Streaming stochastic variational Bayes; An improved approach for Bayesian inference with data streams

Nadheesh Jihan [Corresp., 1], **Malith Jayasinghe** [1], **Srinath Perera** [1]

[1] CTO Office, WSO2, Colombo 03, Sri Lanka

Corresponding Author: Nadheesh Jihan
Email address: nadheesh@wso2.com

Online learning is an essential tool for predictive analysis based on continuous, endless data streams. Adopting Bayesian inference for online settings allows hierarchical modeling while representing the uncertainty of model parameters. Existing online inference techniques are motivated by either the traditional Bayesian updating or the stochastic optimizations. However, traditional Bayesian updating suffers from overconfidence posteriors, where posterior variance becomes too inadequate to adapt to new changes to the posterior. On the other hand, stochastic optimization of variational objective demands exhausting additional analysis to optimize a hyperparameter that controls the posterior variance. In this paper, we present ''Streaming Stochastic Variational Bayes" (SSVB)—a novel online approximation inference framework for data streaming to address the aforementioned shortcomings of the current state-of-the-art. SSVB adjusts its posterior variance duly without any user-specified hyperparameters while efficiently accommodating the drifting patterns to the posteriors. Moreover, SSVB can be easily adopted by practitioners for a wide range of models (i.e. simple regression models to complex hierarchical models) with little additional analysis. We appraised the performance of SSVB against Population Variational Inference (PVI), Stochastic Variational Inference (SVI) and Black-box Streaming Variational Bayes (BB-SVB) using two non-conjugate probabilistic models; multinomial logistic regression and linear mixed effect model. Furthermore, we also discuss the significant accuracy gain with SSVB based inference against conventional online learning models for each task.

Streaming Stochastic Variational Bayes; An Improved Approach for Bayesian Inference with Data Streams

Nadheesh Jihan[1], **Malith Jayasinghe**[1], **and Srinath Perera**[1]

[1]**WSO2 Inc., Mountain View, CA, USA**

Corresponding author:
Nadheesh Jihan[1]

Email address: nadheesh@wso2.com

ABSTRACT

Online learning is an essential tool for predictive analysis based on continuous, endless data streams. Adopting Bayesian inference for online settings allows hierarchical modeling while representing the uncertainty of model parameters. Existing online inference techniques are motivated by either the traditional Bayesian updating or the stochastic optimizations. However, traditional Bayesian updating suffers from overconfidence posteriors, where posterior variance becomes too inadequate to adapt to new changes to the posterior. On the other hand, stochastic optimization of variational objective demands exhausting additional analysis to optimize a hyperparameter that controls the posterior variance. In this paper, we present "Streaming Stochastic Variational Bayes" (SSVB) —a novel online approximation inference framework for data streaming to address the aforementioned shortcomings of the current state-of-the-art. SSVB adjusts its posterior variance duly without any user-specified hyperparameters while efficiently accommodating the drifting patterns to the posteriors. Moreover, SSVB can be easily adopted by practitioners for a wide range of models (i.e. simple regression models to complex hierarchical models) with little additional analysis. We appraised the performance of SSVB against Population Variational Inference (PVI), Stochastic Variational Inference (SVI) and Black-box Streaming Variational Bayes (BB-SVB) using two non-conjugate probabilistic models; multinomial logistic regression and linear mixed effect model. Furthermore, we also discuss the significant accuracy gain with SSVB based inference against conventional online learning models for each task.

INTRODUCTION

More and more applications are required to respond to data as soon as possible. Among real-world applications are sensor networks, stock market systems, market trend analysis, and online recommendation systems. To address such use cases, applications need to source data directly from their sources. Data streams are a useful abstraction for such use cases. We can apply machine learning to such data streams using both online or offline models. The offline models are easier, yet get outdated when new data becomes available, which may affect the accuracy of the predictions. Moreover, offline learning requires storing these large data streams in memory, which is infeasible for some cases. When these limitations are critical, online learning has become an essential tool in such occasions, which updates the model continuously with each data point or mini-batch observed by the model.

On the other hand, Bayesian learning is recognized as an essential workhorse in Machine learning and statistical analysis due to its desirable properties. Those properties include; incremental learning with recursive Bayesian updates to the posterior, flexible feature modeling with hierarchical models, ability to incorporate beliefs and past experience through the prior, and most importantly the ability to estimate the uncertainty of predictions. Hence, extending Bayesian learning for streaming setting enables the online inference of a wide range of models (i.e. simple regression models to complex hierarchical models).

Furthermore, adopting Bayesian learning techniques to online learning enables the ability to express the uncertainty of prediction, which leads to reliable decision making and analytic in most of the domains. Even-though uncertainty was an underappreciated concept in machine learning up until recently, many

real-world applications now shift towards the use of Bayesian uncertainty (Gal, 2016). Especially with endless and non-stationary data streams, the uncertainty of the model parameters can be useful to model the uncertainty from real-world data in predictions.

