

Variational Bayesian Methods for Cognitive Science

Matthew Galdo, Giwon Bahg, and Brandon M. Turner
The Ohio State University

Abstract

Bayesian inference has become a powerful and popular technique for understanding psychological phenomena. However, compared with frequentist statistics, current methods employing Bayesian statistics typically require time-intensive computations, often hindering our ability to evaluate alternatives in a thorough manner. In this article, we advocate for an alternative strategy for performing Bayesian inference, called variational Bayes (VB). VB methods posit a parametric family of distributions that could conceivably contain the target posterior distribution, and then attempt to identify the best parameters for matching the target. In this sense, acquiring the posterior becomes an optimization problem, rather than a complex integration problem. VB methods have enjoyed considerable success in fields such as neuroscience and machine learning, yet have received surprisingly little attention in fields such as psychology. Here, we identify and discuss both the advantages and disadvantages of using VB methods. In our consideration of possible strategies to make VB methods appropriate for psychological models, we develop the differential evolution variational inference algorithm, and compare its performance with a widely used VB algorithm. As test problems, we evaluate the algorithms on their ability to recover the posterior distribution of the linear ballistic accumulator model and a hierarchical signal detection model. Although we cannot endorse VB methods in their current form as a complete replacement for conventional methods, we argue that their accuracy and speed warrant inclusion within the cognitive scientist's toolkit.

Translational Abstract



Bayesian statistics is an alternative statistical framework that has become popular for understanding psychological phenomena. In contrast to the point estimates and confidence intervals of classical statistics, the Bayesian framework provides a distribution (the posterior) that describes our uncertainty about the parameters of interest. Rarely are there closed-form solutions for deriving the posterior, and therefore Bayesians typically rely on computational methods to approximate it. The time-intensive nature of these computational methods can prohibit the application of Bayesian framework. In this article, we advocate for an alternative, and often more efficient, strategy for performing Bayesian inference called variational Bayes (VB). VB methods make assumptions about the functional form of the posterior distribution, and then systematically morph the approximating function's parameters so that it best matches the target posterior. VB methods have enjoyed considerable success in fields such as neuroscience and machine learning, yet have received surprisingly little attention in fields such as psychology. Here, we identify and discuss both the advantages and disadvantages of using VB methods in reference to conventional posterior approximation methods. We investigate a series of algorithmic components to see which, if any, of these components can be packaged into a general purpose algorithm for problems often encountered when fitting psychological models to data by testing them on two popular models from psychology. Although we cannot endorse VB methods in their current form as a complete replacement for conventional methods, we argue that their accuracy and speed warrant inclusion within the cognitive scientist's toolkit.

Keywords: variational Bayes, differential evolution, linear response variational Bayes, linear ballistic accumulator, cognitive modeling

Bayesian methodology has become a leading statistical tool in mathematical and computational psychology over the last decade (Lee & Wagenmakers, 2005; Van de Schoot, Winter, Ryan, Zondervan-Zwijnenburg, & Depaoli, 2017). The reasons for this growth in popularity are numerous, but can be linked to advances in statistical techniques, the wide availability of powerful computing

resources, and most importantly, to the fact that Bayesian techniques work where frequentist methods cannot easily be applied. In particular, Bayesian inference can be performed in the context of models of theoretical interest, whereas frequentist methods often must depend on simplifying asymptotic assumptions (e.g., the central limit theorem). These same models can be embedded in hierarchical structures

This article was published Online First October 10, 2019.

 Matthew Galdo,  Giwon Bahg, and Brandon M. Turner, Department of Psychology, The Ohio State University.

The content of this article was presented at the annual 2019 Midwest Cognitive Science Conference at The Ohio State University and 2019 Annual Meeting of the Society for Mathematical Psychology in Mon-

tréal, Quebec, Canada. The authors would like to thank the reviewers, Alexandra Greenberg, and Oksana Chkrebti for their feedback and suggestions.

Correspondence concerning this article should be addressed to Brandon M. Turner, Department of Psychology, The Ohio State University, 1827 Neil Avenue Columbus, OH 43210. E-mail: turner.826@gmail.com

that permit joint estimation of individual differences and the overall effects of experimental manipulations. Furthermore, inspection of the joint posterior distribution permits us to examine the intimate relationships between parameters that would ordinarily be unobservable without an arduous parameter recovery study (e.g., bootstrapping).

In the Bayesian framework, parameters are considered random variables just as experimental measurements are. The variability of parameters reflects our uncertainty about their true value. Our prior beliefs about the d dimensional vector of parameters $\theta \in \Theta$, are quantified as a probability distribution $\pi(\theta)$ over the possible values in Θ . After the data x are obtained, we use the model's likelihood $\pi(x|\theta)$ and Bayes' theorem to compute the posterior distribution $\pi(\theta|x)$ as follows:

$$\pi(\theta|x) = \frac{\pi(x|\theta)\pi(\theta)}{\pi(x)}, \quad (1)$$

where $\pi(x)$ is the marginal likelihood of the data, often referred to as "model evidence" because it describes the likelihood of the data given the structure of the model.

The posterior distribution $\pi(\theta|x)$ characterizes the probability of observing θ at each value in the continuum of Θ , after considering the data and one's prior beliefs about the a priori plausibility of each θ (for further reading, we recommend Gelman et al., 2013; Lee & Wagenmakers, 2014).

Current Bayesian Method

Although the Bayesian framework is enticingly powerful, there are several limitations that make implementation difficult for some psychological models. For our purposes, the limitations we wish to improve upon are computational inefficiency, and accounting for the correlated nature of the parameters observed in many psychological models.

Computational Inefficiency

Although efficiency is implicitly a relative term, even modern computational techniques for obtaining posterior estimates can be frustratingly slow. From our own efforts, we can often expect to wait for days or even weeks to procure accurate estimates of some complex models. One major difficulty is accurately estimating the marginalizing constant

$$\pi(x) = \int_{\Theta} \pi(x|\theta)\pi(\theta)d\theta. \quad (2)$$

Because the integral is over all parameter dimensions Θ , exact calculation is often limited to only a handful of toy problems, and so it must be approximated for realistic models. Indeed, without the development of the Metropolis algorithm (Chib & Greenberg, 1995; Hastings, 1970; Hitchcock, 2003; Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953) and the advancement of computational resources, Bayesian inference would probably continue to lag behind frequentist inferential techniques.

Perhaps the most widely used method for approximating $\pi(x)$ is through Monte Carlo integration (Casella & Berger, 2002). The central idea behind these techniques is that once the data x have been observed, the marginalizing constant $\pi(x)$ will remain fixed for every value of θ . Hence, it is advantageous to express Equation 1 as

$$\pi(\theta|x) \propto \pi(x|\theta)\pi(\theta).$$

Here, we can easily evaluate the unscaled posterior density at many values of θ , and assess them relative to one another. Techniques such as Markov chain Monte Carlo (MCMC; for a recent introductory article, see Van Ravenzwaaij, Cassey, & Brown, 2018) exploit this proportionality by constructing a long series of random samples of θ with serial dependence, and accept or reject each new sample with Metropolis-Hastings probability. After many samples have been drawn, the frequency of acceptances across the parameter space can then be used as an estimate of the posterior distribution.

Correlated Parameter Dimensions

In the context of psychology, many exciting new models are constructed in a "bottom up" manner: Relatively well-understood neural mechanisms are instantiated and used as the foundation of more complex structures that can produce patterns of data that might be observed in the real world. However, as these models often describe a complex and dynamic system, the parameters depend intimately on one another. While there is no theoretical reason for why parameter dependency should complicate matters, methods such as MCMC have been shown to exhibit drastic inefficiencies as the correlation between parameter dimensions increases (Turner, Sederberg, Brown, & Steyvers, 2013).

Recently, there have been several algorithmic advancements that have addressed the problem of correlated dimensions (Carpenter et al., 2017; Neal, 2011; Turner, Sederberg et al., 2013). One example that will be featured throughout this article is differential-evolution MCMC (DEMCMC; ter Braak, 2006), which has now been productively applied to a wide range of models in psychology (Evans, Steyvers, & Brown, 2018; Heathcote et al., 2019; Molloy, Galdo, Bahg, Liu, & Turner, 2019; Turner, Sederberg et al., 2013). The main difference between DEMCMC and MCMC is the manner in which the two algorithms search the parameter space. DEMCMC relies on a vector-based proposal scheme operating on top of a system of particles. Essentially, each particle in the system contains information about the shape of the likelihood function (i.e., through the Fisher information matrix), and the particles communicate this information collectively through the set of all pairwise differences in their locations. Hence, differential evolution is a coarse way of representing vital information about parameter dependencies that can be exploited for the purposes of either optimization (Turner & Sederberg, 2012) or posterior sampling (Turner, Sederberg et al., 2013).

Outline

Despite the success of DEMCMC within psychology, it does not directly address the aspect of computational efficiency relating to the marginalizing constant in Equation 2. That is, although using DEMCMC reduces the total number of samples necessary to approximate a posterior distribution well, DEMCMC still relies on Monte Carlo integration to approximate Equation 2. However, Monte Carlo integration is not the only way to approximate the posterior distribution in Equation 1. Another approach, called "variational Bayes," turns the goal of approximating a posterior distribution into an optimization problem. The purpose of this article is to investigate a series of algorithmic components to see

which, if any, of these components can be packaged into a general purpose algorithm for problems often encountered when fitting psychological models to data. We begin by first explaining the fundamentals of variational Bayes and the mean-field approximation. Next, we explore the effects of different optimization algorithms (i.e., gradient-based vs. population-based optimization). Third, we examine methods to improve the estimation of the covariance across parameters within the context of the mean-field approximation. Finally, we explore the applicability of our proposed approach to a hierarchical model relevant to psychology.

Variational Bayes

Variational Bayesian methods (also known as variational Bayes, variational inference, variational approximation; VB) are a class of methods that approximate the posterior distribution by searching amongst a prespecified family of distributions for the best approximation. Central to the success of the VB approach is the assumption of the posterior's distributional form q , governed by parameters λ . For example, a common assumption is that q may be a normal distribution, and so λ would be the set of parameters detailing the normal distribution, such as the mean μ and standard deviation σ . The goal then, is to find values of λ such that $q(\theta|\lambda)$ best approximates $\pi(\theta|x)$.

VB first found success in the field of machine learning (Attias, 1999; Jaakkola & Jordan, 2000; Kingma & Welling, 2013; Petersen, 1987), and then garnered further support in neuroscientific applications (Daunizeau, Adam, & Rigoux, 2014; Daunizeau, Friston, & Kiebel, 2009; Friston, Mattout, Trujillo-Barreto, Ashburner, & Penny, 2007; Penny, Kiebel, & Friston, 2003; Starke & Oswald, 2017; Woolrich, 2012). In fields like machine learning and neuroscience, VB is often used out of necessity because MCMC is impractical—if not infeasible—for high dimensional and/or big data problems. The primary issue seems to be the total number of times that the likelihood function $\pi(x|\theta)$ is calculated to form an approximation of the posterior. For big data problems, calculating the likelihood function can be time-consuming, and so each evaluation of the posterior comes with a costly premium. As we hope to show in this article, the accuracy of a VB method improves at a much faster rate than what can be obtained using MCMC methods, although VB methods may not achieve the same level of accuracy as MCMC methods (Blei, Kucukelbir, & McAuliffe, 2017). Though beyond the scope of this article, there is an interesting literature on fusing MCMC and VB algorithms (Salimans, Kingma, & Welling, 2015; Zhang, Shahbaba, & Zhao, 2018).

Despite this growing support, there are only a few uses of VB methods within psychology. For example, Oswald, Kirilina, Starke, and Blankenburg (2014) used VB methods to fit a linear dynamical systems model of perceptual choice response time to data, and suggested the use of VB methods on other popular choice response time models such as the diffusion decision model (Ratcliff & McKoon, 2008) and the linear ballistic accumulator (LBA; Brown & Heathcote, 2008). Annis and Palmeri (2019) used VB to fit an exemplar-based LBA to a continuous recognition memory experiment. In the field of psychometrics, VB has recently been applied to item response theory and was found to be both accurate and efficient in this context (Natesan, Nandakumar, Minka, & Rubright, 2016). VB has also been used in machine-learning

applications to psychological models, such as in the early analysis of topic models (Griffiths, Steyvers, Blei, & Tenenbaum, 2005).

There are likely three reasons for the slow adoption of VB methods within psychology. First, as most psychological applications do not fall within the category of “big data” problems, MCMC methods have continued to be applied successfully on a range of problems with no apparent suffering. Second, a well-known shortcoming of conventional VB methods is their lack of appreciation of correlated parameter dimensions. Because most psychological models exhibit strong parameter dependencies, this feature of VB methods has likely steered many interested parties away. Third, many of the key developments in the VB literature may be difficult to follow due to notational and language difference between machine learning and conventional statistical descriptions within psychology.

Here, our goal is to show how VB methods can be used within psychology. First, as cognitive models continue to integrate additional measures of cognition such as choice, response time, and confidence (Pleskac & Busemeyer, 2010; Ratcliff & Starns, 2009), eye-tracking measures (Mele & Federici, 2012), and high-dimensional neural measures (Mack, Preston, & Love, 2013; Palestro, Bahg et al., 2018; Purcell et al., 2010; Turner, Forstmann, Love, Palmeri, & Van Maanen, 2017; Turner, Van Maanen, & Forstmann, 2015; Turner, Wang, & Merkle, 2017), the inferential problems associated with big data applications loom ominously on the horizon. Second, recent advances in VB techniques have procured simple, post hoc corrections to handle correlated parameter dimensions efficiently. In the end, the set of techniques we advocate should be robust enough to handle the set of problems cognitive modelers typically encounter.

In this section, we discuss three essential components of VB algorithms. First, we discuss how one might choose the q function from which posterior distributions are to be optimized. Second, we discuss a common assumption used in VB methods called the *mean-field approximation*, which is also adopted in this article. Third, we define the discrepancy between the q function and the posterior distribution, such that the distance between the two distributions can be minimized. So, while the first and last sections are essential ingredients to any VB method, we emphasize that the second topic is a simplification that is commonly used for computational convenience.

Choosing a q Function

Suppose, before we attempt to fit a model to data, we give pause to first consider what characteristics the posterior distribution will have. On what values will the posterior tend to center? How wide is the range of possible parameter values within the posterior? Is the posterior likely to be skewed? Of course, in practice we are unlikely to know the answers to these questions, but given that we can usually generate reasonable specifications for our a priori beliefs within the prior distribution $\pi(\theta)$, surely we can provide reasonable guesses about the characteristics the resulting posterior distribution will have.

Describing the characteristics of the posterior is the problem of specifying the distribution function q . There are two types of problems when specifying q : one considers the family of q , and the other considers its parametric form. The family of q refers to the type of distribution we select. For example, we may suspect that

the normal distribution will be a suitable family, or we may prefer a skewed family such as the gamma distribution. On the other hand, the parametric form details family-specific characteristics of the distributions such as the tendencies (e.g., mean, median), spreads (e.g., range, variance), and shapes (e.g., skew, kurtosis) that will be considered within a given family. For example, if we choose a normal family, we may choose only to consider forms with different means, or we may want to consider the full range of forms with different means and variances. However, in the standard normal family, we could not incorporate skew into the set of considerations, because no parameter in the standard normal corresponds to skew. To incorporate skew, we would first have to change the family under consideration (e.g., the more general skewed normal distribution).

The function q is properly written as $q(\theta|\lambda)$, a function defined over the parameter space of interest θ , but controlled by the parameters λ . In the normal example above, q would be specified to be a normal distribution on θ but with parameters $\lambda = \{\mu, \sigma\}$, such that

$$q(\theta|\lambda = \{\mu, \sigma\}) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\theta - \mu)^2}{2\sigma^2}\right].$$

Here, if we only consider distributions with different means, we would set σ to some value (e.g., $\sigma = 10$), and set $\lambda = \mu$. Alternatively, to consider both different means and standard deviations, we would simply set $\lambda = \{\mu, \sigma\}$. Keep in mind that whereas θ are the set of parameters we are trying to estimate, the algorithmic process of VB operates on determining the parameter values λ that allow $q(\theta|\lambda)$ to best match the target posterior $\pi(\theta|x)$. In other words, the parameters λ are auxiliary in the sense that they pertain only to the VB algorithm itself, much like the tuning parameters defining the transition kernel in MCMC (Palestro, Sederberg, Osth, Van Zandt, & Turner, 2018), or the jump scaling parameters in DEMCMC (ter Braak, 2006; Turner, Sederberg et al., 2013).

Although the specification of q may seem daunting as there are an infinite number of possibilities, we have at least one guideline and one assurance. First, the specification of q must match the *support* of the desired posterior. The support Θ is the set of all possible parameter values that the posterior can be placed upon. For example, while the support of the mean parameter μ from the normal distribution is unbounded such that $\Theta_\mu = (-\infty, \infty)$, the probability of success parameter within the binomial distribution p is clearly bounded by zero and one such that $\Theta_p = [0, 1]$. Hence, both the prior distribution and the q function must obey Cromwell's rule, where they each have the same support as the posterior distribution $\pi(\theta|x)$. Rather than working in the space corresponding to the parameters of interest θ , it may be advantageous to consider transformations of θ that would enable more flexible q functions. For example, in the binomial family, it may be easier to estimate the logit-transformed probability of success so that a normal family can be used when defining q and $\pi(\theta)$.

One assurance we have comes from the Bayesian central limit theorem that effectively states that as the number of data points approaches infinity, the posterior converges to a normal distribution given the prior obeys Cromwell's rule and θ is real-valued (Van der Vaart, 2000; Walker, 1969). The number of data points clearly affects the quality of the normal approximation, but one can perform a simulation study on a few example data sets to assess the

quality of the approximation as a function of the size of the data set. Such an analysis would provide some guidance about whether a normal distribution would be an appropriate choice for the q function.

Mean-Field Variational Bayes

As we will discuss in the next section, the key to VB inference is in optimizing the set of parameters λ with respect to the distance between q and the posterior $\pi(\theta|x)$. As with all optimization problems, the complexity of q , as measured by the number of dimensions and the number of possible forms q can take, determines the speed with which optimization can take place. While in general a simpler q function can be optimized more quickly, it can also place limitations on the accuracy of the resulting posterior approximation. It will be important to keep both of these aspects in mind when specifying q in practice.

