Statistics and Computing (2022) 32:4
https://doi.org/10.1007/s11222-021-10062-2

Updating Variational Bayes: fast sequential posterior inference

Nathaniel Tomasetti[1] · **Catherine Forbes**[2] · **Anastasios Panagiotelis**[3]

Received: 30 August 2020 / Accepted: 24 October 2021 / Published online: 17 November 2021

Abstract
Variational Bayesian (VB) methods produce posterior inference in a time frame considerably smaller than traditional Markov Chain Monte Carlo approaches. Although the VB posterior is an approximation, it has been shown to produce good parameter estimates and predicted values when a rich classes of approximating distributions are considered. In this paper, we propose the use of recursive algorithms to update a sequence of VB posterior approximations in an online, time series setting, with the computation of each posterior update requiring only the data observed since the previous update. We show how importance sampling can be incorporated into online variational inference allowing the user to trade accuracy for a substantial increase in computational speed. The proposed methods and their properties are detailed in two separate simulation studies. Additionally, two empirical illustrations are provided, including one where a Dirichlet Process Mixture model with a novel posterior dependence structure is repeatedly updated in the context of predicting the future behaviour of vehicles on a stretch of the US Highway 101.

Keywords Importance sampling · Forecasting · Clustering · Dirichlet process mixture · Variational inference

1 Introduction

Time series data often arrive in high-frequency streams in applications that may require a response within a very short period of time. For example, self-driving vehicles may need to constantly monitor the position of each surrounding vehicle, predict or infer the behaviour of their likely human drivers, and react accordingly. In this context, the most recently received data can be highly informative for very short-term predictions, if the inferred models can be processed very quickly in an online fashion. In order to account for uncertainty in the models or predictions, Bayesian updating methods may be employed by targeting a sequence of posterior distributions, each conditioned on an expanding dataset. The computational demands of such an algorithm may be improved if the incorporation of additional data does not require the re-use of any observations that have previously been conditioned upon.

In many empirical settings, the desired Bayesian posterior distributions are not analytically tractable. In such cases, posterior inference may be obtained using Markov chain Monte Carlo (MCMC) methods, which will eventually produce a (dependent) sample from the posterior. Unfortunately, this approach typically involves relatively slow algorithms that are incompatible with the time frames demanded by streaming data. Further, while particle filtering methods for sequential posterior updating have been developed both for static parameter models (Chopin 2002) and dynamic latent variable models e.g. (Doucet et al. 2001), these available methods appear to be too slow for practical online use. This is particularly the case when they require use of the entire dataset to avoid particle degeneracy and/or when the number of inferred parameters is large. For a recent review of particle filtering methods, see Doucet and

Catherine Forbes acknowledges financial support under the Australian Research Council Discovery Grant No. DP150101728 and the National Science Foundation Grant SES-1921523.

✉ Anastasios Panagiotelis
anastasios.panagiotelis@sydney.edu.au

Nathaniel Tomasetti
nathaniel.tomasetti@coles.com.au

Catherine Forbes
catherine.forbes@monash.edu

[1] Coles Group Ltd., 800-838 Toorak Rd, Melbourne, VIC 3123, Australia

[2] Department of Econometrics and Business Statistics, Monash University, Melbourne, VIC 3800, Australia

[3] Discipline of Business Analytics, University of Sydney, Sydney, NSW 2006, Australia

Lee (2018). An alternative approach appears in Jasra et al. (2010) and Del Moral et al. (2015), who apply Approximate Bayesian Computation (ABC) for sequential posterior updating; however, this involves an embedded particle filter that similarly scales poorly to higher dimensional models. Bhattacharya and Wilson (2018) provide a sequential method to update parameter inference; however, their grid-based posterior evaluation is suitable only for low dimensions. Taking a different approach, Chen et al. (2019) learn the parameters of a so-called flow operator, a neural network that approximates a function which maps a set of particles from a posterior distribution at one time period, and additional data, to a set of particles belonging to an updated posterior distribution.

An alternative approach that has grown in popularity in the recent literature for high-dimensional models is the so-called Variational Bayes (VB) method (see Zhang et al. 2017, for a review). VB approximates the posterior with a tractable family of distributions, and chooses a member of this family by minimising a particular loss function with respect to auxiliary parameters. Early work in VB found an optimal approximation with coordinate descent algorithms for exponential family models, an approach widely known as Mean Field Variational Bayes (MFVB, see (Jordan et al. 1999; Attias 1999; Ghahramani and Beal 2000; Wainwright and Jordan 2008). Recent developments in VB consider gradient-based algorithms (Ranganath et al. 2014; Kingma and Welling 2014), which allow for a much richer class of models and approximating distributions to be utilised. These gradient-based approaches are stochastic, and target the true gradient of a given loss function with an unbiased estimator. We refer to this approach as Stochastic Variational Bayes (SVB).

There is a rich tradition of using only a subset of the complete dataset for certain aspects of VB inference, such as for gradient estimation. Hoffman et al. (2010) and Wang et al. (2011) propose MFVB algorithms for Dirichlet Process Mixture (DPM) models where the optimisation of a subset of the auxiliary parameter vector occurs through gradient-based approaches, using a subsample of the complete data at each iteration. Hoffman et al. (2013) and Titsias and Lázaro-Gredilla (2014) implement this data subsampling into the fully gradient based SVB approaches. Alternatively, Sato (2001) considers an alternative loss function defined as the expected value of the Kullback–Leibler (KL) divergence, with respect to the data generating process. Any realisation from the data generating process may be used within the MFVB coordinate descent algorithm, which is applied online with newly observed data substituted in as it becomes available. However, each of these approaches results in only a single posterior distribution conditioned on data up to some pre-specified time period T_n and do not provide a mechanism for the approximation to be updated at a later time period T_{n+1} following the availability of additional observations.

Smidl (2004) and Broderick et al. (2013) each consider VB approximations for Bayesian updating, resulting in a progressive sequence of approximate posterior distributions that each condition on data up to any given time period T_n. Their approaches update to the time T_{n+1} by substitution of the time T_n posterior with MFVB approximations, which are feasibly obtained due to assuming the model and approximation each adhere to a suitably defined exponential family form. In these special settings, MFVB is able to linearly combine the available optimally converged auxiliary parameters. While Smidl (2004) is concerned with state space models, Broderick et al. (2013) considers application to a latent Dirichlet allocation problem and shows it performs favourably compared to the approach of Hoffman et al. (2010) in terms of predictive log score and computational time.

In this paper, we formalise and extend the SVB approximation approach, developing an algorithm that we call Updating Variational Bayes (UVB). This algorithm can be seen as an application of the framework of Broderick et al. (2013) to the time series setting. UVB can be applied to sequentially update posterior distributions, and in a manner suitable for applications of streaming data. UVB treats data as arriving in a sequence, with the production of recursive, but approximate, posterior distributions obtained from conditioning on past information at nominated time points according to a Bayesian updating scheme. The approach delivers the approximate posterior distributions to the user at any desired point in time, with each new update using only the data observed since the previous update time. UVB requires an optimisation step for each update, which may be too slow for practical use in some situations.

To reduce the computational load of repeated updates, we extend UVB to a second, and completely novel algorithm, called Updating Variational Bayes with Importance Sampling (UVB-IS). Significant gains in computation speed per update can be achieved, albeit with some potential cost in gradient estimator variance and subsequently accuracy. Our proposed UVB-IS shares some similarities with the gradient estimator of Sakaya and Klami (2017); however, the important distinction is that our proposed UVB-IS is developed for the sequential updating setting. To the best of our knowledge, our approach is the first to exploit Importance Sampling in an online Variational Bayes framework.

We provide two simulation studies: a small-scale time series forecasting application, and a larger application clustering time series, to compare the approximation error of each of SVB, UVB, and UVB-IS relative to (asymptotically) exact[1] inference obtained using MCMC. We also compare

[1] Although MCMC is also approximate inference, it is exact in the asymptotic sense, in a way that variational Bayes is not. Hereafter, for brevity we will refer to MCMC as being 'exact' rather than 'asymptotically exact.'

the computational time required by each of the variational approximations and show that UVB-IS is substantially faster than either UVB or the repeated application of SVB, while incurring only a minor cost in performance, dependent on the application. We also demonstrate the application of UVB and UVB-IS to a simple hierarchical model to re-analyse the 'Eight Schools' example of Gelman et al. (1997), and measure the increased approximation error from the updating approaches relative to SVB in this setting.

Finally we demonstrate the application of UVB to the problem of updating posterior inference in the context of a DPM model. Here the aim is to provide Bayesian inference and prediction regarding the heterogeneous behaviour of 500 drivers from the New Generation Simulation dataset (FHWA 2017), according to the distribution of their lateral lane position. In this context, data arrive rapidly. We introduce a new class of dependent approximating distributions and show that the DPM model with UVB-based inference is able to provide more accurate forecasts than those achieved using a standard MFVB-based approach. UVB in this case has accuracy comparable to repeated use of (full data) SVB, but benefits from an ability to process updates sequentially as additional data arrives.

The paper is arranged as follows: in Sect. 2, we review standard VB methods and the available gradient algorithms commonly employed. In Sect. 3, we propose our main UVB approach, with the UVB-IS extension detailed in Sect. 4. Next, Sect. 5 contains simulation studies for time series data and a mixture distribution, while Sect. 6 details applications of the newly proposed methods to the Eight Schools hierarchical model of Gelman et al. (2014). UVB is applied to a vehicle DPM model in Sects. 7, and 8 concludes the paper.

2 Background on Variational Bayes

Before introducing our new algorithms for recursively updating approximations to the posterior, the main ideas associated with the implementation of an SVB approach are introduced. A more detailed description of SVB can be found in Blei et al. (2017), with further references provided therein.

The usual target of Bayesian inference is the posterior distribution for a potentially vector-valued static parameter $\boldsymbol{\theta}$, as characterised by its probability density function (pdf) denoted by $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$. Here $\boldsymbol{y}_{1:T}$ denotes data observed from time 1 to T and the posterior pdf is obtained using Bayes' theorem, given by

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) = \frac{p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})}{\int_{\boldsymbol{\theta}} p(\boldsymbol{y}_{1:T}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}}, \tag{1}$$

where $p(\boldsymbol{\theta})$ denotes the pdf for the prior distribution that characterises belief about $\boldsymbol{\theta}$ prior to the observation of $\boldsymbol{y}_{1:T}$. Although MCMC algorithms are commonly used to produce a (typically dependent) sample from this posterior distribution, these can be computationally intensive.

As an alternative to MCMC, VB aims to approximate the pdf in (1) with another density of given parametric form, denoted by $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$. Here $\boldsymbol{\lambda}$ is a vector of auxiliary parameters associated with the approximation, to be selected via optimisation. We note that the approximating density q is explicitly parameterised by $\boldsymbol{\lambda}$, and so its evaluation does not explicitly require $\boldsymbol{y}_{1:T}$ once an optimal value of $\boldsymbol{\lambda}$ has been found. However, we include the conditioning notation to reinforce that the selected $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ corresponds to an approximation of the posterior distribution in (1). This point is particularly relevant to the process of updating VB as is shown in Sect. 3.

In the SVB context, the family of the approximating density $q_{\boldsymbol{\lambda}}(y_{1:T})$ is held fixed, with the member of that family indexed by the parameter vector $\boldsymbol{\lambda}$ selected to minimise a given loss function. Typically the KL divergence Kullback and Leibler 1951) from $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ to $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$, denoted as $KL[q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \;||\; p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})]$, is used, with

$$KL[q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}) \;||\; p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})] = E_q\left[\log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})) - \log(p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}))\right]. \tag{2}$$

We note that the KL divergence is not symmetric, and reversing $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ and $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$, leads to similarities with assumed density filtering, independently proposed in statistics Lauritzen 1992) and artificial intelligence Boyen and Koller 2013; Opper and Winther 1998) and control Maybeck 1982). Often in practice, the KL divergence in (2) is intractable, with $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ only known up to a proportionality constant due to the difficulties involved in the evaluation of the integral in the denominator of (1). Nevertheless, it has been shown that an equivalent problem to minimising the KL divergence is to maximise the so-called *evidence lower bound* (ELBO Attias 1999), given by

$$\mathcal{L}(q, \boldsymbol{\lambda}) = E_q\left[\log(p(\boldsymbol{\theta}, \boldsymbol{y}_{1:T})) - \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}))\right]. \tag{3}$$

A further complication that typically arises when attempting to implement SVB is that an analytical expression for the expectation in (3) may not be available. In this case, maximisation of the ELBO may be achieved via stochastic gradient ascent (SGA, Bottou 2010). To apply SGA to the problem of maximising the ELBO, an initial value $\boldsymbol{\lambda}^{(1)}$ is selected and is recursively modified to $\boldsymbol{\lambda}^{(m)}$, for $m = 1, 2, \ldots$, according to

$$\boldsymbol{\lambda}^{(m+1)} = \boldsymbol{\lambda}^{(m)} + \rho^{(m)} \frac{\partial \widehat{\mathcal{L}(q, \boldsymbol{\lambda})}}{\partial \boldsymbol{\lambda}}\bigg|_{\boldsymbol{\lambda}=\boldsymbol{\lambda}^{(m)}} \tag{4}$$

with the converged value of $\boldsymbol{\lambda}^{(*)}$ obtained when the change from $\mathcal{L}(q, \boldsymbol{\lambda}^{(m)})$ to $\mathcal{L}(q, \boldsymbol{\lambda}^{(m+1)})$ falls below some pre-specified threshold Hoffman et al. 2013).

The adjustment term in (4) is made of two factors, the so-called *learning rate*, $\rho^{(m)}$, and an estimate of the gradient of the ELBO, $\widehat{\frac{\partial \mathcal{L}(q,\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}}$. A popular estimator of this gradient is the score-based estimator Ranganath et al. 2014), given by

$$\widehat{\frac{\partial \mathcal{L}(q,\boldsymbol{\lambda})}{\partial \boldsymbol{\lambda}}}_{SC} = \frac{1}{S}\sum_{j=1}^{S} \frac{\partial \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T}))}{\partial \boldsymbol{\lambda}} \left(\log(p(\boldsymbol{y}_{1:T}, \boldsymbol{\theta}^{(j)})) - \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T})) - \widehat{\boldsymbol{a}}\right), \tag{5}$$

where the simulated values $\{\boldsymbol{\theta}^{(j)}, \text{ for } j = 1, 2, \ldots, S\}$ are drawn from the presiding approximating density $q_{\boldsymbol{\lambda}^{(m)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$, and $\widehat{\boldsymbol{a}}$ is a vector of control variates with

$$\widehat{a}_k = \widehat{\text{Cov}}\left(\frac{\partial \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}))}{\partial \boldsymbol{\lambda}_k}\left(\log(p(\boldsymbol{y}_{1:T}, \boldsymbol{\theta})) - \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}))\right), \frac{\partial \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}))}{\partial \boldsymbol{\lambda}_k}\right) \Bigg/ \widehat{\text{Var}}\left(\frac{\partial \log(q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T}))}{\partial \boldsymbol{\lambda}_k}\right). \tag{6}$$

As (5) results in an unbiased estimator of the gradient of the ELBO, it is known that the SGA procedure will converge in probability to a local maximum Robbins and Monro 1951), provided that the learning rate sequence.[2] satisfies

$$\sum_{m=1}^{\infty} \rho^{(m)} = \infty \tag{7}$$

and

$$\sum_{m=1}^{\infty} (\rho^{(m)})^2 < \infty. \tag{8}$$

We note that although SGA is itself a recursive procedure, the result in the VB context is the one-time posterior pdf approximation $q_{\boldsymbol{\lambda}^*} \approx p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$, where $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^{(M)}$ is the optimal parameter.

2.1 Dependence in the approximation

Considering the vector $\boldsymbol{\theta} = (\theta_1, \theta_2)'$, the application of SVB often employs the so-called Mean Field approximation Bishop 2006) where the approximating distribution is factorised as

$$q_{\boldsymbol{\lambda}}(\theta_1, \theta_2|\boldsymbol{y}_{1:T}) = q_{\boldsymbol{\lambda}}(\theta_1|\boldsymbol{y}_{1:T}) q_{\boldsymbol{\lambda}}(\theta_2|\boldsymbol{y}_{1:T}). \tag{9}$$

However, SVB allows more general forms of the approximating distribution that may include dependence. In this paper, we also consider approximation families that can exploit cases where the posterior of a subset of parameters conditional on remaining parameters is known. In this case, the full posterior can be approximated by

$$q_{\boldsymbol{\lambda}}(\theta_1, \theta_2|\boldsymbol{y}_{1:T}) = q_{\boldsymbol{\lambda}}(\theta_1|\boldsymbol{y}_{1:T}) p(\theta_2|\theta_1, \boldsymbol{y}_{1:T}). \tag{10}$$

where the second term on the right-hand side is known (hence the use of p rather than q) and only θ_1 requires approximation. For many models, the posterior can be decomposed in this manner. An example is the model we consider in Sect. 7 where we explicitly include the exact conditional distribution. To our knowledge, the potential to exploit this conditional approximation structure—and in particular to include an exact component within that structure—appears to be a novel contribution to the literature.

3 Updating Variational Bayes

We now introduce the proposed algorithm for updating VB when data are observed in an online setting. Let $T_1, T_2, \ldots$ be a sequence of time points, from which a sequence of posterior distributions $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})$, $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_2})$, ..., is desired. Now suppose that the (exact) posterior distribution for the governing (static) parameter vector $\boldsymbol{\theta}$ is available, as given by its pdf $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$. Our objective is to update this posterior distribution, after observing data up to, and including, time T_{n+1}, when the additional $T_{n+1} - T_n$ data points have become available. The pdf of the resulting updated posterior distribution is denoted as $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})$. In an online setting, where new data continues to appear, we will want to repeat this updating procedure sequentially, each time updating the past posterior to reflect all of the data, including the latest available.