Even though Bayesian learning is recognized to be useful in online settings, the exact posterior inference is rarely tractable for both offline and online learning. Thus, sampling techniques such as Markov Chain Monte Carlo (MCMC) sampling or approximation inference techniques such as Variational Inference (VI) (Wainwright and Jordan, 2008) are commonly adopted in practice as an alternative. Especially, VI is shown to be useful with large-scale, finite data streams by Hoffman et al. (2013, 2010); Wang et al. (2011). In these techniques, they have applied "Stochastic Variational Inference" (SVI) (Hoffman et al., 2013), which optimizes the typical variational objective—Evidence Lower Bound (ELBO) based on mini-batches. Usually, SVI demands tedious model specific derivations and implementations (Blei et al., 2017). The black-box inference techniques (Kucukelbir et al., 2017; Ranganath et al., 2014; Kingma and Welling, 2013) extend SVI avoiding such exhausting model specific analysis, allowing practitioners to explore a wide range of models with little additional derivations. However, any of the above approaches are not tailored to use with endless streaming data. Their intended use is to approximate the posteriors for model parameters given a finite dataset with N data-points, where N governs the impact of prior and likelihood to the estimated posterior. Thus, SVI cannot estimate the intermediate posteriors in streaming settings (Broderick et al., 2013).

To solve this problem, Broderick et al. (2013) proposed "Streaming Variational Bayes" (SVB) for online Bayesian inference, which incrementally updates the posterior recursively using incoming data-points from an endless data-stream. Nevertheless, this technique requires tedious model specific derivations and has not been extended to efficient black-box inference. Moreover, as pointed out by McInerney et al. (2015), Bayesian updates on never-ending data lead to point mass posterior densities in almost all cases. Such overconfidence posteriors can be problematic due to two reasons. Firstly, such posteriors are contrary to our motivations to adopt Bayesian inference for online learning; to exploit the uncertainty of the models in online predictive analysis. Secondly, as evident by our analyzes in the Experiments section, overconfidence posteriors result in less responsive Bayesian updates to the changes in data. Therefore, the incremental updates to the posterior that is suggested by the traditional Bayesian framework cannot efficiently handle endless data streams with altering patterns.

McInerney et al. (2015) introduced "Population VI" (PVI) to avoid the overconfidence posteriors with infinite data streams. Their approach can be considered as a reformulation of the SVI to the streaming settings —introducing a new hyperparameter α the number of data points in the population posterior. PVI requires determining a suitable value for α following an appropriate hyperparameter optimization, whereas the original SVI is recovered by setting $\alpha = \mathrm{N}$. Unlike with SVI, α from PVI has no clear relationship with the dataset (McInerney et al., 2015), thus determining α introduces significant additional analyzes compared to rest of the techniques. Moreover, even for the same data stream, the optimal α can vary with the time. Conceptually, the α estimated during parameter optimization can expire after several drift points due to the changes to the population posterior eventually degrading the performance of PVI.

Consequently, existing approaches for online Bayesian inference are rather complex to be of any use to practitioners for real-world applications involving endless streaming data. The expertise and tedious effort required for model specific analysis, inability tackle concept drift due to overconfidence posteriors, and exhausting effort required to understanding and tuning additional hyperparameters have prevented the practitioners from adopting the existing online Bayesian inference approaches to the streaming settings.

We, therefore, propose a novel online variational inference framework —"Streaming Stochastic Variational Bayes" (SSVB) for never-ending streaming data with the following properties.

- SSVB is optimized as stochastic gradient descent, thus enabling online black-box inference for a wide array of models with little additional derivations.
- SSVB does not suffer from overconfidence posteriors, thus the posterior estimated through SSVB reflects the altering patterns in data.
- SSVB can adequately accommodate concepts drift in real-world streaming data without compromising the accuracy.
- SSVB controls the posterior variances considering both the amount of information observed at a given point and the changes to the posterior means.

- SSVB does not enforce any additional hyperparameters in contrast to its offline counterparts. SSVB eliminates the need for user-defined parameters to control the posterior variance.

The proposed technique can be directly applied by practitioners or researchers with endless streaming data in fabricating a wide range of online inference models. Moreover, SSVB provides an online inference framework that is as simple as its offline inference counterparts.

In this paper, we first introduce two modifications to the traditional Bayesian updating framework deriving a streaming Bayesian updating approach that is capable of handling data streams with concept drift. Following the proposed Bayesian updating approach, we then derive a novel black-box inference technique for online settings; "Streaming and Stochastic Variational Bayes" (SSVB).We evaluate the proposed approach against the black-box inference of SVI and PVI objectives, and BB-SVB for two essential models to the online learning; multinomial logistic regression and linear mixed effects models. We conduct an extensive analysis appraising the performance of SSVB against PVI, SVI, and BB-SVB for multinomial logistic regression using three multiclass classification datasets and two real-world data streams. SSVB achieves superior or comparable performance for online classification against the existing state-of-the-art; PVI. In addition, we outline an implementation of a linear mixed effects model with streaming data. In our experiments with a generated mixed-effect data stream, SSVB achieves comparable performance against PVI, avoiding the tedious effort demanded by PVI to tune additional parameters. Furthermore, we evaluate the accuracy gain of SSVB based multiclass classification against the widely adopted conventional online classification techniques such as AROW (Crammer et al., 2009), Passive-Aggressive (PA) classifiers (Crammer et al., 2006) and Stochastic Gradient Descent (SGD) classifier. We observe a significant accuracy gain for SSVB against the above conventional online classifiers.

The rest of the paper is organized as follows. In the later sections, we outline the streaming Bayesian updating and construct SSVB following the black-box inference of typical variational objective, respectively. The experiment results are discussed in the next section. Related work section elaborates the existing literature and the final section concludes this paper.

BAYESIAN UPDATING WITH STREAMING DATA

We now formulate concept drift in an online inference problem while emphasizing the inability of classical Bayesian updating to accommodate such drifts in streaming data. We then propose two modifications to the traditional Bayesian updating framework eliminating its drawbacks with streaming data that evolves over time.

