFT have been neglected for brevity. Details can be found in the supplementary codes (see Appendix A).

5. Examples

In the following sections, this method is applied to four examples of increasing complexity to demonstrate its strengths and weaknesses. In Example 1, the FFT based KDE is presented on its own to demonstrate the density construction process. In Example 2, this is embedded in a MCMC framework to demonstrate its use in parameter inference in a simple case. In Example 3, this method is applied to a more psychologically relevant model, the canonical Linear Ballistic Accumulator Model (LBA). Through this example, the efficacy of the resampling step is demonstrated. In the fourth example, this method is applied to a piecewise extension of the LBA (pLBA) that accounts for a discrete change of information during the course of a trial.

5.1. Example 1: reconstructing a Gaussian distribution

In this example, we will use this FFT based KDE procedure to construct an approximate density function from a collection of samples drawn from a known distribution. A normal distribution with known mean ($\mu = 5$) and variance ($\sigma = 1$) will be used as the test distribution. We begin by first drawing $N_s = 10{,}000$ from the underlying distribution to generate a synthetic data set $\tilde{X}$. Generating a histogram with 2^{10} grid points yields a very noisy distribution (Fig. 3(a), grey). Application of the FFT based smoothing step attenuates the high frequency noise, revealing a smoothed normal distribution that agrees well with the exact distribution (Fig. 3(a), black).

To test the accuracy of log-likelihood estimation, which is critical in MCMC applications, a fixed set of $N_d = 1000$ "observations" from the known normal is drawn. Next, 100 independent reconstructions of this normal are performed, and the approximate density is used to compute $\hat{LL}$ for each. Results show the mean error in this example is 0.3% with a maximum error of 0.8%.

Fig. 3(b) demonstrates the influence of the bandwidth parameter h on the resulting density. Results show that small bandwidths lead to a noisy approximate distribution while large bandwidths attenuate the peak of the distribution. The former is due to insufficient smoothing when h is small. The latter results from over smoothing. Essentially, when h is too large, smoothing transfers some of the mass from the peak into the tails of the distribution. Thus h should 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 produce 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.

5.2. Example 2: fitting a mixture of Gaussians distribution

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

$$X \sim (1-p)N(\mu_1, \sigma) + pN(\mu_2, \sigma), \tag{5.1}$$

where N indicates the normal distribution and p is a weighting parameter indicating the probability that an observation is derived from the normal centered at μ_2. We begin by generating a data set X of $N_d = 1000$ simulated "observations" that the model will be fit to. The parameters used to generate these observations, which we will attempt to recover, are $p = 0.6$, $\mu_1 = -6$, $\mu_2 = 4$, $\sigma = 1$.

To estimate parameters from this data, we first prescribe prior distributions

$$p \sim U(0,1), \qquad \mu_1 \sim U(-10,0), \qquad \mu_2, \sigma \sim U(0,10), \tag{5.2}$$

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) (Storn & Price, 1997; Ter Braak, 2006; 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.

5.2.1. Example 2: results

In Example 1, we saw that h has an important influence on density approximation. To determine how this parameter effects posteriors, the model is fit to data for different values of h. Fig. 4(a) shows the posterior distribution for (μ_1, μ_2) for two values of h, where h_{Silv} is the value derived from "Silverman's rule of thumb" (Silverman, 1986)