The usual application of Bayes' rule at a given time T_{n+1} involves a likelihood made up of T_{n+1} factors. However, with the availability of the posterior at time T_n, given by its density $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$, the updated time T_{n+1} posterior is given by

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) \propto p(\boldsymbol{y}_{T_n+1:T_{n+1}}|\boldsymbol{y}_{1:T_n}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n}), \tag{11}$$

where $p(\boldsymbol{y}_{T_n+1:T_{n+1}}|\boldsymbol{\theta}, \boldsymbol{y}_{1:T_n})$ on the right-hand side of (11) is comprised of only $T_{n+1} - T_n$ factors.

We propose the Updating Variational Bayes (UVB) algorithm for use when the evaluation of the online posterior updating is computationally demanding. Our UVB algorithm, detailed in Algorithm 1, is initialised by forming the

[2] The learning rate used for all implementations of SGA in this paper is provided by the Adaptive Moment (Adam) algorithm of Kingma and Ba (2014).

variational approximation at a given time T_1 as $q_{\boldsymbol{\lambda}_1^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})$ where

$$\boldsymbol{\lambda}_1^* = \arg\min_{\boldsymbol{\lambda}_1} \quad KL[q_{\boldsymbol{\lambda}_1}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1}) \quad || \quad p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})]. \tag{12}$$

At this first stage, we simply approximate the first posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})$ with the optimised distribution as in SVB, namely $q_{\boldsymbol{\lambda}_1^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})$. Importantly, this first update depends only on the first set of observations, $y_{1:T_1}$, through the choice of the optimal parameter, $\boldsymbol{\lambda}_1^*$.

In general then, after approximating the posterior at time T_n with $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ and observing additional data up to time T_{n+1}, UVB replaces the posterior construction described by (11) with the available approximation,

$$\widetilde{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) \propto p(\boldsymbol{y}_{T_n+1:T_{n+1}}|\boldsymbol{\theta}) q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n}). \tag{13}$$

This defines an alternate target distribution, $\widetilde{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})$, referred to as the 'pseudo-posterior' at time T_{n+1}.

The objective for each update is to find $\boldsymbol{\lambda}_{n+1}^*$ (and hence $q_{\boldsymbol{\lambda}_{n+1}^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})$) through the minimisation of the KL divergence to the corresponding pseudo-posterior, resulting in

$$\boldsymbol{\lambda}_{n+1}^* = \arg\min_{\boldsymbol{\lambda}_{n+1}} \quad KL[q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) \quad || \quad \widetilde{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})], \tag{14}$$

for each $n = 1, 2, \ldots$. The sequence of distributional families $q_{\boldsymbol{\lambda}_1}, q_{\boldsymbol{\lambda}_2}, \ldots,$ may differ at each time period, though we note it is convenient to hold the family fixed.

Algorithm 1: Updating Variational Bayes (UVB)

Input: Prior, Likelihood.
Result: Posterior approximation at T_τ.
Observe $\boldsymbol{y}_{1:T_1}$.;
Minimises $KL[q_{\boldsymbol{\lambda}_1}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1}) \quad || \quad p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})]$ using SGA via (5).;
for n *in* $1, \ldots, \tau - 1$ **do**
  Observe next data $\boldsymbol{y}_{T_n+1:T_{n+1}}$.;
  Use $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ and (13) to construct the UVB pseudo-posterior up to proportionality.;
  Minimise $KL[q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) \quad || \quad \widetilde{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})]$ using SGA via (5).;
end

We note some important features of the proposed UVB algorithm compared with an SVB implementation. First, at time T_{n+1} an SVB implementation would target the exact posterior $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) \propto p(\boldsymbol{y}_{1:T_{n+1}}|\boldsymbol{\theta})p(\boldsymbol{\theta})$ whereas UVB instead targets an alternate pseudo-posterior distribution in (13). Second, at time T_{n+1}, the evaluation for UVB corresponding to (5), which conditions on all available data $y_{T_{n+1}}$, is composed of only $T_{n+1} - T_n$ factors, since the earlier data $y_{1:T_n}$ have already been incorporated in the previous update which forms the new prior, as shown in (13). Hence, the computational complexity of UVB has rate $O(T_{n+1} - T_n)$ rather than rate $O(T_{n+1})$, i.e. computing UVB is not increasing in the number of observations for equally spaced intervals, as is the case for SVB. Third, unlike SVB, the UVB algorithm can begin even when only part of the data has been observed, making it well-suited to online applications. Further, the prevailing optimal value of $\boldsymbol{\lambda}_n$, denoted $\boldsymbol{\lambda}_n^*$, could be used as the UVB starting value for the optimisation at time T_{n+1} as long as the class of approximating distributions q is the same for each update. This may reduce the number of SGA iterations required for the UVB algorithm to converge.

While posterior parameter distributions that result from each iteration of UVB are relatively fast to compute, with each update there will likely be some loss of accuracy, particularly with regard to the tails of these distributions. However, a loss of accuracy in posterior distributions for parameters need not imply a large loss of accuracy in posterior prediction distributions, as measured by a scoring rule. This sort of finding has been seen before in other approximate inferential settings, including for SVB (see, for example (Gefang et al. 2019; Gunawan et al. 2021). We investigate the trade-off between computational speed and forecast accuracy in a simulation setting in Sect. 5.1.

For applications involving very long time series, the deterioration of the accuracy of UVB and UVB-IS relative to SVB will eventually offset any earlier computational gains. In these settings, it is advised to run either SVB or MCMC at regular intervals to 'refresh' the approximation. For instance, if updates are required every minute, perhaps a full MCMC could be run offline at the start of each day to avoid the accumulation of approximation errors. The regularity with which to run exact inference will be context specific.

Where the objective of the analysis is classification using a mixture model, the misclassification rate may be similarly robust to inaccuracies in the tail, such that the computational gains of our proposed sequential method are worthwhile. This is explored in a simulation setting in in a simulation setting in Sect. 5.2.

Before exploring these aspects, we introduce a modified approach whereby the computational speed may be further improved, albeit potentially with some additional loss in accuracy. This modified approach, referred to as UVB with Importance Sampling (UVB-IS), is described in the next section.

4 UVB with importance sampling

An application of UVB up to some time T_n involves SVB inference at time T_1 followed by $n - 1$ updates, for a total of n applications of SGA optimisation. Repeated updates may incur a significant computational overhead relative to

SVB, which applies only a single SGA algorithm using all data up to time T_n. In this section, we address this problem and explore the possibility of achieving large computational gains per update through the incorporation of ideas from importance sampling. (For a general overview of importance sampling, see Gelman et al. (2014).) Before introducing our UVB with Importance Sampling (UVB-IS) algorithm, we briefly review the incorporation of importance sampling into SGA, as introduced by Sakaya and Klami (2017).

Temporarily suppressing the subscript n on the given time period T, the m^{th} iteration in the SGA algorithm for a given target VB posterior changes $\boldsymbol{\lambda}^{(m)}$ to $\boldsymbol{\lambda}^{(m+1)}$ via S simulations of $\boldsymbol{\theta}^{(m)}$ from $q_{\boldsymbol{\lambda}^{(m)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ as per (5). For each of these simulations, the log-likelihood, log-prior, and additional terms involving $q_{\boldsymbol{\lambda}^{(m)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ must be evaluated. Note that, for large scale applications this computation is dominated by the T terms in the log-likelihood.

In the subsequent SGA iteration from $\boldsymbol{\lambda}^{(m+1)}$ to $\boldsymbol{\lambda}^{(m+2)}$, the evaluation of the log-likelihood requires a new set of S simulations $\boldsymbol{\theta}^{(m+1)}$ from $q_{\boldsymbol{\lambda}^{(m+1)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$. Sakaya and Klami (2017) note that as the change from $\boldsymbol{\lambda}^{(m)}$ to $\boldsymbol{\lambda}^{(m+1)}$ is likely to be small, the distributions $q_{\boldsymbol{\lambda}^{(m)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ and $q_{\boldsymbol{\lambda}^{(m+1)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ will likely be similar. Using this motivation, an alternative gradient estimator is suggested for each iteration $k = m+1, m+2, \ldots, m+r$ via an importance sampler that uses $q_{\boldsymbol{\lambda}^{(m)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$ as a proposal distribution, rather than generating new draws of $\boldsymbol{\theta}$ from each $q_{\boldsymbol{\lambda}^{(k)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T})$. This approach retains the set of samples $\boldsymbol{\theta}^{(m)}$ and their associated log-likelihood values, only resampling $\boldsymbol{\theta}$ and re-evaluating the corresponding log-likelihood at iteration $m+r+1$. In the SVB context, the value of r should not be taken to be too large, as substantial differences between $\boldsymbol{\lambda}^{(m)}$ and $\boldsymbol{\lambda}^{(m+r)}$ may lead to a corresponding increase in the variance of the resulting gradient estimator.

In the context of UVB, we sequentially update the posterior approximation at each time T_n via repeated applications of SGA. As before UVB-IS holds the family of the approximating distribution $q_{\boldsymbol{\lambda}}$ fixed between each update, and sets the initial value of the parameter vector at time T_{n+1} equal to the optimal value from the previous update, i.e. we set $\boldsymbol{\lambda}_{n+1}^{(1)} = \boldsymbol{\lambda}_n^*$. During the subsequent application of SGA, the sequence of parameter vectors $\boldsymbol{\lambda}_{n+1}^{(1)}, \boldsymbol{\lambda}_{n+1}^{(2)}, \ldots, \boldsymbol{\lambda}_{n+1}^*$ corresponds to a sequence of distributions moving from $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ to $q_{\boldsymbol{\lambda}_{n+1}^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})$. For repeated updates with small values of $T_{n+1} - T_n$, the new information about $\boldsymbol{\theta}$ in $\boldsymbol{y}_{T_n+1:T_{n+1}}$ will typically be relatively small, and unless there is a structural change in the data process, we expect the approximating distributions will become similar.

The above observation motivates the addition of an importance sampling gradient estimator to be applied for each update. In each update using the SGA algorithm at time T_{n+1}, all of the requisite gradients are estimated via importance sampling, using the previous UVB posterior $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ as the (identical) proposal distribution. The consequence of this approach is that only S samples of $\boldsymbol{\theta}$ are required for the entire SGA algorithm, and thus the likelihood is evaluated S times in total, rather than S times per iteration (or S times per r iterations in the case of Sakaya and Klami (2017)).

Suppressing the SGA iteration superscript index (m), the UVB-IS gradient estimator is derived from the score-based estimator implied by (5). In this case, the updated joint distribution, given by $p(\boldsymbol{y}_{T_n+1:T_{n+1}}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$, is replaced by an expression proportional to (13), with

$$\frac{\partial \mathcal{L}(q, \boldsymbol{\lambda}_{n+1})}{\partial \boldsymbol{\lambda}_{n+1}} = \int_{\boldsymbol{\theta}} q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) f(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{15}$$

where

$$f(\boldsymbol{\theta}) = \frac{\partial \log(q_{\boldsymbol{\lambda}_{n+1}}|\boldsymbol{y}_{1:T_{n+1}})}{\partial \boldsymbol{\lambda}_{n+1}} \left(\log \left(\frac{\widetilde{p}(\boldsymbol{y}_{T_n+1:T_{n+1}}, \boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})}{q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})} \right) - \widehat{\boldsymbol{a}} \right).$$

Multiplication and division of the integrand in (15) by $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ allows it to be written as an expectation with respect to $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$,

$$\frac{\partial \mathcal{L}(q, \boldsymbol{\lambda}_{n+1})}{\partial \boldsymbol{\lambda}_{n+1}} = \int_{\boldsymbol{\theta}} q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n}) \frac{q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})}{q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})} f(\boldsymbol{\theta}) d\boldsymbol{\theta}, \tag{16}$$

Hence, (16) may be estimated via a Monte Carlo average,

$$\widehat{\frac{\partial \mathcal{L}(q, \boldsymbol{\lambda}_{n+1})}{\partial \boldsymbol{\lambda}_{n+1}}}_{IS} = \frac{1}{S} \sum_{j=1}^{S} w(\boldsymbol{\theta}^{(j)}) f(\boldsymbol{\theta}^{(j)}) \tag{17}$$

since $\boldsymbol{\theta}^{(j)} \sim q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ and

$$w(\boldsymbol{\theta}^{(j)}) = \frac{q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T_{n+1}})}{q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T_n})}, \tag{18}$$

with $\widehat{\boldsymbol{a}}$ estimated as per Eq. (6).

Since only the value of $\boldsymbol{\lambda}_{n+1}$ changes in each iteration of SGA, and the S sampled values $\boldsymbol{\theta}^{(j)}$ are held fixed, only the terms involving $\boldsymbol{\lambda}_{n+1}$, namely $\frac{\partial}{\partial \boldsymbol{\lambda}_{n+1}} \log(q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T_{n+1}}))$ and $q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T_{n+1}})$, are required to be calculated.

The variance of the UVB-IS gradient estimator is increased relative to the score-based gradient estimator in (5) due to the presence of the importance sampling weights. This increased variance may result in a reduction in the accuracy of $q_{\boldsymbol{\lambda}_{n+1}^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})$. This is due to the fact that the algorithm stopping criterion, which is a sufficiently small value of

$|\mathcal{L}(q, \boldsymbol{\lambda}^{(m+1)}) - \mathcal{L}(q, \boldsymbol{\lambda}^{(m)})|$ can only be evaluated approximately by a noisy estimator, also produced via an importance sampler. As the computation per iteration is extremely small, S may be set to a larger value to reduce the variance, thereby allowing the user the capacity to balance the inevitable trade-off between computational time and approximation accuracy to suit their requirements. Provided there is no major structural change in the data generating process, it is expected that the distributions $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ and $q_{\boldsymbol{\lambda}_{n+1}^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})$ become more similar as n increases, subsequently reducing the UVB-IS gradient estimator variance.

A potential disadvantage of using importance sampling is for the variance of the gradient to increase with each iteration of stochastic gradient ascent. This is likely to be offset by a reduction in the variance of the gradient as more data are observed with each update. As a check on whether the importance sampling is working well, we recommend running stochastic gradient ascent without importance sampling on the first block of data and computing the variance of the gradient after a small number of iterations (e.g. 30-50). This can then provide a threshold which the variance of the gradient computed by importance sampling should not exceed.

The proposed UVB-IS algorithm is summarised in Algorithm 2. Figure 1 provides a diagram to help illustrate the differences between the approach of Sakaya and Klami (2017) to UVB-IS. In panel (a) of Fig. 1, each block indicates r separate iterations of SGA, each undertaken over an entire sample of length T, with arrows indicating that the final iteration of each block is used as an importance sampling proposal distribution for the entire next block. That is, there is one SGA algorithm applied for all data, but the importance sampling distribution changes every r^{th} iteration until convergence is reached. In panel (b) of Fig. 1, three distributional updates using UVB-IS are depicted. In this case, the posterior itself is updated periodically, as indicated by arrows and corresponding to times T_1, T_2, and T_3, with the same importance sampling distribution used for *all* SGA iterations needed to complete a single distributional update.

Algorithm 2: UVB with Importance Sampling (UVB-IS)

Input: Prior, Likelihood.
Result: Approximating distribution at T_τ.
Observe $\boldsymbol{y}_{1:T_1}$.;
Minimises $KL[q_{\boldsymbol{\lambda}_1}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1}) \;||\; p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_1})]$ using SGA via (5).;
for n *in* $1, \ldots, \tau - 1$ **do**
  Observe next data $\boldsymbol{y}_{T_n+1:T_{n+1}}$.;
  Sample $\boldsymbol{\theta}^{(j)} \sim q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ for $j = 1, 2, \ldots S$.;
  Evaluate $p(\boldsymbol{y}_{T_n+1:T_{n+1}}|\boldsymbol{\theta}^{(j)})$ and $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}^{(j)}|\boldsymbol{y}_{1:T_n})$ for each j.;
  Set $\boldsymbol{\lambda}_{n+1}^{(1)}$ to $\boldsymbol{\lambda}_n^*$.;
  Minimise $KL[q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}}) \;||\; \widetilde{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_{n+1}})]$ using SGA via (17).;
end

5 Simulation studies

To investigate the trade-off between the computational efficiency and accuracy of different methods, we consider two simulated examples. The first is a time series forecasting application, while the second is a clustering example based on a mixture model. As well as considering both of the proposed algorithms (i.e. UVB and UVB-IS) we also consider a standard SVB approach and an exact MCMC algorithm, based on a Random Walk Metropolis-Hastings strategy (see (Gilks et al. 1995a, b), and (Garthwaite et al. 2016), employed using all data observed up to each relevant time point.