Until now, our discussion has focused on the estimation of a single parameter θ , but in practice θ will often be multidimensional such that $\theta = \{\theta_1, \dots, \theta_d\}$, consisting of d parameters. In this case, specifying q becomes complicated because the support of Θ is multidimensional, and so the q function will also need to be multidimensional. While there are plenty of multivariate functions we could choose when specifying q , doing so would exacerbate the number of parameters within λ , making the optimization problem more difficult and often computationally inefficient. For reasons of computational complexity, most VB research relies on the mean-field approximation. The mean-field approximation assumes that the q function over the set of latent variables θ can be safely factorized into a set of marginal distributions, namely

$$q(\theta|\lambda) = \prod_{i=1}^d q_i(\theta_i|\lambda_i),$$

where $\lambda = \{\lambda_1, \dots, \lambda_d\}$, and each λ_i denotes the subset of parameters defining the parametric form of each q_i family corresponding to parameter θ_i along dimension i . Here, each dimension i can have a separate q_i family, depending on any number of considerations (e.g., the support of θ_i).

By assuming the variational distribution q can be factorized across dimensions, we are inherently assuming that all of the information in the joint distribution of θ can be accessed by combining each of the marginal distributions of θ_i . Although this is a strong assumption, especially for psychological models, it provides some compelling advantages. First, it suggests that we can optimize each λ_i by holding all other λ parameters at their current values (i.e., optimizing λ_i and holding $\lambda_{-i} := \{\lambda_j \in \lambda : j \in (1, \dots, i-1, i+1, \dots, d)\}$ constant). A further convenience is when the functional form of each q_i is from an exponential family, such as a normal, gamma, or beta distribution. When using exponential families, one can analytically solve for the exact update equations for performing gradient ascent.

Kullback-Leibler Divergence

Once we have chosen the family and parametric form of $q(\theta|\lambda)$, we must identify the set of parameter values λ that minimize the discrepancy between q and the target posterior distribution $\pi(\theta|x)$. However, to do this, we need to define a metric to assess the distance between the two distributions. Although there are a vari-

ety of metrics that can be used (e.g., Dieng, Tran, Ranganath, Paisley, & Blei, 2017), in this article we focus on the Kullback-Liebler (KL) divergence (Kullback & Leibler, 1951). The KL divergence comes from information theory and describes how divergent, or different, one probability distribution is in reference to another probability distribution. The KL divergence between q and the posterior is

$$KL[q(\theta|\lambda) \parallel \pi(\theta|x)] = \int_{\theta} q(\theta|\lambda) \log \left[\frac{q(\theta|\lambda)}{\pi(\theta|x)} \right] d\theta, \quad (3)$$

where $KL(a \parallel b)$ denotes the KL divergence between distributions a and b . The KL divergence will always be larger than zero unless $q(\theta|\lambda) = \pi(\theta|x)$. Additionally, the KL divergence value depends on what the reference distribution and is therefore not symmetric (i.e., $KL(a \parallel b) \neq KL(b \parallel a)$). Figure 1 shows example KL divergences between two normal distributions: one shown as a histogram and the other shown as a blue density curve. Moving from left to right, the blue density's variance decreases in 0.5 increments and the mean increases in increments of 1.0, while the distribution depicted as a histogram remains fixed. In the top centermost panel the two densities

are equivalent and produce a zero-valued KL divergence. Any deviations in the blue density's mean or variance from the target histogram results in a positive KL divergence. Additionally, Figure 1 contains a contour plot of the KL divergence over a grid for different standard deviations and means of q . We can see the change in KL divergence for a shift in the mean parameter is highly dependent on the value of the standard deviation and vice versa.

Although Equation 3 defines the distance between q and the posterior, we have yet to detail how it can be used as an objective measure on which to optimize λ . After all, if we could easily evaluate the marginalizing constant in Equation 2, there would be no need to approximate the posterior using any of the aforementioned techniques. First, we can replace the posterior distribution in Equation 3 with the terms from Equation 1:

$$\begin{aligned} KL[q(\theta|\lambda) \parallel \pi(\theta|x)] &= \int_{\theta} q(\theta|\lambda) \log \left[\frac{q(\theta|\lambda)\pi(x)}{\pi(x|\theta)\pi(\theta)} \right] d\theta \\ &= \int_{\theta} q(\theta|\lambda) \left[\log(\pi(x)) + \log \left(\frac{q(\theta|\lambda)}{\pi(x|\theta)\pi(\theta)} \right) \right] d\theta. \end{aligned} \quad (4)$$

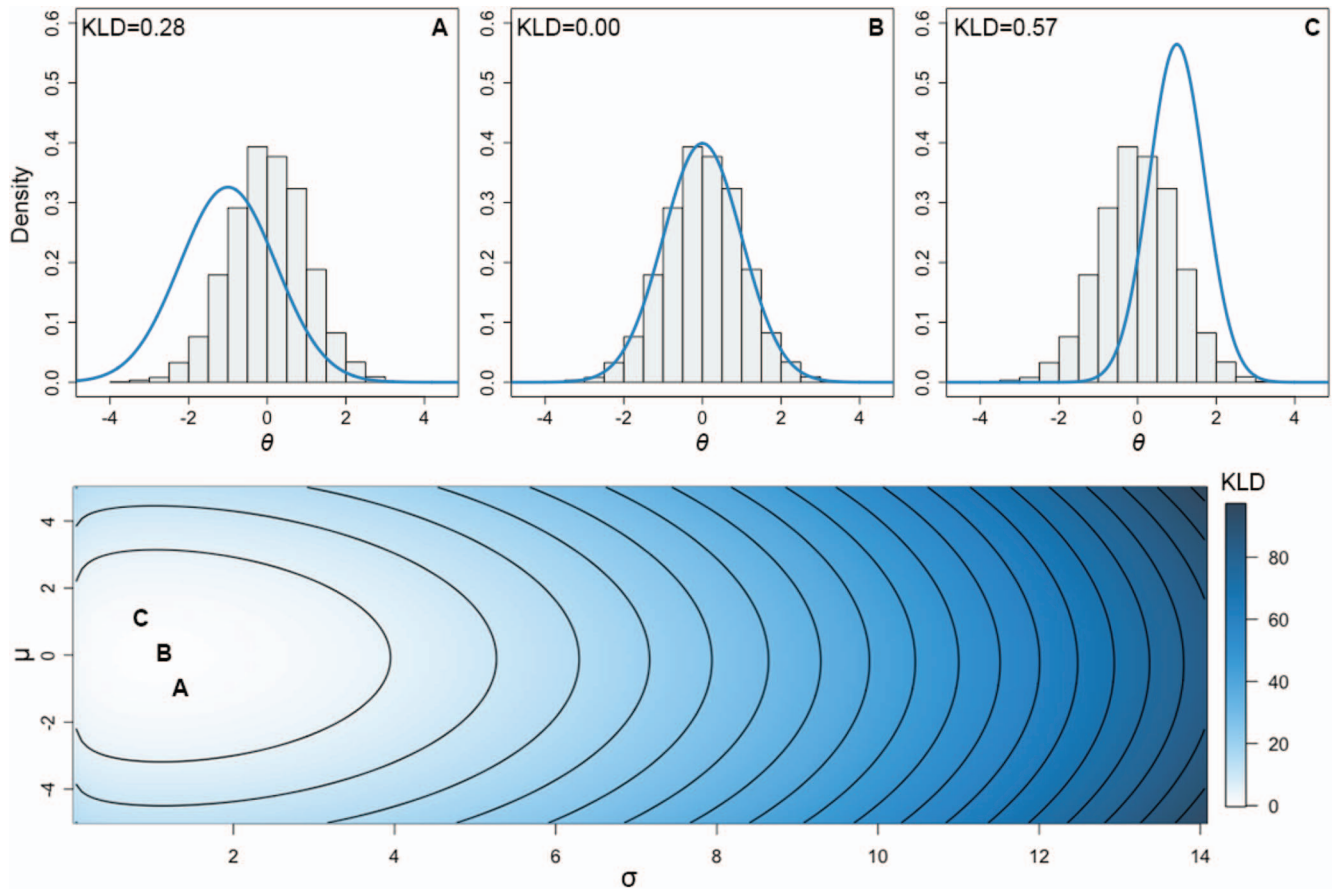


Figure 1. Example KL divergences. The top three panels show two normal distributions—one illustrated as a histogram and one illustrated as a blue density (i.e., q)—and their corresponding KL divergences (KLD). For the top three panels, the mean of q increases with respect to the x-axis in increments of 1, and the variance of q decreases in increments of 0.5. Below is a contour plot which illustrates the KL divergence for a grid of different means (y-axis) and standard deviations (x-axis) for q . Each of the letters in the contour plot correspond to the letters in the top three panels. See the online article for the color version of this figure.

Using the “Law of the Unconscious Statistician” (Casella & Berger, 2002), we can define the expectation of a function $f(\theta)$ where $\theta \sim q(\theta|\lambda)$ as

$$\mathbb{E}_q[f(\theta)] = \int_{\Theta} q(\theta|\lambda) f(\theta) d\theta.$$

Hence, Equation 4 can be rewritten in terms of expectation:

$$\begin{aligned} KL[q(\theta|\lambda) \parallel \pi(\theta|x)] &= \mathbb{E}_q[\log(\pi(x))] + \mathbb{E}_q\left[\log\left(\frac{q(\theta|\lambda)}{\pi(x|\theta)\pi(\theta)}\right)\right] \\ &= \log(\pi(x)) + \mathbb{E}_q\left[\log\left(\frac{q(\theta|\lambda)}{\pi(x|\theta)\pi(\theta)}\right)\right] \\ &= \log(\pi(x)) + \mathbb{E}_q\{\log[q(\theta|\lambda)]\} - \mathbb{E}_q\{\log[\pi(x|\theta)\pi(\theta)]\}. \end{aligned} \quad (5)$$

Although Equation 5 helps to relate the KL divergence to a computable quantity, because we still do not know the marginalizing constant $\pi(x)$, we cannot effectively use this expression to find the set of λ values that minimize it. Yet, as we have indicated above, $\pi(x)$ does not depend on the q function; $\log(\pi(x))$ is a constant and therefore does not influence which values of λ are extrema of the KL divergence. Therefore, the left-most term in Equation 5 can safely be ignored (Blei et al., 2017; Jordan, Ghahramani, Jaakkola, & Saul, 1999). Hence, to minimize the KL divergence in Equation 5, an equivalent operation would be to find the values of λ that maximize

$$-\Omega(\lambda) := \mathbb{E}_q\{\log[q(\theta|\lambda)]\} - \mathbb{E}_q\{\log[\pi(x|\theta)\pi(\theta)]\}. \quad (6)$$

Evidence Lower Bound

An alternative strategy for optimizing λ is based on the marginalizing constant itself. For a single data point, we can reexpress Equation 2 in terms of the KL divergence by rearranging Equation 5:

$$\begin{aligned} \log(\pi(x)) &= \mathbb{E}_q\{\log[\pi(x|\theta)\pi(\theta)]\} - \mathbb{E}_q\{\log[q(\theta|\lambda)]\} \\ &\quad + KL[q(\theta|\lambda) \parallel \pi(\theta|x)] \end{aligned}$$

Because the KL divergence is a nonnegative constant lying on the interval $[0, \infty)$, we can substitute it with C and write $\log(\pi(x))$ as follows:

$$\begin{aligned} \log(\pi(x)) &= \mathbb{E}_q\{\log[\pi(x|\theta)\pi(\theta)]\} - \mathbb{E}_q\{\log[q(\theta|\lambda)]\} + C \\ &\geq \mathbb{E}_q\{\log[\pi(x|\theta)\pi(\theta)]\} - \mathbb{E}_q\{\log[q(\theta|\lambda)]\} \equiv \Omega(\lambda). \end{aligned} \quad (7)$$

Equation 7 suggests that the right-hand side will always be less than or equal to the log of the model evidence, which is why the term $\Omega(\lambda)$ is often referred to as the “evidence lower bound” (ELBO). Finally, by comparing Equation 7 to Equation 6, we see that maximizing the ELBO is equivalent to minimizing the KL divergence.

Although Equation 7 relates the value of λ directly to the KL divergence, computing the integrals or expectations in Equation 7 is nontrivial. However, one can approximate Equation 7 with Monte Carlo integration (Weinzierl, 2000). Specifically, when maximizing $\Omega(\lambda)$, the first step is to generate S random samples of θ denoted as θ^* from the distribution $q(\theta|\lambda^i)$ using the current best values for λ on the i th iteration. Then, Equation 7 can be approximated with the unbiased Monte Carlo estimator $\hat{\Omega}(\lambda)$:

$$\hat{\Omega}(\lambda) := \frac{1}{S} \sum_{s=1}^S \{\log[\pi(x|\theta_s^*)\pi(\theta_s^*)] - \log[q(\theta_s^*|\lambda^i)]\}. \quad (8)$$

Hence, the function values for the likelihood, prior, and q function are evaluated for each θ_s^* , combined, and averaged to produce an estimate of the evidence lower bound. Although specifying $\hat{\Omega}(\lambda)$ provides a tractable function we can optimize, because $\hat{\Omega}(\lambda)$ is a Monte Carlo estimator, the function and optimization process are inherently stochastic. Although there are a variety of other distribution-comparison measures we could use to specify our VB objective function (see Dieng et al., 2017; Knoblauch, Jewson, & Damoulas, 2019; Saha, Bharath, & Kurtek, 2019), these other metrics are beyond the scope of this article. Here we focus on the KL divergence, or equivalently ELBO, because it is the most widely used.

Benchmark Data and Estimation Method

Having described the fundamental components of VB, we now turn to validating the method and examining more complex algorithmic considerations. To create a benchmark test from which each algorithm can be evaluated, there are three considerations. First, the ideal scenario is knowing the data generating process so that we can obtain a “true” posterior distribution. Having a method for obtaining the true posterior distribution assures us that the only differences observed in the posterior approximations when using VB algorithms are due to the VB methods themselves. Second, we wished to test the algorithms on models with psychological plausibility and broad applicability across the field. Third, we require a benchmark problem that is difficult enough along the appropriate dimensions. By being “difficult enough,” we hope to create separation among the performances of the various algorithms. By being the “appropriate” type of difficulty, we hope to test the algorithms in ways that are valuable to our field. Namely, we want the benchmark test to be high-dimensional with strong correlations between parameter dimensions.

With these considerations in mind, the subsequent analyses will first focus on the LBA; (Brown & Heathcote, 2008) model. We hope to use the LBA model as a benchmark test to evaluate the algorithms (e.g., following Turner & Sederberg, 2014) in an analogous way as machine-learning benchmark tests on standard data sets, such as MNIST data set (Deng, 2012). The LBA model is quite suitable for our purposes as it is (a) analytically tractable, meaning that its likelihood function is known and can be evaluated efficiently; (b) has several parameters (i.e., five); and (c) has been shown to exhibit strong correlations among the parameter dimensions (Turner, Sederberg et al., 2013). Because of the aforementioned problems when using VB algorithms on problems with correlated parameter dimensions, approximating the LBA model’s posterior distribution should serve as a difficult challenge for conventional mean-field VB algorithms.

A large part of the success of Bayesian statistics in psychology is due to the ease of fitting hierarchical models (also known as multilevel models or random-effects models) to data. Hierarchical models assume that parameters vary randomly on some level (e.g., from subject to subject or from condition to condition) but come from a common distribution. Hierarchical models afford us enough flexibility to appreciate heterogeneity in data (e.g., individual differences), but also provide enough constraint to allow for gen-

eralizable inference. In the Bayesian framework, prior distributions are the mechanic by which constraints from different levels (i.e., hierarchies) of inference can be imposed. In place of a distinct fixed prior for each replicate's set of lower-level parameter(s) (e.g., mean response for a given subject), a shared set of hyperparameters is freely estimated. The free estimation of hyperparameters facilitates the flow of information between replicates' parameter estimates resulting in a reduction in the variance of lower-level posteriors coined "shrinkage." Additionally, because the hyperparameters are freely estimated, hyperpriors must be specified. Hierarchical models are inherently higher dimensional, providing a further inferential challenge. With the challenge of a high dimensional posterior and broad applicability in mind, we wanted to demonstrate the suitability of our VB approach for a hierarchical model but also for a model other than the LBA. Therefore, for our third analysis we test our VB approach on a hierarchical signal detection model (HSD; Egan, 1958; Green & Swets, 1966; Lee, 2008; Rouder & Lu, 2005).

Linear Ballistic Accumulator Model

Model description. The LBA model was designed to explain how speeded decisions are made by capturing the joint distribution of choice and response time. Relative to previously developed models such as the diffusion decision model (DDM; Ratcliff & McKoon, 2008), and the leaky, competing accumulator (LCA; Usher & McClelland, 2001) model, the LBA model assumes a noiseless momentary integration of evidence through the deliberation period. The "ballistic" and independent nature of the accumulation process facilitates the derivation of the likelihood function, and the ease with which this likelihood function can be calculated has undoubtedly facilitated the model's widespread success (Annis, Miller, & Palmeri, 2017; Bogacz, Wagenmakers, Forstmann, & Nieuwenhuis, 2010; Rodriguez, Turner, & McClure, 2014; van Maanen et al., 2011). Like all sequential sampling models, the LBA assumes that on the presentation of a stimulus, evidence is accumulated for each alternative until the amount of evidence for a particular alternative first exceeds a threshold amount b . Each alternative is represented as a separate accumulator, whose rate of accumulation (i.e., the drift rate) is sampled from a normal distribution with mean v_i for the i th accumulator and a common standard deviation s . Each accumulator is also assumed to have some preliminary evidence prior to stimulus presentation (i.e., bias), and these starting points are sampled from a uniform distribution on the interval $(0, A)$. Both the rates of accumulation and the starting points are assumed to be resampled on every trial for each accumulator, creating variability in the choice response time distributions even when the same stimulus is presented. Finally, the LBA model assumes the presence of processes such as visual encoding and motor movements that are modeled with a nondecision time parameter τ . Hence, for the two alternative cases we will use below, the model has six parameters, b, A, v_1, v_2, s , and τ . However, due to mathematical scaling properties of the model, we assume that the parameter $s = 1$ throughout.

Prior specification. Within the LBA model, there are clear dependencies among the parameters that are often advantageous to incorporate into the parameter estimation process. Doing so facilitates general algorithm implementation, as idiosyncratic details of the model should not impede our ability to fit the model to data.

For example, the upper bound of the starting point A should never exceed the response threshold b , and so it follows that $A \in (0, b)$ is a natural constraint as opposed to $A \in (0, \infty)$. A common technique for incorporating parameter dependencies is to reparameterize A to be a proportion of b . For our purposes, we assume $A = zb$, where $z \in (0, 1)$ and we estimate z rather than A . Another restriction is that the nondecision time parameter τ must be less than the minimum of the response time distribution. Similar to z , we can set $\tau = y \min(RT)$, and estimate $y \in (0, 1)$ rather than τ .