$$h_{Silv} = 1.06\hat{\sigma}N_s^{-0.2}. \tag{5.3}$$

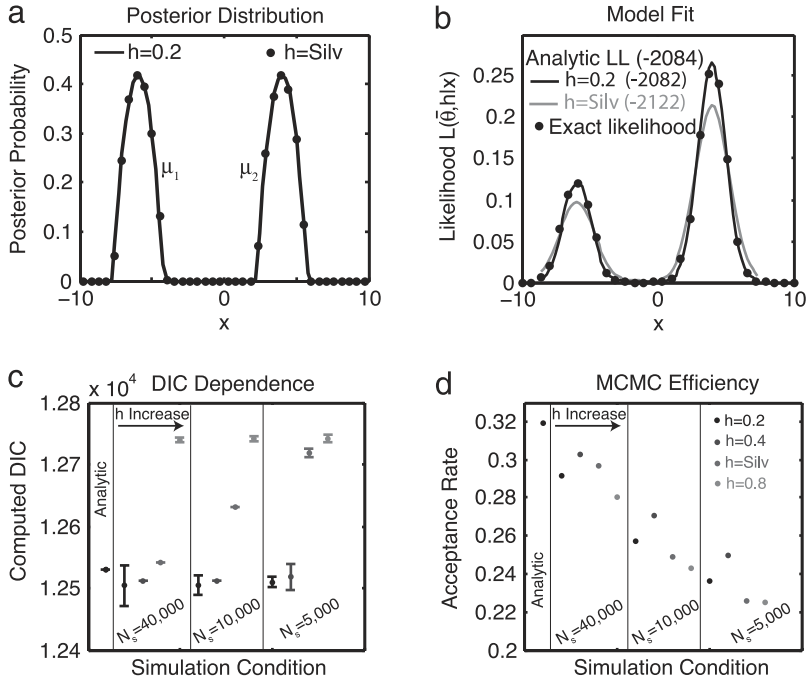

Fig. 4. Fitting a bimodal distribution: *Panel* (a) Posterior distribution obtained using $N_s = 10{,}000$ and two separate values of $h = 0.2, 0.68$, the latter is computed from Silverman's rule of thumb. Posterior for the two means μ_1, μ_2 are shown. *Panel* (b) Quality of model fit. For each value of h (and $N_s = 10{,}000$) the mean of the posterior for each parameter was computed. Using the mean parameter set, 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 is 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 right, 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 3.1.

Here, $\hat{\sigma}^2$ is the sample variance of the data and $N_s = 10{,}000$, yielding $h_{Silv} = 0.67$. Results of posterior computations show the posteriors are visually identical for two values of h. To investigate the influence of h further, (1) the quality of posterior model fit to the data and (2) the log-likelihood error were determined by comparing to the analytic solution, Fig. 4(b).

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 computations. Yet, when the model's PDF is constructed from these parameter sets, the resulting "best fit" model densities differ significantly. This results from over-smoothing of the density in the h_{Silv} case, since the underlying distribution is bimodal. As before, when the smoothing kernel is applied, the peaks of the distribution are attenuated leading to fatter tails.

The choice of h further influences log-likelihood estimates. While the difference between the computed and actual log-likelihood is small in a relative sense ($\sim$2% for h_{Silv}), it is still large in an absolute (e.g. not normalized) sense ($\sim$38). To determine the extent to which posterior approximation errors influence model comparison statistics, DIC was computed for different pairings of (N_s, h). For each, 100 independent posterior estimation simulations were performed (all for the same "observations" X), from which the DIC was computed for each. Fig. 4(c) shows that while even the worst model fit ($N_s = 5000$, $h = 0.8$) leads to a relative DIC error of $<2\%$, the resulting absolute DIC error is >200, which is significantly greater than the DIC differential commonly used to support or reject a model in a comparison context. These results confirm that DIC and similar measures are inadequate for model comparison in this context. This however is a problem with approximate methods in general since the resulting distribution is the posterior of an augmented version of the model under consideration.

5.2.2. The influence of the bias–variance tradeoff

These results (Fig. 4(c)) further illustrate the influence of h (and the bias variance tradeoff it mediates) on results. As h increases, error in the DIC increases, largely independent of N_s which has no influence on estimation bias. As h decreases, the DIC error decreases but the DIC variance increases. This is consistent with the fact that increasing h reduces estimation variance but increases estimation bias. 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 the bandwidth.

Next, consider the influence of h and N_s on proposal acceptance rates (Fig. 4(d)). For each simulation in Fig. 4(c) (which did not incorporate the resampling step), the acceptance rate was recorded. 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 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. Furthermore, as will be shown next, it is less effective than regular resampling of the likelihood.