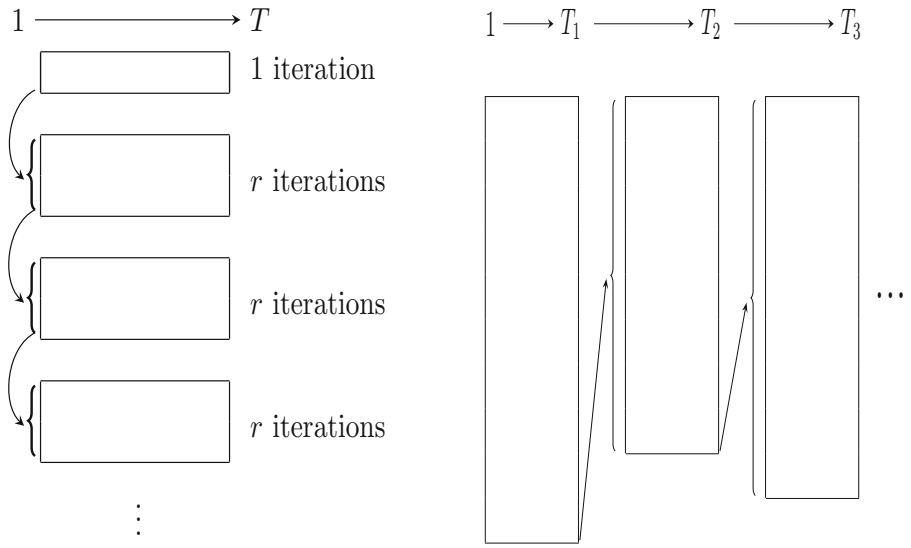
5.1 Time series forecasting

In this first simulation study, we consider $R = 500$ replications of time series data, with each comprised of T = 500 observations simulated from the following auto-regressive order 3 (AR3) model, given by

$$\begin{aligned} y_t = \; & \mu + \phi_1(y_{t-1} - \mu) + \phi_2(y_{t-2} - \mu) \\ & + \phi_3(y_{t-3} - \mu) + e_t \end{aligned} \tag{19}$$

where $e_t \sim N(0, \sigma^2)$. For each replication, the true values of the parameters are obtained by drawing μ and each auto-regressive coefficient, ϕ_1, ϕ_2, and ϕ_3 from an independent $N(0, 1)$ distribution, accepting only draws where each ϕ lies in the AR3 stationary region. The precision parameter, σ^{-2}, is drawn from a Gamma distribution with both shape and rate equal to five.

The inferential objective is to progressively produce the one-step ahead predictive densities, each based on a UVB approximation to the target posterior distribution that results from assuming data arises from the AR3 model above, with a prior distribution specified for $\boldsymbol{\theta} = \{\mu, \phi_1, \phi_2, \phi_3, \log(\sigma^2)\}$. The prior distribution for the parameter vector is taken as $\boldsymbol{\theta} \sim N(\mathbf{0}_5, 10\mathbb{I}_5)$, where $\mathbf{0}_d$ and $\mathbb{I}_d$ denote, respectively, the $d-$dimensional zero vector and identity matrix. In particular, we aim to produce UVB-based approximate one-step ahead predictive distributions progressively, using at time T_n all (and only) data up to and including time period T_n, recursively for each of the 21 time periods given by $T_n = 100, 125, 150, \ldots 500$. That is, the first target predictive distribution is given by $p(y_{101}|\boldsymbol{y}_{1:100})$, followed by $p(y_{126}|\boldsymbol{y}_{1:125})$, and continuing on to the final predictive $p(y_{501}|\boldsymbol{y}_{1:500})$. For each update, predictive distributions are approximated with $q_{\boldsymbol{\lambda}}$ taken as a $K-$component mixture of multivariate normal distributions. Diagonal covariance matrices for each normal are assumed; alternatively, a sparse structure Tan and Nott 2018) or a factor structure (Ong et al.



(a) Sakaya and Klami (2017) **(b)** UVB-IS

Fig. 1 Graphical illustrations for importance sampling in VB algorithms. **a**: The approach of Sakaya and Klami (2017). Each block indicates r iterations of a single implementation of the SGA algorithm, with arrows indicating that the final iteration of each block is used as an importance sampling proposal distribution for the next r iterations contained in the subsequent block. **b**: The UVB-IS algorithm, where each block indicates that SGA is applied three times, once for each of three distributional updates corresponding to an increase in data. For an update, indicated by an arrow, a sample from the pseudo-posterior distribution corresponding to the previous update is used as proposal draws in every iteration of the SGA algorithm

2018) could be employed. The results are compared using three different choices of K, with $K = 1, 2$ and 3. This strategy allows us to compare the approximation accuracy of the simple $K = 1$ distribution that may not adequately capture the entire posterior distribution as well as more complex approximations. In all cases, the convergence criterion is to compare the mean of objective function from the last five iterations to the five iteration before that, and to stop if the difference is less than 10^{-4} times the number of parameters.

For the cases involving SVB and UVB, the score-based gradient estimator (5) uses $S = 25$ draws of $\boldsymbol{\theta}$; however, we use a larger number of draws for UVB-IS to offset the increased variance, setting $S = 100$. Finally the MCMC benchmark comparison is based on 15000 posterior draws, with the first 10000 discarded for 'burn in'. In each approach, we allow $\{\phi_1, \phi_2, \phi_3\}$ to take any value in $\mathbb{R}^3$, so the posterior distribution for these parameters is not restricted to the AR3 stationary region.

Under the posterior given by $p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ together with the conditional predictive densities implied by (19), the one-step ahead predictive density is given by

$$p(y_{T_n+1}|\boldsymbol{y}_{1:T_n}) = \int_{\boldsymbol{\theta}} p(y_{T_n+1}|\boldsymbol{y}_{1:T_n}, \boldsymbol{\theta}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n}) d\boldsymbol{\theta}. \quad (20)$$

Given our UVB approximation to the posterior at time T_n, we approximate the integral in (20) using M draws $\boldsymbol{\theta}^{(1)} \dots \boldsymbol{\theta}^{(M)} \sim q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$, with the resulting marginal predictive density estimate given by

$$\widehat{p}(y_{T_n+1}|\boldsymbol{y}_{1:T_n}) \approx \frac{1}{M} \sum_{i=1}^{M} p(y_{T_n+1}|\boldsymbol{y}_{1:T_n}, \boldsymbol{\theta}^{(i)}). \quad (21)$$

The forecast accuracy associated with the resulting approximate predictive density is measured using the cumulative predictive log score (CLS) for the update at time T_n, given by

$$CLS_n = \sum_{j=1}^{n} \log(\widehat{p}(y_{T_j+1}^{(obs)}|\boldsymbol{y}_{1:T_j})), \quad (22)$$

for $n = 1, 2, ..., 17$, where $y_{T_n+1}^{(obs)}$ denotes the realised (observed) value of y_{T_n+1}. In particular, we compare the mean CLS (MCLS) over $R = 500$ Monte Carlo replications, for each approximation method and each given value of K, at consecutive update times $T_n \in \{100, 125, \dots, 500\}$. The results are displayed in Fig. 2, where each row indicates a different (known) value of K. Panel (a), on the left-hand side, the MCLS value is displayed relative to the MCLS value obtained using MCMC inference at each incremental values of $T_n + 1 = 101, 126, \dots, 501$. As greater values of indicate better forecast accuracy, it is not surprising to find that each of the approximate VB method produces a lower relative to

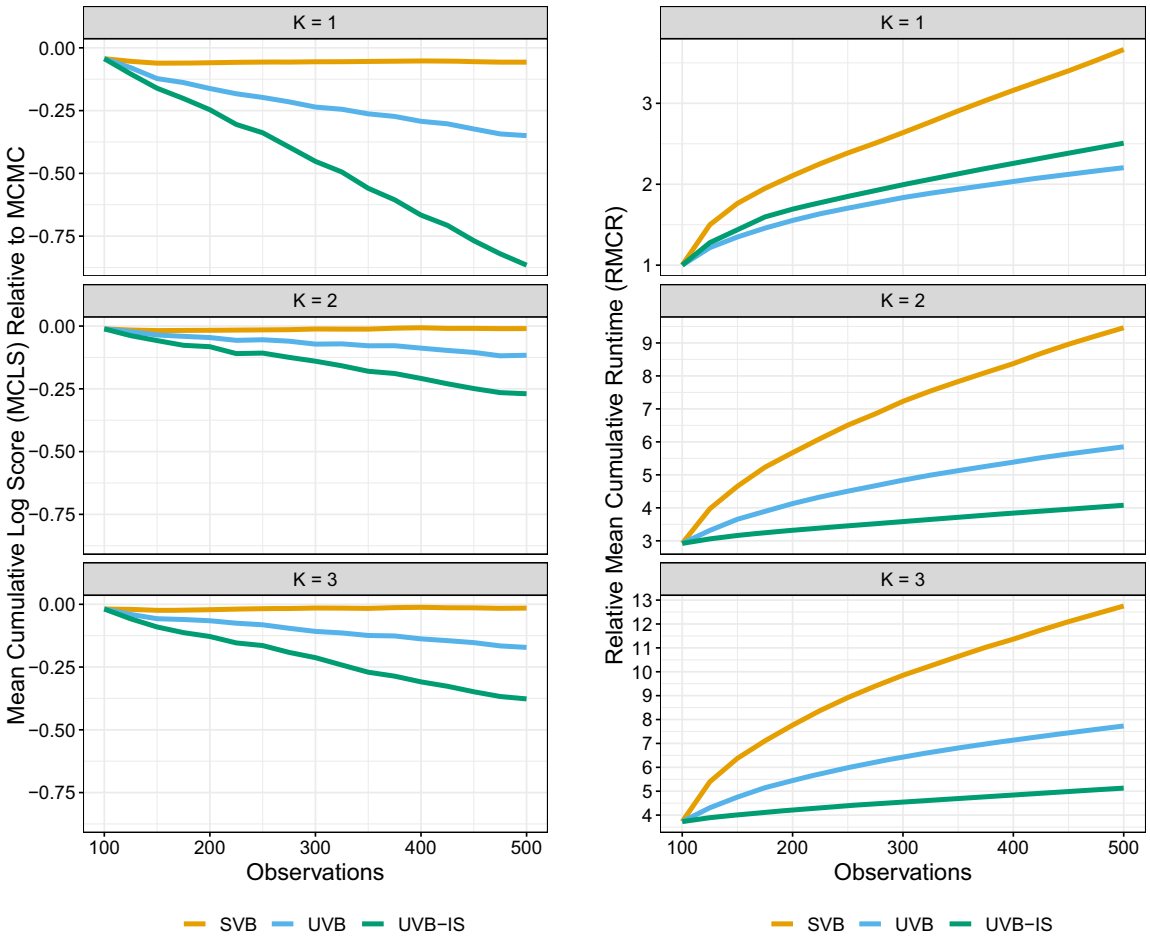


Fig. 2 AR3 Simulation from Sect. 5.1. (left): Forecast accuracy, indicated by one-step-ahead mean cumulative predictive log scores (MCLS), corresponding to incremental updates under competing methods (SVB, UVB and UVB-IS) relative to MCMC. Higher values of MCLS indicate better forecast accuracy. (right): Computational efficiency, indicated by relative mean cumulative runtime (RMCR), again corresponding to incremental updates under competing method (SVB, UVB and UVB-IS), each reported relative to the mean runtime for SVB when $K = 1$ and $T_n = 100$. Lower values of RMCR reflect improvements in computational efficiency

exact (MCMC) inference. Amongst the approximate methods, repeated SVB performs the best, in terms of, followed by UVB and UVB-IS at $K = 1$, though these differences are less severe as K (and the model complexity) increases.[3]

To investigate the computational efficiency of different methods, we compute the relative cumulative mean runtime (RMCR), for each considered algorithm (SVB, UVB and UVB-IS), again over updating times $T_n \in \{100, 125, 150, \ldots, 500\}$. Each of these sequences represents an average over $R = 500$ independent Monte Carlo cumulative runtimes from the relevant algorithm, reported as a multiple of the average runtime of the SVB algorithm for a single update at $T_1 = 100$ and with a single mixture component $K = 1$. For all three methods considered, at update time T_{n+1} the optimal value of the variational parameter obtained at time T_n is used as the starting value for the optimisation. The RMCR values obtained are reported in Panel (b) of Fig. 2. Note that for $n > 1$, the SVB approximation at time T_n requires an application of the SGA algorithm using all data observed up to T_n, while each of the updating methods begin with an SVB approximation at T_1, followed by $n - 1$ progressive updates each using only the new data observed since the previous update period. As can be seen in the top row of Panel (b), the RMCRs are all identical and equal to one at the first update time $T_1 = 100$ and all increase over consecutive updates. While all three methods show an increase in RMCR with each update, the SVB method appears to be least efficient, while substantial improvements in computational time accrue from using UVB-IS.

In this setting, the amount of data in each update is relatively small, and UVB increases the runtime compared to

[3] To check that these differences are not a result of variability across the 500 replications we conduct non-parametric Friedman and post-hoc Nemenyi tests with details discussed in "Appendix A".

SVB. This is due to the computational overhead of n SGA applications not being offset by a reduction in the number of log-likelihood calculations. In contrast, UVB-IS achieves sizeable computational gains despite showing minimal loss in the corresponding MCLS for $K > 1$.

To illustrate the reduced variability in subsequent UVB gradient estimators, Fig. 3 displays the median variance of the gradient estimator for the posterior mean parameter μ, for UVB with $S = 25$, and for UVB-IS with each of $S = 25, 50, 100$ and 200, all monitored over six selected update periods. At T_1, all algorithms implement SVB with arbitary starting values for $\boldsymbol{\lambda}_1^{(1)}$. This causes extreme, but declining, variance until convergence is reached. This pattern is typical for SVB inference. In subsequent time periods, each updating method sets the starting value at $\boldsymbol{\lambda}_n^{(1)} = \boldsymbol{\lambda}_{n-1}^*$. The estimated variance is subsequently orders of magnitude smaller than SVB. For small values of n the distributions $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ and $q_{\boldsymbol{\lambda}_{n+1}^{(m)}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:T_n})$ may differ as m increases, causing a reduction in the effective sample size associated with the gradient estimator, and an increase in the UVB-IS estimator variance. This effect is visible at times T_2, T_3, and T_5, though the UVB-IS estimator variance is low relative to SVB despite this inefficiency. Furthermore although the variances of the gradients estimated by Importance Sampling are high (with a median value reaching around 30 for T_2 and T_3), this is still orders of magnitude lower than for SVB without importance sampling at T_1. For this application, UVB-IS passes the check recommended in Sect. 4.

5.2 Mixture model clustering

In the second simulated example, we consider the case where repeated measurements are simulated on $N = 100$ cross-sectional units at each of $T = 100$ times. The measurements for a given unit follows one of two possible DGPs, with the objective being to cluster the units into the correct groups, according to the underlying DGP, with additional observations of each cross-sectional unit accumulating in an online fashion as time increases. Each of these scenarios was then replicated $R = 500$ times.

For each independent replication, we generate data $y_{i,t}$ as the measurement of unit i at time t, for $i = 1, 2, \ldots, N$ and $t = 1, 2, ..., T$ as follows. We first define the cluster indicator for unit i as k_i, and generate these for a given probability $0 < \pi < 1$ according to

$$k_i|\pi \overset{i.i.d.}{\sim} Bernoulli(\pi), \tag{23}$$

where $i.i.d$ abbreviates *independent and identically distributed.* Then, conditional on k_i we let

$$y_{i,t}|(k_i = j), \mu_j, \sigma_j^2 \overset{ind}{\sim} N(\mu_j, \sigma_j^2), \tag{24}$$

for $j = 0, 1$, with ind short for *independent*. For this exercise, we set $\pi = 0.5$, with the replicated values of μ_0 and μ_1 independently drawn from an $N(0, 0.25)$ distribution, while σ_0^2 and σ_1^2 are independently drawn from a uniform distribution over the interval $(1, 2)$.

Having simulated the data, the actual values k_i are retained for each replication. We then use the UVB algorithm of the described model with the simulated data, as if all N units were being observed online at increasing times $T_n = 10, 20, 30, \ldots 100$. The aim of the exercise is to cluster the units into two groups aligning with the true, but 'unobserved' value of k_i.

The Bayesian updating analysis proceeds as follows. Denoting the collective parameter vector as $\boldsymbol{\theta} = \{\log(\sigma_0^2), \log(\sigma_1^2), \mu_0, \mu_1\}$, the joint prior for $\boldsymbol{\theta}$ and π used at T_1 is given by independent components

$$\boldsymbol{\theta} \sim N(\mathbf{0}_4, 10\mathbb{I}_4), \text{ and} \tag{25}$$

$$\pi \sim Beta(\alpha, \beta). \tag{26}$$

Note that the model for π in (23) and the prior in (26) imply that the k_i are independent *a priori*, with marginal probabilities given by

$$\Pr(k_i = j) = \frac{\mathcal{B}(j + \alpha, \beta - j + 1)}{\mathcal{B}(\alpha, \beta)}, \tag{27}$$

for $j = 0, 1$, where $\mathcal{B}(\cdot, \cdot)$ denotes the Beta function. Hence we have marginalised out the 'unknown' value of π, and can now proceed to updating the prior in (27), for each $i = 1, 2, \ldots, N$, on the basis of information at times $T_n = 10, 20, ..., 100$.

Denoting $\boldsymbol{y}_{i,1:T_n} = \{y_{i,t}|t = 1, \ldots, T_n\}$ and $\boldsymbol{y}_{1:N,1:T_n} = \{\boldsymbol{y}_{i,1:T_n}; i = 1, \ldots N\}$, the initial *augmented* posterior distribution is given by

$$p(\boldsymbol{\theta}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_1}) \propto p(\boldsymbol{\theta}) \prod_{i=1}^{N} p(\boldsymbol{y}_{i,1:T_1}|\boldsymbol{\theta}, k_i = j) \times Pr(k_i = j), \tag{28}$$

with each likelihood $p(\boldsymbol{y}_{i,1:T_1}|\boldsymbol{\theta}, k_i = j)$ given by the product of densities associated with (24) and the value of j.