Let us consider an independent and identically distributed (i.i.d.) dataset $\mathrm{x} = \{x_i\}_{i=1}^{\mathrm{N}}$ generated using unobserved D random variables $\mathrm{z} = \{z_i\}_{i=1}^{\mathrm{D}}$ following a conditional distribution $p(\mathrm{x}|\mathrm{z})$. The traditional inference tackles the problem of computing the conditional probability $p(\mathrm{z}|\mathrm{x})$ given a batch of data.

Traditional Bayesian Updating

In online settings, data is continuously arriving from various sources in batches or one-by-one. Assuming that data is generated i.i.d., the inference task can be extended to streaming data as estimating the conditional probability $p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1)$ given the first b batches of data $\mathrm{c}_1 \dots \mathrm{c}_b$ each having M data-points. Since we are dealing with i.i.d. data, this task is equivalent to incrementally learning randomly sampled mini-batches from a large dataset. Therefore, we can adopt traditional Bayesian updating to estimate the probability of $p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1)$ as below.

$$p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1) \propto \prod_{i=1}^{b} \big[p(\mathrm{c}_i|\mathrm{z})\big] p(\mathrm{z}) \tag{1}$$

Real-world streams sometimes evolve over time due to various external factors that dynamically change the underlying probability distributions of the random variables that generate the data. We call this phenomenon as *concept drift*. The occurrences of such drifts are unpredictable for most of the cases. Gama et al. (2014); Webb et al. (2015) provide a formal definition of concept drift between time t_0 and time t_1 as,

$$\exists\, x_i : p_{t_0}(x_i, \mathrm{z}) \neq p_{t_1}(x_i, \mathrm{z}) \tag{2}$$

where p_{t_0} and p_{t_1} represent the probability distributions at time t_0 and time t_1, respectively. Therefore, data-points from such streams may not be identically distributed or exchangeable. However, the traditional Bayesian updating framework illustrated in equation 1 is ill-suited for data that does not hold i.i.d assumptions.

Therefore, Bayesian updating lacks a built-in mechanism to handle concept drift (McInerney et al., 2015); it is intended for incrementally updating the posteriors of the random variables that generate data continuously, assuming that the underlying distributions of those random variables are fixed. Consequently, the posteriors tend to shrink with constantly arriving data, eventually resulting in overconfidence posteriors. Such posteriors undermine the ability to adapt to changes forcing themselves to remain unaffected disregarding any changes to the new data. Therefore, the Bayesian updating fails to recover from drifting patterns found with real-world data streams.

Streaming Bayesian Updating

Conventional Bayesian updating suggests that the uncertainty of the priors after each update should be the exact uncertainty of preceding posterior. However, such premises do not hold if the underlying probability distributions of the random variables are susceptible to changes over time due to various factors; then we cannot be as confident as the previous posteriors regarding the current state of the random variables. Therefore, we can employ a fixed uncertainty to our priors for each update if each batch has an equal probability of being subjected to concept drift.

Moreover, the uncertainty of the posteriors and priors are strongly correlated to their variances. Especially with unimodal priors, we can maintain the uncertainty unchanged by fixing the variance of the priors. However, we can expect the current posteriors to be around the preceding posterior unless a sudden drift has occurred. Furthermore, embedding the location of the previous posterior to the priors is important to detect and accommodate the changes to the current batch. Therefore, we only restrict the modifications to the variance of the priors, we still decide the expectation of the priors as prescribed by classical Bayesian updating.

On the contrary, by having a fixed uncertainty for the priors, the posteriors are unable to assess the amount of data used during the updates. Hence, the posteriors will fail to improve their confidence with continuously arriving data. Analogous to McInerney et al. (2015), we can control the posterior variance by scaling the likelihood of each batch. Therefore, we can embed the amount of information observed during the updates by scaling the likelihood of each batch with a suitable measure.

Consider a stream $\mathrm{c}_1 \dots \mathrm{c}_b$ after generating b batches for the case where underlying distributions of the random variables are susceptible to changes. Nevertheless, assume that no concept drift arises within the batches, thus preserving i.i.d. assumptions internally; all the changes to the underlying distributions are occurred in-between the batches. Since it is difficult to accurately anticipate the occurrence of such changes, we assume that each batch has an equal probability of being subjected to concept drift. Moreover, suppose that the underlying distributions drift slowly without any rapid changes. Let us denote such sequence of b batches using the notation $< \mathrm{c}_1 \dots c_b >$. Under the online inference tasks, we are interested in estimating the conditional probability of unobserved random variables $p(\mathrm{z}| < \mathrm{c}_1 \dots c_b >)$.

Therefore, we introduce two modifications to the conventional Bayesian updating presented in equation 1 to enable its ability to approximate $p(\mathrm{z}| < \mathrm{c}_1 \dots c_b >)$. First, we maintain a fixed variance for the priors during Bayesian updating permitting posteriors to adapt to the drifting patterns; we only transmit the information concerning the posterior expectations through the priors during the incremental updates. Secondly, we scale the likelihood of each batch as it is estimated using the total number of data-points employed during all the posterior updates including the current update.