5.3. Example 3: fitting the LBA model

Here, the canonical LBA model (Brown & Heathcote, 2008) is considered as an example of an evidence accumulator model in decision making literature. A number of accumulator models, including Ratcliff's drift diffusion model (Ratcliff, 1978; Ratcliff & Rouder, 1998), the leaky competing accumulator (Usher & McClelland, 2001), the ballistic accumulator (Brown & Heathcote, 2005), and decision field theory (Busemeyer & Townsend, 1993),

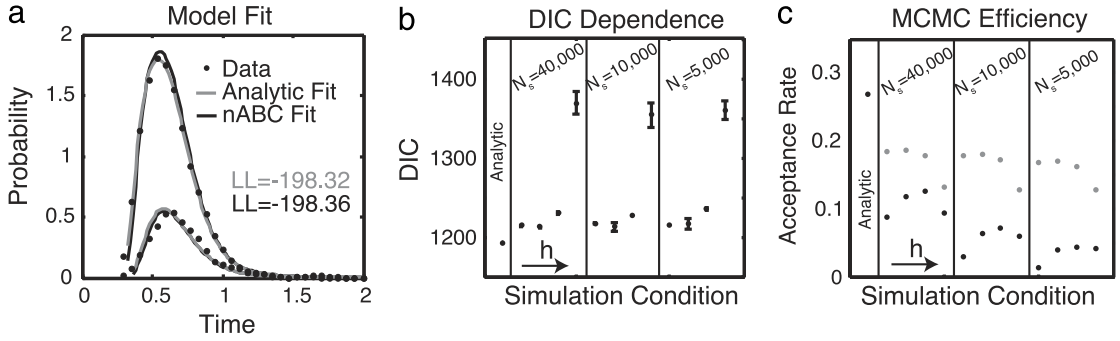

Fig. 5. Fitting the LBA: *Panel* (a) Quality of model fit using the analytic likelihood and approximate likelihood methods (with resampled PDA). For the PDA fit, the approximate likelihood function was used to reconstruct the likelihood using the same value of h, but $N_s = 1{,}000{,}000$ to reduce variance. 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 are shown for each. The resampled PDA is again used here. *Panel* (c) Acceptance rates as a function of N_s, h for the standard (black) and resampled PDA (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 PDA.

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. This example will thus be used to demonstrate the potential power of these methods in response time modeling. 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 ith 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 $\{\mu_i\}$, which comprise the parameter vector θ for this model. The likelihood $L(\theta|c_i, \tau_i)$ of an option c_i being chosen at time τ_i can then be described by an analytic function (Brown & Heathcote, 2005).

To assess the properties of this method in a psychologically relevant context, parameter recovery will be performed using this method for the LBA. To begin, a set of "observations" is generated by simulating the model $N_d = 1000$ times 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

$$b, A \sim U(0, 10), \qquad \mu_1, \mu_2 \sim U(0, 10), \qquad \tau_{er} \sim U(0, 1). \quad (5.4)$$

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

5.3.1. Example 3: results

The posterior of this model is fit both with the analytic likelihood and using this FFT based PDA procedure. Again, for each combination of (N_s, h), 100 independent fits are performed to determine how estimation variability influences various quantities, Fig. 5. 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 PDA (using $N_s = 10{,}000$ and h_{Silv} indicates the bandwidth $h = 0.028$ determined by 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 statistic, Fig. 5(b). For all but the worst case fits ($h = 0.07$), the relative DIC error is ∼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 see a similar problem as before with acceptance rates, Fig. 5(c), which generally decrease when either N_s or h decreases (black dots). This is again consistent with the fact that increased estimation variance reduces acceptance rates. 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 ∼6%. Fortunately, this can be ameliorated to a significant extent by introducing the resampling step into the PDA.