Due to the conditional independence of the components of $\boldsymbol{\theta}$ and the cluster indicators, subsequent posteriors at times T_{n+1} are approximated by

$$\widehat{p}(\boldsymbol{\theta}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}}) \propto \prod_{i=1}^{N} p(\boldsymbol{y}_{i,T_n+1:T_{n+1}}|\boldsymbol{\theta}, k_i = j) \times \widehat{\Pr}(k_i = j|\boldsymbol{y}_{1:N,1:T_n}) p(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_n}), \tag{29}$$

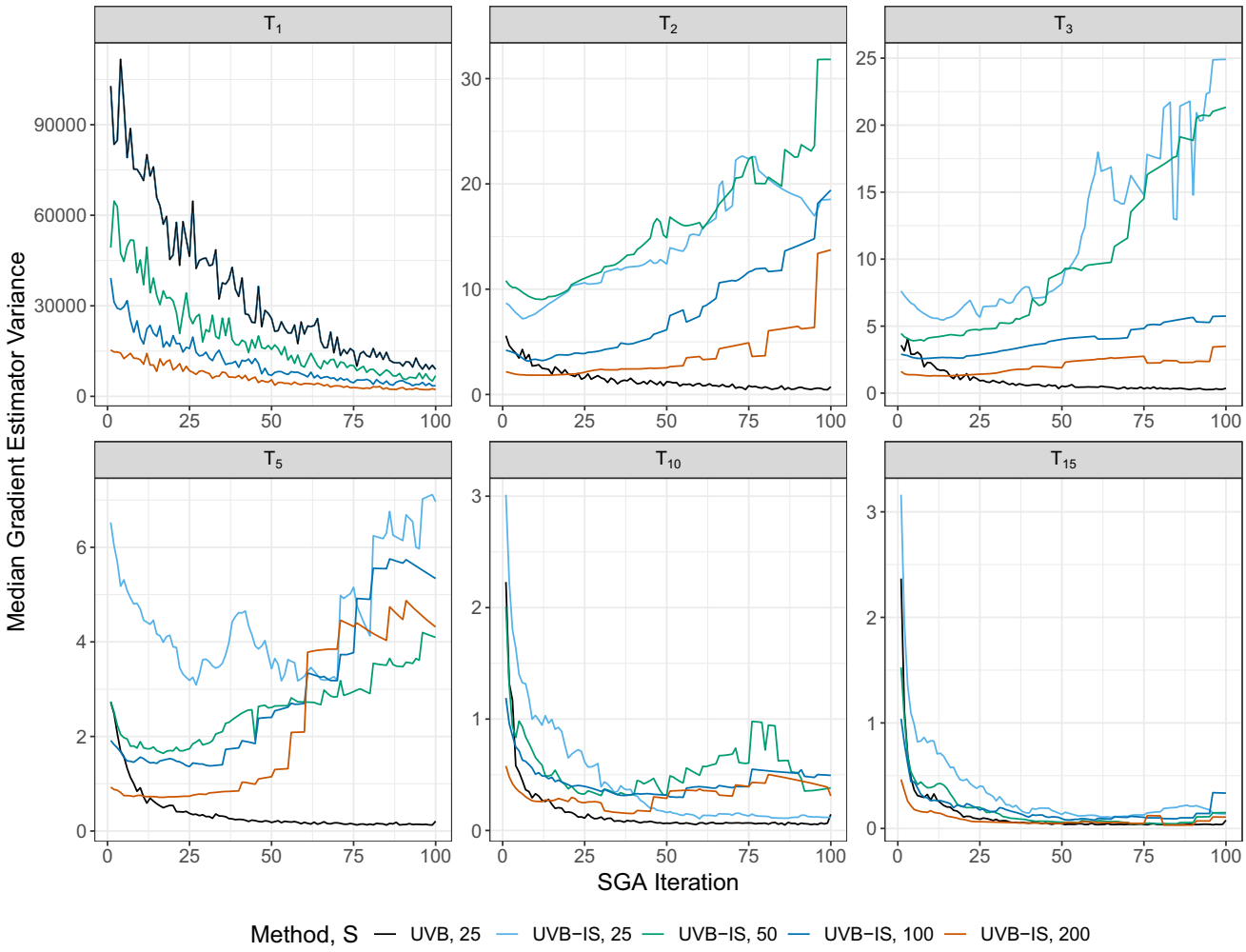


Fig. 3 AR3 example in Sect. 5.1. Median gradient estimator variance for the first 100 SGA iterations, with colour indicated by the acronym (either UVB or UVB-IS) followed by the value of S. Both UVB and UVB-IS algorithms have arbitrary starting values at T_1, denoted as $\boldsymbol{\lambda}_1^{(1)}$, where the estimated gradient exhibits high variance. The starting value at time T_n is set to the previous optimal value, i.e. $\boldsymbol{\lambda}_n^{(1)} = \boldsymbol{\lambda}_{n-1}^*$. Since only a subset of time periods are presented here, the variance corresponding to the previously converged value of λ is not shown for each update. For example, the gradient variance corresponding to $\lambda_5^{(1)}$ is shown but not λ_4^*. Since the variance falls dramatically with each update, different y-axis scales are used at at subsequent update times. Consequently, the variance of the UVB gradient estimator is reduced relative to SVB, though the UVB-IS variance increases slightly for small n and large iteration index m

where the latent class probabilities, $\Pr(k_i = j|\boldsymbol{y}_{1:N,1:T_n})$, are estimated before updating with

$$\widehat{\Pr}(k_i = j|\boldsymbol{y}_{1:N,1:T_n}) \propto \frac{1}{M}\sum_{l=1}^{M} p(\boldsymbol{y}_{1:N,1:T_n}|\boldsymbol{\theta}^{(l)}, k_i = j) \times \Pr(k_i = j), \tag{30}$$

with $\boldsymbol{\theta}^{(l)} \sim p(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_n})$ for $l = 1, 2, \ldots, M$.

As in Section 5.1, the UVB and UVB-IS algorithms are compared to standard SVB and MCMC. Each of these approaches utilises an approximation to the augmented posterior of the form

$$q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}}) = q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_{n+1}}) \times \prod_{i=1}^{N} \widehat{\Pr}(k_i = j|\boldsymbol{y}_{1:N,1:T_n}), \tag{31}$$

where $q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_{n+1}})$ is a $K = 1, 2$, or 3 component mixture of multivariate normal distributions and the $\boldsymbol{\theta}^{(l)}$ samples used to estimate (30) are simulated from the previous approximation $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_n})$.

The form of the approximation used in (31) is chosen due to the fact that the gradient of the augmented divergence, $KL[q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n}) \;||\; \widehat{p}(\boldsymbol{\theta}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n})]$ is equivalent to the gradient of the marginal divergence,

$KL[q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_n}) \;||\; \widehat{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_n})]$, and hence the same approximation can be found by instead targeting the marginal posterior distribution,

$$p(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_1}) \propto p(\boldsymbol{\theta}) \times \prod_{i=1}^{N}\left(\sum_{j=0}^{1} p(\boldsymbol{y}_{i,1:T_1}|\boldsymbol{\theta}, k_i = j)\Pr(k_i = j)\right), \tag{32}$$

or its updated form

$$\widehat{p}(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_{n+1}}) \propto \prod_{i=1}^{N}\Bigg(\sum_{j=0}^{1} p(\boldsymbol{y}_{i,T_n+1:T_{n+1}}|\boldsymbol{\theta}, k_i = j) \times \widehat{\Pr}(k_i = j|\boldsymbol{y}_{1:N,1:T_n})p(\boldsymbol{\theta}|\boldsymbol{y}_{1:N,1:T_n})\Bigg). \tag{33}$$

At each update, we estimate class labels for k_i according to

$$\widehat{k}_{i,n} = \arg\max_j \widehat{\Pr}(k_i = j|\boldsymbol{y}_{1:N,1:T_n}), \tag{34}$$

and assign a classification accuracy (CA) score at T_n, given by

$$CA_n = \max\left(\frac{1}{N}\sum_{i=1}^{N} I(\widehat{k}_{i,n} = k_i),\;\; \frac{1}{N}\sum_{i=1}^{N} I(\widehat{k}_{i,n} \neq k_i)\right), \tag{35}$$

the proportion of successful classifications up to label switching. SVB and UVB gradients are estimated from $S = 25$ samples of $\boldsymbol{\theta}$ per iteration, while UVB-IS sets $S = 100$.

The results for this problem are displayed in Fig. 4, where each row corresponds to a different value of K. Panel (a) displays the mean classification accuracy (MCA), corresponding to updates at times $T_n = 10, 20, ..., 100$ and across R replications. As in the previous study, each variational approximation reduces accuracy relative to exact inference. In this example, UVB and in some instances UVB-IS are more accurate than SVB, with little change apparent in any variational approach between different values of K. This result is somewhat puzzling since UVB and UVB-IS do introduce further approximation compared to SVB. One possible explanation is that by failing to capture the thickness of the tails in the posterior, UVB 'implicitly' imposes an empirical Bayes prior with thinner tails for the following update. Priors (and posteriors) with thinner tail may improve the performance with respect to classification accuracy, especially for observations near the decision boundary.[4] This finding may be idiosyncratic to this example, and in general, it seems unrealistic to expect UVB and UVB-IS to substantially outperform SVB.

As in the previous section, Panel (b) of Fig. 4 displays the RMCR for each VB method using data up to T_n, for $T_n = 10, 20, \ldots, 100$, calculated relative to the mean run time of the SVB algorithm fitting a single mixture at the initial update time, when $T_n = 10$. As the problem features a large number of cross-sectional units, the computational cost of calculating the log-likelihood dominates the gradient estimation. Processing smaller amounts of data, and having a reduced gradient variance lead to reduced computational time for both UVB and UVB-IS relative to SVB, particularly in the case of UVB-IS. Despite the updating methods consisting of 10 SGA applications while SVB uses only one, UVB and UVB-IS require, on average, 14.7%, and 4.6% of the computational time of SVB, respectively, in the top right panel when $K = 1$ at time $T_{10} = 100$.

6 Eight schools example

In this section, the so-called Eight Schools problem described in Gelman et al. (2014) is considered. This problem analyses the effectiveness of a short-term coaching program, implemented independently by each of eight studied schools, for the SAT-V test.[5] For students $i = 1, 2, \ldots, N_j$ in each school $j = 1, 2, \ldots, 8$, consider the linear regression

$$SAT\text{-}V_{i,j} = \beta_{0,j} + \beta_{1,j}Coach_{i,j} + \beta_{2,j}PSAT\text{-}V_{i,j} + \beta_{3,j}PSAT\text{-}M_{i,j} + \epsilon_{i,j} \tag{36}$$

where $Coach_{i,j}$ is a dummy variable indicating a student's inclusion (or not) in a coaching program run by their school, alongside control variables $PSAT\text{-}V_{i,j}$ and $PSAT\text{-}M_{i,j}$, corresponding to each student's scores in the verbal and mathematical preliminary SAT, respectively.

Following Gelman et al. (2014), the estimated school-level coaching coefficients that correspond to the ordinary least squares estimators are taken as the observations, $y_j = \widehat{\beta}_{1,j}$, for $j = 1, 2, \ldots 8$, and have approximate sampling distributions given by

$$y_j|\theta_j, \sigma_j^2 \sim \mathcal{N}(\theta_j, \sigma_j^2), \tag{37}$$

where θ_j is the latent 'true' effectiveness of school j's coaching program. The standard deviation of the sampling distribution, σ_j, is assumed to be known and is held fixed at the standard error estimated by the relevant regression, with each having taken account of the individual school sample size N_j.

[4] We thank an anonymous referee for making this point.

[5] The SAT-V is a standardised aptitude test commonly taken by high school students in the USA.

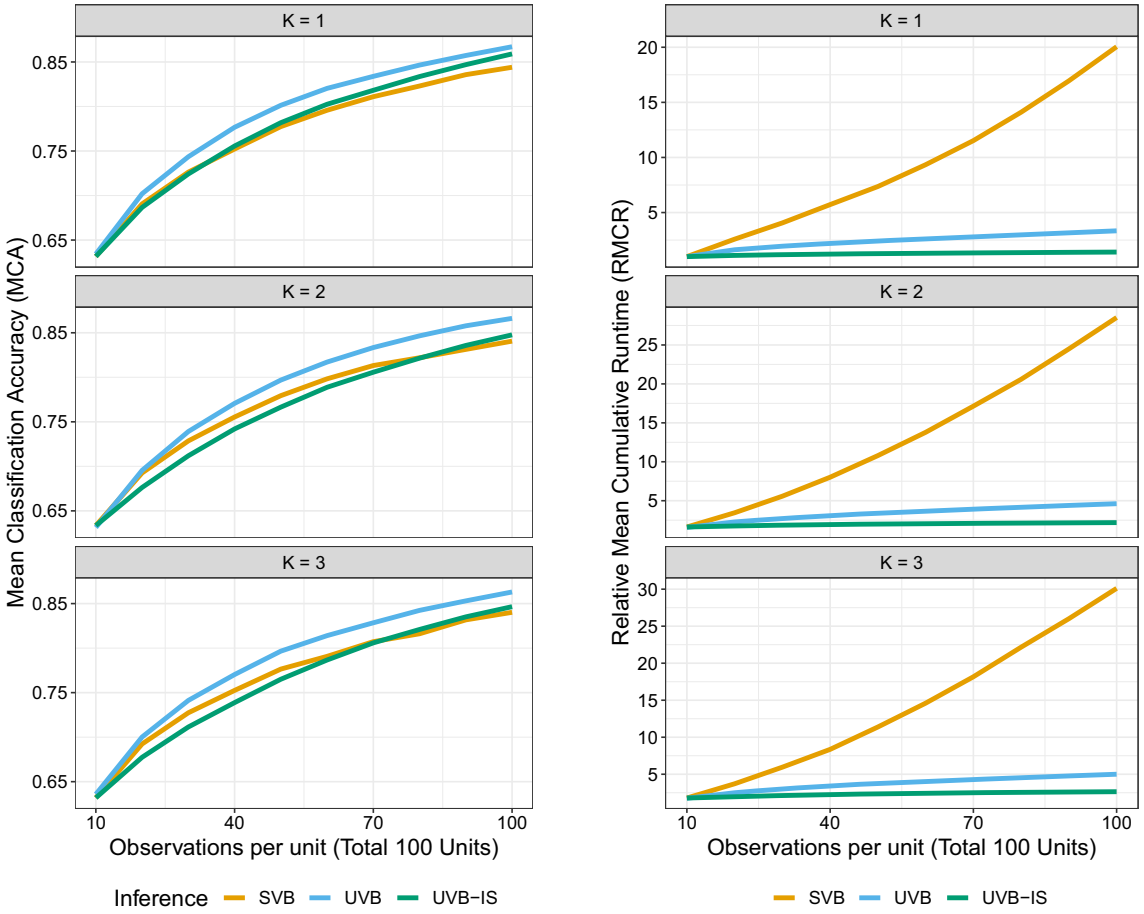


Fig. 4 Classification example in Sect. 5.2. (left): Mean Classification Accuracy (MCA) for each inference method, higher is better. (right): Relative mean cumulative runtime (RMCR) for each updating method. Average runtimes for each updating algorithm are reported relative to the average runtime required for SVB at T_1 and with $K = 1$, with lower RMCR values preferred. Both UVB and UVB-IS perform better than SVB in terms of classification accuracy and are also much faster, as computation of the data likelihood is a large part of the gradient calculation in this scenario

Again following Gelman et al. (2014), we apply a hierarchical prior to the population mean values in (37), assuming that the θ_j themselves are random and iid from a Student-t distribution,

$$\frac{\theta_j - \mu}{\tau} \sim t(\nu) \tag{38}$$

where ν is the degrees of freedom, fixed at $\nu = 4$. The hierarchical model also employs the uninformative hyper-prior

$$p(\mu, \tau) \propto 1, \tag{39}$$

over positive values of τ, and both positive and negative values of μ.

Collecting the unknown school means together and denoting by $\boldsymbol{\theta}_{1:8} = \{\theta_1, \theta_2, \ldots, \theta_8\}$, the posterior distribution of all unknowns and based on the observed values from all schools is then given by

$$p(\boldsymbol{\theta}_{1:8}, \mu, \tau | \boldsymbol{y}_{1:8}) \propto \prod_{j=1}^{8} p(y_j | \theta_j, \sigma_j^2) p(\theta_j | \tau, \mu). \tag{40}$$

It is feasible to obtain this posterior exactly, via MCMC, for example using the algorithm provided in the statistical modelling platform Stan, (Stan Development Team 2018).