Accordingly, we propose a streaming Bayesian updating framework that is capable of approximating the posterior $p(\mathrm{z}| < \mathrm{c}_1 \dots c_b >)$ after b batches as shown below.

$$p(\mathrm{z}| < \mathrm{c}_1 \dots c_b >) \propto \prod_{i=1}^{b} \left[p(\mathrm{c}_b|\mathrm{z})\right] p(\mathrm{z})^* \tag{3}$$

The priors $p(\mathrm{z})^*$ are resolved for b^{th} batch s.t.,

$$\mathrm{E}[\mathrm{z}] = \begin{cases} \mathrm{E}[\mathrm{z}| < \mathrm{c}_1 \ldots c_{b-1} >], & \text{if } b > 1 \\ \mu_0, & \text{otherwise} \end{cases}$$
$$\mathrm{Var}[\mathrm{z}] = \sigma_0^2, \ \forall\, b > 0 \tag{4}$$

where μ_0 and σ_0^2 are user-specified parameters typically based on their initial belief.

In equation 3, we have deliberately omitted the normalization term. Understating the normalization term is needless; because we derive the variational objectives independent of the intractable normalization term. The b^{th} update performed following the proposed Bayesian updating is equivalent to a Bayesian inference using $b \times M$ data-points similar to the current batch c_b whilst expecting posteriors to be closer to the expectation of posterior approximated using the previous batch c_b. Therefore, if no change has occurred in-between two adjacent batches, then the estimated posterior will be identical to a posterior estimated using traditional Bayesian framework presented in equation 1. Nevertheless, in the case of drifting patterns, the likelihood may suggest posteriors shifted from our beliefs that are embedded via priors $p(\mathrm{z})^*$. Such a disagreement between likelihood and priors will result in higher posterior variance, especially if we have assigned considerable variance σ_0^2 to the priors $p(\mathrm{z})^*$.

The specifications that are followed (in equation 4) to form the priors during Bayesian updating with streaming data may constrain the type of distributions that can be approximated as posteriors or employed as priors. The streaming Bayesian updating may require additional derivations to identify suitable priors, especially when encountered multimodal posterior densities. However, we consider such complex scenarios are beyond the scope of this work; we are mostly interested in understanding the behaviour of the streaming Bayesian updating as an approximation inference in online settings. Most of the distributions that we consider with approximation inference are unimodal distributions that allow modifying expectation and variance, individually. We will further discuss this concern after deriving the streaming variational objectives.

Accordingly, in stream settings we cannot assume identically distributed or exchangeable data due to dynamically evolving data; therefore traditional Bayesian updating framework fails to handle real-world data streams with concept drift. We tailor-made the proposed Bayesian updating framework to handle such streaming data while adapting drifting patterns more efficiently.

STREAMING STOCHASTIC VARIATIONAL BAYES

We now derive an online inference objective fusing Bayesian updating with the traditional variational objective. Such objective still lacks the ability to efficiently accommodate the changes with conventional Bayesian updating. Thus, we extend our initial objective with two amendments enabling the ability to adopt drifting patterns in data as suggested by the streaming Bayesian updating framework. Then we outline black-box inference for both streaming variation objective based on stochastic gradient updates formulating BB-SVB and SSVB frameworks for online inference. Furthermore, to conduct a fair comparison, we outline the black-box inference of both SVI and PVI objectives following an identical approach to the black-box inference of SSVB.

Variational Lower Bound

In variational inference, a family of distribution $q_\theta(.)$ that is parameterized by θ is specified over each unobserved random variable z. Then the exact posteriors densities $p(\mathrm{z}|\mathrm{x})$ for unobserved random variables are approximated to a distribution $q_\theta(\mathrm{z})$ from the selected family of distribution by determining θ that minimize the Kullback-Leibler (KL) divergence to the exact posterior $p(\mathrm{z}|\mathrm{x})$. The KL divergence between the approximated posterior $q_\theta(\mathrm{z})$ and the exact posterior $p(\mathrm{z}|\mathrm{x})$ can be expressed as;

$$\mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z}|\mathrm{x})] = \log p(\mathrm{x}) - \mathcal{L}(\theta;\mathrm{x}) \tag{5}$$

The $\mathcal{L}(\theta;\mathrm{x})$ term denotes the evidence lower bound (ELBO) which we will discuss shortly. The objective $\mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z}|\mathrm{x})]$ is non-negative and the log marginal likelihood $\log p(\mathrm{x})$ is fixed for a given x. Hence, the ELBO acts as a lower bound to the log marginal likelihood. Since the term $\log p(\mathrm{x})$ is not computable in most of the cases, the ELBO is maximized as a proxy to minimizing the KL divergence.

Therefore, the variational parameters θ that maximize the ELBO given data x, minimize the KL divergence between $q_\theta(\mathrm{z})$ and the exact posterior $p(\mathrm{z}|\mathrm{x})$. Accordingly, we maximize the ELBO shown below as the variational objective.

$$\mathcal{L}(\theta;\mathrm{x}) = \mathrm{E}[\log p(\mathrm{x}|\mathrm{z})] - \mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z})] \tag{6}$$

As illustrated in equation 6, maximizing the ELBO maximizes the likelihood of the observed data simultaneously forcing $q_\theta(\mathrm{z})$ to be closer to the prior distribution. In other words, maximizing the likelihood fits the model to data, whereas maximizing the negative $\mathrm{D}_{KL}[q_\theta(\mathrm{z})|p(\mathrm{z})]$ regularizes the estimated posteriors avoiding the overfitting to the data.