To test the efficacy of the resampling step, we performed an identical numerical experiment with the likelihood of each chain resampled every third iteration, independent of the history of the chain Fig. 5(c) (grey dots). Why was this frequency chosen? It is well established that the theoretical acceptance rate for this form of MCMC is ∼25% for five or more parameters. The resampling rate was chosen to be faster than the theoretical frequency of chain movement. Results show substantially improved acceptance rates, which increase to ∼17%–18%. This improvement 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 causes this, but this value is well above any reasonable choice for h. These results additionally demonstrate likelihood resampling is more efficient than increasing N_s. Even with a small value of $N_s = 5000$, the acceptance rate with resampling is significantly higher than the rate using $N_s = 40{,}000$ without resampling. Thus resampling is both more computationally efficient and more effective than increasing N_s.

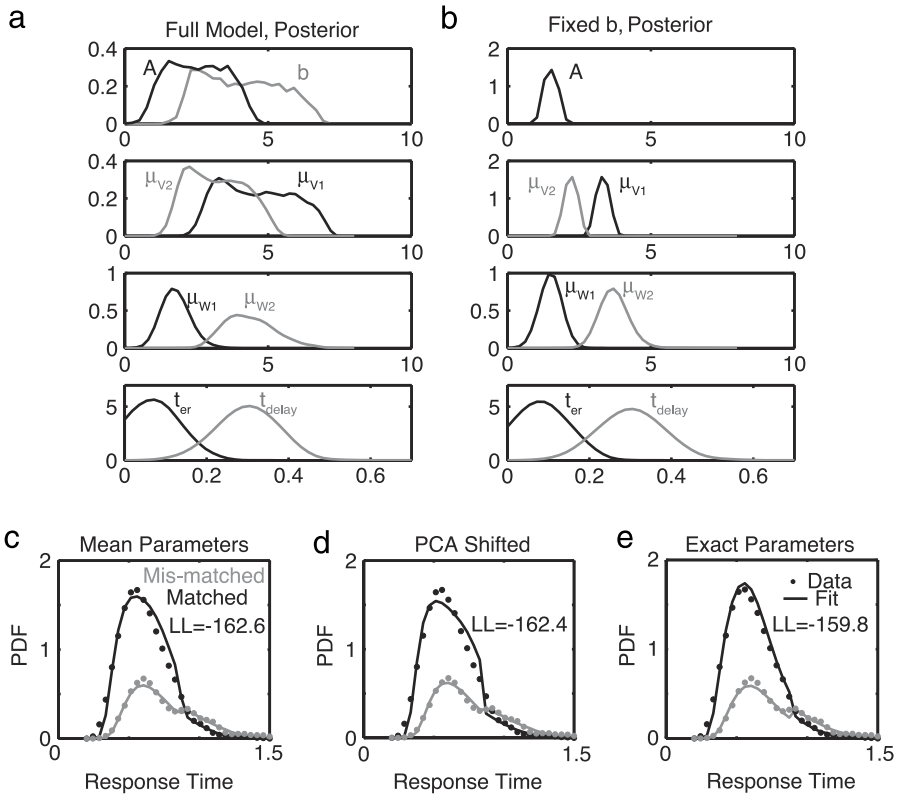

Fig. 6. Fitting the piecewise LBA: *Panels* (a, b) Posteriors for each parameter in the full piecewise LBA model and the restricted model with $b = 2.7$ fixed. *Panels* (c, d, e) Fit to data for three parameter sets: (c) the mean parameter set taken from (a), (d) a translation from the mean parameter 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.

5.4. Example 4: fitting the pLBA

In this final example, a piecewise LBA type model will be considered. The canonical LBA describes models decisions made on the basis of information that is fixed and unchanging in time. In many cases however, information may change during the course of the decision process. In Huk and Shadlen (2005), Kiani, Hanks, and Shadlen (2008), Thura, Beauregard-Racine, Fradet, and Cisek (2012), Tsetsos, Gao, McClelland, and Usher (2012), and Winkel, Keuken, Van Maanen, Wagenmakers, and Forstmann (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, information changes over time and one would expect the decision process to incorporate the new information. To account for this, a piecewise variant of the standard LBA was first presented in Holmes, Trueblood, and Heathcoat (submitted for publication). 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, leading to two sets of rates before ($v_i \sim N(\mu_{vi}, \sigma)$) and after ($w_i \sim N(\mu_{wi}, \sigma)$) the change respectively. This model is referred to as "piecewise LBA" since evidence accumulation is piecewise linear and deterministic. 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, μ_{v1}, μ_{v2}, μ_{w1}, μ_{w2}, t_{er}, and t_{delay}. See Holmes et al. (submitted for publication) for further details. To begin, a data set consisting of $N_d = 1000$ observations is created, assuming $A = 1.6$, $b = 2.7$, $\mu_{v1} = 3.4$, $\mu_{v2} = 2.5$, $\mu_{w1} = 1.5$, $\mu_{w2} = 3.6$, $t_{er} = 0.1$, $t_{delay} = 0.3$.