As our goal was to obtain posterior estimates in a Bayesian context, we must also specify prior distributions for the model parameters. To estimate each of the model's parameters, we first transformed each parameter so that they had infinite support (i.e., all real values are possible points in the posterior). For example, for parameters bounded by $(0, \infty)$ such as the threshold parameter b , a logarithmic transformation was used, whereas for parameters bounded by $(0, 1)$, a logit transformation was used. Because the effects of the above transformations may be unclear, we recommend, for example, first drawing samples from the prior distribution for z , transforming them to A , and then evaluating whether or not the prior specification on z results in a reasonable specification for a prior on A . Performing such a test helps to make the prior specification of z more transparent, as well as avoiding any change-of-variable complications that arise from parameter transformations within the model (e.g., deriving the Jacobian). Following the transformation, we settled on the following informative priors:

$$\log(b), \log(v_1), \log(v_2) \sim N(0, 1/2)$$

$$\text{logit}(z), \text{logit}(y) \sim N(0, 1/3).$$

Ultimately, with the exception of degenerate cases, the prior specification should not affect the generality of our results as these priors were maintained across all algorithms.

Data generation. We used the LBA model to generate 750 choice response times from a two-alternative forced-choice task, using the following parameter values: $b = 1$, $A = .5$, $v_1 = 1.2$, $v_2 = 0.8$, and $\tau = 0.15$. These parameter values were chosen such that they would generate data that resembled a real experiment. The choice proportion of the first alternative was 0.624. The mean and standard deviation of the response time distribution were 0.914 s and 1.720 s, respectively.

Hierarchical Signal Detection Theory

Model description. First, we will introduce signal detection theory (SDT; Green & Swets, 1966), as the hierarchical SDT is simply the hierarchical Bayesian extension of the classic SDT framework. SDT aims to explain how a decision maker discriminates a meaningful signal from random noise when one is forced to make an (often binary) choice under uncertainty. In a typical signal detection task, subjects are presented with a perceptual stimulus and asked to determine whether a signal is present (i.e., a "yes" response) or not (i.e., a "no" response). Due to the perceptual overlap between stimuli that either contain a signal or not, there is clear uncertainty in the decision accuracy that is often proportional to the degree of overlap between the two physical states (i.e., signal present vs. signal absent).

Given a situation where we need to discriminate signals from noises based on a feature whose values are continuous, the most

basic version of SDT assumes that the signal and noise are represented as two separate normal distributions with different means but equal variance. With this representation, we make a decision by referring to a decision criterion which distinguishes signal-like and noise-like feature values. The SDT framework formalizes the decision-making performances using two parameters: discriminability D and response bias B . Discriminability is the standardized distance between means of the signal and noise distributions and represents an observer's ability to separate signal stimuli from pure noise. Response bias represents the location of a decision criterion with respect to the signal and noise distributions. If the stimuli is perceived to have sensory affect that is above the criterion, a "yes" response will be given, whereas if the stimulus is perceived to be below the criterion, a "no" response will be given. Bias reflects the tendency of the observer to place their criterion away from the ideal observer, and can be either liberal (i.e., allowing more "yes" responses) or conservative (i.e., allowing more "no" responses).

A hierarchical Bayesian extension of SDT (HSD; e.g., Lee, 2008; Rouder & Lu, 2005) allows us to incorporate individual differences in the discriminability and response bias parameters. The typical modeling approach in the HSD framework is to assume that individual-level model parameters are samples from normal distributions whose means (m_B , m_D) and variances (s_B^2 , s_D^2) are the group-level parameters. Even further, Rouder and Lu (2005) proposed an additive mean model that explains the individual-level SDT parameters to be the sum of the group-level mean plus individual- and stimulus-wise effect terms. In this article, we reduce the problem to the level of individual differences for simplicity.

Prior specification. Like the LBA, we first transformed our parameters to ensure they were real-valued and matched the support of the normal distribution. Because the discriminability parameters D_i 's and the hypervariances s_D^2 and s_B^2 are positively constrained we estimated them on the log scale. We then chose informative priors that obeyed Cromwell's rule. We specified the following hyperpriors:

$$m_B, m_D, \log(s_B), \log(s_D) \sim N(0, 1).$$

The structure of the model then implies the following priors for the subject-level parameters:

$$B_i \sim N(m_B, s_B) \quad \forall i \in \{1, 2, \dots, 10\};$$

$$\log(D_i) \sim N(m_D, s_D) \quad \forall i \in \{1, 2, \dots, 10\}.$$

Data generation. We used the HSD model to sample $\log(D)$ and B parameters for 10 hypothetical subjects. We sampled each subject parameter $\log(D_i)$ and B_i parameter by setting the hyperparameters to $m_B = m_D = 0$ and $s_B = s_D = 1/2$. Then for each subject we simulated 750 choices for a signal detection task with 350 stimuli with a signal present, and 350 stimuli with no signal present.

Differential Evolution MCMC (DEMCMC)

To obtain a benchmark estimate of the joint posterior distribution of the LBA model's parameters, we used a genetic variant of the MCMC algorithm—DEMCMC (ter Braak, 2006; Turner, Sederberg et al., 2013). DEMCMC has been shown to be a highly

efficient algorithm for sampling from posteriors with correlated dimensions, such as those seen in the LBA model (Turner, Rodriguez, Norcia, Steyvers, & McClure, 2016; Turner, Sederberg et al., 2013), the DDM (Turner et al., 2015), and the LCA model (Turner, Rodriguez et al., 2018; Turner & Sederberg, 2014; Turner, Sederberg, & McClelland, 2016). In this application, we maintained similar specifications as used in Turner, Sederberg et al. (2013), but increased the number of chains and samples to remedy issues of sampler auto-correlation. We ran the algorithm for 5,500 iterations with 30 chains. Chains were initialized by drawing random samples around the true parameter values, with the variance of these draws being slightly less than the variance of their corresponding marginal priors. Within the crossover step, we set γ to $\frac{2.38}{\sqrt{2d}}$, where $d = \dim(\theta)$, and $\eta \sim U(-.001, .001)$. We treated the first 500 iterations as a burn-in period and discarded these samples. The posterior samples were thinned such that only every fourth sample was retained to reduce autocorrelation. In total, 37,500 samples were used to estimate the joint posterior distribution. Visual assessment was used for determining convergence.

To obtain a benchmark estimate of the joint posterior distribution of the HSD model's parameters, we used DEMCMC in an identical way as for the LBA model, except where stated. Common with hierarchical applications, we used blocked-DEMCMC (ter Braak, 2006; Turner, Sederberg et al., 2013). Within one iteration, subject-level parameters were sampled in parallel conditional on the current values of hyperparameters and then hyperparameters were sampled conditional on newly sampled subject-level parameters. We ran the algorithm for 3,000 iterations with 20 chains. We treated the first 1,000 iterations as a burn-in period and discarded these samples. In total, 10,000 samples were used to estimate the joint posterior distribution.

Analysis I: Optimization

The key to the success of any VB algorithm is the efficiency with which a variational distribution q is morphed into the target distribution (i.e., the posterior in our case). Because the posterior distribution is fixed, albeit unknown, and the functional form of the variational distribution q is fixed but mutable via λ , VB is an optimization problem with respect to λ . Nearly all VB algorithms use gradient-based algorithms to maximize $\Omega(\lambda)$ in Equation 7. However, gradient-based optimization is sensitive to initial values and often gets stuck in local extrema and saddle-points. Furthermore, the Monte Carlo approximation of the gradient of $\Omega(\lambda)$ can often be too noisy to be useful (Ranganath, Gerrish, & Blei, 2014; M.-N. Tran, Nott, & Kohn, 2017). In this article, we examine the utility of a population-based optimization strategy called differential evolution (DE; ter Braak, 2006; Storn & Price, 1997) instead of gradient ascent. To do this, we will compare our DE-based optimization method against the widely accessible algorithm automatic differentiation variational inference (ADVI; Kucukelbir, Tran, Ranganath, Gelman, & Blei, 2017) used by common statistical software packages such as Stan (Carpenter et al., 2017) in terms of accuracy, computational consistency, and computation time. Both algorithms will use the same analytic expressions for the likelihood, prior, and functional form for q , and so their only difference is in how updates are made when maximizing $\Omega(\lambda)$.

ADVI

As most VB algorithms rely on gradient ascent, early implementations required model specific optimization functions, which were tedious and ungeneralizable. Recent advances have endeavored to develop more general “black-box” VB algorithms (Blei et al., 2017; Kucukelbir et al., 2017; Ranganath et al., 2014; M.-N. Tran et al., 2017). These black-box algorithms are so named because of their agnostic approach to optimization that does not depend on the specific model configuration, and can be used in a modular manner (i.e., the same algorithm can be used for different q functions and different models). To maximize $\Omega(\lambda)$, extent black-box VB algorithms define specific rules about how each λ parameter should be updated from one iteration to the next. For example, given the current status of a parameter vector λ^i on iteration i , we could update λ^i for the next iteration according to the following:

$$\lambda^{i+1} = \lambda^i + \alpha \nabla \hat{\Omega}(\lambda^i), \quad (9)$$

where α is a tuning parameter intended to control the speed of convergence, and ∇ is a gradient operator. The gradient is a vector of the first order partial derivatives with respect to the vector λ . Although Equation 9 is a deterministic function, because the right-hand term is a Monte Carlo approximation of the true gradient in Equation 7, the path that λ_i takes over time is noisy.

The simple updating scheme in Equation 9 ensures that changes in λ are always along the path that is proportional to

the gradient of the objective function. For example, the left panel of Figure 2 illustrates how updates to λ might work on a two-dimensional estimation problem. The gradient, illustrated as the vector field across the parameter space, details the overall landscape of the objective $\Omega(\lambda)$ as a function of λ , where blue regions provide larger values of $\Omega(\lambda)$. The vector field map shows how updates across λ would locally improve the estimates of λ such that higher $\Omega(\lambda)$ values were obtained, yet there is no guarantee that the local improvements would result in global improvements. In other words, gradient ascent can sometimes converge to local extrema depending on initial starting values (i.e., λ_0).

ADVI (Kucukelbir et al., 2017) is a highly accessible variant within the family of gradient-based black-box VB algorithms, as it is built into Stan and automates the implementation of VB. ADVI automates the derivation of $\Omega(\lambda)$'s gradient, transformation of parameters, and tuning of the algorithm. While the algorithm in the current version of Stan (RStan 2.18.2 as of November 2018) is still listed as experimental, it has been implemented in Stan since 2015, and is the most accessible VB algorithm to practitioners and experimenters in psychology. Additionally, we chose to use ADVI to minimize the potential implementation errors that could have led to confounds in our analyses. Because we explicitly specified all parameter transformations (see “prior specification”) in our model before using ADVI, any differences between ADVI and the algorithm using DE below will be due to the efficiency

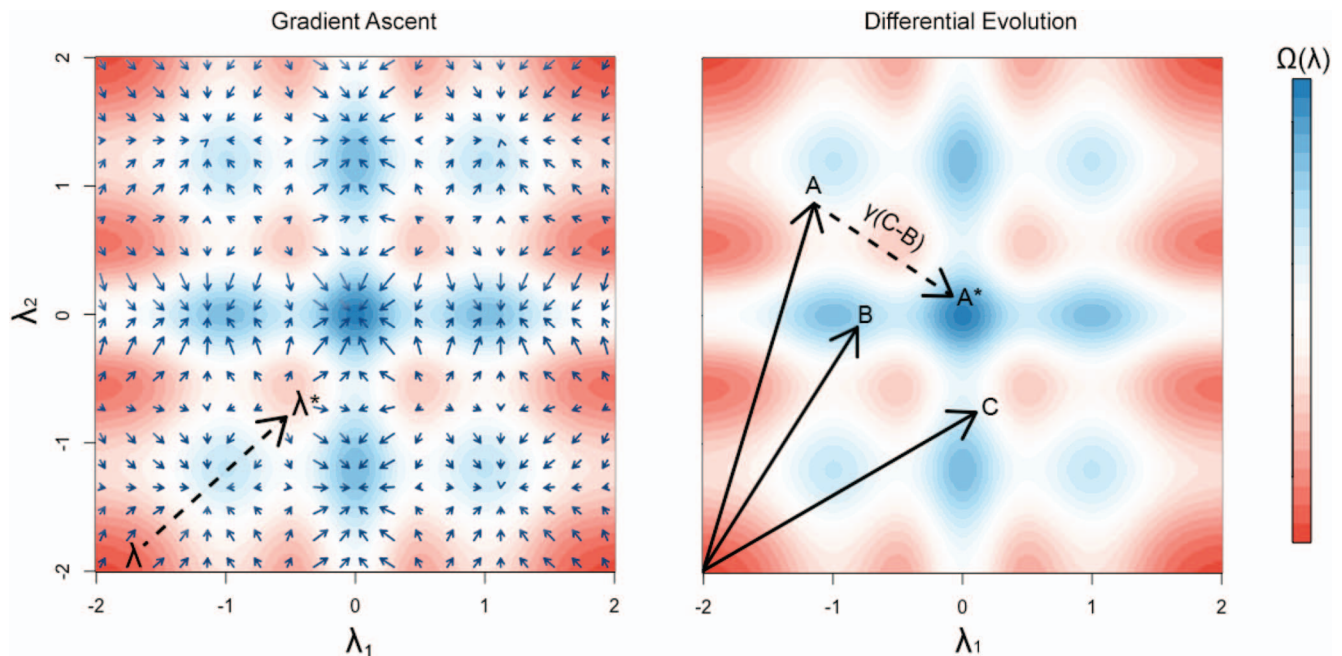


Figure 2. Illustrations of optimization algorithms. Each panel illustrates potential paths that algorithms might take when optimizing a particularly complicated objective function $\Omega(\lambda)$ = gradient ascent (left) and differential evolution (right). The global max of $\Omega(\lambda)$ is located at (0, 0), and blue colors reflect larger (i.e., better) values of $\Omega(\lambda)$. In the left panel, arrows depict the vector field of the objective function in terms of magnitude (i.e., length of the arrow) and direction (i.e., orientation). In the right panel, proposals are generated based on a consultation with other particles in the system, allowing the proposals to traverse the parameter space in a more informed way. The * denotes the proposal vector an algorithm generates. See the online article for the color version of this figure.

of the updating process for λ . In this article, the LBA was fit using the ADVI code modified from the original Stan implementation, which is released in [Annis et al. \(2017\)](#).

DEVI

Differential evolution (DE) is a population-based, evolutionary algorithm that was designed to deal with noisy real-valued objective functions ([Elsayed & Sarker, 2016](#); [Storn & Price, 1997](#)). DE updates a population of candidate solutions, each of which is referred to as a “particle,” by generating new proposals with the weighted difference between two other members of the population, similar to particle swarm optimization. The primary advantages of DE is that it does not require well-behaved objective functions, and so optimization does not depend on convexity, differentiability, or continuity. Instead, DE approximates the shape of the objective function by simply computing pairwise differences between the particles, and these differences become continually better approximations of the true objective function as the group of particles evolve. Although DE shares some similarities with the highly successful natural gradient approach ([Amari, 1998](#); [Blei et al., 2017](#); [Ranganath et al., 2014](#); [M.-N. Tran et al., 2017](#)), because it relies on a coarse particle-based approximation, it does not require the analytic derivation or numerical approximation of the gradient. By contrast, because DE is a population-based algorithm, it is less susceptible to problems of local extrema, and saddle points. However, there is still no asymptotic guarantee for convergence ([Elsayed & Sarker, 2016](#)).

While there are innumerable articles on VB using gradient-based optimization, to our knowledge, there is only one other published work using DE within VB algorithms. [Wang, Xia, and Feng \(2011\)](#) fit Gaussian mixture models with a DE-based variational expectation-maximization algorithm to segment brain images. Their application focused on unsupervised learning and prediction, and their DE implementation has some differences with the specific approach we detail below. Although we do find these differences to be interesting, as our primary motivation is to compare the basic DE approach with ADVI, we do not compare our DE approach with the algorithm used in [Wang et al. \(2011\)](#). For clarity, we refer to our specific flavor of DE-based VB as differential evolution variational inference (DEVI). In line with other DE algorithms for performing MCMC ([Turner & Sederberg, 2012](#); [Turner, Sederberg et al., 2013](#)), DEVI is comprised of three steps: crossover, purification, and migration (see [Figure 3](#)).

Crossover. DE constructs a population of J particles, each containing a candidate solution for λ . These particles communicate with each other to search the objective function and over time, gravitate toward values of λ that improve the objective function $\Omega(\lambda)$. In DEVI, the crossover step is the primary means for communication. Each particle is a K -dimensional vector containing candidate values for λ . We can represent our population as a $(K \times J)$ dimensional matrix λ . Each j -th column of the matrix, $\lambda_{1:K,j}$ is a candidate solution vector from the population. Each k -th row of the matrix, $\lambda_{k,1:J}$ is the current population of solutions for one dimension of q , namely the parameter set λ_k . The matrix λ evolves over iterations, and so we refer to the set of particles on the i th iteration as λ^i . On iteration i , when proposing a new value $\lambda_{1:K,j}^*$ as a potential

update to the vector $\lambda_{1:K,j}$, we sample two unique particles a and b such that $a \neq b \neq j$, and

$$\lambda_{1:K,j}^* = \lambda_{1:K,j}^{i-1} + \gamma(\lambda_{1:K,a}^{i-1} - \lambda_{1:K,b}^{i-1}) + \eta, \quad (10)$$

where γ is a positive value that scales the distance between the a th and b th particle, and η is a zero-centered symmetrically distributed random variable. For example, a common assumption is that $\eta \sim U(-c, c)$ where c is a small value relative to the width of the target distribution (e.g., $c = 0.001$).

Although [Equation 10](#) details how proposals are generated, not all proposals should be taken seriously. To decide whether or not $\lambda_{1:K,j}^*$ should be replaced with $\lambda_{1:K,j}^*$, we must define an acceptance rule. In a typical Bayesian application of DE ([ter Braak, 2006](#); [Turner, Sederberg et al., 2013](#)), we would define a rule for acceptance based on the Metropolis-Hastings probability. However, as VB is an optimization problem and not a posterior sampling problem, we can instead define the following greedy, deterministic rule:

$$\lambda_{1:K,j}^i = \begin{cases} \lambda_{1:K,j}^* & \text{if } \hat{\Omega}(\lambda_{1:K,j}^*) > \hat{\Omega}(\lambda_{1:K,j}^{i-1}) \\ \lambda_{1:K,j}^i & \text{if } \hat{\Omega}(\lambda_{1:K,j}^*) \leq \hat{\Omega}(\lambda_{1:K,j}^{i-1}) \end{cases}. \quad (11)$$

Because each iteration should update all J particles and these updates do not depend on one another, the pool-update operation can be parallelized and we recommend this procedure for efficient implementation.