Streaming Variational Objective

In streaming settings, ELBO is to be optimized, once each batch c_b arrives. Suppose that the underlying distributions of the random variables z that generate the data x are fixed for duration being considered; therefore continuously generating i.i.d. data. Based the traditional Bayesian updates (and the proof in Appendix 1), we can consider approximated posterior $q_{\theta_{b-1}}(\mathrm{z})$ after observing $b-1$ batches as the prior when approximating the posterior $q_\theta(\mathrm{z})$ with the current batch.

Hence, the ELBO after observing b^{th} batch can be re-written as shown below;

$$\mathcal{L}(\theta;\mathrm{c}_b,\theta_{b-1}) = \mathrm{E}[\log p(\mathrm{c}_b|\mathrm{z})] - \mathrm{D}_{KL}[q_\theta(\mathrm{z})||q_{\theta_{b-1}}(\mathrm{z})] \tag{7}$$

It should be emphasized that the above objective is different from SVB (Broderick et al., 2013); SVB suggests recursively updating the offline approximation inference primitives that are derived using ELBO, whereas we have embedded such Bayesian updating to the ELBO allowing us to construct online probabilistic models directly. Therefore, we optimize the streaming variational objective in equation 7 as a single inference problem instead of decomposing each update to an offline inference task.

We will later construct BB-SVB for black-box online inference based on Bayesian updating following the objective illustrated in equation 7.

Streaming Variational Objective with Drift Adaptation

As discussed earlier, traditional Bayesian updating collapses with drifting patterns in streaming data, thus the streaming variational objective illustrated in equation 7 cannot handle data generated using the random variables with evolving underlying distributions. We now derive a truly streaming variational objective based on the proposed Bayesian updating framework in equation 3.

Accordingly, considering the proposed Bayesian updating (and the proof in Appendix 1) the improved streaming variational objective can be formulated as;

$$\mathcal{L}(\theta;\mathrm{c}_b,b) = b \times \mathrm{E}[\log p(\mathrm{c}_b|\mathrm{z})] - \mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z})^*] \tag{8}$$

We need to express the priors $p(\mathrm{z})^*$ in terms of an appropriate known family of distribution. The ideal selection of priors allows us to scale the posterior distributions to the desired variance without altering the shape the posterior (e.g. Gaussian distributions are an ideal candidate to represent the priors $p(\mathrm{z})^*$ regarding Gaussian posteriors). Let us consider a family of distribution $\hat{q}_{(\mu,\sigma^2)}(.)$ that is parameterized by the expected value μ and the variance σ^2. Suppose $\hat{q}_{(\mu_{b-1},\sigma_0^2)}(\mathrm{z})$ as the priors $p(\mathrm{z})^*$ for streaming Bayesian updating after observing (b-1)$^{\text{th}}$ batch, where μ_{b-1} and σ_0^2 are respectively the expectation of the preceding posteriors and the initial variance as suggested by equation 4.

Additionally, we employ a scaling function S_b instead of the number of batches b to scale the likelihood term in the variational objective. We define S_b s.t.,

$$\mathrm{S}_b = \frac{n_b}{M \times \phi} = \frac{b}{\phi}, \quad \text{s.t. } \phi > 0 \tag{9}$$

where the n_b is the total number of data-points used during all the updates including the current update and ϕ is a normalization constant, which is useful to adjust regularization to avoid overfitting. Recall that

M is the batch size. The purpose of introducing this scaling function is to control the regularization to the posteriors accordingly. However, in our experiments, we have always considered $\phi = 1$ unless specified otherwise. Therefore, with default settings the $S_b = b$ as recommended by the streaming Bayesian updating.

Accordingly, the improved variational objective can be re-written as;

$$\mathcal{L}(\theta; \mathrm{c}_b, \mathrm{S}_b, \mu_{b-1}) = \mathrm{S}_b \times \mathrm{E}[\log p(\mathrm{c}_b|\mathrm{z})] - \mathrm{D}_{KL}[q_\theta(\mathrm{z})||\hat{q}_{(\mu_{b-1}, \sigma_0^2)}(\mathrm{z})] \tag{10}$$

Therefore, the proposed streaming variational objective (eq. 10) scales the likelihood proportionally to the total number of data-points used to update the model until the b^{th} batch inclusively. McInerney et al. (2015) control the posterior variance by scaling the likelihood term relative to the KL-divergence term in the variational objective. Their findings further justify our streaming variational objective; scaling the likelihood term with S_b controls the variance of the posterior as it is updated using n_b data-points. However, unlike the scale employed by Hoffman et al. (2013) (SVI) and McInerney et al. (2015) (PVI), S_b is not a constant; it is updated with each batch based amount of new data observed by the model. An additional benefit of employing such dynamic scaling is that by resetting the S_b (e.g. setting $\mathrm{S}_b = 1$) we can refresh the posteriors by forgetting irrelevant information. Such resetting can be also useful for an occasional re-calibration of the posterior uncertainty.

As discussed earlier, one downside of employing the proposed objective compared to tradition Bayesian updates is identifying a suitable distribution for the priors that allows modifying the mean and std, separately. Even though most of the distributions are not explicitly parameterized as mean and std, we can still obtain the distributions having any given mean and std > 0. As an example, we can easily obtain the Gamma prior with given mean and std by defining the shape and rate parameters of the Gamma distribution in terms of the mean and std. Alternatively, we could handle such constraints by using appropriate distributions for the priors that allow explicit parameterization of mean and std. We identify Gaussian priors as a suitable candidate for most of the cases irrespective the family of the posteriors.