To fit this model to this data set (e.g. parameter recovery), we set the KDE parameters to $h = 0.02, N_s = 10{,}000$ and utilize 24 chains to account for the increased number of parameters. To generate the synthetic data needed to construct the likelihood at each proposal, N_s pairs of accumulators are simulated to determine the choice and response time for each. To improve sampling performance, we further use a blocked MCMC, as in Holmes et al. (submitted for publication). Parameters are grouped into two blocks: (A, b, μ_{v1}, μ_{v2}, t_{er}), which describes the accumulation process prior to the change of information, and (μ_{w1}, μ_{w2}, t_{delay}), which describes the process after the change.

5.4.1. Example 4: results

Estimated posteriors for the pLBA are shown in Fig. 6(a). 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, Fig. 6(c). Second, it is known that the LBA model exhibits significant parameter correlations (Turner et al., 2013), which commonly lead to poorly localized posteriors. In biological and physics literature, this is commonly referred to as a "sloppy model" (Apgar, Witmer, White, & Tidor, 2010; Gutenkunst et al., 2007) since the likelihood is nearly unchanged over a wide range of parameters, Fig. 7. To confirm parameter correlations are the source of this posterior spread, principal component analysis (PCA) was performed on the recorded chain data, as done in Meulders, De Boeck, Van Mechelen, Gelman, and Maris (2001). This reveals that the first and second principal components account for ~92% and 5% of the variability in the posterior respectively. Furthermore, the eigenvector of the principal component shows

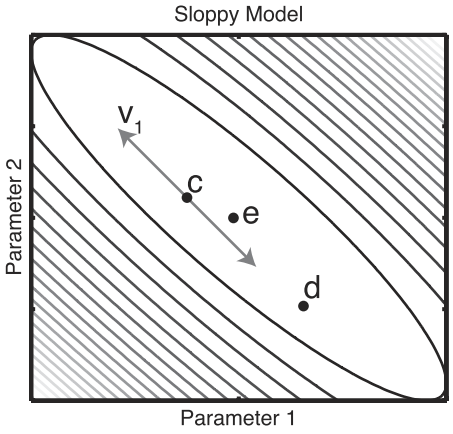

Fig. 7. Sloppy models: Schematic depiction of a sloppy model showing a strong one dimensional correlation in the likelihood space. v_1 indicates the first principal component while the point c would be akin to the mean parameter set determined from MCMC chain samples. Point d indicates a point along 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 Fig. 6(c)–(e) respectively.

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

$$\vec{p} = \mu + k\vec{v_1}, \tag{5.5}$$

see Fig. 7 for a schematic depiction. For even a relatively large displacement ($k = 4$, Fig. 6(d)), 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, Fig. 6(e). 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 (Fig. 6(c)), 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.

5.4.2. PDA and sloppy models

These observations raise an important issue. Recall from the previous two examples that this approximation procedure yields small errors in the log-likelihood on the order of ∼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 PDA method 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 goal is to extract a single parameter set, more must be done. Strategies for dealing with sloppiness in models have been discussed extensively in other literature (Apgar et al., 2010; Gutenkunst et al., 2007), but such exposition is beyond the scope of this article. It is important to reiterate though that the sloppiness of posterior estimates here is a reflection of the sloppiness of the underlying model that prevents accurate estimation.