Purification. Computationally, it is far more efficient to store the value of the objective function associated with each particle. For example, to evaluate [Equation 11](#), we need both $\hat{\Omega}(\lambda_{1:K,j}^*)$ and $\hat{\Omega}(\lambda_{1:K,j}^i)$. However, because of the recursive structure of the updates, we have inevitably computed $\hat{\Omega}(\lambda_{1:K,j}^i)$ at some iteration in the past. Hence, if we simply stored the value of the objective function by setting $W_j^i = \hat{\Omega}(\lambda_{1:K,j}^i)$, [Equation 11](#) would only require one calculation of the objective function, not two. Note that the “weight” matrix W_j^i does not have an index for the K parameters, as the vector of parameters corresponds to a single weight.

It is also important to realize that [Equation 8](#) has Monte Carlo error associated with it, making it somewhat unstable. In other words, by approximating [Equation 7](#) with random draws from q , and evaluation of [Equation 11](#), different weights can be obtained even when the exact same value for λ is used. This feature of [Equation 8](#) makes optimization tricky because some particles $\lambda_{1:K,j}^i$ may have spuriously high weights W_j^i associated with them.

As a remedy, we advocate for a “purification” move ([Holmes, 2015](#)) to avoid spuriously high particle weights. The implementation of purification is quite simple. Periodically (e.g., every 10 iterations), we simply recompute a subset of the weights associated with the pool of particles, and update their values. Let R be the number of particles to purify, and let \mathcal{R} be the set of particle indices such that $\mathcal{R} = \{r_1, \dots, r_R\}$ where $r_q \sim \{1, \dots, J\} \forall q \in \{1, \dots, R\}$ without replacement. A purification move is made by setting $W_{r_q}^i = \hat{\Omega}(\lambda_{1:K,r_q}^i)$.

Migration. Although the crossover step is a very efficient means of proposal generation, it can perform quite poorly when the individual particles are initialized poorly. A similar problem occurs when a single particle or a minority of the particles get

“stuck” in regions of the parameter space that are far from the target density (referred to as “outlier” particles). When updating nonoutlier particles, the outlier particles are still selected with uniform probability, making the difference vector that generates the new proposal large. As a consequence, the new proposal tends to land in worse areas of the parameter space, resulting in rejection of the new proposal. Similarly, when updating the outlier particles, the difference vectors will tend to be too small to move the outlier chain back into appropriate regions of the parameter space.

To remedy the problem of outlier particles, [Hu and Tsui \(2005\)](#) proposed a migration step taken from the distributed genetic algorithm framework ([Tanese, 1989](#)) to efficiently circulate the states of the particles. The idea is to propose a jump from one particle’s current state to another particle’s current state. The proposal often includes multiple particle states being swapped in a cyclical fashion, so that if three particles are selected, Particle 1 moves to Particle 2’s location, Particle 2 moves to Particle 3’s location, and Particle 3 moves to Particle 1’s location.

Following [Turner and Sederberg \(2012\)](#) and [Turner, Sederberg et al. \(2013\)](#), we can use migration to propose particle swaps and avoid outlier chains and local extrema. Using similar notation

as in the purification section, let R be the number of particles to propose a swap, and let \mathcal{R} be the set of particle indices such that $\mathcal{R} = \{r_1, \dots, r_R\}$ where $r_q \sim \{1, \dots, J\} \forall q \in \{1, \dots, R\}$ without replacement. We can then construct a cyclical chain of proposed swaps such as

$$\lambda_{1:K,r_1}^i \leftarrow \lambda_{1:K,r_R}^i$$

$$\lambda_{1:K,r_2}^i \leftarrow \lambda_{1:K,r_1}^i$$

...

$$\lambda_{1:K,r_R}^i \leftarrow \lambda_{1:K,r_{R-1}}^i$$

However, the proposed sequence of swaps is only a proposal. To determine if any of the proposed swaps should be executed, we use the deterministic acceptance rule in [Equation 11](#).

Algorithm Settings

Specification of q . Following convention, we used the mean-field approximation and factorized the variational distribution q by

Algorithm 1: Differential Evolution Variational Inference

initialize λ^1 and Ω^1 ;

for $i \leftarrow 2$ to I do

 for $j \leftarrow 1$ to J do

 Sample 2 particles, a and b , such that $a \neq b \neq j$;

$$\lambda_{1:K,j}^* = \lambda_{1:K,j}^{i-1} + \gamma(\lambda_{1:K,a}^{i-1} - \lambda_{1:K,b}^{i-1}) + \eta$$

 for $s \leftarrow 1$ to S do

$\theta_s \sim q(\lambda_{1:K,j}^*)$ //sample from q

$$\hat{\Omega}^* = \hat{\Omega}^* + \frac{1}{S} [\log \pi(x|\theta_s) + \log \pi(\theta_s) - \log q(\theta_s|\lambda_{1:K,j}^*)]$$

 end

 if $\hat{\Omega}^* > \hat{\Omega}_j^{i-1}$ then

$\hat{\Omega}_j^i = \hat{\Omega}^*$ and $\lambda_{1:K,j}^i = \lambda_{1:K,j}^*$

 else

$\hat{\Omega}_j^i = \hat{\Omega}_j^{i-1}$ and $\lambda_{1:K,j}^i = \lambda_{1:K,j}^{i-1}$

 end

end

$u \sim U(0, 1)$

if $u > \text{migrationProbability}$ then

 | $\text{Migrate}(\lambda^i, \hat{\Omega}^i)$

end

if purifyTime then

 | $\text{Purify}(\lambda^i, \hat{\Omega}^i)$

end

end

Figure 3. Differential evolution variational inference. Pseudocode for a basic scheme of the DEVI algorithm is presented.

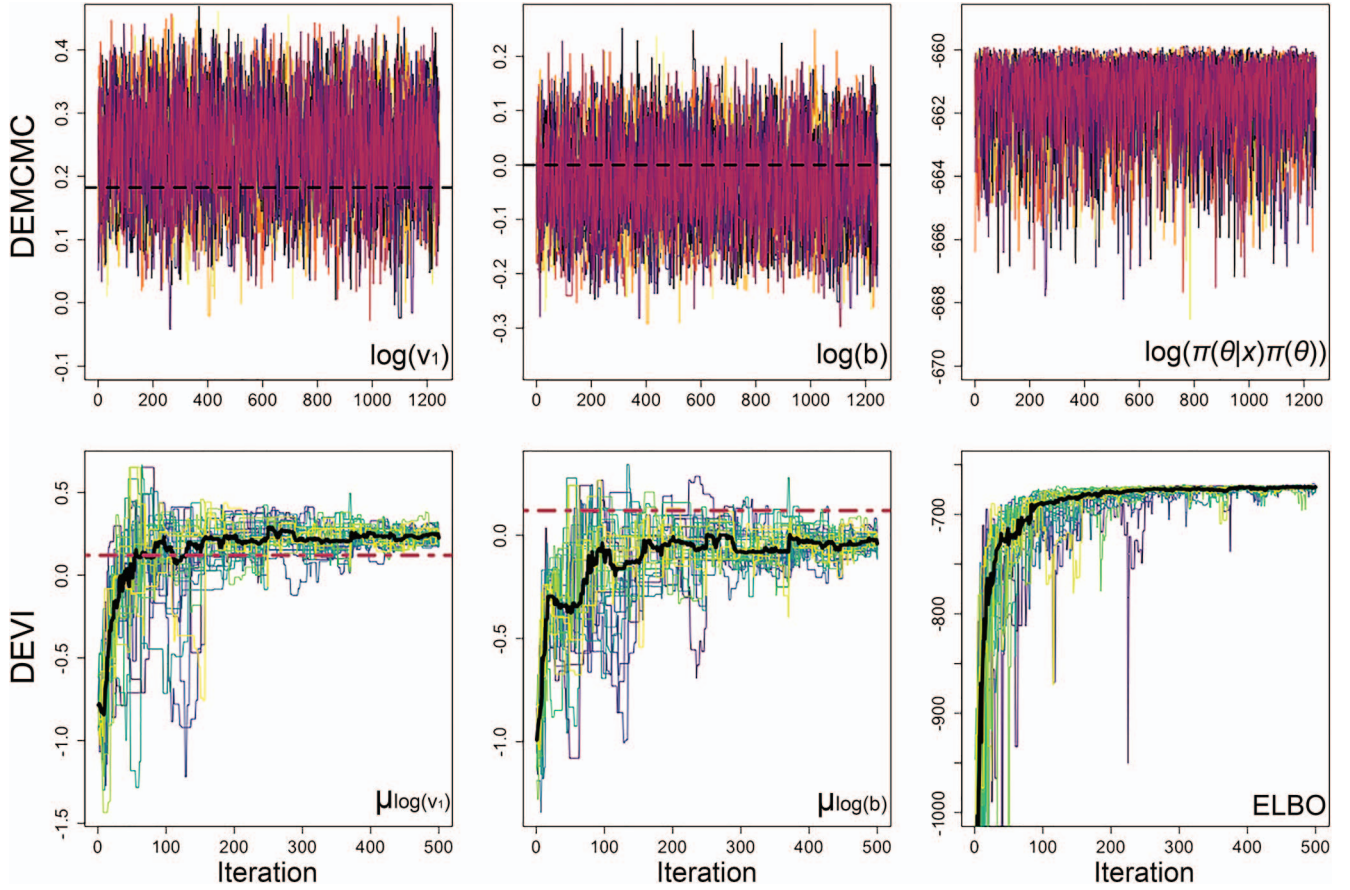


Figure 4. Particle traceplots. Trace plots of DEMCMC chains for two LBA parameters and the log of the unnormalized posterior sample densities (top three panels) and the trajectory of DEVI particles for two variational means and the ELBO (bottom three panels). For the DEMCMC trace plots, the true parameter values are denoted by a dashed, black line. For the DEVI trace plots, the median particle value for an iteration is shown by black line and the benchmark posterior mean is shown with a dashed magenta line. DEMCMC = differential-evolution Markov chain Monte Carlo; DEVI = differential evolution variational inference; ELBO = evidence lower bound; LBA = linear ballistic accumulator. See the online article for the color version of this figure.

each parameter dimension. Hence, a separate q_i function was used corresponding to each LBA model parameter $\theta = \{\log(b), \text{logit}(y), \log(v_1), \log(v_2), \text{logit}(z)\}$. We further assumed that each variational distribution was from the normal family, such that

$$q(\theta|\lambda) = \prod_{i=1}^d N(\theta_i | \mu_{\theta_i}, \exp(\zeta_{\theta_i})).$$

Here, ζ_{θ_i} is a log-transformation of the typical standard deviation σ_{θ_i} (i.e., $\zeta_{\theta_i} = \log(\sigma_{\theta_i})$). The logarithmic transformation allows the optimization algorithms to search the space of ζ_{θ_i} unconstrained because ζ_{θ_i} has infinite support. Considering the set of model parameters to be estimated, we have two variational parameters per dimension: $\lambda_i = \{\mu_{\theta_i}, \zeta_{\theta_i}\}$. Hence, the problem each VB algorithm faces is the optimization of Equation 7 with respect to $\lambda = \{\lambda_1, \dots, \lambda_d\}$.

Variational inference with differential evolution (DEVI). We approximated the joint posterior distribution for the LBA model by running DEVI for 500 iterations. We used 30 particles. For each particle, we used six samples to approximate $\Omega(\lambda)$ in

Equation 8. Hence, Equation 8 was evaluated 180 times per each crossover step. To provide the final estimate of λ , we calculated the mean of the particle states across the last 10 iterations, such that

$$\hat{\lambda}_{1:K} = \frac{1}{10J} \sum_j \sum_{i \in [491:500]} \lambda_{1:K,j}^i.$$

Within each crossover step, we set the scaling parameter $\gamma = \frac{2.38}{\sqrt{4d}}$ and $\eta \sim U(-.001, .001)$.¹ On each iteration, we performed a migration step with probability of 0.10. We performed a purification step every five iterations. After DEVI reached its maximum iteration, we assessed the median ELBO of the particles convergence (i.e., convergence to a singular point) through visual inspection. See Figure 4 for examples of converged chains for DEVI and DEMCMC. Except in rare cases, convergence occurred much earlier than the maximum iteration (e.g., around 150 iterations).

¹ Note we use $4d$ (in contrast to $2d$ for DEMCMC) because the dimension of the mean-field posterior is twice that of the model.

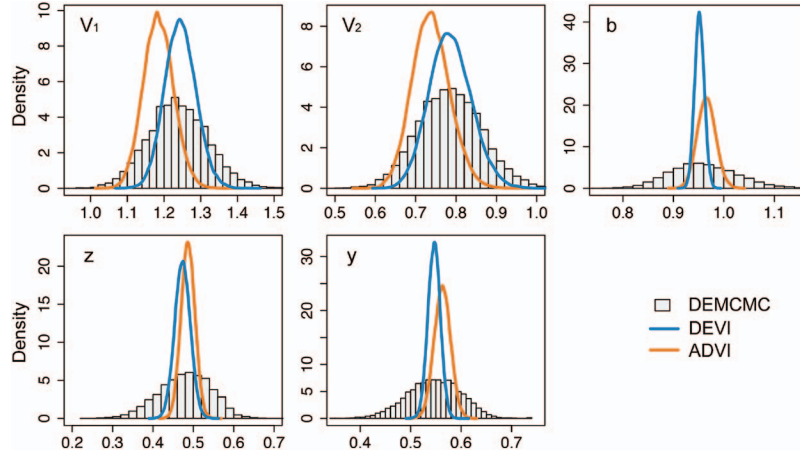


Figure 5. Estimated marginal posterior distributions. Each panel corresponds to the estimated marginal posterior distribution of the LBA model’s parameters obtained using different algorithms: benchmark estimate (DEMCMC; histograms), differential evolution variational inference (DEVI; blue lines), and automatic differentiation variational inference (ADVI; orange lines). DEMCMC = differential-evolution Markov chain Monte Carlo; LBA = linear ballistic accumulator. See the online article for the color version of this figure.

Variational inference with gradient ascent (ADVI). Where possible, we used the default settings provided in Stan when using the ADVI algorithm. Specifically, we used the default stopping criterion of a 0.01 change in the relative value of $\hat{\Omega}$ from one iteration to the next, and this criterion was calculated every 100 iterations. We set the number of samples for approximating $\nabla \Omega$ to 150, and so it had a comparable number of samples when approximating Equation 8 compared with the DEVI algorithm while still maintaining a significant computational advantage over DEMCMC.²

Results

As an initial evaluation, Figure 5 shows the estimated marginal posterior distributions for each of the five model parameters obtained using the two VB algorithms (lines) against the benchmark estimates obtained using DEMCMC (histograms). The blue lines correspond to the DEVI algorithm, whereas the orange lines correspond to the ADVI algorithm. Across all panels, the two VB algorithms center on the appropriate mean of each posterior, and show generally good agreement with the possible exception of v_1 and v_2 . However, the most striking trend is the VB algorithms’ poor recovery of the marginal posterior variances. This underestimation of posterior variance is a known problem associated with VB algorithms, and is related to the KL divergence and the appropriateness of the mean-field approximation. For the LBA model here, one could argue that the mean-field approximation is a poor assumption, given the documented dependencies among the model’s parameters (Turner, Sederberg et al., 2013). We will return to this problem in our second analysis below.

Although Figure 5 provides a qualitative evaluation of each algorithm’s accuracy, it does not provide a sense of either the robustness or the computational consistency of each algorithm. To assess robustness and computational consistency, we refit DEVI and ADVI to the data 25 times, and calculated the resulting mean estimate μ_{θ_i} for each parameter dimension. The variability in the

VB approximations comes from both the noise in the optimization procedure (i.e., stochastic search and initialization) and the stochastic nature of the objective function. As we increase the number of samples S in our approximation of the $\Omega(\lambda)$, the variability of our estimators of the $\Omega(\lambda)$ and $\nabla \Omega(\lambda)$ will decrease and likewise the variability of the distribution of VB approximations. Because we have already noted that the VB algorithms do poorly when estimating the variance of the posteriors, we did not perform a similar analyses on ζ_{θ_i} , but will address this issue in the following section. Figure 6 shows a boxplot of the distribution of means across the 25 runs obtained using DEVI (blue lines) and ADVI (orange lines). The benchmark estimate obtained with DEMCMC is shown as the black “X” symbol. Across panels, Figure 6 shows that the distribution of means is more computationally consistent (i.e., has less variance in the result obtained), and is more accurate (i.e., is closer to the benchmark) for the DEVI algorithm. The differences between the algorithms are particularly compelling for the starting point z and nondecision time y parameters, where ADVI often misestimates the mean considerably.

A final consideration is the computational burden for obtaining the posterior estimates. We performed our simulations using R on a 2.8 GHz Intel quad-core i7 processor and obtained the benchmark results using DEMCMC in 223.57 s. Across the 25 runs, DEVI took on average 32.34 s to complete with a standard deviation of 0.68, whereas ADVI (mean-field) took on average 99.76 s to complete with a standard deviation of 26.98. Hence, both algorithms are quite fast relative to the benchmark, with DEVI performing 3.08 times faster than ADVI on average. Considering both accuracy and computational speed, this set of results provides

² We reran this analysis of ADVI using 180 samples. However, our results were not sensitive to this change, therefore for brevity we do not report these results in detail.

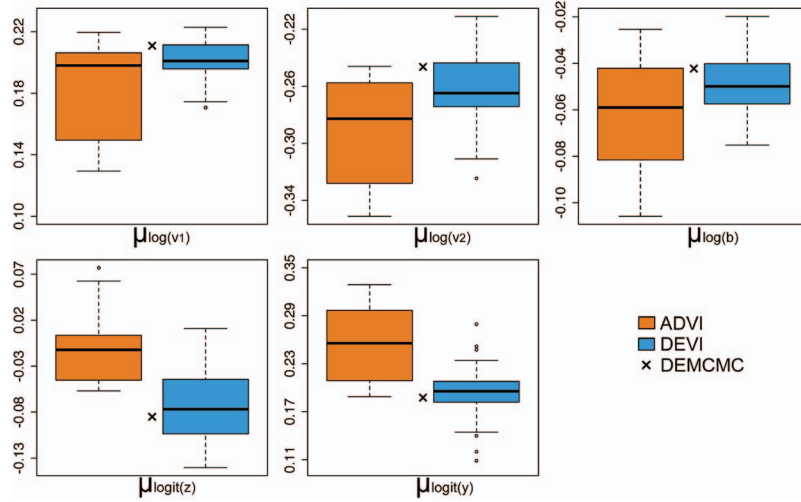


Figure 6. LBA mean estimates across runs. Each panel shows a boxplot of the mean estimate obtained using ADVI (orange) and DEVI (blue) across 25 replications. Within each panel, the benchmark mean estimate is shown as the black “X” symbol. ADVI = automatic differentiation variational inference; DEVI = differential evolution variational inference; DEMCMC = differential-evolution Markov chain Monte Carlo; LBA = linear ballistic accumulator. See the online article for the color version of this figure.

some initial evidence that DEVI is a promising alternative to gradient ascent algorithms such as ADVI.