Accordingly, we have derived an improved streaming variational objective by fusing the streaming Bayesian Updating with the variational objective. The proposed objective allows online Bayesian inference while accommodating drifting patterns in data more effectively compared the initial streaming variational objective. Moreover, the obtained objective can be justified using the existing state-of-the-art variational objective adopted to streaming settings. In the next section, we will outline SSVB for black-box online inference following the improved streaming variational objective presented in equation 10.

Black-Box Inference of Streaming Variational Objectives

The recent approaches to the black-box inference of the variational objective are mostly performed by optimizing the variational objectives collectively using Monte-Carlo gradient estimators and stochastic gradient descent (Ranganath et al., 2014; Zhang et al., 2018; Rezende et al., 2014). Thus, we adopt those strategies to conduct black-box inference of the streaming variational objectives. We will discuss the implementation of black-box inference of the proposed objective as a gradient descent optimization.

Let us first derive a streaming variational gradient estimator by differentiating the streaming variational objectives in equations 7 and 10 w.r.t. variational parameters θ_b. The acquired streaming variational gradient estimator after observing b^{th} mini-batch is as follows.

$$\nabla_\theta \mathcal{L}(\theta; \mathrm{c}_b, \theta_{b-1}) = \nabla_\theta \mathrm{E}[\log p(\mathrm{c}_b|\mathrm{z})] - \nabla_\theta \mathrm{D}_{KL}[q_\theta(\mathrm{z})||q_{\theta_{b-1}}(\mathrm{z})] \tag{11}$$

$$\nabla_\theta \mathcal{L}(\theta; \mathrm{c}_b, \mathrm{S}_b, \mu_{b-1}) = \mathrm{S}_b \times \nabla_\theta \mathrm{E}[\log p(\mathrm{c}_b|\mathrm{z})] - \nabla_\theta \mathrm{D}_{KL}[q_\theta(\mathrm{z})||\hat{q}_{(\mu_{b-1}, \sigma_0^2)}(\mathrm{z})] \tag{12}$$

Since θ_{b-1} and μ_{b-1} are determined based on preceding posterior, only θ_b is considered as the variational parameters to be optimized. Hence, we have further simplified the notation in the equation 12 by replacing the variational parameter θ_b with θ.

Computing the Gradients

The generic Monte Carlo gradient estimator is typically used to compute the gradients of the variational objective (Paisley et al., 2012; Ranganath et al., 2014). Nevertheless, the gradient estimated using Monte

Algorithm 1: Black-Box Streaming Variational Bayes - BB-SVB

Inputs : $c_1 \dots c_b$, θ_0
Initialize : θ
foreach $c_i \in c_1 \dots c_b$ **do**
  $\bar{\theta} \leftarrow \theta_{i-1}$
  $g \leftarrow \nabla_\theta \mathcal{L}(\theta; c_i, \bar{\theta})$ (Eq. 11)
  $\theta_i \leftarrow$ Update parameters using gradients g (Eq. 14 with ADAM)
end
return θ

Algorithm 2: Streaming Stochastic Variational Bayes - SSVB

Inputs : $c_1 \dots c_b$, μ_0, σ_0^2, M
Initialize : θ
foreach $c_i \in c_1 \dots c_b$ **do**
  $\bar{\mu} \leftarrow \mu_{i-1}$
  $n_i \leftarrow n_{i-1} + M$
  $S_i \leftarrow n_i / M$
  $g \leftarrow \nabla_\theta \mathcal{L}(\theta; c_i, S_i, \bar{\mu})$ (Eq. 12)
  $\theta_i \leftarrow$ Update parameters using gradients g (Eq. 14 with ADAM)
end
return θ

Carlo gradient estimator usually exhibits a very high variance (Paisley et al., 2012). The reparameterization trick is shown to be useful to obtain a differential estimator of the variational lower bound with less variance than the generic estimator by Kingma and Welling (2013). Furthermore, the recent work by Figurnov et al. (2018) proposes *implicit reparameterization gradients*, which extends reparameterization trick to most of the commonly used families of distributions such as Gamma and Dirichlet etc. Hence, we adopt the "reparameterization gradient VI" (Zhang et al., 2018; Gal, 2016) to optimize each variational objective described above.

We express each random variable z as deterministic variable $z = h(\theta, \varepsilon)$, where ε is an auxiliary variable with independent marginal $\varepsilon \sim p(\varepsilon)$. We compute the gradients for both BB-SVB and SSVB by applying the reparameterization trick to the gradient estimators illustrated in equations 11 and 12, respectively. Nevertheless, the KL divergence $D_{KL}[q_\theta(z)||p(z)]$ often can be integrated analytically (Kingma and Welling, 2013), such that only the likelihood term requires sampling. In such cases, only the first RHS terms of the gradient estimators are computed based on reparameterization trick.

Gradient Descent Steps

In the process of stochastic gradient descent, the objective is differentiated w.r.t each variable and the gradient of each variable is evaluated at the current point.

$$g(\theta) = \nabla_\theta \mathcal{L}(\theta; \dots) \tag{13}$$

$$\theta_t = \theta_{t-1} - \mathcal{F}(\rho, g(\theta_{t-1})) \tag{14}$$

Equation 13 represents the gradients computed using b^{th} batch for a given variational objective. Hence, $g(\theta_{t-1})$ in equation 13 denotes the gradient of the objective evaluated at the current point θ_{t-1}. Equations 13 and 14 are followed during each pass t to take a single gradient step. Each update may consist of several passes (or iterations). The stochastic gradient optimizer decides the operations that are performed by $\mathcal{F}(.)$, here ρ is known as the step size or the learning rate. Since we transform all the random variables to deterministic variables via reparameterization trick, the optimization shown in equation 14 can be performed in conjunction with any stochastic gradient optimizer such as Adagrad (Duchi et al., 2011) or ADAM (Kingma and Ba, 2014).