6. Profiling and performance

To demonstrate the performance gains of this method, performance of both the PDA and this variant have been profiled for the LBA and pLBA examples. Before continuing, it is important to note that there is no universally accepted way of determining the runtime of a program. Here, I use what is usually referred to as CPU time which measures the total time a CPU is in operation. With multi-core processors, this is different than the wall clock time (aka. runtime), which measures the time from start to finish. If a program utilizes four cores for 1 min, then the runtime would be 1 min while the CPU time would be 4 min. In the following data CPU time has been chosen to reflect the amount of computational load used by an algorithm, and MATLAB's built in profiler has been used to determine the fraction of total time spent on different parts of the computation. All computations were performed on a 3.2 GHz Core i5 mac. Similar results (not presented) were found when computations were performed on a less powerful Macbook Pro laptop.

There are two critical computational steps that are responsible for almost the entire computational load in the PDA: (i) simulating the model a large number of times and (ii) synthesizing those samples into a likelihood. To assess performance of the basic and improved PDA algorithms (Table 1), the total time to complete a single Metropolis–Hastings update for a collection of N_c chains (15 for LBA and 24 for pLBA), with $N_s = 10{,}000$ samples generated for each chain parameter set, was computed. The fraction of that time spent on (i) and (ii) respectively was then determined. For the LBA model, we see that the FFT based PDA method is significantly faster (∼98% faster) than the original PDA using Matlab's built in kernel density estimation function ("ksdensity"). Ten independent replicates of this numerical experiment were run with near identical results. Results further show the improvement in performance is derived from dramatically faster likelihood estimation. Similar results were found when profiling the pLBA code as well. Taken together, these results suggest that for LBA based models, likelihood estimation is a computational bottleneck, and this procedure removes that bottleneck.

To assess performance of this method relative to a standard MH algorithm with an analytic likelihood, the performance of the latter is determined for the LBA model. Results show the PDA variant is faster than the MH algorithm with the analytic likelihood. Profiling 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 to consider functions such as the normal CDF to be "analytic", the term analytic has little meaning to a computer. To be clear, this is not meant to advocate for abandoning the standard methods. 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.

Of course, these results will depend on the language being used, the core libraries being called, and the computational platform. Taken together though, they suggest that within the MATLAB environment on a standard desktop or laptop computer, this augmentation can significantly speed the likelihood estimation process. Furthermore, this requires only 30–40 extra lines of code beyond the standard PDA.

Table 1
Profiling different posterior estimation methods. Here, the computational cost of each estimation method is reported. Results for two examples (3 and 4) are shown. For example 3 (the LBA model), the posterior is estimated using (i) the known likelihood, (ii) the PDA procedure utilizing Matlab's "ksdensity" implementation of the kernel density estimation, and (iii) the augmented PDA method discussed here utilizing an FFT for improved performance. For example 4 (the pLBA model), estimation is performed using only the PDA method and the augmented PDA method, since an analytic likelihood is not available to compare against. In each case, the time reported is for a single MCMC iteration. For each iteration, the following times are reported: (A) The time required to generate model samples for density estimation, (B) the time required to generate an approximate likelihood function from those samples, and (C) the total time required for the MCMC iteration. In each case, 100 iterations are performed and the average times are reported. Additionally, ten independent replicates of this numerical experiment were performed to ensure initial conditions do not influence results. These results indicate the FFT based density estimation is dramatically faster than the build in kernel density estimator.

Model	Method	Sample generation	Likelihood estimation	Total
LBA	Analytic likelihood	–	–	0.07
	PDA	0.02	2.91	2.95
	PDA variant	0.02	0.02	0.05
pLBA	PDA	0.04	5.65	5.76
	PDA variant	0.04	0.04	0.11

7. Discussion

This article presents an enhancement of the Probability Density Approximation method originally introduced in Turner and Sederberg (2014). This method, which combines non-parametric statistical methods with Bayesian inference techniques, is an extensible methodology for performing posterior parameter estimation. The purpose of this article is to discuss this methods properties, 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, Carlin, Stern, & Rubin, 2003; Robert & Casella, 2004) to particle filtering methods (Cappé, Guillin, Marin, & Robert, 2004; Del Moral, Doucet, & Jasra, 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 tractable expression 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 summary 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 likelihood estimation methods have been used to circumvent this requirement (Turner & Sederberg, 2014).