Our aim here is to demonstrate the application of UVB and UVB-IS to this hierarchical model, where with each update we sequentially 'observe' an additional school, as indicated by the inclusion of an additional observation y_j. Each variational algorithm approximates the progressive posterior by the multivariate normal distribution $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}_{1:n}, \mu, \tau | \boldsymbol{y}_{1:n})$, for $n = 1, 2, \ldots, 8$. The initial distribution approximation at T_1 for UVB and UVB-IS is given by the multivariate normal distribution $q_{\boldsymbol{\lambda}_1^*}(\theta_1, \mu, \tau | y_1)$ where

$$\boldsymbol{\lambda}_1^* = \arg\min_{\boldsymbol{\lambda}_1} \quad KL[q_{\boldsymbol{\lambda}_1}(\theta_1, \mu, \tau|y_1) \quad || \quad p(\theta_1, \mu, \tau|y_1)]. \tag{41}$$

Updates at further 'times' $T_{n+1} = n+1$, for $n = 1, 2, \ldots, 7$, involves sequentially adding schools to the model targetting the pseudo-posterior distribution, given by the decomposition

$$\widetilde{p}(\boldsymbol{\theta}_{1:n+1}, \mu, \tau|\boldsymbol{y}_{1:n+1}) \propto p(y_{n+1}|\theta_{n+1})\, p(\theta_{n+1}|\mu, \tau) \times q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}_{1:n}, \mu, \tau|\boldsymbol{y}_{1:n}). \tag{42}$$

Either UVB or UVB-IS then may be used to obtain the updated approximate posterior, given by $q_{\boldsymbol{\lambda}_{n+1}^*}(\boldsymbol{\theta}_{1:n+1}, \mu, \tau|\boldsymbol{y}_{1:n+1})$, with

$$\boldsymbol{\lambda}_{n+1}^* = \arg\min_{\boldsymbol{\lambda}_{n+1}} \quad KL[q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}_{1:n+1}, \mu, \tau|\boldsymbol{y}_{1:n+1}) \quad || \quad \widetilde{p}(\boldsymbol{\theta}_{1:n+1}, \mu, \tau|\boldsymbol{y}_{1:n+1})], \tag{43}$$

for $n = 1, 2, ..., y$. As each update adds a new variable θ_{n+1} to the model, the optimal vector $\boldsymbol{\lambda}_{n+1}^*$ updates the auxiliary parameters associated with the pseudo-posterior distribution for θ_{n+1} together with the previously included variables μ, τ, and $\boldsymbol{\theta}_{1:n}$. We note that our implementation of UVB-IS here employs a hybrid strategy utilising importance sampled gradients (17) for simulations of μ, τ, and $\boldsymbol{\theta}_{1:n}$ from the previous $q_{\boldsymbol{\lambda}_n^*}(\boldsymbol{\theta}_{1:n}, \mu, \tau|\ \boldsymbol{y}_{1:n+1})$, and score-based gradients for $\boldsymbol{\theta}_{n+1}$, as per (5). The score-based gradients use samples generated from $\boldsymbol{\theta}_{n+1} \sim q_{\boldsymbol{\lambda}_{n+1}}(\theta_{n+1}|\boldsymbol{y}_{1:n+1}, \mu, \tau, \boldsymbol{\theta}_{1:n})$, which is available as this variational approximation was chosen to be a multivariate normal distribution.

We compare approximations that result from using UVB and UVB-IS, relative to the sequential implementation of SVB, following the incorporation of data from each new school. As the ordering of the inclusion of schools is arbitrary in this example, we report results that are averaged over a randomly selected 100 of the $8! = 40,320$ possible permutations of school sequences. For each ordering, the variational posterior $q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_{1:n+1}, \mu, \tau|\boldsymbol{y}_{1:n+1})$ is compared to the exact posterior $p((\boldsymbol{\theta}_{1:n+1}, \mu, \tau|\boldsymbol{y}_{1:n+1})$ in (40), each calculated using 10,000 MCMC sample draws retained following a burn-in period of 10,000 iterations.

The average Hellinger distances between different variational marginal posteriors and their exact counterparts, for each school specific effect are summarised in Table 1. In all cases, the Hellinger distance is computed between each marginal posterior produced using MCMC and the corresponding posterior produced using one of the three variational methods. The Hellinger distance is estimated by noting that squared Hellinger distance is, up to a multiplicative constant, a special case of a Tsallis divergence. We then use the $\hat{T}_{lin}$ estimator based on a von Mises expansion proposed by Krishnamurthy et al. (2014). We note that while this estimator is consistent, in finite samples, negative values of the estimate are possible when two distributions are extremely close in Hellinger distance. Table 1 shows that for an example where the objective of the analysis is parameter inference, rather than prediction or classification, the resulting approximate posterior inference can deteriorate when UVB or UVB-IS is used relative to SVB. This is likely due to the errors in the variational approximation, particularly in the tails, accumulating with each update. We therefore recommend caution when using the UVB and UVB-IS purely for parameter estimation rather than forecast accuracy.

Table 1 Eight Schools example from Sect. 6. Average Hellinger distance between the marginal posteriors obtained via MCMC and those obtained by SVB, UVB and UVB-IS. Lower values are better.

Parameter	SVB	UVB	UVB-IS
θ_1	0.022	0.218	0.612
θ_2	0.002	0.048	0.590
θ_3	−0.007	0.147	0.539
θ_4	0.004	0.054	0.548
θ_5	0.012	0.096	0.511
θ_6	0.008	0.084	0.470
θ_7	0.006	0.106	0.657
θ_8	0.004	0.119	0.571

The differences between methods are statistically significant for all parameters—see the discussion in "Appendix A"

7 Lane position example

Vehicle drivers may exhibit a tendency to move laterally (i.e. side-to-side) within their designated lane on a highway. Figure 5 displays this notion, by plotting the trajectory of five drivers as they travel along a section of the US Route 101 Highway, as taken from the Next Generation Simulation (NGSIM, FHWA (2017)) dataset. In this figure, the vehicles—whose trajectories are indicated in black—are travelling towards the right, with each (estimated) lane centre line given by the red dashed line. Drivers likely adapt their position in real time, in at least partial response to the perceived position of vehicles that are travelling nearby.

The aim of this section is to apply the UVB methodology to analyse a model of the lateral position of vehicles. The model incorporates driver heterogeneity, while the analysis itself produces sequential, per-vehicle distributional forecasts of a large number of future car positions. The methodology suggests that a smart vehicle (i.e. one without a human driver) may be able to repeatedly 'observe' neighbouring vehicle positions, predict their positions in real time as they travel along the road, and appropriately respond to those forecasts.

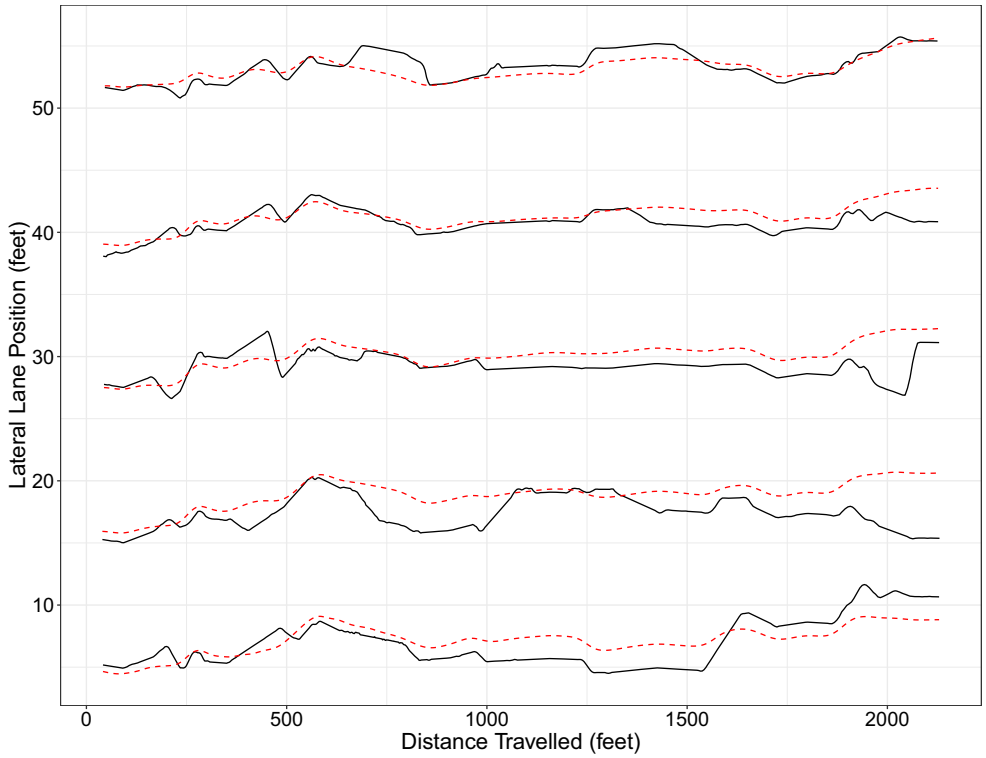


Fig. 5 Car lane position example from Sect. 7. The path of five selected vehicles from the NGSIM dataset, travelling from left to right, with each black line representing a unique vehicle, with estimated lane centre lines in red. This section of US Route 101 is comprised of five main lanes, with a sixth entry/exit lane not shown

To set up the scenario, we randomly select, from the NGSIM dataset, trajectories associated with $N = 500$ vehicles that do not change lane. We note that the NGSIM dataset is the result of a project conducted by the US Federal Highway Administration (FHWA), and includes data recorded from 6101 vehicles traveling along a 2235 foot long section of the US 101 freeway in Los Angeles, California from 7:50 am to 8:35 am on June 15th, 2005. Though initially collected by static cameras, the data were then processed by Cambridge Systematics Inc. to produce coordinates of the centre of the front of each vehicle at 100 millisecond intervals.

7.1 A hierarchical model

In developing a model for the position of cars, we consider a number of issues. First, we view each vehicle/driver as having its own idiosyncratic behaviour, captured by its own parameter values. Let $y_{i,t}$ denote the lateral deviation from the lane centre of vehicle i at time t, with details on calculating the lateral deviation provided in "Appendix B". For $i = 1, \ldots, N$ and $t = 1, \ldots, T$, we assume

$$y_{i,t} \mid \mu_i, \sigma_i^2 \overset{ind}{\sim} \mathcal{N}\left(\mu_i, \sigma_i^2\right), \tag{44}$$

where μ_i and σ_i^2 are parameters specific to vehicle i. For simplicity, we collect the individual vehicle-specific parameters into a single vector, $\boldsymbol{\theta}_i$, by defining $\boldsymbol{\theta}_i = (\mu_i, \log(\sigma_i^2))$, for $i = 1, 2, .., N$. We note that alternative parametric models could be used here, including a time series model for vehicle i, with little loss in generality.

Multiple cars may display similar behaviour, a phenomenon that can be modelled by allowing different cross-sectional units to share parameters. This structure, whereby cross-sectional units belong to mixture components, leads to predictions that 'borrow strength' from the full sample of vehicles. To make this idea explicit, let k_i denote an indicator variable such that vehicle i belongs to mixture component j if $k_i = j$. All vehicles within the same mixture component share parameters, that is $\theta_i = \theta_j^*$, for all i such that $k_i = j$. Note that the star superscript and j subscript are generally used to index the mixture component that the parameters belong to, while the subscript i is generally used to index the cross-sectional unit, i.e. vehicle.

Since the number of components are unknown and since there is a possibility that a new vehicle will be observed with behaviour that cannot be well described by any of the prevailing parameters, we consider an infinite mixture model. In particular, we use an infinite mixture model induced by a

Dirichlet Process (DP) Prior for the distribution of the parameters. The DP prior is given by

$$G \sim DP(\alpha, G_0)\,, \tag{45}$$

where G_0 is the DP base distribution, assumed here to be $N(\mathbf{0}_2, 10\mathbb{I}_2)$, and the DP concentration parameter α is fixed here and equal to one. The prior for the collection of $\boldsymbol{\theta}_i$ values represent a draw from the DP, with

$$\boldsymbol{\theta}_i | G \overset{iid}{\sim} G, \text{ for } i = 1, 2, \ldots, N\,. \tag{46}$$

Combining (44), (45) and (46) leads to the hierarchy

$$\begin{aligned} &G \sim DP(\alpha, G_0) \\ &\boldsymbol{\theta}_i | G \overset{iid}{\sim} G, \text{ for } i = 1, 2, \ldots, N \\ &y_{i,t} \mid \boldsymbol{\theta}_i \overset{ind}{\sim} \mathcal{N}\left(\mu_i, \sigma_i^2\right), \text{ for } i = 1, 2, \ldots, N \text{ and} \\ &\quad t = 1, 2, \ldots, T. \end{aligned} \tag{47}$$

We note that the DP prior induces clustering on the observation sequences, as described by the Chinese Restaurant Process (CRP, Aldous 1985) representation. The CRP provides a mechanism for drawing from the prior of $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_n$, marginal of the random G, via the introduction of discrete variables that act as component indicators. Define s_i as the number of unique values in $k_1, k_2, \ldots, k_i$, and let $n_{ij} = \sum_{m=1}^{i} I(k_m = j)$. Then, the indicator variables can be simulated from $p(k_i = j | \alpha, \boldsymbol{k}_{1:i-1})$ where

$$p(k_1 = 1 \mid \alpha, G_0) = 1, \tag{48}$$

$$p(k_i = j | \alpha, G_0, \boldsymbol{k}_{1:i-1}) = \begin{cases} \frac{n_{i-1,j}}{\alpha+i-1} & \text{for } j = 1, 2, \ldots, s_{i-1} \\ \frac{\alpha}{\alpha+i-1} & \text{for } j = s_{i-1}+1, \end{cases} \tag{49}$$

for $i = 2, \ldots, N$. Note that although simulation of the indicators does not require knowledge of G_0, we include explicit conditioning on both α and G_0 in (48) and (49) to emphasise the marginalisation over G. Under the CRP, unique values of $\boldsymbol{\theta}_i$, denoted as $\boldsymbol{\theta}_j^*$, for $j = 1, 2, \ldots, s_N$ are drawn from the base distribution G_0, and if we set $\boldsymbol{\theta}_i = \boldsymbol{\theta}_j^*$ for all i such that $k_i = j$, then $(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_N)$ is a draw from the hierarchical setup in (47). Note that although the model is an infinite component mixture model, under the CRP the maximum number of unique clusters, s_N, can be no greater than the number of vehicles in the sample, N. For simplicity, we retain the full vector $\boldsymbol{\theta}_{1:N}^*$, noting that some values $\boldsymbol{\theta}_{1:N}^*$, may not be associated with any vehicle.

The overall model may be seen as a Dirichlet Process Mixture (DPM) model for the lane deviations. Background material regarding Bayesian analysis of DPM models, including many references and detailed discussions relating to MCMC-based techniques for sampling from the relevant posterior, is given in Müller et al. (2015). Online VB-based inference for DPMs has been established using a Mean Field approach see, e.g. Hoffman et al. (2010); Wang et al. (2011) and Kabisa et al. (2016)). In contrast, our approach incorporates SVB, which allows for greater flexibility regarding the form of the approximating posterior distribution. Another important distinction between our analysis and this literature is that our setting involves multiple observations over time, on each cross-sectional unit. Rather than updating as new cross-sectional units are observed, we update parameters relating to the same cross-sectional units observed periodically over a period of time.

7.2 Implementation of SVB at time T_1

Before discussing how UVB is applied to this problem it is instructive to discuss how SVB is implemented for the DPM in (47) that targets the posterior conditional on all cross-sectional units N over just the first time period from $t = 1$ to $t = T_1$. For notational convenience, conditional dependence on α and G_0 is suppressed in all notation for the remainer of this section. The objective is to minimise the KL divergence between a suitable variational approximation and a posterior that is augmented by indicator variables. To implement SVB, we must evaluate

$$\begin{aligned} p(\boldsymbol{y}_{1:N,1:T_1}, \boldsymbol{\theta}_{1:N}^*, \boldsymbol{k}_{1:N}) = &\left[\prod_{i=1}^{N}\prod_{t=1}^{T_1} p(y_{i,t} | \boldsymbol{\theta}_{1:N}^*, k_i)\right] \\ &\times \left[\prod_{i=1}^{N} p(k_i | \boldsymbol{k}_{1:i-1})\right] p(\boldsymbol{\theta}_{1:N}^*) \end{aligned} \tag{50}$$

for given values of $\boldsymbol{y}_{1:N,1:T_1}, \boldsymbol{\theta}_{1:N}^*$, and $\boldsymbol{k}_{1:N}$,. Each of the three main components on the right-hand side of (50) can be computed from the hierarchical structure in (47) and the CRP, as Sethuraman (1994) shows that the unique values $\boldsymbol{\theta}_{1:N}^*$ are *a priori* independent and identically distributed according to the base distribution G_0.

A second required input into SVB is an approximate posterior density structure, given by $q_{\boldsymbol{\lambda}}$, and for this we propose

$$\begin{aligned} q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}_{1:N}^*, \boldsymbol{k}_{1:N} | \boldsymbol{y}_{1:N,1:T_1}) = &\left[\prod_{j=1}^{N} q_j(\theta_j^* | \boldsymbol{y}_{1:N,1:T_1})\right] \\ &\times \left[\prod_{i=1}^{N} p(k_i | \boldsymbol{y}_{1:N,1:T_1}, \boldsymbol{\theta}_{1:N}^*, \boldsymbol{k}_{1:i-1})\right]. \end{aligned} \tag{51}$$

Each $q_j(.)$ on the right-hand side is a bivariate normal distribution with unique means, variances and covariances for each $i = 1, 2, .., N$, leading to a total of $5N$ auxiliary parameters in the approximation. In the second product term on the right-hand side of (51), the notation p is used instead of q since $p(k_i|\boldsymbol{y}_{1:N,1:T_1}, \boldsymbol{k}_{1:i-1}, \boldsymbol{\theta}^*_{1:N})$ is known exactly and can be computed recursively using

$$p(k_i = j|\boldsymbol{y}_{1:N,1:T_1}, \boldsymbol{k}_{1:i-1}, \boldsymbol{\theta}^*_{1:N}) \propto p(k_i = j|\boldsymbol{k}_{1:i-1}) \times p(\boldsymbol{y}_{i,1:T_1}|\boldsymbol{\theta}^*_{1:N}, k_i), \tag{52}$$

for $i = 1, 2, ..., N$.

The use of the so-called full conditional distribution for $\boldsymbol{k}_{1:N}$, given by the second product in (51), is a novel inclusion that enables our model to capture some of the dependence structure of the posterior. In contrast, a MFVB approximation would force posterior independence between each k_i and every θ^*_j, as in, for example, Wang et al. (2011).