Analysis II: Covariance Estimation

In Analysis I we saw that although the VB algorithms gave computationally consistent and accurate approximations of the marginal posterior means, they were highly inaccurate with respect to the marginal posterior variances. Given the importance of assessing the variation of effects in cognitive psychology such as individual differences, VB posterior approximations will have quite limited utility in practice. A second, but related, issue is the misestimation of the covariance between the model parameters. In using the mean-field approximation, we are explicitly assuming that the parameter dimensions are independent. In the case where independence is unwarranted, the dependence among the parameters will affect our ability to accurately estimate θ , and so our approximations become inaccurate. Unfortunately, as we have already emphasized, correlated dimensions among model parameters is the norm, not the exception, in psychological models (Turner, Dennis, & Van Zandt, 2013; Turner & Sederberg, 2014; Turner, Sederberg et al., 2013; Turner & Van Zandt, 2012, 2018).

In this section, we explore a newly developed remedial method for correcting the variance and covariance of a VB-acquired posterior estimate called linear response variational Bayes (LRVB; R. J. Giordano, Broderick, & Jordan, 2015; R. Giordano, Broderick, & Jordan, 2018). Although we explored alternative strategies such as a copula (D. Tran, Blei, & Airoldi, 2015), Gaussian process (D. Tran, Ranganath, & Blei, 2015), and a full rank multivariate Gaussian (Kucukelbir et al., 2017), these alternatives come with significantly greater computational costs because they increase the dimensionality of the optimization problem and can bias the posterior estimates (R. J. Giordano et al., 2015; R. Giordano et al., 2018). Because a full rank multivariate Gaussian q is readily available in Stan’s ADVI imple-

mentation, we chose to include it our analysis to examine its computational and statistical utility relative to the other algorithms.

Linear Response Variational Bayes

LRVB (R. J. Giordano et al., 2015; R. Giordano et al., 2018) is a post hoc covariance correction method that relies on the direct relationship between derivatives and moments (i.e., means, covariances). The basic intuition of the linear response approximation is to detect the covariance between model parameters by small (linear) perturbations (e.g., derivatives) of the log posterior density. Considering hypothetical perturbations is productive as it allows us to compare the posterior to any alternative distribution, such as our q function. To assess how small perturbations affect the posterior, we can use the cumulant generating function (i.e., the log-transformed moment generating function) of the target posterior $\pi(\theta|x)$:

$$C(t) = \log[\mathbb{E}_{\pi(\theta|x)}\{\exp(t^T\theta)\}]. \quad (12)$$

The cumulant generating function (Equation 12) is extremely useful, as we can derive characteristics of the posterior distribution (cumulants) by simply differentiating with respect to t . Importantly for us, we can derive the posterior mean (i.e., the first moment or first cumulant) $\mathbb{E}_{\pi(\theta|x)}[\theta]$ and posterior covariance (i.e., the second central moment or second cumulant) $\Sigma_{\pi(\theta|x)}$ by taking the first and second derivatives of $C(t)$ with respect to t and evaluating each at $t = 0$:

$$\mathbb{E}_{\pi(\theta|x)}[\theta] = \frac{\partial}{\partial t} C(t) \Big|_{t=0}, \quad (13)$$

$$\Sigma_{\pi(\theta|x)} = \frac{\partial^2}{\partial t \partial t^T} C(t) \Big|_{t=0}. \quad (14)$$

Now suppose that we perturb the posterior distribution on the log scale. We denote the perturbation function as $\rho(\theta, t)$, and we only consider linear perturbations or shifts of the form $\rho(\theta, t) = t^T\theta$, which

we will apply to the posterior on the log scale. These specific settings come from fundamental assumptions enabling the linear covariance approximation (R. Giordano et al., 2018), but also simplify the calculations below. Under these assumptions, we can shift the posterior along dimensions of θ , such as

$$\exp(\log(\pi(\theta|x)) + t^T\theta) = \pi(\theta|x)\exp(t^T\theta). \quad (15)$$

Equation 15 shows that by shifting the posterior in the log space, we arrive at a new distribution that often may not be properly normalized (i.e., $\int \pi(\theta|x)\exp(t^T\theta)d\theta \neq 1$). To normalize the posterior, we define

$$\begin{aligned} \pi_t(\theta|x) &:= \frac{\pi(\theta|x)\exp(t^T\theta)}{\int_{\theta^*} \pi(\theta^*|x)\exp(t^T\theta^*)d\theta^*} \\ &= \frac{\pi(\theta|x)\exp(t^T\theta)}{\mathbb{E}_{\pi(\theta|x)}\{\exp(t^T\theta)\}}. \end{aligned}$$

By log-transforming $\pi_t(\theta|x)$, we get

$$\begin{aligned} \log \pi_t(\theta|x) &= \log \frac{\pi(\theta|x)\exp(t^T\theta)}{\mathbb{E}_{\pi(\theta|x)}\{\exp(t^T\theta)\}} \\ &= \log \pi(\theta|x) + t^T\theta - \log[\mathbb{E}_{\pi(\theta|x)}\{\exp(t^T\theta)\}] \\ &\equiv \log \pi(\theta|x) + t^T\theta - C(t). \end{aligned}$$

Hence, the cumulant generating function (i.e., Equation 12) serves as the normalizing constant when we assume linear log-perturbation. This gives us the intuition that cumulants can be surmised from log-linear perturbations of the posterior density. Note that when we evaluate $\pi_t(\theta|x)$ at $t = 0$, we are guaranteed by definition that

$$\pi(\theta|x) \equiv \pi_t(\theta|x)|_{t=0}. \quad (16)$$

Now suppose that we arrive at a local minimum λ^* when using VB to estimate the target $\pi(\theta|x)$, and let $q^* = q(\theta|\lambda^*)$ for notational convenience. We can analogously express a cumulant generating function of q^* and express the expectation of q^* as

$$C_{q^*}(t) = \log[\mathbb{E}_{q^*}\{\exp(t^T\theta)\}],$$

$$\mathbb{E}_{q^*}[\theta] = \frac{\partial}{\partial t} C(t) \Big|_{t=0}.$$

If the mean-field approximation q^* is successful, then $\mathbb{E}_{q^*} \approx \mathbb{E}_{\pi(\theta|x)}$, and we should be able to find the estimate of $\Sigma_{\pi(\theta|x)}$ denoted \mathbf{S} by applying Equations 13, 14, and 16:

$$\begin{aligned} \Sigma_{\pi(\theta|x)} &= \frac{\partial^2}{\partial t \partial t^T} \{C(t)\} \Big|_{t=0} \\ &= \frac{\partial}{\partial t^T} \frac{\partial}{\partial t} \{C(t)\} \Big|_{t=0} \\ &\equiv \frac{\partial}{\partial t^T} \mathbb{E}_{\pi_t}[\theta] \Big|_{t=0} \\ &\approx \frac{\partial}{\partial t^T} \mathbb{E}_{q^*}[\theta] \Big|_{t=0} =: \mathbf{S}. \end{aligned}$$

Assuming λ^* is a local minimum and the KL divergence is twice continuously differentiable at λ^* , the Hessian of the KL divergence evaluated at $\lambda = \lambda^*$ is positive definite and thus invertible. If q is a mean-field approximation and we configure the variational parameter λ as

$$\lambda = (\mu_{\theta_1}, \dots, \mu_{\theta_k}, \zeta_{\theta_1}, \dots, \zeta_{\theta_k})^T,$$

then \mathbf{S} is the upper left submatrix of the inverse Hessian of the KL divergence (denoted $H_{\lambda\lambda}^{-1}$) with respect to λ evaluated at the local minimum λ^* (R. J. Giordano et al., 2015; R. Giordano et al., 2018):

$$\mathbf{S} \subseteq H_{\lambda\lambda}^{-1} := \left[\frac{\partial^2}{\partial \lambda \partial \lambda^T} KL\{q(\theta|\lambda) \parallel \pi(\theta|x)\} \Big|_{\lambda=\lambda^*} \right]^{-1}.$$

\mathbf{S} and the covariance of $\pi(\theta|x)$ differ only when there are at least some moments of $\pi(\theta|x)$ that q fails to accurately estimate.

Although linear response variational Bayes provides an easy way to correct the estimate of the variance-covariance matrix, this method should be used with careful attention to the assumptions of the model and its posterior. First, if one uses a mean-field approximation that is not normal, λ must be reparameterized in terms of the mean and variance. Furthermore, the LRVB posteriors are exact only when (a) q gives an exact approximation of the mean and (b) the true posterior is defined only by its first and second moments (e.g., multivariate normal distribution). Otherwise, the LRVB posteriors are simply approximations. Although these concerns are important, we have the assurance that with enough data points, the Bayesian central limit theorem guarantees that the posterior will have a multivariate normal distribution (Van der Vaart, 2000; Walker, 1969). As examples of how the LRVB derivations are performed on a univariate and multivariate normal distribution, see Appendices A and B, respectively.

In sum, LRVB leverages the practical success of mean-field approximations in approximating posterior means and uses the derivative of its log-linear perturbation to find a more accurate and (sometimes exact) covariance estimate. For more mathematical details of LRVB, we refer readers to R. Giordano, Broderick, and Jordan (2018).

Implementation

If one has already specified the KL divergence (at least up to a constant), the LRVB correction is easy to implement. To numerically approximate $H_{\lambda\lambda}$, we used the R package `numDeriv` which uses Richardson Extrapolation (Gilbert & Varadhan, 2019). For a stable approximation of the KL divergence's partial derivatives, we generated 10 quasirandom samples from q using the Sobol sequence (Dutang & Savicky, 2019; Leobacher & Pillichshammer, 2014). Quasirandom sequences produce deterministic—but representative—samples over θ , and therefore can make the approximation of Equation 8 more stable, expediting the convergence of numerical differentiation, though classical Monte Carlo is also applicable for LRVB implementation.

Additionally, we ran Stan's ADVI using a full-rank multivariate Gaussian (ADVI-FR) for q using the same settings as reported in the previous analysis for ADVI using the mean-field approximation. Supplementary code for implementing DEVI + LRVB for the LBA model is available on <https://github.com/MbCN-lab> to accelerate learning and future work.

Results

We applied the LRVB correction post hoc to our VB posterior approximations from Analysis I. Note that the LRVB correction

does not affect the mean, only the covariances. Applying the LRVB correction increased computation time by about 1.70 s. In contrast, ADVI-FR took on average 151.23 s to complete with a standard deviation of 55.48. On average, the computation time of ADVI-FR is a 49% increase in computation time compared with our standard ADVI settings, and 4.44 times slower the DEVI + LRVB. Furthermore, 12% of the 25 ADVI-FR replications were slower than our benchmark DEMCMC procedure. Figure 7 shows the benchmark posterior means (left panel) and standard deviations (middle panel) obtained using DEMCMC plotted against estimates obtained from DEVI (blue dots) and ADVI (orange dots). The right panel of Figure 7 shows the standard deviation estimates from each VB algorithm once the LRVB correction has been applied. Interestingly, following the LRVB correction, Figure 7 shows that DEVI provides better estimates of the standard deviations than does ADVI. The Pearson correlation between the DEMCMC SD and DEVI + LRVB SD was 0.991, whereas the Pearson correlation between the DEMCMC SD and ADVI + LRVB SD was 0.607. Importantly, ADVI-FR achieved nearly identical accuracy to DEVI + LRVB, with a Pearson correlation between the DEMCMC SD and ADVI-FR SD of 0.985. Therefore, for the sake of visual clarity, we chose to omit it from our figures. Between ADVI + LRVB and DEVI + LRVB, we suspect the difference arose because DEVI gave more consistent and accurate approximations of the posterior mean and variance. Specifically, four of our 25 ADVI approximations did not arrive numerically close to a local minimum of the KL divergence. The poor performance of ADVI was also reported in R. Giordano et al. (2018), where ADVI alone failed to converge to a local minimum in their applications, and secondary optimization routines were necessary. Although the authors recommend using a second-order optimization algorithm (i.e., second-order Newton trust region methods), these second-order algorithms will come with an additional computational cost. By contrast, we found that DEVI cannot only obtain faster run-times compared with the first-order algorithms ADVI, but DEVI did not require additional optimization to arrive at a local minimum. In contrast to ADVI + LRVB, ADVI-FR achieved similar accuracy to DEVI + LRVB. However, in contrast to DEVI + LRVB, ADVI-FR required much greater computation time. With respect to the number of model parameters, the dimension of ADVI-FR increases quadratically, while the dimension of

mean-field approximation increases linearly. Therefore, we expect the disparity in computation time between mean-field approximations and ADVI-FR and the similarity between the computation time of DEMCMC and ADVI-FR to increase in higher dimensional problems.

To further illustrate the quality of the LRVB corrections, Figure 8 shows the effect of the LRVB correction on the DEVI algorithm. The diagonal elements show the marginal posterior distributions, the upper diagonal elements show the pairwise joint posterior estimates, and the bottom diagonal shows the estimated raw correlation values. Across panels, blue colors correspond to the DEVI algorithm without the LRVB correction, red colors correspond to DEVI with the LRVB correction, and black corresponds to the benchmark estimate obtained with DEMCMC. Figure 8 shows that while the DEVI estimates obtain good approximations for the mean for each parameter, they fail to properly estimate the variance and covariance among the parameter dimensions. By contrast, once the DEVI estimates have been LRVB corrected (red), they are in close agreement with the benchmark posteriors, in terms of both the marginal and joint distributions.

Analysis III: Hierarchical Modeling Application

Hierarchical models are a powerful tool for psychologists, allowing for the characterization of variability in data such as subject-level differences and trial-level fluctuations in model parameters (Ahn, Krawitz, Kim, Bussemeyer, & Brown, 2013; Haaf & Rouder, 2019; Lee, 2011; Ly et al., 2017; Rodriguez et al., 2014; Wiecki, Sofer, & Frank, 2013). While powerful, they can drastically increase the dimensionality of a model, increasing the difficulty of model fitting. The widespread use of hierarchical models and their associated computational challenges make hierarchical estimation problems an interesting testbed for the variational Bayes approach. In this analysis, we explored the applicability of DEVI and LRVB to a hierarchical signal detection model by testing DEVI + LRVB's accuracy and computational advantage in estimating subject-level signal-detection parameters, hypermeans and hypervariances. We applied the model to a simulated experiment containing 10 subjects. In specifying the hierarchical SDT model from above, the number of parameters is 24, including lower and hyperlevel parameters.

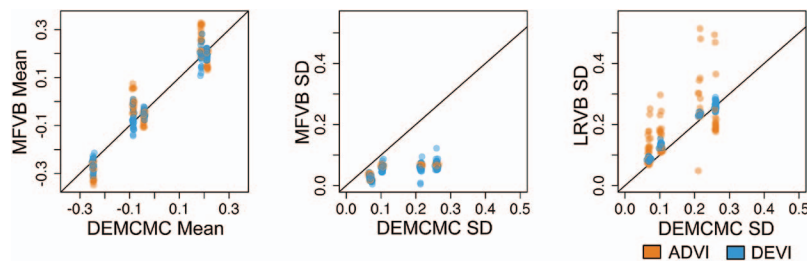


Figure 7. DEMCMC = differential-evolution Markov chain Monte Carlo; ADVI = automatic differentiation variational inference; DEVI = differential evolution variational inference; LRVB = linear response variational Bayes; MFVB = Mean Field Variational Bayes. LRVB corrected posterior statistics. Each panel compares statistical summaries of the joint posterior distribution for DEVI (blue dots) and ADVI (orange dots). Estimates for the means are shown in the left panel, whereas estimates for the standard deviations are shown in the middle panel. The right panel shows the estimates of the standard deviations after the LRVB correction. See the online article for the color version of this figure.

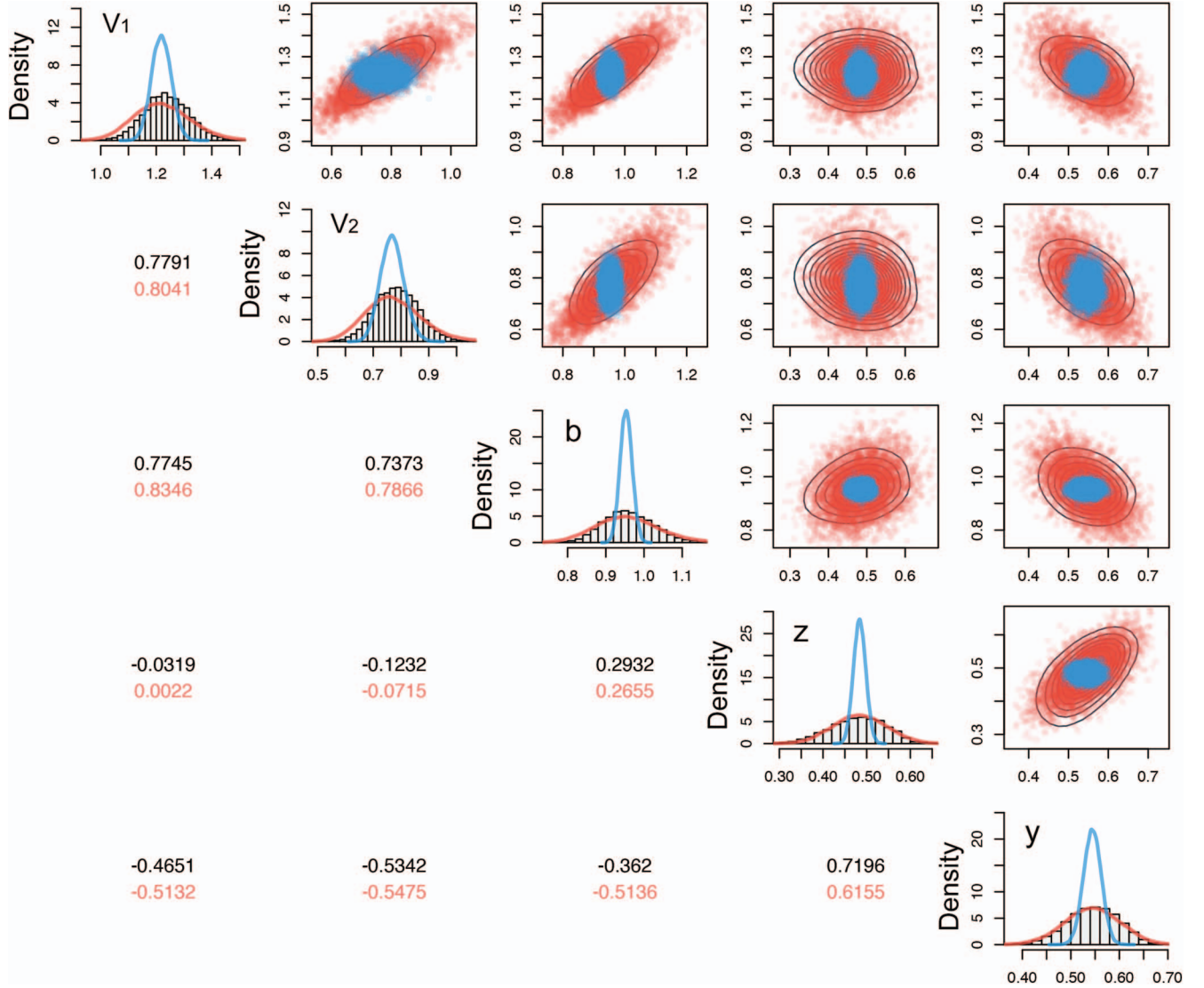


Figure 8. Estimated posterior distributions for LBA. Estimates of the joint posterior distribution obtained using DEVI with the mean-field approximation (blue) and the linear response approximation (red) are shown against the benchmark DEMCMC marginal posteriors (black contours) for the LBA model. The marginal posteriors are along the diagonal, and each pairwise joint distribution is pictured on the upper triangle. The lower triangle shows the pairwise Pearson correlations between parameters predicted by the linear response approximation (red text) and the benchmark DEMCMC posteriors (black text). See the online article for the color version of this figure.