Algorithms 1 and 2 respectively illustrate BB-SVB and SSVB that is obtained by performing black-box inference on the initial and improved variational objectives.

Algorithm 3: Black-Box Variational Inference - VI

Inputs : x, p(z)
Initialize : θ
repeat
  $g \leftarrow \nabla_\theta \mathcal{L}(\theta; \mathrm{x})$ (VGE Eq. 15)
  $\theta \leftarrow$ Update parameters using gradients g (Eq. 14 with ADAM)
until θ *converges*;
return θ

Algorithm 4: Black-Box Stochastic Variational Inference - SVI

Inputs : $\mathrm{c}_1 \ldots \mathrm{c}_b$, p(z), N, M
Initialize : θ
foreach $c_i \in c_1 \ldots c_b$ **do**
  $g \leftarrow \nabla_\theta \mathcal{L}(\theta; \mathrm{c}_i, \mathrm{N}, \mathrm{M})$ (SVGE Eq. 16)
  $\theta \leftarrow$ Update parameters using gradients g (Eq. 14 with ADAM))
end
return θ

Single Pass Updates

A typical online learning algorithm learns from each data point exactly once, which is known as single pass online learning. Assuming that the parameters are updated strictly once per each mini-batch based on equation 14, we can expect any variational parameters θ_{b-1} to be equivalent θ_{t-1} for $b \geq 2$. For $b = 1$ the variational parameters θ_0 should be initialized through the hyperparameters to the model.

When such a relationship holds, the objective of BB-SVB exhibits some special characteristics. The KL divergence term $\mathrm{D}_{KL}[q_\theta(\mathrm{z})||q_{\theta_{b-1}}(\mathrm{z})]$ from equation 7 becomes zero once evaluated at current point. As a result, the KL divergence of most of the commonly adopted families of distributions (e.g. Normal and Gamma etc) has zero gradients during single pass updates with BB-SVB.

One can assume that using BB-SVB with single pass updates completely eliminates the effect of the priors on the model for $b \leq 2$. However, such behaviour is not an intentional elimination of the effect of the prior instead it is exactly the effect of the prior on the model as a result of the streaming and stochastic nature of the BB-SVB. Alternatively, we could consider that single-pass updates with BB-SVB to be equivalent to maximizing the likelihood of some model variables that are defined using a mean and variance.

Black-Box Inference of VI, SVI and PVI

To conduct a fair evaluation, we derive black-box inference for VI, SVI (Hoffman et al., 2013) and PVI (McInerney et al., 2015) objectives following same approach employed by SSVB and BB-SVB.

A variational gradient estimator (VGE) can be constructed by differentiating the ELBO w.r.t. to the variational parameters θ (Hoffman et al., 2013; Ranganath et al., 2014; Kingma and Welling, 2013; Paisley et al., 2012) as shown below.

$$\nabla_\theta \mathcal{L}(\theta; \mathrm{x}) = \nabla_\theta \mathrm{E}[\log p(\mathrm{x}|\mathrm{z})] - \nabla_\theta \mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z})] \tag{15}$$

The VGE in equation 15 uses the full dataset to evaluate the gradient in a single iteration. The usual approach to construct a stochastic variational gradient estimator (SVGE) for randomly sampled mini-batches from a dataset with N data-points requires scaling the likelihood term by $\frac{\mathrm{N}}{\mathrm{M}}$ (Hoffman et al., 2013; Kucukelbir et al., 2017). Thus, the likelihood is scaled to as it is computed using the full dataset suppressing the overwhelming priors or in this case the overwhelming KL divergence term. We obtain SVGE for mini-batches randomly sampled from the full dataset as follows.

$$\nabla_\theta \mathcal{L}(\theta; \mathrm{x}) \simeq \nabla_\theta \mathcal{L}(\theta; \mathrm{c}_b, N, M) = \frac{N}{M} \nabla_\theta \mathrm{E}[\log p(\mathrm{c}_b|\mathrm{z})] - \nabla_\theta \mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z})] \tag{16}$$

We can optimize the VGE and SVGE following *reparameterization VI* to construct the black-box inference for VI and SVI. Accordingly, algorithms 3 and 4 respectively present the black-box VI and black-box SVI.

Conceptually, SVI cannot approximate the intermediate posterior densities, SVI rather estimates the posterior for the full dataset with N data points. Nevertheless, McInerney et al. (2015) introduce PVI justifying the use of SVI with streaming data by interpreting N as an additional parameter α that controls the posterior variance. The α is configured by the practitioners s.t. PVI reaches the optimum. Therefore, henceforth PVI denotes the instance that uses optimal α instead of the size of the dataset N with the SVGE illustrated by equation 16.

Recall that KL divergence term serves as the regularization to the posteriors thus, we can interpret that the role of α is to control the regularization to the posteriors. Therefore, in addition to controlling the posterior variance, α also adjust the regularization to the posterior mean; and estimating the optimal α correspond to finding ideal regularization to posteriors.

We have obtained black-box counterparts of VI, SVI and PVI following their original objectives in this section. We will be using them throughout our experiments in contrast with SSVB and BB-SVB.