Furthermore, in addition to minimising the KL divergence to the augmented posterior, our choice has the benefit of ensuring minimisation of the KL divergence to the corresponding marginal posterior. That is, the augmented gradients are given by

$$\frac{\partial KL[q_{\boldsymbol{\lambda}_1}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_1}) \;||\; p((\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_1})]}{\partial \boldsymbol{\lambda}_1} \tag{53}$$

and are equal to the marginal gradients

$$\frac{\partial KL[q_{\boldsymbol{\lambda}_1}(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_1}) \;||\; p(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_1})]}{\partial \boldsymbol{\lambda}_1}, \tag{54}$$

and so the optimisation procedure is equivalent to one where the indicator variables used to construct the DPM have been marginalised out. The proof of this result is shown in "Appendix C".

7.3 Iterating UVB

Using data up to time T_1, the first UVB posterior is obtained using SVB, as described in Sect. 7.2. For updating at time T_{n+1}, we construct a pseudo-posterior using information from the previous variational approximation $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n})$ in two distinct ways. First, the base distribution in the DP as the prior distribution for $\boldsymbol{\theta}^*_{1:N}$ is updated to reflect the clustering present in the previously obtained posterior, and so is replaced with $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_n})$. Second, retaining the form of the approximation in (51) for the update is complicated by the use of the full conditional distribution for k_i, given by

$$p(k_i = j|\boldsymbol{y}_{1:N,1:T_{n+1}}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1}) \propto p(\boldsymbol{y}_{i,T_n+1:T_{n+1}}|\boldsymbol{\theta}^*_{1:N}, k_i) \times p(k_i = j|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1}), \tag{55}$$

as all currently observed data up to time T_{n+1} is required for each new $\boldsymbol{\theta}^*_{1:N}$ value simulated within the SGA algorithm. Instead our approach is to marginalise the variational distribution using

$$q(k_i = j|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{k}_{1:i-1}) = \int_{\boldsymbol{\theta}^*_{1:N}} q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_n}) \times p(k_i = j|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1}) d\boldsymbol{\theta}^*_{1:N}, \tag{56}$$

before each update, estimating (56) from a sample average of $p(k_i = j|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1})$ using M samples $\boldsymbol{\theta}^*_{1:N}$ and $\boldsymbol{k}_{1:i-1}$ simulated from the available approximate distribution. This requires use of all observed data at T_n, for each of the M samples, but is independent of $\boldsymbol{\theta}^*_{1:N}$ and thus data up to T_n is not required as new $\boldsymbol{\theta}^*_{1:N}$ values are simulated in the SGA algorithm. The component of the variational approximation for k_i is then replaced by

$$\widehat{p}(k_i = j|\boldsymbol{y}_{1:N,1:T_{n+1}}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1}) \propto p(\boldsymbol{y}_{i,T_n+1:T_{n+1}}|\boldsymbol{\theta}^*_{1:N}, k_i = j) \times q(k_i = j|\boldsymbol{y}_{1:N,1:T_{n+1}}, \boldsymbol{k}_{1:i-1}), \tag{57}$$

which may be calculated using only the newly observed data $\boldsymbol{y}_{1:N,T_n+1:T_{n+1}}$ in the SGA algorithm. Note that the marginalisation step for all updates uses the exact full conditional distribution from the CRP representation, $p(k_i = j|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1})$, rather than the marginalised form $\widehat{p}(k_i = j|\boldsymbol{y}_{1:N,1:T_{n+1}}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1})$ from the previous update.

The targeted pseudo-posterior distribution for the update at T_{n+1} is given by

$$\widetilde{p}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}}) \propto \prod_{i=1}^{N} \big[p(\boldsymbol{y}_{i,T_n+1:T_{n+1}}|\boldsymbol{\theta}^*_{1:N}, k_i) \times q(k_i|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{k}_{1:i-1})\big] q_{\boldsymbol{\lambda}^*_n}(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_n}), \tag{58}$$

where the base distribution of the DP posterior in the DPM (and its corresponding CRP) is replaced with its associated variational approximation at time T_n. The approximating distribution for the update at time T_{n+1} is given by

$$q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}}) = \prod_{j=1}^{N} q_{j,n+1}(\theta^*_j|\boldsymbol{y}_{1:N,1:T_{n+1}}) \times \prod_{i=1}^{N} \widehat{p}(k_i|\boldsymbol{y}_{1:N,1:T_{n+1}}, \boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:i-1}). \tag{59}$$

Given the pseudo-posterior (50), form of approximating distribution (59), and components of the time T_n approximation: $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_n})$ and $q(k_i = j|\boldsymbol{y}_{1:N,1:T_n}, \boldsymbol{k}_{1:i-1})$, the optimal parameter vector at time T_{n+1}, $\boldsymbol{\lambda}^*_{n+1}$, may be obtained via Algorithm 3.

Algorithm 3: UVB for the DPM

Input: DP base distribution G_0 or updated approximating distribution at T_n.
Result: Approximating distribution at T_{n+1}.
Calculate (56) for all i.;
Observe $\boldsymbol{y}_{1:N,T_n+1:T_{n+1}}$.;
Set $\mathcal{L}(q, \boldsymbol{\lambda}^{(0)}_{n+1}) = -\infty$.;
Set initial values $\boldsymbol{\lambda}^{(1)}_{n+1}$.;
Set $m = 1$.;
while $|\mathcal{L}(q, \boldsymbol{\lambda}^{(m)}_{n+1}) - \mathcal{L}(q, \boldsymbol{\lambda}^{(m-1)}_{n+1})| < \epsilon$ **do**
  Simulate $\theta^{*(s)}_{1:N} \sim q_{\boldsymbol{\lambda}^{(m)}_{n+1}}(\boldsymbol{\theta}^*_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}})$ for $s = 1, 2, \ldots, S$.;
  Simulate $\boldsymbol{k}^{(s)}_{1:N}$ with probabilities (52) or (57).;
  Evaluate $\widetilde{p}(\boldsymbol{y}_{1:N,1:T_{n+1}}, \boldsymbol{\theta}^{*(s)}_{1:N}, \boldsymbol{k}^{(s)}_{1:i-1})$.;
  Evaluate $q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}^{*(s)}_{1:N}, \boldsymbol{k}^{(s)}_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}})$.;
  Evaluate $\partial q_{\boldsymbol{\lambda}_{n+1}}(\boldsymbol{\theta}^{*(s)}_{1:N}, \boldsymbol{k}^{(s)}_{1:N}|\boldsymbol{y}_{1:N,1:T_{n+1}})/\partial \boldsymbol{\lambda}_{n+1}$.;
  Update auxiliary parameter
  $\boldsymbol{\lambda}^{(m+1)}_{n+1} = \boldsymbol{\lambda}^{(m)}_{n+1} + \rho^{(m)} \frac{\partial \widehat{\mathcal{L}(q,\boldsymbol{\lambda}_{n+1})}}{\partial \boldsymbol{\lambda}_{n+1}}\Big|_{\boldsymbol{\lambda}_{n+1}=\boldsymbol{\lambda}^{(m)}_{n+1}}$.;
  Calculate $\mathcal{L}(q, \boldsymbol{\lambda}^{(m+1)}_{n+1})$.;
  Set $m = m + 1$.;
end

7.4 Predicting Lane Positions

Given a posterior approximation $q_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n})$, we may obtain the approximate predictive distribution for vehicle i at some future time $T_n + h$ as

$$q(y_{i,T_n+h}|\boldsymbol{y}_{1:N,1:T_n}) = \int p(y_{i,T_n+h}|\boldsymbol{\theta}^*_{1:N}, k_i) \, q^*_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n}) d\boldsymbol{\theta}^*_{1:N} d\boldsymbol{k}_{1:N}. \quad (60)$$

After obtaining this distribution from samples $\{\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}\}^{(j)} \sim q^*_{\boldsymbol{\lambda}_n}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n})$, for $j = 1, 2, \ldots, M$, we calculate the predictive log score (LS),

$$LS_{i,n,h} = \log(q(y^{(obs)}_{i,T_n+h}|\boldsymbol{y}_{1:N,1:T_n})), \quad (61)$$

where $y^{(obs)}_{i,T_n+h}$ is the observed value of y_{i,T_n+h}. The performance of the UVB algorithm is evaluated by comparing its RMCR relative to those produced by competing methods.

We also infer the DPM model via MFVB using the so-called stick-breaking representation of the Dirichlet Process, as in Wang et al. (2011). This approach estimates the fully factorised posterior approximation, given by

$$q_{\boldsymbol{\lambda}}(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N}|\boldsymbol{y}_{1:N,1:T_n}) = \prod_{j=1}^{N} q(\theta^*_j|\boldsymbol{y}_{1:N,1:T_n}) \, q(k_j|\boldsymbol{y}_{1:N,1:T_n}). \quad (62)$$

This may be used to build a predictive distribution in the same manner as (60). Details of the MFVB approximation are provided in "Appendix 1".

To illustrate the benefits of including posterior dependence in the approximation, we also introduce a parametric and *independent* model, which retains a normal likelihood for each vehicle, i.e.

$$y_{i,t} \sim \mathcal{N}(\mu_i, \sigma^2_i) \quad (63)$$

and assumes for each vehicle an independent uninformative prior, given by

$$p(\mu_i, \sigma^2_i) \propto \sigma^{-2}_i. \quad (64)$$

For this model, the predictive distribution for vehicle i is analytically available as

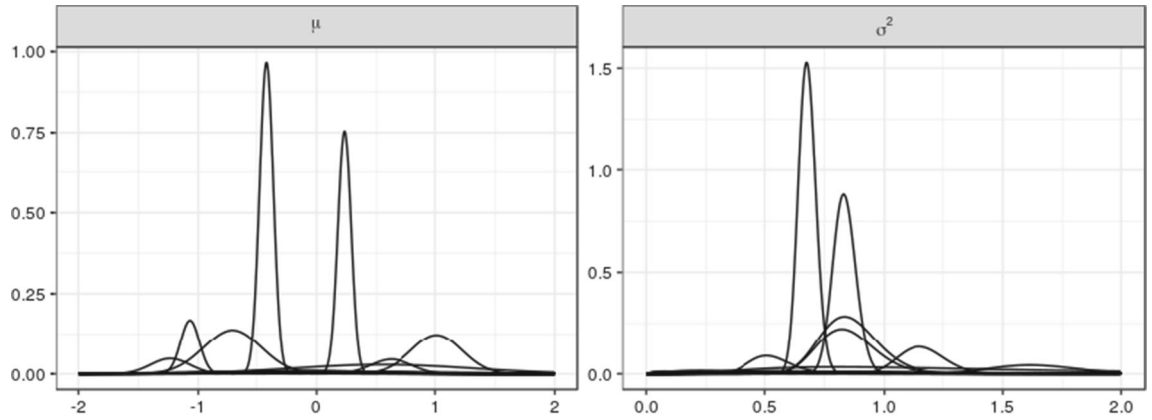
$$p(y_{i,T_n+h}|\boldsymbol{y}_{i,1:T_n}) = \frac{\Gamma\left(\frac{\nu+1}{2}\right)}{\Gamma\left(\frac{\nu}{2}\right)\sqrt{\frac{\pi\nu(T_n+1)s^2_{i,n}}{T_n}}} \left(1 + \frac{T_n\left(y_{i,T_n+h} - \bar{y}_{i,n}\right)^2}{\nu(T_n+1)s^2_{i,n}}\right)^{\frac{-(\nu+1)}{2}} \quad (65)$$

a location-scale transform of the usual Student-t distribution with $\nu = T_n - 1$ degrees of freedom, where $\bar{y}_{i,n}$ and $s^2_{i,n}$ denote the sample mean and variance of $\boldsymbol{y}_{i,1:T_n}$, respectively. Note that this model ignores any information from all other vehicles and is similarly evaluated by the corresponding cumulative predictive log score.

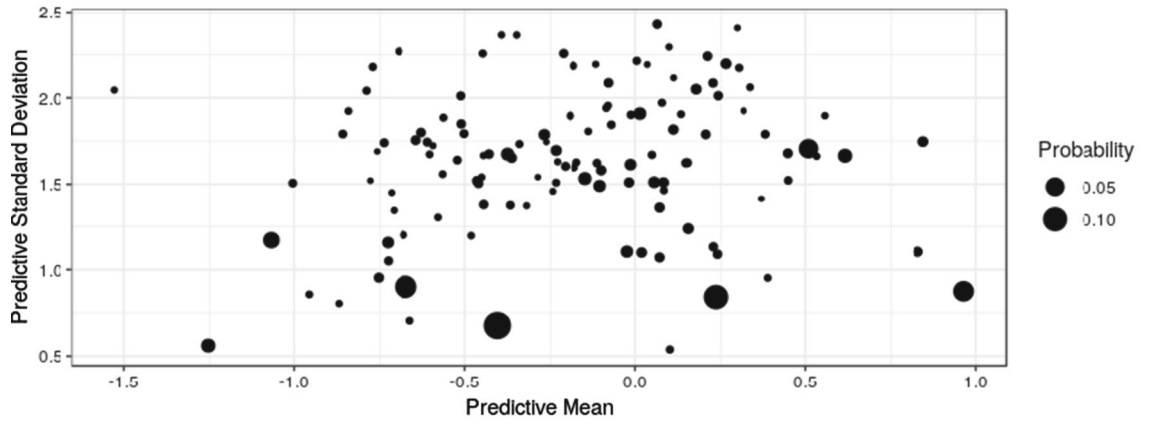
7.5 Analysis of the NGSIM Data

We now discuss the empirical application results from the UVB algorithm for the DPM model described above for the NGSIM data. The posterior updates for both the cluster locations, $\boldsymbol{\theta}^*_{1:N}$, and the indicator variables, $\boldsymbol{k}_{1:N}$, occur at a sequence of pre-determined time periods, given by $T_1 = 50, T_2 = 75, T_3 = 100, T_4 = 125, T_5 = 150$, and $T_6 = 175$.

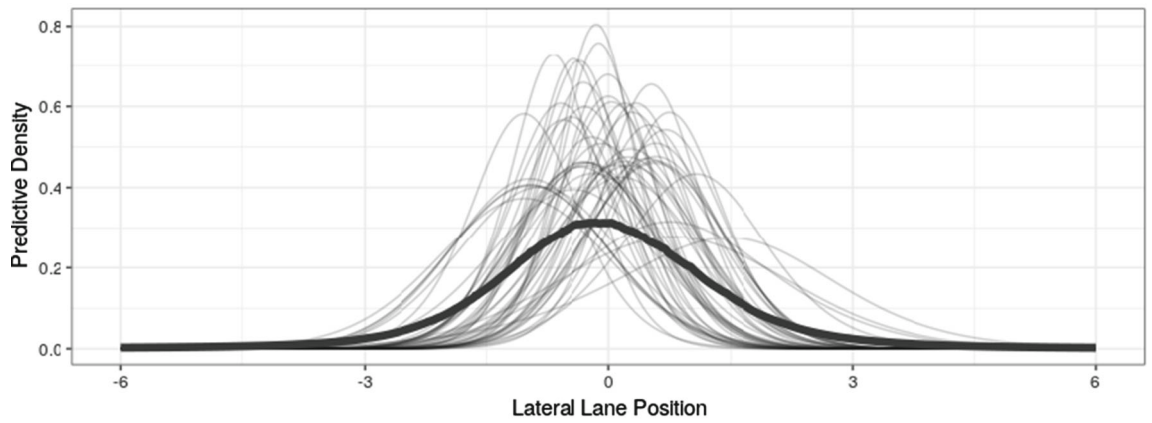
Consider first the two graphs shown in the top panel (panel (a)) of Fig. 6. In each graph, the approximate marginal posterior distributions for each unique value μ^*_j (on the left) and $\sigma^{2,*}_j$ (on the right). Noting there are $N = 500$ marginal densities for each of μ^* and $\sigma^{2,*}$, the plotted densities for each



(a) Weighted θ^* marginal posterior approximation, based on UVB, at time T_6.



(b) UVB Predictive moments for high probability groups, at time T_6.



(c) Individual vehicle and average predictive densities from UVB at time T_6.

Fig. 6 Car lane position example from Section 7. **a**: Posterior approximation for each $\boldsymbol{\theta}^*$, weighted by proportion of $\boldsymbol{k}_{1:N}$ draws. Two groups have high posterior precision with numerous groups showing more uncertainty. **b**: Posterior predictive distribution means and standard deviations, sized according to the top 80% of $\boldsymbol{k}_{1:N}$ draws. **c**: Averaged predictive distribution for all groups in dark blue, with a random subset of fifty per vehicle distributions in grey

parameter are weighted according to the proportion of vehicles in a sample of $M = 100$ draws of $(\boldsymbol{\theta}^*_{1:N}, \boldsymbol{k}_{1:N})$ obtained from the UVB approximation. That is, the weights are calculated according to

$$w_j = \sum_{m=1}^{M} \sum_{i=1}^{N} \frac{I(k_i^{(m)} = j)}{MN}, \tag{66}$$

so that w_j represents the proportion of the MN many sampled k_i values, denoted by $k_i^{(m)}$ for $i = 1, 2, \ldots, N$ and $m = 1, 2, ..., M$, that correspond to the given value of j. The weights suggest that only six of the $\boldsymbol{\theta}^*_j$ values account for the majority of the vehicles, with the six weighted densities associated with μ^*_k and σ^*_k most prominent in the figures shown in panel (a). In contrast, the sample of $\boldsymbol{\theta}^*_j$ values that are seldom (if ever) allocated to a vehicle and hence receive little or no weight appear in these figures as flat lines indistinguishable from zero.

Now turning to panel (b) of Fig. 6, a predictive distribution for new values of y is estimated for each cluster location j, using the M previously simulated values $\boldsymbol{\theta}^*_{1:N}$. The mean of each predictive distribution is plotted against the corresponding predictive standard deviation, with the size of each point given by w_j. The fifty pairs of means and standard deviations shown correspond to 80% of all simulated k_i values, with the results showing that the majority of vehicles belong to a relatively small number of large and cohesive groups, each associated with a distinct predictive mean value coupled with low predictive standard deviation. Members of these groups appear to stay in the same region of their lane, but with these regions spread across both sides of the centre line. There are also many smaller groups, having predictive means closer to zero but with larger standard deviations, perhaps describing idiosyncratic vehicle positioning in the region of the centre lane.