Algorithm Settings

Specification of q . We used the mean-field approximation and factorized the variational distribution q by each parameter dimension. Hence, a separate q_i function was used corresponding to each j th subject parameters $\theta_{j,1:2} = \{B_j, \log(D_j)\}$ and each of the global parameters contained in $\phi = \{m_B, m_D, \log(s_B), \log(s_D)\}$. We'll denote the variational parameters for the k th parameter for subject j as $\lambda_{j,k,1:2} = \{\mu_{\theta_{j,k}}, \exp(\zeta_{\theta_{j,k}})\}$ and the variational parameters for the i th hyperparameter as $\psi_i = \{\mu_{\phi_i}, \exp(\zeta_{\phi_i})\}$. To be concise, we'll refer to the vector containing all variational parameters as Λ . We again assumed that each variational distribution was from the normal family, such that:

$$q(\theta, \phi | \Lambda) = \prod_{i=1}^4 [N(\phi_i | \mu_{\phi_i}, \exp(\zeta_{\phi_i}))] \prod_{j=1}^{10} \prod_{k=1}^2 N(\theta_{j,k} | \mu_{\theta_{j,k}}, \exp(\zeta_{\theta_{j,k}})).$$

Variational inference with differential evolution (DEVI). We approximated the joint posterior distribution for the HSD model by running DEVI for 300 iterations. Because a particle-based algorithm's performance is related to the number of particles relative to the dimension of the objective function (Chen, Montgomery, & Bolufé-Röhler, 2015), we increased the number of particles from 30 to 60 particles. For each particle, we used six samples to approximate $\Omega(\lambda)$ using Equation 8. Hence, Equation 8 was evaluated 360 times across all crossover steps in an iteration.

To provide the final estimate of λ , we calculated the mean of the particle states across the last 10 iterations.

Within each crossover step, we set the scaling parameter $\gamma = \frac{2.38}{\sqrt{4d}}$ and $\eta \sim U(-.001, .001)$. On each iteration, we performed a migration step with probability 0.10. We performed a purification step every five iterations. After DEVI reached its maximum iteration, we assessed convergence of the median ELBO (i.e., convergence to a singular point) through visual inspection.

LRVB implementation. To numerically approximate $H_{\lambda\lambda}$, we again used the R package `numDeriv` and 10 quasirandom samples from Sobol's sequence. Furthermore, we used the sparsity of Hessian to make its calculation more efficient (R. J. Giordano et al., 2015; R. Giordano et al., 2018). Due to the conditional independence assumption of local (subject-level) parameters, the second partial derivative of the KL divergence with respect to two local parameters for two different locations (i.e., two different subjects) is guaranteed to be 0. More explicitly, because we know

$$\frac{\partial^2}{\partial \lambda_{a,b,c} \partial \lambda_{d,e,f}} KL[q(\theta, \phi | \Lambda) \| \pi(\theta, \phi | x)] \Big|_{\Lambda=\Lambda^*} = 0 \quad \forall \quad a \neq d,$$

we do not need to waste time numerically computing derivatives along those dimensions. By taking advantage of the sparsity of $H_{\Lambda\Lambda}$, the number of partial derivatives one needs to calculate for LRVB increases linearly with respect to subjects instead of quadratically. Recall that the submatrix of $H_{\Lambda\Lambda}^{-1}$ we are interested in corresponds to second partial derivatives with respect to the mean parameters μ_{ϕ_i} and $\mu_{\theta_{j,k}}$. If we arrange Λ as follows:

$$\Lambda = \{\mu_{\theta_{1,1}}, \mu_{\theta_{1,2}}, \mu_{\theta_{2,1}}, \dots, \mu_{\theta_{10,2}}, \mu_{\phi_1}, \dots, \mu_{\phi_4}, \zeta_{\theta_{1,1}}, \zeta_{\theta_{1,2}}, \zeta_{\theta_{2,1}}, \dots, \zeta_{\theta_{10,2}}, \zeta_{\phi_1}, \dots, \zeta_{\phi_4}\},$$

then the upper left quadrant of $H_{\Lambda\Lambda}^{-1}$ will be **S**, our corrected covariance matrix approximation.

Results

We applied the LRVB correction post hoc to our VB posterior approximations obtained using DEVI and compared them to posteriors obtained using DEMCMC for the HSD model. DEVI + LRVB ran for 62.6 s, the vast majority of which was spent optimizing Equation 7. Our DEMCMC implementation took 220.2 s. DEVI + LRVB provided a 3.5 times increase in speed.

Figure 9 shows the DEMCMC posterior means and standard deviations plotted against their mean-field and linear response approximations. The Pearson's correlation between the DEVI posterior means and DEMCMC posterior means was 0.9948. The Pearson's correlation between the DEMCMC posterior means and DEVI standard deviations before and after the LRVB correction were 0.8237 and 0.9643, respectively.

Figure 10 shows a partial joint posterior plot for subject 10 and the hyperparameters. In general, the DEVI + LRVB posterior approximations were close to target DEMCMC posterior contours. The majority of mean-field approximations underestimated marginal variances, the only exception being $\log(S_B)$. Though still reasonably accurate, there was relatively more error in the approximation of posterior means of m_D , m_B , and S_D . As a result, LRVB tended to overestimate the covariance between these parameters and other model parameters. This occurred because of LRVB's reliance on the assumption that $\mathbb{E}_{q^*}[\theta] \approx \mathbb{E}_{\pi(\theta|x)}[\theta]$; Parameters

whose means were more accurately approximated had more accurate LRVB posterior approximations. In general, we found DEVI + LRVB resulted in better approximations of subject-level parameters. We speculate that the improved estimation accuracy is attributable to the size of subject-level data (750 trials per subject) relative to the number of subjects (10) in this example. The subject-level posteriors were less influenced by group-level constraint and therefore $\Omega(\Lambda)$ was less sensitive to misses in the group-level posteriors. Though not perfect, DEVI + LRVB provided a high-quality approximation of the HSD posterior in a fraction of the time.

Discussion

For a Bayesian, conclusions about the data and model of interest are often driven by the inferred posterior distribution. To trust these conclusions, computational consistency and accurate approximations are necessary. We found that a population-based algorithm like DEVI enables modelers to obtain more computationally consistent and accurate approximations of the posterior mean. However, as expected, we found that although mean-field VB algorithms provided good approximations of the posterior mean for the Linear Ballistic Accumulator model, but they failed to accurately estimate the posterior variance (i.e., the uncertainty about each model parameter). Because a primary advantage of the Bayesian framework is the ability to quantify uncertainty in our estimates, it would seem that the VB algorithms using the mean-field approximation would have limited effect in psychological research. However, after applying post hoc posterior corrections using the recently developed LRVB (R. J. Giordano et al., 2015; R. Giordano et al., 2018), we showed that VB algorithms can provide powerful solutions to the problem of fitting complex mathematical models of cognitive processes to data.

Applicability to Cognitive Modeling

Compared with Markov chain Monte Carlo, in exchange for some accuracy, VB reduces the opportunity cost of the Bayesian framework. Faster computation time can allow researchers to fit

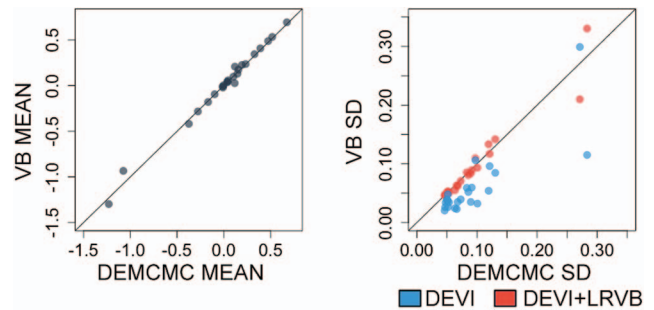


Figure 9. Hierarchical posterior approximation statistics. Each panel compares statistical summaries of the joint posterior distribution for mean-field DEVI (blue dots) and DEVI + LRVB (red dots). Estimates for the means are shown in the left panel, whereas estimates for the standard deviations (corrected and uncorrected) are shown on the right panel. DEMCMC = differential-evolution Markov chain Monte Carlo; DEVI = differential evolution variational inference; LRVB = linear response variational Bayes; VB = variational Bayes. See the online article for the color version of this figure.

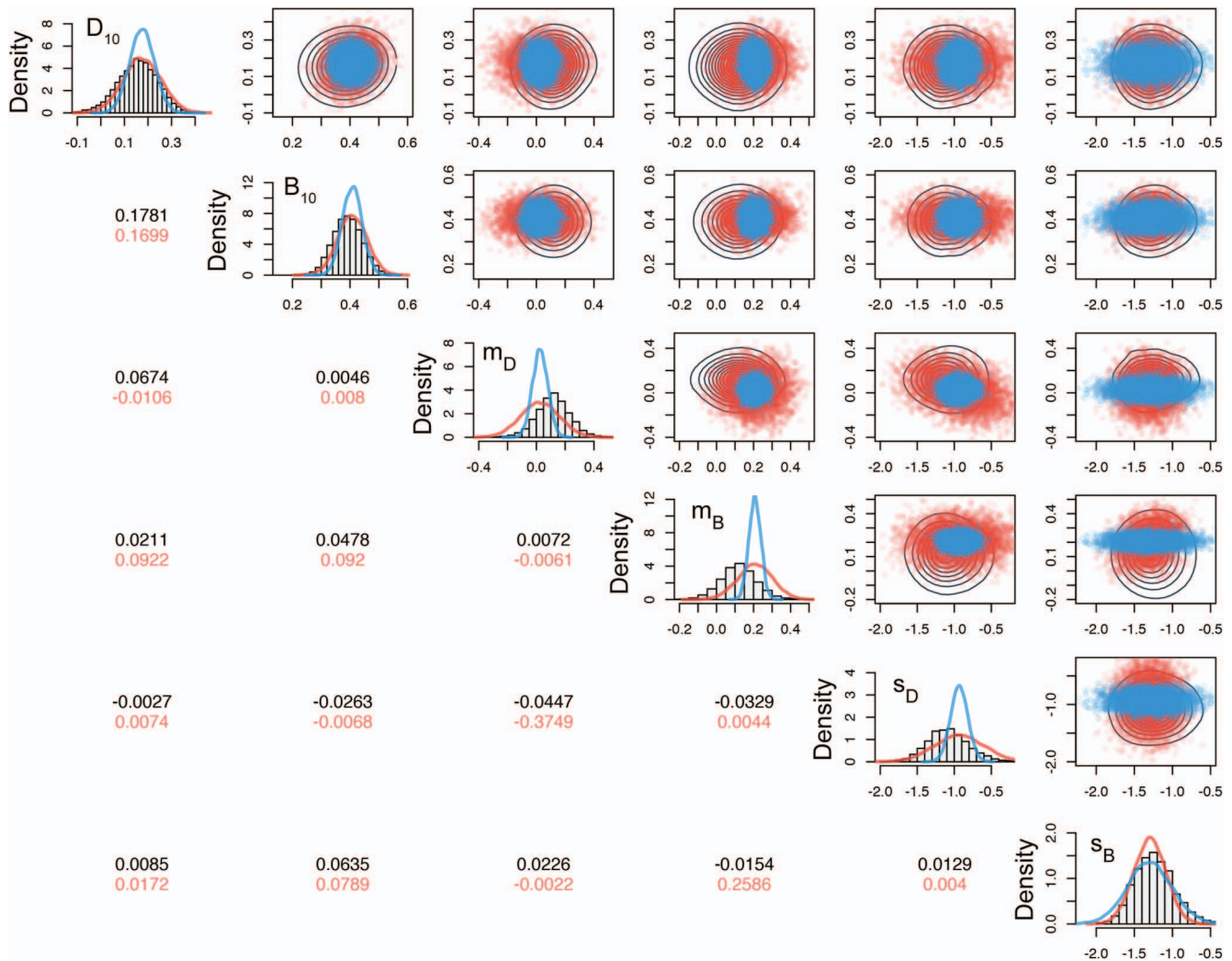


Figure 10. Estimated posterior distributions for HSD. Partial joint posterior plot for estimates obtained using DEVI with the mean-field approximation (blue) and the linear response approximation (red) are shown against the benchmark DEMCMC marginal posteriors (black contours) for the HSD model. Subject 10's individual parameters and the hyperparameter's marginal posteriors are along the diagonal, and each pairwise joint distribution is pictured on the upper triangle. The lower triangle shows the pairwise Pearson correlations between parameters predicted by the linear response approximation (red text) and the benchmark DEMCMC posteriors (black text). Posteriors were depicted on their transformed scaled. See the online article for the color version of this figure.

larger quantities of models, and also fit models that are too computationally demanding within the Bayesian framework. As such, the number of psychological research questions we can ask can grow, possibly enabling a better understanding of cognition.

As scientists, we often answer the question “What model best fits the data?”, when really the question of theoretical interest is “Why does this model best fit the data?” Ultimately, cognitive theories should be concerned with model mechanisms, not specific model parameterizations, because a model fit is insufficient for the validation of cognitive theory in and of itself. To provide guidance on the latter question, we are often required to fit many different models, systematically examining each possible combination of theoretical mechanisms. This type of “switchboard” analysis allows one to integrate out the influence of atheoretical model specifications, and examine the complex

interaction between model mechanisms (Turner, in press; Turner, Schley, Muller, & Tsetsos, 2018). However, switchboard analyses can be incredibly computationally demanding, which is why they are often—and understandably—not done. Some have implemented switchboard analyses using maximum likelihood estimates because they are relatively computational efficient (Grootswagers, Ritchie, Wardle, Heathcote, & Carlson, 2017; Heathcote, Loft, & Remington, 2015; Heathcote, Suraev et al., 2015; Palada et al., 2016; Provost & Heathcote, 2015; Rae, Heathcote, Donkin, Averell, & Brown, 2014; Strickland, Heathcote, Remington, & Loft, 2017), though this fails to take uncertainty and prior information into account. If full Bayesian switchboard analyses were to become a standard cognitive modeling approach, we believe it would be much easier to synthesize why cognitive mechanisms perform better in one context, but not in others. However,

without algorithms that can evaluate a model's credentials quickly and effectively, switchboard analyses will remain too computationally demanding to be enforced as the standard.

More conservatively, VB methods could be used as a way to initialize posterior sampling with more accurate methods such as MCMC or DEMCMC. Whereas VB methods have not yet been proven to always converge to the correct posterior distribution, MCMC methods have been given infinite time. Hence, an effective strategy for obtaining the most accurate posterior estimates in the shortest amount of time is to first use VB methods to arrive at reasonably good starting estimates. Then we can use this initial posterior estimate to generate a set of chains and obtain posterior samples using MCMC. Such an approach would eliminate the sometimes difficult task of locating good initial values as well as executing costly burn-in procedures.

Another application of VB methodology is the assessment of the robustness of the posterior to hyperparameters (e.g., the prior distribution). A naive estimation of the robustness of the posterior is to rerun the algorithm using different hyperparameters. VB makes this naive prior-sensitivity analysis feasible, as many different settings of hyperparameters can be evaluated much faster compared to MCMC methods. Like MCMC approximations, there is some variability in VB approximations. So we recommend running sensitivity analyses under a variety of prior parameter values to examine systematic trends in changes in the posterior. Better yet, $H_{\lambda\lambda}$ is directly related to the computation of sensitivity (i.e., the derivative of the posterior expectation with respect to the hyperparameters), an important concept in the Bayesian robustness literature (R. Giordano et al., 2018).

Another interesting application of VB methodology is in the context of large data sets or computationally costly approximation algorithms. In these contexts, researchers often must rely on sub-optimal procedures such as principal component analysis to interpret data or subsampling to model the data. These alternative strategies tend to fail to account for uncertainty, and are arguably less intuitive than standard Bayesian modeling approaches. Worse yet, computational constraints can force modelers to remove theoretically interesting mechanisms of a model. Importantly, VB makes the intuitive Bayesian modeling framework accessible to more data sets and model structures, and thus more researchers.

Limitations and Future Directions

While VB is well poised to accelerate theoretical developments in psychology, it should not be trivialized that there are no theoretical guarantees for accuracy—only practical ones—and so we recommend that VB methods should be used with some skepticism. DEVI also has no guarantees for convergence or accuracy, and for many applications it has little utility without the LRVB correction. Furthermore, the LRVB correction only works when the mean-field approximation accurately approximates the mean. In the cases that mean-field approximation is not accurate, LRVB still does perform better than standard mean-field and Laplace approximations, but fails to garner the near-perfect posterior approximation we illustrated for the LBA model or HSD models (R. J. Giordano et al., 2015; R. Giordano et al., 2018).