Both PDA 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. They differ in how they determine this plausibility. Whereas summary statistics are used to compare the simulated data to observations in ABC, in PDA, a non-parametric approximation of the underlying likelihood function is computed. 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 is to construct an approximation of the models density function $L(\theta|x)$. This is accomplished using the KDE (Silverman, 1986), which is a method of directly computing an approximation of the likelihood of any particular observation $L(\theta|x_i)$ 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 $LL(\theta|X)$, 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., submitted for publication; Turner & Sederberg, 2014) show that 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 computationally inefficient and can itself become a computational bottleneck. To circumvent this, a highly efficient implementation of the KDE is presented here to improve performance of the PDA. In the present applications, this improved performance dramatically. 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 PDA" procedure is proposed, which accounts for the variability in this estimator and substantially improves performance.

While this method is an effective option for likelihood free models, like any approximate method, it does come with drawbacks. First, the KDE likelihood estimator is inherently biased. This bias is quite small, being on the order of 1%. Unfortunately, these errors have a profound effect on model comparison and hypothesis testing. The essential problem is that standard quantities such as AIC, BIC, or DIC are absolute measures of model comparison whereas estimation errors are relative errors. 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 1000 (which is common), approximation errors on the order of 1%–2% will overwhelm these comparison statistics. Thus, small errors introduced in the approximation process can render comparisons null and void.

A second issue is that this method can have difficulties with models containing very strong parameter correlations, 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, 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 the PDA method 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 PDA are somewhat 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, Thomas, Best, & Spiegelhalter, 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.

Acknowledgments

This work was supported by the National Science Foundation grant SES1530760. I would like to thank Joachim Vandekerckhove and Jennifer S. Trueblood for comments on this manuscript.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.jmp.2015.08.006.

References

Apgar, J. F., Witmer, D. K., White, F. M., & Tidor, B. (2010). Sloppy models, parameter uncertainty, and the role of experimental design. *Molecular BioSystems*, *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.

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), 907–929.

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.

Dawid, A. P. (1979). Conditional independence in statistical theory. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *41*(1), 1–31.

de Boor, C. (1972). *Elementary numerical analysis*. McGraw-Hill.

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.

Fermanian, J.-D., & Salanie, B. (2004). A nonparametric simulated maximum likelihood estimation method. *Econometric Theory*, *20*(04), 701–734.

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

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis* (2nd ed.). Chapman and Hall/CRC.

Gutenkunst, R. N., Waterfall, J. J., Casey, F. P., Brown, K. S., Myers, C. R., & Sethna, J. P. (2007). Universally sloppy parameter sensitivities in systems biology models. *PLoS Computational Biology*, *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. (2015). A new framework for modeling decisions about changing information: The Piecewise Linear Ballistic Accumulator model (submitted for publication).

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.

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.

Mengersen, K. L., Pudlo, P., & Robert, C. P. (2013). Bayesian computation via empirical likelihood. *Proceedings of the National Academy of Sciences*, *110*(4), 1321–1326.

Meulders, M., De Boeck, P., Van Mechelen, I., Gelman, A., & Maris, E. (2001). Bayesian inference with probability matrix decomposition models. *Journal of Educational and Behavioral Statistics*, *26*(2), 153–179.

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*. New York, NY: Springer.

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.

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), 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*, 1–17.

Turner, B., & Sederberg, P. (2014). A generalized, likelihood-free method for posterior estimation. *Psychonomic Bulletin & 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.

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. *Psychonomic Bulletin & Review*, *21*, 777–784.

Zhu, W., Diazaraque, J.M.M., & Leisen, F. (2014). A bootstrap likelihood approach to Bayesian computation. Statistics and econometrics working papers, Universidad Carlos III, Departamento de Estadística y Econometría. URL: http://EconPapers.repec.org/RePEc:cte:wsrepe:ws142517.