The bottom panel plots, in grey, the individual predictive densities associated with fifty randomly selected vehicles, with the average predictive density over all $N = 500$ vehicles in the sample shown in dark blue. Note that the predictive distribution associated with an individual vehicle will typically itself be comprised of a mixture of components. Importantly, many of the individual predictive densities display reduced uncertainty, relative to the overall average.

We now consider the performance of UVB against several competing methods. Using data up to each time period T_n, we predict the future position y_{i,T_n+h}, $h = 1, 2, \ldots, 50$ for each vehicle using four different predictive distributions described in Sect. 7.4:

1. The DPM predictive distribution (60), with approximate inference provided via UVB.
2. The DPM predictive distribution (60), with approximate inference provided via MFVB,
3. The DPM predictive distribution (60), with approximate inference provided via SVB,
4. The independent model predictive distribution (65), with exact inference.

The mean cumulative predictive log scores (MCLS), averaged across each of the $N = 500$ vehicles, and associated with each of the four types of predictive distributions for individual cars enumerated above, are plotted in Fig. 7.

The results show that, while in each case both approximate implementations of the DPM model outperform the analytically exact independent model, the posterior dependency in the SVB and UVB approximations greatly improves forecasts relative to MFVB. The UVB and SVB lines coincide, and there is no evidence of accumulating approximation error through the UVB recursion relative to the single model fit of SVB. As the amount of data increases, the MFVB and independent model log scores similarly increase. In contrast, the UVB inference MCLS stays at the same level: the $N \times (T_6 - T_1) = 62,500$ additional observations included in T_6 has not provided much marginal information to improve forecasts relative to the original T_1 fit with $N \times T_1 = 25,000$ observations. By construction, the DPM shares information between vehicles, so forecasts of vehicle i are accurate even with only $T_1 = 50$ observations of that particular vehicle. When MFVB inference is employed forecasts are only slightly stronger than the fully independent model that does not share information, implying that the MFVB implementation did not successfully include behaviour of other vehicles.

8 Conclusions

This paper proposes UVB, a framework for SVB inference implemented in a sequential posterior updating setting. UVB is a variational analogue to exact Bayesian updating, where the previous posterior distribution, taken as the prior for the update, is replaced with an approximation itself derived from an earlier SVB approximation. The resulting sequence of posterior distributions can be computed substantially faster than those produced using repeated applications of SVB on the expanding dataset and are amenable to many different types of inferential activities, such as parameter estimation, classification and prediction. In addition, the UVB-IS method provides a further reduction in computational overheads by exploiting information from previously updated UVB posteriors through importance sampling.

The relative inferential and computational performance of posteriors resulting from UVB and UVB-IS are studied in against those that result from 'exact' MCMC, and from repeated SVB, through two simulation experiments.

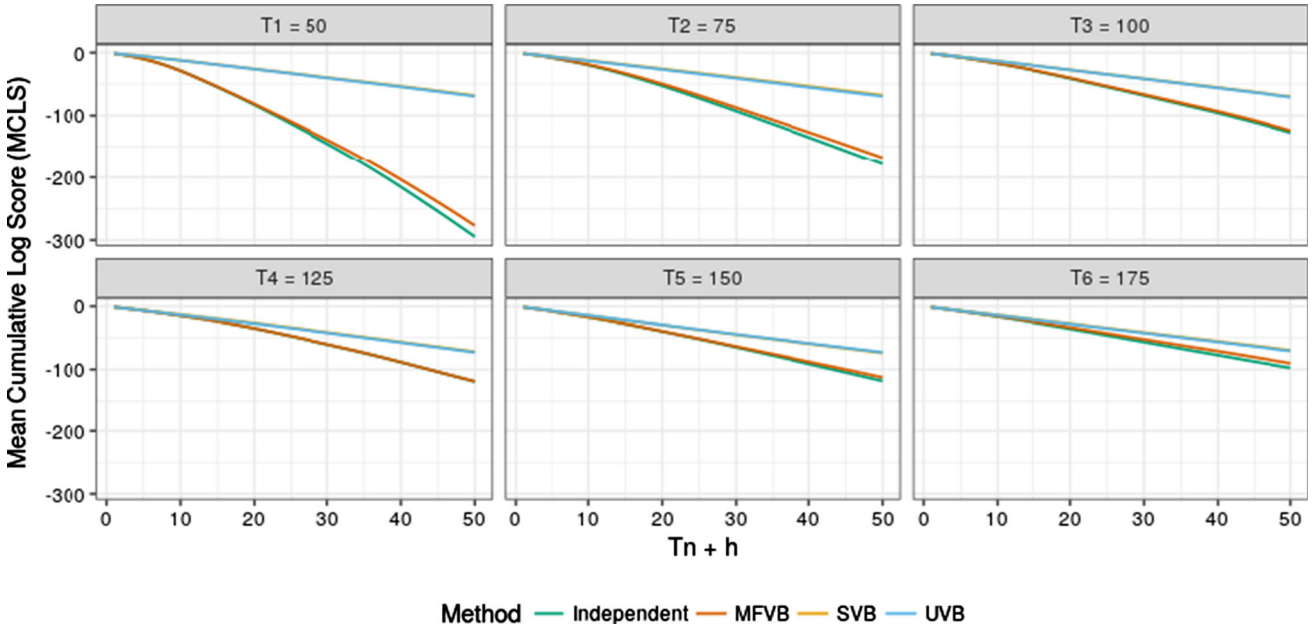


Fig. 7 Car lane position example from Section 7. Mean cumulative predictive log scores (MCLS) for each model averaged across $N = 500$ vehicles. Each model is fit using data up to T_n, then forecasts are made for each of the following fifty observations. The SVB and UVB implementations are visually indistinguishable, while the MFVB implementation performs only slightly better than the fully independent model

One experiment is focused on sequential forecasting of time series data and the other on the progressive clustering of observations from a mixture model. In addition, the UVB approaches are considered in the context of the well-known 'Eight Schools' example, where individual school-based information is incorporated successively. In all cases, large computational savings can be obtained with the UVB methods; however, there is some cost in terms of predictive and parameter inference particularly after several rounds of updating. Whether in other applications the loss of inferential accuracy over several implementations of UVB or UVB-IS will be acceptable will of course depend on the context and corresponding urgency for fast updates, it may be prudent to 'refresh' the updated distributions periodically with an SVB approximation. It should also be noted that the cost in terms of accuracy from using UVB and UVB-IS may ultimately depend on the objective of the analysis. In cases where accuracy is needed in the tails of the posterior, the cost may be large, for predictive accuracy as measured by log score, the cost is negligible, while in our classification setting, UVB actually outperforms SVB in terms of classification accuracy.

The proposed UVB and UVB-IS algorithms are well-suited to situations where up-to-date inference for complex probabilistic models is required whenever data arrive so rapidly as to render MCMC or SVB infeasible, and especially so when inference involves classification and prediction. To illustrate this type of situation, an empirical illustration regarding observed lane positions of vehicles on the US-101 Highway is presented using a Dirichlet Process Mixture. In this implementation of UVB, an approximating distributional family that exploits dependence between cluster locations and indicator variables is detailed. Forecasts of future lane positions produced using UVB are comparable to an SVB approach. Posterior dependence is induced by exploiting the known full conditional distribution for the discrete indicator variables by using these as a component of the approximating distribution. Inferring the model through UVB and SVB outperform inference using MFVB, as this method requires an independent posterior approximation. Future research involves the application of UVB to build a more sophisticated heterogeneous model to provide forecasts of vehicle movement from this dataset in an online fashion—where UVB facilitates model updates and forecasts in a short time-frame after data arrives.

A Post-Hoc Nemenyi tests

To ensure that the differences in performance between methods, both with respect to accuracy via the log score and run time, are significant, we employ the following procedure. First a non-parametric Friedman test is performed on the ranks of each method. Then post-hoc Nemenyi tests are used to detect whether each method is significantly worse than the

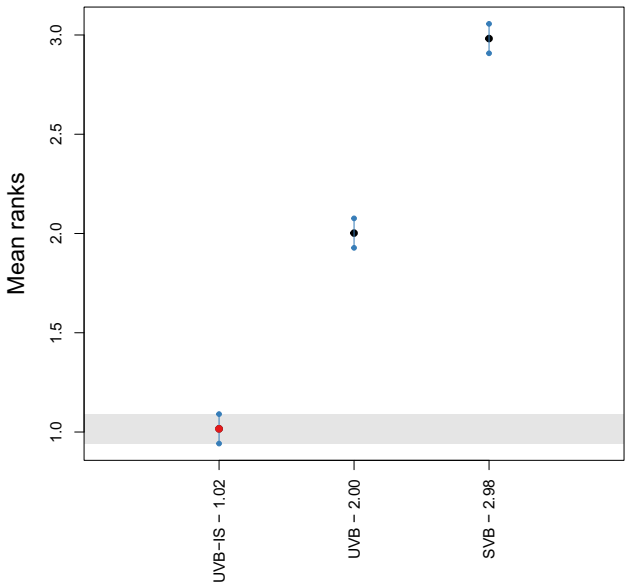


Fig. 8 AR3 example from Section 5.1. Plot showing multiple comparisons from the best method (according to log score) based on ranks. Results are shown for $T_n = 400$ and $K = 3$. Here SVB is the most accurate method according to log score. Since the intervals for UVB and UVB-IS do not overlap with the grey region, these differences are significant. A 95% significance level was used; however, there is no overlap with the grey bands even for a 99% level of significance

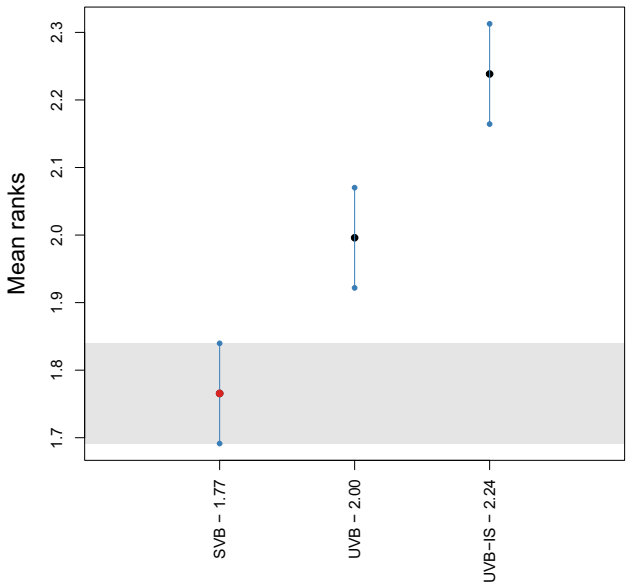


Fig. 9 AR3 example from Section 5.1. Plot showing multiple comparisons from the best method (according to computational time) based on ranks. Results are shown for $T_n = 400$ and $K = 3$. Here UVB-IS is the fastest method. Since the intervals for SVB and UVB do not overlap with the grey region, these differences are significant. A 95% significance level was used; however, there is no overlap with the grey bands even for a 99% level of significance

best performing method. All p-values were less than 0.0001 indicating differences between methods were significant.

The results are presented graphically in Fig. 8 for the log score and in Fig. 9 for run time. These results are from the AR(3) in the simulation study in Sect. 5.1. Results are shown for forecasts made at $T_n = 400$ and with $K = 3$ components in mixture of normals used for the variational approximation. In these plots, if any method has an interval that overlaps with the grey band, then the difference in forecasting accuracy, (or running time) is not significantly different from the best method. It can be clearly seen that the bands do not overlap. As such, we are confident that the number of replications used in the simulation study is sufficiently large to ensure differences in performance are not simply due to variation across the replications. While omitted for brevity, similar plots are obtained for different values of T_n and K and for other simulation studies conducted in this paper.

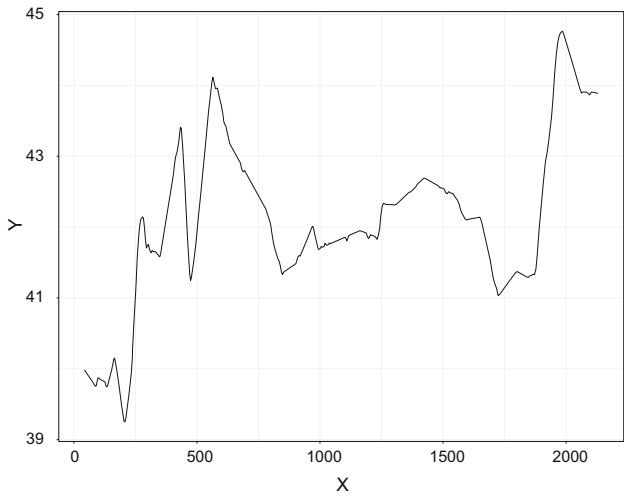


Fig. 10 Car lane position example from Section 7. Coordinate system for one vehicle. The $X-$axis denotes distance travelled along the lane, and the $Y-$axis denotes the relative vertical position in the lane

B Calculation of Lateral Lane Deviation

Let $x_{i,t}$ denote the position of vehicle i along the direction of travel at time t, and $y_{i,t}$ denote the position across the lane, as in Fig. 10 for one vehicle travelling to the right.

For each vehicle i and time t since entering the road, with travel originating at $t = 1$, the total distance travelled up to time t is given by

$$d_{i,t} = \sum_{s=2}^{t} \sqrt{(x_{i,s} - x_{i,s-1})^2 + (y_{i,s} - y_{i,s-1})^2}. \tag{67}$$

Using this distance measure and 100 randomly sampled vehicles per lane, the two-dimensional coordinates corresponding to the centre line of each lane are estimated via independent

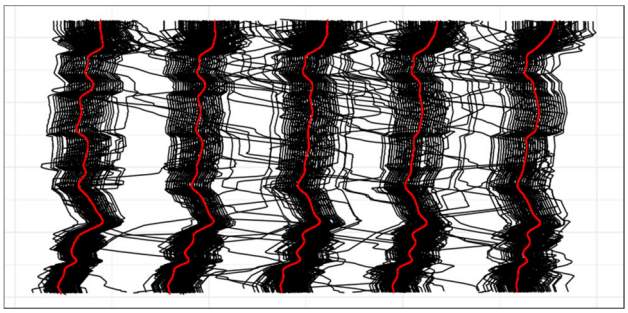


Fig. 11 Car lane position example from Sect. 7. Raw data of paths taken by cars in the five lanes of the highway corresponding to the application in Sect. 7. Each black line charts the path of a single car, with 100 randomly selected cars per lane shown on the figure. The fitted spline models for each lane used to correct for the geometry of the road are overlaid as red lines

smoothing splines, where each coordinate is a function of the distance travelled to that point. Each smoothing spline is calculated using the 'R stats' package R Core Team 2017). The estimated centre line for lane k, is denoted by the curve $\{\widehat{x}_{d,k} = f_x^k(d), \widehat{y}_{d,k} = f_y^k(d)\}$, for $d \geq 0$. The fitted spline models are shown in red overlaying the raw data in Fig. 11

Excluding the vehicles used to estimate the spline models, each of the vehicles in the dataset uses the relevant lane centre line estimate fit from the spline model associated with its starting lane to calculate relative coordinates $\{x^*_{i,t}, y^*_{i,t}\}$. $x^*_{i,t}$ denotes the distance travelled along the road, and $y^*_{i,t}$ denotes the deviation from the lane centre line, and are calculated by

$$\widehat{d}_{i,t} = \arg\min_d \sqrt{(x_{i,t} - f_x^k(d))^2 + (y_{i,t} - f_y^k(d))^2} \tag{68}$$

$$\widehat{x}_{i,t} = f_x^k(\widehat{d}_{i,t}) \tag{69}$$

$$\widehat{y}_{i,t} = f_y^k(\widehat{d}_{i,t}) \tag{70}$$

$$y^*_{i,t} = \text{sign}\left(\tan\left(\frac{\widehat{x}_{i,t} - x_{i,t}}{\widehat{y}_{i,t} - y_{i,t}}\right)^{-1} - \tan\left(\frac{f_x'^{,k}(\widehat{d}_{i,t})}{f_y'^{,k}(\widehat{d}_{i,t})}\right)^{-1}\right) \sqrt{(x_{i,t} - \widehat{x}_{i,t})^2 + (y_{i,t} - \widehat{y}_{i,t})^2} \tag{71}$$

The coordinate pair $(\widehat{x}_{i,t}, \widehat{y}_{i,t})$ denotes the closest position of the spline model to the actual vehicle position given by the pair $(x_{i,t}, y_{i,t})$. Lateral deviation $y^*_{i,t}$ has magnitude equal to that of the vector from $(\widehat{x}_{i,t}, \widehat{y}_{i,t})$ to $(x_{i,t}, y_{i,t})$. A negative sign on $y^*_{i,t}$ indicates that the vehicle is to the left of the lane centre, and a positive sign indicates that the vehicle is to the right of the lane centre.