Bayes factor. In this article, we focused on posterior estimation, which is one vehicle of inference. Inference can also be driven by Bayes factors, which are more sensitive to priors. The Bayes factor is

a likelihood ratio of $\pi(x)$ for two competing hypotheses. Though to our knowledge there is no literature on calculating Bayes factors with variational Bayes, one can calculate Bayes factors using the ELBO as an estimate of $\pi(x)$, though LRVB does not directly correct the ELBO. However, one can recompute the ELBO given the LRVB-corrected covariance matrix and Equation 7. Additionally, one can merge both inference driven by posteriors and inference driven by Bayes factors using spike-and-slab priors (Rouder, Haaf, & Vandekerckhove, 2018). Though we didn't explore spike-and-slab priors here, there are VB algorithms that successfully implement spike-and-slab approaches (Goodfellow, Courville, & Bengio, 2012; Titsias & Lázaro-Gredilla, 2011).

Higher dimensions. For gradient-based VB algorithms of hierarchical models, researchers often subsample the data for each iteration to minimize the number of likelihood calculations and improve the scalability of the algorithm (Hoffman, Blei, Wang, & Paisley, 2013). While the subsampling procedure increases the noise of the gradient estimator, the estimates remain unbiased. Although we did not explore this technique here, DEVI could be implemented with a similar subsampling procedure to enable scalability for higher-dimensional, hierarchical problems.

Additionally, we only demonstrated the suitability of DEVI + LRVB for problems of moderately high dimensions (i.e., 10 and 48). Some analyses have reported that DE will have issues scaling to higher dimensions (Chen et al., 2015). Though susceptible to problems of saddle points, local minima, and inefficiencies due to noisy gradient estimates, gradient-based algorithms like ADVI (Kucukelbir et al., 2017) or black box variational inference (Ranganath et al., 2014) may be the only option for very high dimensional optimization problems. However, Elsayed and Sarker (2016) demonstrated that DE improved its scalability when used in tandem with a local search algorithm (e.g., gradient ascent) while preserving some immunity to problems associated with local search (e.g., local minima, saddle points). Algorithms combining local and population-based search are referred to as "memetic optimization" algorithms and future work should explore the utility of the memetic approach for VB.

Faster computation for LRVB. Though not implemented in this article, there are likely more efficient ways of computing $H_{\lambda\lambda}$ than numDeriv. While numDeriv is highly accessible, numDeriv computes each second partial derivative of $H_{\lambda\lambda}^{-1}$ in a serial progression. There are no explicit dependencies among these derivatives and so they could be computed in parallel. In Voglis, Hadjidoukas, Lagaris, and Papageorgiou (2009), the authors provide a software library for parallel numerical differentiation.

Additionally, one could use automatic differentiation to accelerate the computation of $H_{\lambda\lambda}$. Automatic differentiation is readily accessible or is in development for high-level languages like Python, Matlab, and R (Maclaurin, Duvenaud, & Adams, 2015; Revels, Lubin, & Papamarkou, 2016). Once the KL's gradient is derived, one can use finite differences of the gradient to approximate $H_{\lambda\lambda}$ with less computations. Lastly, for high-dimensional problems the inversion of $H_{\lambda\lambda}$ can be costly, the authors in R. J. Giordano et al. (2015) discuss methods for efficient inversion.

Misspecification of q . When using variational Bayes, we assume a specific form of approximating distribution q (e.g., the product of independent normal distributions) having no definitive idea about the target posterior. Therefore, the wrong specification of q could fail to capture the precise form of the target posterior.

So then, a natural concern is on how our posterior approximation depends on the misspecification of q .

R. Giordano et al. (2018) tested the performance of LRVB and Laplace approximations with three cases where normal distributions could not perfectly match the target distribution (i.e., a univariate skewed, a univariate overdispersed, and a bivariate overdispersed distribution) by imposing a range of perturbations to the target. In these examples, the perturbation approach justifies the linear response method above as the effect of linear perturbation (also called “tilting” in R. Giordano et al., 2018) on the expectation of the target parameter is equivalent to its variance. In general, the result showed that the LRVB variance correction reasonably approximated the variance of the target distributions and outperformed the Laplace approximation. Even in the case of bivariate overdispersed distribution where LRVB underestimated the target variance, the LRVB estimate of variance was nearer to the ground truth than the estimate from the Laplace approximation.

Although the simulation experiments presented in R. Giordano et al. (2018) do not consider all possible scenarios in which we can misspecify q , their results suggest that if the target posterior is expected to be unimodal and have only minor deviations from q (e.g., skewness, overdispersion), misspecification of q will not be detrimental to the accuracy of the LRVB approach. However, as the posterior distribution is never known in practice, we unfortunately cannot guarantee that the misspecification of q can always be remedied by LRVB. For these reasons, caveats should always exist when important interpretations of the posterior estimate are made.

Conclusion

In this article, we have demonstrated that fusing population-based optimization methods such as differential evolution with recent advances in covariance estimation, mean-field variational algorithms can be used effectively to yield highly accurate estimates of posterior distributions that are relevant to the field of psychology. If used with the proper precautions, VB algorithms can be used as a means to accelerate the advancement of cognitive models and theory.

References

- Ahn, W.-Y., Krawitz, A., Kim, W., Busemeyer, J. R., & Brown, J. W. (2013). A model-based fMRI analysis with hierarchical Bayesian parameter estimation. *Journal of Neuroscience Psychology and Economics*, 4, 95–110.
- Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10, 251–276.
- Annis, J., Miller, B. J., & Palmeri, T. J. (2017). Bayesian inference with stan: A tutorial on adding custom distributions. *Behavior Research Methods*, 49, 863–886.
- Annis, J., & Palmeri, T. J. (2019). Modeling memory dynamics in visual expertise. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 45, 1599–1618. <http://dx.doi.org/10.1037/xlm0000664>
- Attias, H. (1999). Inferring parameters and structure of latent variable models by variational Bayes. In K. B. Laskey & H. Prade (Eds.), *Proceedings of the fifteenth conference on uncertainty in artificial intelligence* (pp. 21–30). San Francisco, CA: Morgan Kaufmann Publishers Inc.
- Blei, D. M., Kucukelbir, A., & McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112, 859–877.
- Bogacz, R., Wagenmakers, E.-J., Forstmann, B. U., & Nieuwenhuis, S. (2010). The neural basis of the speed–accuracy tradeoff. *Trends in Neurosciences*, 33, 10–16.
- Brown, S. D., & Heathcote, A. (2008). The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, 57, 153–178.
- Carpenter, B., Gelman, A., Hoffman, M. D., Lee, D., Goodrich, B., Betancourt, M., . . . Riddell, A. (2017). Stan: A probabilistic programming language. *Journal of statistical software*, 76(1), 1–32.
- Casella, G., & Berger, R. L. (2002). *Statistical inference* (Vol. 2). Pacific Grove, CA: Duxbury.
- Chen, S., Montgomery, J., & Bolufé-Röhler, A. (2015). Measuring the curse of dimensionality and its effects on particle swarm optimization and differential evolution. *Applied Intelligence*, 42, 514–526.
- Chib, S., & Greenberg, E. (1995). Understanding the Metropolis-Hastings algorithm. *The American Statistician*, 49, 327–335.
- Daunizeau, J., Adam, V., & Rigoux, L. (2014). VBA: A probabilistic treatment of nonlinear models for neurobiological and behavioural data. *PLoS Computational Biology*, 1, 1–16.
- Daunizeau, J., Friston, K. J., & Kiebel, S. J. (2009). Variational Bayesian identification and prediction of stochastic nonlinear dynamic causal models. *Physica D: Nonlinear Phenomena*, 238, 2089–2118.
- Deng, L. (2012). The MNIST database of handwritten digit images for machine learning research [best of the web]. *IEEE Signal Processing Magazine*, 29, 141–142.
- Dieng, A. B., Tran, D., Ranganath, R., Paisley, J., & Blei, D. (2017). Variational inference via χ^2 upper bound minimization. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems* (Vol. 3, pp. 2732–2741). Red Hook, NY: Curran Associates Inc.
- Dutang, C., & Savicky, P. (2019). randtoolbox: Generating and testing random numbers. *R package version 1.30.0*. Retrieved from <https://CRAN.R-project.org/package=randtoolbox>
- Egan, J. P. (1958). Recognition memory and the operating characteristic. *USAF Operational Applications Laboratory Technical Note*, 58-51, 32.
- Elsayed, S., & Sarker, R. (2016). Differential evolution framework for big data optimization. *Memetic Computing*, 8, 17–33.
- Evans, N., Steyvers, M., & Brown, S. (2018). Modelling the covariance structure of complex data sets using cognitive models: An application to individual differences and the heritability of cognitive ability. *Cognitive Science*, 44, 1925–1944.
- Friston, K., Mattout, J., Trujillo-Barreto, N., Ashburner, J., & Penny, W. (2007). Variational free energy and the Laplace approximation. *Neuroimage*, 34, 220–234.
- Gelman, A., Stern, H. S., Carlin, J. B., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2013). *Bayesian data analysis*. London, UK: Chapman and Hall/CRC.
- Gilbert, P., & Varadhan, R. (2019). *The number package*. *R package version 2016.8-1.1*. Retrieved from <https://cran.r-project.org/package=numDeriv>
- Giordano, R. J., Broderick, T., & Jordan, M. I. (2015). Linear response methods for accurate covariance estimates from mean field variational Bayes. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28, pp. 1441–1449). Red Hook, NY: Curran Associates Inc.
- Giordano, R., Broderick, T., & Jordan, M. I. (2018). Covariances, robustness and variational Bayes. *The Journal of Machine Learning Research*, 19, 1981–2029.
- Goodfellow, I., Courville, A., & Bengio, Y. (2012). Large-scale feature learning with spike-and-slab sparse coding. *arXiv preprint arXiv:1206.6407*. Retrieved from <https://arxiv.org/abs/1206.6407>
- Green, D. M., & Swets, J. A. (1966). *Signal detection theory and psychophysics* (Vol. 1). New York, NY: Wiley.
- Griffiths, T., Steyvers, M., Blei, D., & Tenenbaum, J. (2005). Integrating topics and syntax. In Y. Weiss, B. Schölkopf, & J. C. Platt (Eds.), *Advances in neural information processing systems* (Vol. 17, pp. 537–544). Cambridge, MA: MIT Press.

- Grootswagers, T., Ritchie, J. B., Wardle, S. G., Heathcote, A., & Carlson, T. A. (2017). Asymmetric compression of representational space for object animacy categorization under degraded viewing conditions. *Journal of Cognitive Neuroscience*, 29, 1995–2010.
- Haaf, J. M., & Rouder, J. N. (2019). Some do and some don't? accounting for variability of individual difference structures. *Psychonomic Bulletin & Review*, 26, 772–789.
- Hastings, W. K. (1970). Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57, 97–109.
- Heathcote, A., Lin, Y.-S., Reynolds, A., Strickland, L., Gretton, M., & Matzke, D. (2019). Dynamic models of choice. *Behavior Research Methods*, 51, 961–985.
- Heathcote, A., Loft, S., & Remington, R. W. (2015). Slow down and remember to remember! A delay theory of prospective memory costs. *Psychological Review*, 122, 376–410.
- Heathcote, A., Surave, A., Curley, S., Gong, Q., Love, J., & Michie, P. T. (2015). Decision processes and the slowing of simple choices in schizophrenia. *Journal of Abnormal Psychology*, 124, 961–974.
- Hitchcock, D. B. (2003). A history of the Metropolis–Hastings algorithm. *The American Statistician*, 57, 254–257.
- Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *The Journal of Machine Learning Research*, 14, 1303–1347.
- Holmes, W. R. (2015). A practical guide to the probability density approximation (PDA) with improved implementation and error characterization. *Journal of Mathematical Psychology*, 68, 13–24.
- Hu, B., & Tsui, K.-W. (2005). *Distributed evolutionary Monte Carlo with applications to Bayesian analysis*. Madison: Department of Statistics, University of Wisconsin-Madison.
- Jaakkola, T. S., & Jordan, M. I. (2000). Bayesian parameter estimation via variational methods. *Statistics and Computing*, 10, 25–37.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37, 183–233.
- Kingma, D. P., & Welling, M. (2013). Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*. Retrieved from <https://arxiv.org/abs/1312.6114>
- Knoblauch, J., Jewson, J., & Damoulas, T. (2019). Generalized variational inference. *arXiv preprint arXiv:1904.02063*. Retrieved from <https://arxiv.org/abs/1904.02063>
- Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., & Blei, D. M. (2017). Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18, 430–474.
- Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22, 79–86.
- Lee, M. D. (2008). Three case studies in the Bayesian analysis of cognitive models. *Psychonomic Bulletin & Review*, 15, 1–15.
- Lee, M. D. (2011). How cognitive modeling can benefit from hierarchical Bayesian models. *Journal of Mathematical Psychology*, 55, 1–7.
- Lee, M. D., & Wagenmakers, E.-J. (2005). Bayesian statistical inference in psychology: Comment on Trafimow (2003). *Psychological Review*, 112, 662–668.
- Lee, M. D., & Wagenmakers, E.-J. (2014). *Bayesian cognitive modeling: A practical course*. Cambridge, MA: Cambridge University Press.
- Leobacher, G., & Pillichshammer, F. (2014). *Introduction to quasi-Monte Carlo integration and applications*. New York, NY: Springer.
- Ly, A., Boehm, U., Heathcote, A., Turner, B. M., Forstmann, B., Marsman, M., & Matzke, D. (2017). A flexible and efficient hierarchical Bayesian approach to the exploration of individual differences in cognitive-model-based neuroscience. *Computational Models of Brain and Behavior*. Advance online publication. <http://dx.doi.org/10.1002/9781119159193.ch34>
- Mack, M. L., Preston, A. R., & Love, B. C. (2013). Decoding the brain's algorithm for categorization from its neural implementation. *Current Biology*, 23, 2023–2027.
- Maclaurin, D., Duvenaud, D., & Adams, R. P. (2015). Autograd: Effortless gradients in Numpy. *ICML 2015 AutoML Workshop*, 238, 1–3. Retrieved from <https://indico.lal.in2p3.fr/event/2914/contributions/6483/subcontributions/180/attachments/6060/7185/automl-short.pdf>
- Mele, M. L., & Federici, S. (2012). Gaze and eye-tracking solutions for psychological research. *Cognitive Processing*, 13(S1), 261–265. <http://dx.doi.org/10.1007/s10339-012-0499-z>
- Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21, 1087–1092.
- Molloy, M. F., Galdo, M., Bahg, G., Liu, Q., & Turner, B. M. (2019). What's in a response time?: On the importance of response time measures in constraining models of context effects. *Decision*, 6, 171.
- Natesan, P., Nandakumar, R., Minka, T., & Rubright, J. D. (2016). Bayesian prior choice in IRT estimation using MCMC and variational Bayes. *Frontiers in Psychology*, 7, 1–11.
- Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2, 2.
- Ostwald, D., Kirilina, E., Starke, L., & Blankenburg, F. (2014). A tutorial on variational Bayes for latent linear stochastic time-series models. *Journal of Mathematical Psychology*, 60, 1–19.
- Palada, H., Neal, A., Vuckovic, A., Martin, R., Samuels, K., & Heathcote, A. (2016). Evidence accumulation in a complex task: Making choices about concurrent multiattribute stimuli under time pressure. *Journal of Experimental Psychology: Applied*, 22, 1–23.
- Palestro, J. J., Bahg, G., Sederberg, P. B., Lu, Z.-L., Steyvers, M., & Turner, B. M. (2018). A tutorial on joint models of neural and behavioral measures of cognition. *Journal of Mathematical Psychology*, 84, 20–48.
- Palestro, J. J., Sederberg, P. B., Osth, A. F., Van Zandt, T., & Turner, B. M. (2018). *Likelihood-free methods for cognitive science*. New York, NY: Springer.
- Penny, W., Kiebel, S., & Friston, K. (2003). Variational Bayesian inference for fMRI time series. *NeuroImage*, 19, 727–741.
- Peterson, C. (1987). A mean field theory learning algorithm for neural networks. *Complex Systems*, 1, 995–1019.
- Pleskac, T. J., & Busemeyer, J. R. (2010). Two stage dynamic signal detection theory: A dynamic and stochastic theory of confidence, choice, and response time. *Psychological Review*, 117, 864–901.
- Provost, A., & Heathcote, A. (2015). Titrating decision processes in the mental rotation task. *Psychological Review*, 122, 735–754.
- Purcell, B., Heitz, R., Cohen, J., Schall, J., Logan, G., & Palmeri, T. (2010). Neurally-constrained modeling of perceptual decision making. *Psychological Review*, 117, 1113–1143.
- Rae, B., Heathcote, A., Donkin, C., Averell, L., & Brown, S. (2014). The hare and the tortoise: Emphasizing speed can change the evidence used to make decisions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 40, 1226–1243.
- Ranganath, R., Gerrish, S., & Blei, D. (2014). Black box variational inference. In S. Kaski & J. Corander (Eds.), *Proceedings of the sevenieth conference on artificial intelligence and statistics* (pp. 814–822). Internet: PMLR.
- Ratcliff, R., & McKoon, G. (2008). The diffusion decision model: Theory and data for two-choice decision tasks. *Neural Computation*, 20, 873–922.
- Ratcliff, R., & Starns, J. (2009). Modeling confidence and response time in recognition memory. *Psychological Review*, 116, 59–83.
- Revels, J., Lubin, M., & Papamarkou, T. (2016). Forward-mode automatic differentiation in Julia. *arXiv preprint arXiv:1607.07892*. Retrieved from <https://arxiv.org/abs/1607.07892>
- Rodriguez, C. A., Turner, B. M., & McClure, S. M. (2014). Intertemporal choice as discounted value accumulation. *PLoS ONE*, 9, 1–9.
- Rouder, J. N., Haaf, J. M., & Vandekerckhove, J. (2018). Bayesian inference for psychology, part iv: Parameter estimation and Bayes factors. *Psychonomic Bulletin & Review*, 25, 102–113.

- Rouder, J. N., & Lu, J. (2005). An introduction to Bayesian hierarchical models with an application in the theory of signal detection. *Psychonomic Bulletin & Review*, 12, 573–604.
- Saha, A., Bharath, K., & Kurtek, S. (2019). A geometric variational approach to Bayesian inference. *Journal of the American Statistical Association*. Advance online publication. <http://dx.doi.org/10.1080/01621459.2019.1585253>
- Salimans, T., Kingma, D., & Welling, M. (2015). Markov chain Monte Carlo and variational inference: Bridging the gap. In F. Bach & D. Blei (Eds.), *Proceedings of the 32nd international conference on machine learning* (pp. 1218–1226). Internet: PMLR.
- Starke, L., & Ostwald, D. (2017). Variational Bayesian parameter estimation techniques for the general linear model. *Frontiers in Neuroscience*, 11, 1–23.
- Storn, R., & Price, K. (1997). Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization*, 11, 341–359.
- Strickland, L., Heathcote, A., Remington, R. W., & Loft, S. (2017). Accumulating evidence about what prospective memory costs actually reveal. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 43, 2197–2207.
- Tanese, R. (1989). *Distributed genetic algorithms* (Doctoral dissertation). Retrieved from <https://dl.acm.org/citation.cfm?id=915973>
- 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, 239–249.
- Titsias, M. K., & Lázaro-Gredilla, M. (2011). Spike and slab variational inference for multi-task and multiple kernel learning. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems* (Vol. 24, pp. 2339–2347). Red Hook, NY: Curran Associates Inc.
- Tran, D., Blei, D., & Airoldi, E. M. (2015). Copula variational inference. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, & R. Garnett (Eds.), *Advances in neural information processing systems* (Vol. 28, pp. 3564–3572). Red Hook, NY: Curran Associates Inc.
- Tran, D., Ranganath, R., & Blei, D. M. (2015). The variational gaussian process. *arXiv preprint arXiv:1511.06499*. Retrieved from <https://arxiv.org/abs/1511.06499>
- Tran, M.-N., Nott, D. J., & Kohn, R. (2017). Variational Bayes with intractable likelihood. *Journal of Computational and Graphical Statistics*, 26, 873–882.
- Turner, B. M. (in press). Toward a common representational framework for adaptation. *Psychological Review*.
- Turner, B. M., Dennis, S., & Van Zandt, T. (2013). Bayesian analysis of memory models. *Psychological Review*, 120, 667–678.
- Turner, B. M., Forstmann, B. U., Love, B. C., Palmeri, T. J., & Van Maanen, L. (2017). Approaches to analysis in model-based cognitive neuroscience. *Journal of Mathematical Psychology*, 76, 65–79.
- Turner, B. M., Rodriguez, C. A., Liu, Q., Molloy, M. F., Hoogendijk, M., & McClure, S. M. (2018). On the neural and mechanistic bases of self-control. *Cerebral Cortex*, 29, 1–19.
- Turner, B. M., Rodriguez, C. A., Norcia, T., Steyvers, M., & McClure, S. M. (2016). Why more is better: A method for simultaneously modeling EEG, fMRI, and behavior. *NeuroImage*, 128, 96–115.
- Turner, B. M., Schley, D. R., Muller, C., & Tsetsos, K. (2018). Competing theories of multialternative, multiattribute preferential choice. *Psychological Review*, 125, 329–362.
- Turner, B. M., & Sederberg, P. B. (2012). Approximate Bayesian computation with differential evolution. *Journal of Mathematical Psychology*, 56, 375–385.
- Turner, B. M., & Sederberg, P. B. (2014). A generalized, likelihood-free method for posterior estimation. *Psychonomic Bulletin & Review*, 21, 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, 368–384.
- Turner, B. M., Sederberg, P. B., & McClelland, J. L. (2016). Bayesian analysis of simulation-based models. *Journal of Mathematical Psychology*, 72, 191–199.
- Turner, B. M., Van Maanen, L., & Forstmann, B. U. (2015). Informing cognitive abstractions through neuroimaging: The neural drift diffusion model. *Psychological Review*, 122, 312–336.
- Turner, B. M., & Van Zandt, T. (2012). A tutorial on approximate Bayesian computation. *Journal of Mathematical Psychology*, 56, 69–85.
- Turner, B. M., & Van Zandt, T. (2018). Approximating Bayesian inference through model simulation. *Trends in Cognitive Science*, 22, 826–840.
- Turner, B. M., Wang, T., & Merkle, E. C. (2017). Factor analysis linking functions for simultaneously modeling neural and behavioral data. *NeuroImage*, 153, 28–48.
- Usher, M., & McClelland, J. L. (2001). The time course of perceptual choice: The leaky, competing accumulator model. *Psychological Review*, 108, 550–592.
- Van der Vaart, A. W. (2000). *Asymptotic statistics* (Vol. 3). Cambridge, MA: Cambridge University Press.
- Van de Schoot, R., Winter, S. D., Ryan, O., Zondervan-Zwijnenburg, M., & Depaoli, S. (2017). A systematic review of Bayesian articles in psychology: The last 25 years. *Psychological Methods*, 22, 217–239.
- van Maanen, L., Brown, S. D., Eichele, T., Wagenmakers, E. J., Ho, T., Serences, J., & Forstmann, B. U. (2011). Neural correlates of trial-to-trial fluctuations in response caution. *Journal of Neuroscience*, 31, 17488–17495.
- Van Ravenzwaaij, D., Cassey, P., & Brown, S. D. (2018). A simple introduction to Markov chain Monte Carlo sampling. *Psychonomic Bulletin & Review*, 25, 143–154.
- Vogl, C., Hadjidakis, P. E., Lagaris, I. E., & Papageorgiou, D. G. (2009). A numerical differentiation library exploiting parallel architectures. *Computer Physics Communications*, 180, 1404–1415.
- Walker, A. (1969). On the asymptotic behaviour of posterior distributions. *Journal of the Royal Statistical Society: Series B*, 31, 80–88.
- Wang, J., Xia, Y., & Feng, D. D. (2011, December). Differential evolution based variational Bayes inference for brain PET-CT image segmentation. In J. Patton (Ed.), *2011 international conference on digital image computing: techniques and applications* (pp. 330–334). Red Hook, NY: IEEE & Curran Associates Inc.
- Weinzierl, S. (2000). Introduction to Monte Carlo methods. *arXiv preprint hep-ph/0006269*.
- Wiecki, T. V., Sofer, I., & Frank, M. J. (2013). Hddm: Hierarchical Bayesian estimation of the drift-diffusion model in python. *Frontiers in Neuroinformatics*, 7, 1–10.
- Woolrich, M. W. (2012). Bayesian inference in fMRI. *NeuroImage*, 62, 801–810.
- Zhang, C., Shahbaba, B., & Zhao, H. (2018). Variational Hamiltonian Monte Carlo via score matching. *Bayesian Analysis*, 13, 485–506.

(Appendices follow)

Appendix A

Linear Response Variational Bayes for Univariate Normals

Suppose our variational distribution Q is a normal distribution with mean μ_Q and standard deviation σ_Q , and suppose our target distribution (e.g., the posterior distribution) P is a normal distribution with mean μ_P and standard deviation σ_P . The KL divergence $KL(Q \parallel P)$ between these distributions is

$$\begin{aligned}
 KL(Q \parallel P) &= \int_{\mathcal{X}} q(x | \mu_Q, \sigma_Q) \log \left[\frac{q(x | \mu_Q, \sigma_Q)}{p(x | \mu_P, \sigma_P)} \right] dx \\
 &= \int_{\mathcal{X}} \left[-\log(\sigma_Q) - \frac{1}{2} \left(\frac{x - \mu_Q}{\sigma_Q} \right)^2 + \log(\sigma_P) + \frac{1}{2} \left(\frac{x - \mu_P}{\sigma_P} \right)^2 \right] \\
 &\quad \times \frac{1}{\sqrt{2\pi\sigma_Q^2}} \exp \left[-\frac{1}{2} \left(\frac{x - \mu_Q}{\sigma_Q} \right)^2 \right] dx \\
 &= \mathbb{E}_Q \left\{ \log \left(\frac{\sigma_P}{\sigma_Q} \right) + \frac{1}{2} \left[\left(\frac{x - \mu_P}{\sigma_P} \right)^2 - \left(\frac{x - \mu_Q}{\sigma_Q} \right)^2 \right] \right\} \\
 &= \log \left(\frac{\sigma_P}{\sigma_Q} \right) + \frac{1}{2\sigma_P^2} \mathbb{E}_Q \{ (x - \mu_P)^2 \} - \frac{1}{2\sigma_Q^2} \mathbb{E}_Q \{ (x - \mu_Q)^2 \} \\
 &= \log \left(\frac{\sigma_P}{\sigma_Q} \right) + \frac{1}{2\sigma_P^2} \mathbb{E}_Q \{ (x - \mu_Q + \mu_Q - \mu_P)^2 \} - \frac{1}{2} \\
 &= \log \left(\frac{\sigma_P}{\sigma_Q} \right) + \frac{1}{2\sigma_P^2} [\mathbb{E}_Q \{ (x - \mu_Q)^2 \} + \mathbb{E}_Q \{ 2(x - \mu_Q)(\mu_Q - \mu_P) \} \\
 &\quad + \mathbb{E}_Q \{ (\mu_Q - \mu_P)^2 \}] - \frac{1}{2} \\
 &= \log \left(\frac{\sigma_P}{\sigma_Q} \right) + \frac{1}{2\sigma_P^2} [\sigma_Q^2 + 2(\mu_Q - \mu_P)\mathbb{E}_Q \{ (x - \mu_Q) \} + (\mu_Q - \mu_P)^2] - \frac{1}{2} \\
 &= \log \left(\frac{\sigma_P}{\sigma_Q} \right) + \frac{\sigma_Q^2 + (\mu_Q - \mu_P)^2}{2\sigma_P^2} - \frac{1}{2}.
 \end{aligned} \tag{17}$$

From our definition of the Hessian matrix of the Kullback-Leibler divergence,

$$H = \begin{bmatrix} \frac{\partial^2 KL(Q \parallel P)}{\partial \mu_Q^2} & \frac{\partial^2 KL(Q \parallel P)}{\partial \mu_Q \partial \sigma_Q} \\ \frac{\partial^2 KL(Q \parallel P)}{\partial \mu_Q \partial \sigma_Q} & \frac{\partial^2 KL(Q \parallel P)}{\partial \sigma_Q^2} \end{bmatrix}.$$

We must now evaluate the second order partial derivatives of Equation 17 with respect to the variational parameters μ_Q and σ_Q . The first order partial derivatives are as follows:

$$\begin{aligned}
 \frac{\partial KL(Q \parallel P)}{\partial \mu_Q} &= \frac{\partial}{\partial \mu_Q} \left(\frac{\mu_Q^2 - 2\mu_Q\mu_P + \mu_P^2}{2\sigma_P^2} \right) \\
 &= \frac{\mu_Q - \mu_P}{\sigma_P^2} \\
 \frac{\partial KL(Q \parallel P)}{\partial \sigma_Q} &= -\frac{1}{\sigma_Q} + \frac{\sigma_Q}{\sigma_P^2}.
 \end{aligned}$$

The second order partial derivatives within the Hessian are then

$$\begin{aligned}
 \frac{\partial^2 KL(Q \parallel P)}{\partial \mu_Q^2} &= \frac{\partial}{\partial \mu_Q} \left(\frac{\mu_Q - \mu_P}{\sigma_P^2} \right) \\
 &= \frac{1}{\sigma_P^2} \\
 \frac{\partial^2 KL(Q \parallel P)}{\partial \sigma_Q^2} &= \frac{1}{\sigma_Q^2} + \frac{1}{\sigma_P^2} \\
 \frac{\partial^2 KL(Q \parallel P)}{\partial \mu_Q \partial \sigma_Q} &= \frac{\partial}{\partial \mu_Q} \left(-\frac{1}{\sigma_Q} + \frac{\sigma_Q}{\sigma_P^2} \right) \\
 &= 0.
 \end{aligned}$$

Our final step is to obtain the upper left quadrant of the inverse Hessian matrix H^{-1} . Letting

$$H = \begin{bmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{bmatrix}$$

and using the identity

$$H^{-1} = \begin{bmatrix} (H_{11} - H_{12}H_{22}^{-1}H_{21})^{-1} & -(H_{11} - H_{12}H_{22}^{-1}H_{21})^{-1}H_{12}H_{22}^{-1} \\ -H_{22}^{-1}H_{21}(H_{11} - H_{12}H_{22}^{-1}H_{21})^{-1} & H_{22}^{-1} + H_{22}^{-1}H_{21}(H_{11} - H_{12}H_{22}^{-1}H_{21})^{-1}H_{12}H_{22}^{-1} \end{bmatrix}$$

the term we require reduces to the following:

$$\begin{aligned}
 (H_{11} - H_{12}H_{22}^{-1}H_{21})^{-1} &= \left(\frac{1}{\sigma_P^2} - 0 \right)^{-1} \\
 &= \sigma_P^2.
 \end{aligned}$$

(Appendices continue)

Appendix B

Linear Response Variational Bayes for Multivariate Normals

Suppose our variational distribution Q is a k -dimensional multivariate normal distribution with mean vector μ_Q and variance-covariance matrix Σ_Q , and suppose our target distribution (e.g., the posterior distribution) P is a k -dimensional multivariate normal distribution with mean vector μ_P and variance-covariance matrix Σ_P , such that

$$p_k(x|\mu_P, \Sigma_P) = (2\pi)^{-k/2} |\Sigma_P|^{-1/2} \exp\left[-\frac{1}{2}(x - \mu_P)^T \Sigma_P^{-1} (x - \mu_P)\right].$$

The KL divergence $KL(Q \| P)$ between these distributions can be obtained in a similar way as for the univariate case. First, note that

$$\begin{aligned} KL(Q \| P) &= \int_{\mathcal{X}} q(x|\mu_Q, \sigma_Q) \log \left[\frac{q(x|\mu_Q, \sigma_Q)}{p(x|\mu_P, \sigma_P)} \right] dx \\ &= \frac{1}{2} \log \frac{|\Sigma_P|}{|\Sigma_Q|} - \frac{1}{2} \mathbb{E}_Q \{ (x - \mu_Q)^T \Sigma_Q^{-1} (x - \mu_Q) \} \\ &\quad + \frac{1}{2} \mathbb{E}_Q \{ (x - \mu_P)^T \Sigma_P^{-1} (x - \mu_P) \}. \end{aligned}$$

At this point, we can use the fact that the trace $tr(x)$ of a matrix x allows for convenient rearrangements because it is not affected by expectations as the trace operation results in a scalar: $\mathbb{E}(x) = \mathbb{E}(tr(x))$. Hence, we can derive the following result:

$$\begin{aligned} \mathbb{E}_Q \{ (x - \mu_Q)^T \Sigma_Q^{-1} (x - \mu_Q) \} &= \mathbb{E}_Q \{ tr[(x - \mu_Q)^T \Sigma_Q^{-1} (x - \mu_Q)] \} \\ &= \mathbb{E}_Q \{ tr[(x - \mu_Q)(x - \mu_Q)^T \Sigma_Q^{-1}] \} \\ &= tr[\mathbb{E}_Q \{ (x - \mu_Q)(x - \mu_Q)^T \} \Sigma_Q^{-1}] \\ &= tr[\Sigma_Q \Sigma_Q^{-1}] \\ &= tr[I_k] \\ &= k. \end{aligned} \tag{18}$$

Similar to the trick used in the univariate case, we can add zero to the right-hand term of Equation 18 to complete the expectation:

$$\begin{aligned} \mathbb{E}_Q \{ (x - \mu_P)^T \Sigma_P^{-1} (x - \mu_P) \} &= \mathbb{E}_Q \{ (x - \mu_Q + \mu_Q - \mu_P)^T \Sigma_P^{-1} (x - \mu_Q + \mu_Q - \mu_P) \} \\ &= \mathbb{E}_Q \{ (x - \mu_Q)^T \Sigma_P^{-1} (x - \mu_Q) + (\mu_Q - \mu_P)^T \Sigma_P^{-1} (\mu_Q - \mu_P) \} \\ &\quad + \mathbb{E}_Q \{ 2(x - \mu_Q)^T \Sigma_P^{-1} (\mu_Q - \mu_P) \} \\ &= tr(\Sigma_P^{-1} \Sigma_Q) + (\mu_Q - \mu_P)^T \Sigma_P^{-1} (\mu_Q - \mu_P). \end{aligned}$$

Hence, Equation 18 becomes

$$KL(Q \| P) = \frac{1}{2} \left[\log \frac{|\Sigma_P|}{|\Sigma_Q|} - k + tr(\Sigma_P^{-1} \Sigma_Q) + (\mu_Q - \mu_P)^T \Sigma_P^{-1} (\mu_Q - \mu_P) \right].$$

If we impose the mean-field approximation on the variational distribution Q , then we can write the probability density function as

$$q(x|m, v) = \prod_k N(x_k|m_k, v_k),$$

where now the variational parameters m_k and v_k are assigned to each dimension of our parameter space. Letting $\text{diag}(v)$ denote a diagonal matrix whose entries are v_k , Equation 19 greatly simplifies, as now the variational covariance matrix $\Sigma_Q = \text{diag}(v)$:

$$\begin{aligned} KL(Q \| P) &= \frac{1}{2} \left[\log \frac{|\Sigma_P|}{|\Sigma_Q|} - k + tr(\Sigma_P^{-1} \Sigma_Q) + (\mu_Q - \mu_P)^T \Sigma_P^{-1} (\mu_Q - \mu_P) \right] \\ &= -\frac{1}{2} \log \left(\prod_k v_k \right) + \frac{1}{2} tr(\Sigma_P^{-1} \text{diag}(v)) + \frac{1}{2} m^T \Sigma_P^{-1} m \\ &\quad - m^T \Sigma_P^{-1} \mu_P + \frac{1}{2} \mu_P^T \Sigma_P^{-1} \mu_P + C, \\ &= -\frac{1}{2} \sum_k (\log v_k) + \frac{1}{2} tr(\Sigma_P^{-1} \text{diag}(v)) + \frac{1}{2} m^T \Sigma_P^{-1} m - m^T \Sigma_P^{-1} \mu_P + C, \end{aligned}$$

where C denotes a constant. To calculate the Hessian, we begin by computing the first order partial differential equations with respect to m and v :

$$\frac{\partial KL(Q \| P)}{\partial m_i} = (\Sigma_P^{-1})_{ii} m_i - (\Sigma_P^{-1} \mu_P)_i$$

$$\frac{\partial KL(Q \| P)}{\partial v_i} = \frac{1}{2} (\Sigma_P^{-1})_{ii} - \frac{1}{2v_i}.$$

The second order partial derivatives are

$$\frac{\partial^2 KL(Q \| P)}{(\partial m_i)^2} = (\Sigma_P^{-1})_{ii}$$

$$\frac{\partial^2 KL(Q \| P)}{(\partial v_i)^2} = \frac{1}{2v_i^2}$$

$$\frac{\partial^2 KL(Q \| P)}{\partial v_i \partial m_j} = 0 \quad \forall \quad i \neq j.$$

It follows then that the upper left quadrant of the inverse Hessian matrix is the inverse of the variance-covariance matrix for the target posterior distribution:

$$\begin{aligned} (H^{-1})_{11} &= (H_{11} - H_{12} H_{22}^{-1} H_{21})^{-1} \\ &= (H_{11})^{-1} \\ &= \Sigma_P. \end{aligned}$$

Received March 25, 2019

Revision received July 11, 2019

Accepted August 1, 2019 ■