C Equivalence of Augmented and Marginal KL Divergence Gradients

Consider the augmented posterior distribution

$$p(\theta, k|y) \propto p(y|\theta, k)p(k|\theta)p(\theta) \tag{72}$$

and variational approximation given by

$$q_\lambda(\theta, k|y) = q_\lambda(\theta|y)p(k|y, \theta). \tag{73}$$

The corresponding KL divergence, $KL[q_\lambda(\theta, k|y) \;||\; p(\theta, k|y)]$, is indirectly minimised using the gradient

$$\frac{\partial KL[q_\lambda(\theta, k|y) \;||\; p(\theta, k|y)]}{\partial \lambda} = -\frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda}, \tag{74}$$

where the gradient $\partial\mathcal{L}(q, \lambda)/\partial\lambda$ is the score-based gradient of the ELBO, given by

$$\frac{\partial \mathcal{L}(q, \lambda)}{\partial \lambda} = \int_{\theta,k} q_\lambda(\theta, k|y) \frac{\partial \log(q_\lambda(\theta, k|y))}{\partial \lambda} (\log(p(y, \theta, k)) - \log(q_\lambda(\theta, k|y))\, d\theta dk. \tag{75}$$

Next, consider the associated marginal posterior distribution,

$$p(\theta|y) \propto p(y|\theta)p(\theta), \tag{76}$$

and consider using as the variational approximation $\widetilde{q}_\lambda(\theta|y)$ given by the first component (only) on the right-hand side of (73), i.e. $\widetilde{q}_\lambda(\theta|y) \equiv q_\lambda(\theta|y)$. Note that, as a consequence, $\log q_\lambda \equiv \log \widetilde{q}_\lambda$ and $\frac{\partial \log q_\lambda}{\partial \lambda} \equiv \frac{\partial \log \widetilde{q}_\lambda}{\partial \lambda}$. The KL divergence in this case, $KL[\widetilde{q}_\lambda(\theta|y) \;||\; p(\theta|y)]$, has gradient given by

$$\frac{\partial KL[\widetilde{q}_\lambda(\theta|y) \;||\; p(\theta|y)]}{\partial \lambda} = -\frac{\partial \mathcal{L}(\widetilde{q}, \lambda)}{\partial \lambda}, \tag{77}$$

where $\partial\mathcal{L}(\widetilde{q}, \lambda)/\partial\lambda$ is

$$\frac{\partial \mathcal{L}(\widetilde{q}, \lambda)}{\partial \lambda} = \int_\theta \widetilde{q}_\lambda(\theta|y) \frac{\partial \log(\widetilde{q}_\lambda(\theta|y))}{\partial \lambda} (\log(p(y, \theta)) - \log(\widetilde{q}_\lambda(\theta|y)))\, d\theta. \tag{78}$$

Here we show that (75) is equal to (78), and hence the gradient of both KL divergences are equal, and must share local minima.

Begin by expanding each joint density in (75) by (72) and (73),

$$\begin{aligned}\frac{\partial \mathcal{L}(q,\lambda)}{\partial \lambda} &= \int_{\theta,k} q_\lambda(\theta|y)p(k|\theta,y) \\ &\quad \times \frac{\partial(\log(q_\lambda(\theta|y)) + \log(p(k|\theta,y))}{\partial\lambda} \\ &\quad \times (\log(p(\theta)p(y|k,\theta)p(k|\theta)) \\ &\quad - \log(q_\lambda(\theta|y)p(k|\theta,y)))\,d\theta dk \qquad (79) \\ &= \int_{\theta,k} q_\lambda(\theta|y)p(k|\theta,y)\frac{\partial \log(q_\lambda(\theta|y))}{\partial\lambda} \\ &\quad \times \left(\log(p(\theta) + \log\left(\frac{p(y|k,\theta)p(k|\theta))p(y|\theta)}{p(y|\theta)}\right)\right. \\ &\quad \left. - \log(q_\lambda(\theta|y)) - \log(p(k|\theta,y))\right)d\theta dk \qquad (80)\end{aligned}$$

as the term $\log(p(k|\theta, y))$ is independent of λ. Then

$$\begin{aligned}\frac{\partial \mathcal{L}(q,\lambda)}{\partial \lambda} &= \int_{\theta,k} q_\lambda(\theta|y)p(k|\theta,y)\frac{\partial \log(q_\lambda(\theta|y))}{\partial\lambda} \\ &\quad \times (\log(p(\theta) + \log(p(y|\theta)) + \log(p(k|y,\theta)) \\ &\quad - \log(q_\lambda(\theta|y)) \\ &\quad - \log(p(k|\theta,y)))\,d\theta dk \qquad (81)\end{aligned}$$

by Bayes' Rule. Cancelling $\log(p(k|y, \theta))$ results in

$$\begin{aligned}\frac{\partial \mathcal{L}(q,\lambda)}{\partial \lambda} &= \int_{\theta,k} q_\lambda(\theta|y)p(k|\theta,y)\frac{\partial \log(q_\lambda(\theta|y))}{\partial\lambda} \\ &\quad (\log(p(\theta) + \log(p(y|\theta)) - \log(q_\lambda(\theta|y)))\,d\theta dk \qquad (82) \\ &= \int_\theta \left(\int_k p(k|\theta,y)dk\right) \quad q_\lambda(\theta|y)\frac{\partial \log(q_\lambda(\theta|y))}{\partial\lambda} \\ &\quad (\log(p(y,\theta) - \log(q_\lambda(\theta|y)))\,d\theta. \qquad (83)\end{aligned}$$

The final expression (83) is equivalent to the marginal model gradient (78) and the proof is complete.

D Mean Field Variational Bayes Implementation of the Dirichlet Process Mixture

Implementation of MFVB for this model follows the offline coordinate ascent approach of Wang et al. (2011), employing the stick-breaking construction of the Dirichlet Process as

$$\theta_j^* \overset{iid}{\sim} G_0, \qquad (84)$$

$$\beta_j' \overset{iid}{\sim} Beta(1,\alpha), \qquad (85)$$

$$\beta_N' = 1, \qquad (86)$$

$$\beta_j = \beta_j' \prod_{l=1}^{j-1}(1-\beta_l'), \qquad (87)$$

$$G = \sum_{j=1}^{N} \beta_j \delta(\theta_j^*), \qquad (88)$$

where δ is the Dirac Delta function. The stick-breaking construction is equivalent to the CRP representation of the DP, after marginalisation over β Miller 2018), and is similarly augmented with the set of indicator variables $k_{1:N}$. In this case, the prior distribution is given by

$$k_i \sim Multinomial(\boldsymbol{\beta}_{1:N}). \qquad (89)$$

The contribution to the likelihood from observation i is then determined by

$$\boldsymbol{y}_{i,1:T_1}|\boldsymbol{\theta}_{1:N}^*, k_i \sim \mathcal{N}\left(\mu_{k_i}^*, \sigma_{k_i}^{2*}\right). \qquad (90)$$

To maintain the analytical tractability of the MFVB approximation, we replace the base distribution G_0 with a conjugate prior for the normal likelihood,

$$\mu^*|G_0 \sim \mathcal{N}(0,10) \qquad (91)$$

$$\sigma^{2*}|G_0 \sim InverseGamma(shape = \alpha_0, scale = \kappa_0) \qquad (92)$$

where α_0 and κ_0 are chosen to be the MLE values for the inverse gamma distribution, estimated from 100,000 samples of the implied lognormal(0, 10) distribution for σ^{2*} that was used in the SVB and UVB approaches. These values are estimated by the second algorithm of Llera and Beckmann (2016) as

$$\alpha_0 = 0.15275 \qquad (93)$$

$$\kappa_0 = 0.00102. \qquad (94)$$

The variational approximation employed is of the form

$$q_{\boldsymbol{\lambda}_n}(\boldsymbol{k}_{1:N}, \boldsymbol{\beta}_{1:N}', \boldsymbol{\theta}_{1:N}^*) = \prod_{i=1}^{N} q(k_i)q(\beta_i')q(\mu_i^*)q(\sigma_i^{2*}) \qquad (95)$$

with

$$k_i \sim Multinom(\boldsymbol{\rho}_i) \tag{96}$$
$$\beta_i' \sim Beta(a_i, b_i) \tag{97}$$
$$\mu_i^* \sim \mathcal{N}(\gamma_i, \tau_i) \tag{98}$$
$$\sigma_i^{2*} \sim Inv.Gamma(\alpha_i, \kappa_i) \tag{99}$$

Coordinate ascent algorithms for MFVB consists of cycling through in a set of equations for each parameter until the change in each parameter is below some threshold. For this model, the equations are given by

$$a_j = 1 + \sum_{i=1}^{N} \rho_{ij}, \tag{100}$$
$$b_j = \alpha + \sum_{i=1}^{N} \sum_{l=j+1}^{N} \rho_{il}, \tag{101}$$
$$\rho_{ij} \propto -\frac{T_n}{2} E_q[\log(\sigma_j^{2*})] - \frac{\alpha_j}{2\kappa_j}\left(\sum_{t=1}^{T_n} y_{it}^2 - 2\gamma_j \sum_{t=1}^{T_n} y_{it} + T_n(\gamma_j^2 + \tau_j)\right) + E_q[\log(\beta_j)], \tag{102}$$
$$\gamma_j = \frac{10\frac{\alpha_j}{\kappa_j}\sum_{i=1}^{N}\sum_{t=1}^{T_n}\rho_{ij}y_{it}}{10T_n\frac{\alpha_j}{\kappa_j}\sum_{i=1}^{N}\rho_{ij} + 1}, \tag{103}$$
$$\tau_j = \frac{10}{10T_n\frac{\alpha_j}{\kappa_j}\sum_{i=1}^{N}\rho_{ij} + 1}, \tag{104}$$
$$\alpha_j = \alpha_0 + \frac{T_n}{2}\sum_{i=1}^{N}\rho_{ij}, \tag{105}$$
$$\kappa_j = \kappa_0 + \frac{1}{2}\left(\sum_{i=1}^{N}\rho_{ij}\left(\sum_{t=1}^{T_n} y_{it}^2 + T_n(\gamma_j^2 + \tau_j) - 2\gamma_j\sum_{t=1}^{T_n} y_{it}\right)\right). \tag{106}$$

The expectations $E_q[\log(\beta_j)]$ are available in closed form as

$$E_q[\log(\beta_j)] = E_q[\log(\beta_j')] + \sum_{l=1}^{j-1} E_q[\log(1-\beta_j')] \tag{107}$$

where

$$E_q[\log(\beta_j')] = \Psi(a_j) - \Psi(a_j + b_j) \tag{108}$$

and

$$E_q[\log(1-\beta_j')] = \Psi(b_j) - \Psi(a_j + b_j) \tag{109}$$

where Ψ is the digamma function. The expectation $E_q[\log(\sigma_j^{2*})]$ does not have a closed-form solution but is estimated from samples of $q(\sigma_j^{2*})$.

References

Aldous, D. J.: Exchangeability and related topics. In: Ecole d'Ete de Probabilities de Saint-Flour XIII 1983 (1985)

Attias, H.: A variational Bayesian framework for graphical models. In: Advances in Neural Information Processing Systems 12 (1999)

Bhattacharya, A., Wilson, S.P.: Sequential Bayesian inference for static parameters in dynamic state space models. Comput. Stat. Data Anal. **127**, 187–203 (2018)

Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Berlin (2006)

Blei, D.M., Kucukelbir, A., McAuliffe, J.D.: Variational inference: a review for statisticians. J. Am. Stat. Assoc. **112**(518), 859–877 (2017)

Bottou, L.: Large-scale machine learning with stochastic gradient descent. In: Proceedings of COMPSTAT'2010 (2010)

Boyen, X., Koller, D.: Tractable inference for complex stochastic processes arXiv:1301.7362, (2013)

Broderick, T., Boyd, N.: Wibisono A, Wilson AC, Jordan MI, Streaming variational Bayes. In: Advances in Neural Information Processing Systems 26 (2013)

Chen, X., Dai, H., Song, L.: Meta particle flow for sequential Bayesian inference. arXiv:1902.00640 [csLG], (2019)

Chopin, N.: A sequential particle filter method for static models. Biometrika **89**, 539–551 (2002)

Del Moral, P., Jasra, A., Lee, A., Yau, C., Zhang, X.: The alive particle filter and its use in particle Markov chain Monte Carlo. Stochast. Analy. Appl. **33**(6), 943–974 (2015)

Doucet, A., Lee, A.: Sequential Monte Carlo methods. Chapman and Hall, chap **7**, 165–189 (2018)

Doucet, A., de Freitas, N., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer, Berlin (2001)

FHWA: Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data. Available online at https://data.transportation.gov/Automobiles/Next-Generation-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj, (2017)

Garthwaite, P.H., Fan, Y., Sisson, S.A.: Adaptive optimal scaling of Metropolis-Hastings algorithms using the Robbins–Monro process. Commun. Stat. Theory Methods **45**(17), 5098–5111 (2016). https://doi.org/10.1080/03610926.2014.936562

Gefang, D., Koop, G., Poon, A.: Variational Bayesian inference in large vector autoregressions with hierarchical shrinkage (2019)

Gelman, A., Gilks, W.R., Roberts, G.O.: Weak convergence and optimal scaling of random walk Metropolis algorithms. Ann. Appl. Probab. **7**, 110–120 (1997)

Gelman, A., Carlin, J.B., Stern, H.S., Dunson, D.B., Vehtari, A., Rubin, D.B.: Bayesian Data Analysis, 3rd edn. CRC Press, Cambridge (2014)

Ghahramani, Z., Beal, M.J.: Propagation algorithms for variational Bayesian learning. In: Advances in Neural Information Processing Systems 13 (2000)

Gilks, W.R., Best, N.G., Tan, K.K.C.: Adaptive rejection Metropolis sampling within Gibbs sampling. J. R. Stat. Soc. Ser. C Appl. Stat. **44**(4), 445–472 (1995)

Gilks, W.R., Richardson, S., Spiegelhalter, D.: Markov Chain Monte Carlo in Practice. Chapman and Hall, London (1995)

Gunawan, D., Kohn, R., Nott, D.: Variational Bayes approximation of factor stochastic volatility models. Int. J. Forecast. (2021)

Hoffman, M., Bach, F.R., Blei, D.M.: Online learning for latent Dirichlet allocation. In: Advances in Neural Information Processing Systems 23 (2010)
Hoffman, M., Blei, D.M., Wang, C., Paisley, J.: Stochastic variational inference. J. Mach. Learn. Res. **14**, 1303–1347 (2013)
Jasra, A., Singh, S.S., Martin, J.S.: Filtering via approximate Bayesian computation. Stat. Comput. **22**, 1223–1237 (2010)
Jordan, M.I., Ghahramani, Z., Jaakola, T.S., Saul, L.K.: An introduction to variational methods for graphical models. Mach. Learn. **37**, 183–233 (1999)
Kabisa, S.T., Dunson, D.B., Morris, J.S.: Online variational Bayes inference for high-dimensional correlated data. J. Comput. Graph. Stat. **25**, 426–444 (2016)
Kingma, D.P., Ba, J.L.: arXiv:1412.6980v9 [csLG], (2014)
Kingma, D.P., Welling, M.: Auto-encoding variational Bayes. arXiv:1312.6114 [statML], (2014)
Krishnamurthy, A., Kandasamy, K., Poczos, B., Wasserman, L.: Nonparametric estimation of Rényi divergence and friends. In: International Conference on Machine Learning, PMLR, pp 919–927, (2014)
Kullback, S., Leibler, R.A.: On information and sufficiency. Ann. Math. Stat. **22**(1), 79–86 (1951)
Lauritzen, S.L.: Propagation of probabilities, means, and variances in mixed graphical association models. J. Am. Stat. Assoc. https://doi.org/10.1080/01621459.1992.10476265, (1992)
Llera, A., Beckmann, C.F.: Estimating an inverse gamma distribution. arXiv:1605.01019 [statME], (2016)
Maybeck, P.S.: Stochastic models, estimation, and control. Academic press, (1982)
Miller, J.W.: An elementary derivation of the chinese restaurant process from Sethuraman's stick-breaking process. arXiv:1801.00513 [mathST], (2018)
Müller, P., Quintana, F.A., Jara, A., Hanson, T.: Bayesian Nonparametric Data Analysis. Springer, Berlin (2015)
Ong, V.M.H., Nott, D.J., Smith, M.S.: Gaussian variational approximation with a factor covariance structure. J. Comput. Graph. Stat. **27**(3), 465–478 (2018)
Opper, M., Winther, O.: A Bayesian approach to on-line learning. Online learning in neural networks pp 363–378, (1998)
R Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, available online at https://www.R-project.org/, (2017)
Ranganath, R., Gerrish, S., Blei, M. David: Black box variational inference. In: Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics, (2014)
Robbins, H., Monro, S.: A stochastic approximation method. Ann. Math. Stat. **22**(3), 400–407 (1951)
Sakaya, J., Klami, A.: Importance sampled stochastic optimization for variational inference. In: Uncertainty in Artificial Intelligence, (2017)
Sato, M.: Online model selection based on variational Bayes. Neural Comput. **13**, 1649–1681 (2001)
Sethuraman, J.: A constructive definition of Dirichlet priors. Statistica Sinica **4**, 639–650 (1994)
Smidl, V.: The variational Bayes approach in signal processing. PhD thesis, Trinity College, The University of Dublin (2004)
Stan Development Team: RStan Getting Started. Available online at https://github.com/stan-dev/rstan/wiki/RStan-Getting-Started, (2018)
Tan, L.S., Nott, D.J.: Gaussian variational approximation with sparse precision matrices. Stat. Comput. **28**(2), 259–275 (2018)
Titsias, M.K., Lázaro-Gredilla, M.: Doubly stochastic variational Bayes for non-conjugate inference. In: Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32 (2014)
Wainwright, M.J., Jordan, M.I.: Graphical models, exponential families, and variational inference. Found. Trends Mach. Learn. **1**(1–2), 1–305 (2008)
Wang, C., Paisley, J., Blei, D.: Online variational inference for the hierarchical Dirichlet process. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (2011)
Zhang, C., Bütepage, J., Kjellström, H., Mandt, S.: Advances in variational inference. arXiv:1711.05597 [csLG], (2017)