DISCUSSION

In this section, we provide empirical evidence to establish the superiority of SSVB against the existing online inference techniques such as PVI (McInerney et al., 2015), SVB (Broderick et al., 2013), SVI (Hoffman et al., 2013) and lastly BB-SVB, which we derived. Our analysis includes three experiments. First, we will demonstrate the deficiencies in extending the SVB to the black-box VI techniques. Based on our observations during the first experiments, we justify omitting SVB from further analysis against SSVB. Then we discuss the properties of the SSVB and BB-SVB using two supervised non-conjugate probabilistic models; multinomial logistic regression and linear mixed effect model. We select each model considering their importance as an online inference approach. As the second experiment, we conduct an extensive evaluation of the performance of SSVB compared to PVI, SVI and BB-SVB using multinomial logistic regression. The second experiment is consist of four phases. In the first phase, we use three diverse multiclass-classification datasets to evaluate the ability to learn non-drifting patterns. We extend these experiments to the second phase by adopting two real-world streaming datasets that have drifting patterns. Apart from the performance of SSVB, we also analyze the posteriors estimated by each technique to understand the behaviour of the posteriors under drifting patterns as the third phase of the experiment. Then we analyze the performance of SSVB against conventional online classifiers as the last phase of the second experiment. As the final experiment, we further investigate the performance of SSVB against PVI, SVI, and BB-SVB based on a different and more complex task; linear mixed effect regression. Using a generated data-stream with random drift-points, we attempt to generalize the competitive accuracy observed with SSVB with the previous experiment to a wide-range of probabilistic inference tasks.

Experiment 1 - SVB with Black-box Inference

We now demonstrate the deficiencies in extending the SVB (Broderick et al., 2013) to the black-box VI techniques. As seen in the literature, the black-box inference techniques are mostly motivated by the ability to perform gradient descent updates on the variational objective (Kucukelbir et al., 2017; Ranganath et al., 2014; Kingma and Welling, 2013; Zhang et al., 2018). Therefore, we use black-box VI presented in algorithm 3 as the offline approximation primitive of SVB. We analyze the estimated posteriors using SVB against the posterior approximated by SVI (algorithm 4) for a simple logistic regression task. We perform single-pass updates on each data points from 1e3 generated data-points with five regressors. Figure 1 illustrates the approximated posteriors for the five regression coefficients after each two hundred data points.

The posteriors estimated using SVB are either failing to converge to the true coefficients or suffering from a high variance when using BB-VI as the approximation inference primitive. This is mainly due to the properties of the steepest descent; for each mini-batch, it initiates the stochastic search from a new random point, which results in a much slower and poor convergence for SVB framework. Since SVB is not extendable as an efficient black-box inference alternative, we do not consider SVB in our future analysis. We consider PVI and SVI as the existing state-of-the-art to perform black-box inference with data streams. Moreover, we consider the BB-SVB as the black-box inference equivalent of SVB in the following experiments.

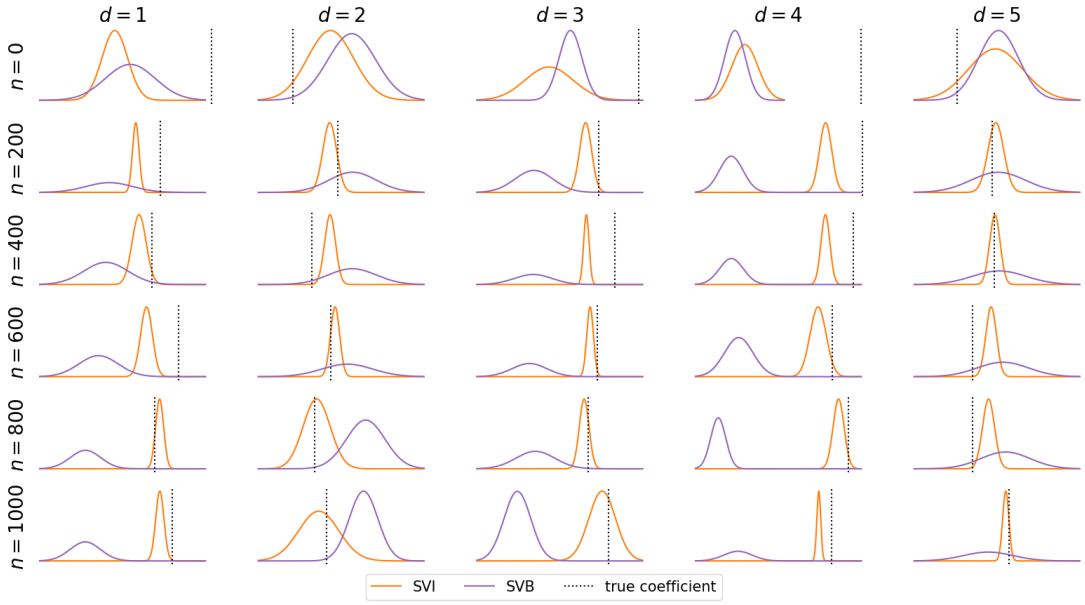

Figure 1. Convergence of posteriors estimated using traditional SVB with black-box inference primitives and SVI

Dataset	#samples	#features	#classes
20News	11314	100000	20
MNIST	60000	785	10
Otto Products	61878	95	9
Airline	5810462	13	2
Poker	829201	11	10

Table 1. Summery of datasets

Experiment 2 - Linear Classification

Classification is a necessary tool in stream analytic. Especially, multinomial logistic regression one of the simplest yet need approximate inference. Therefore, we investigate the applicability of the proposed objectives for online classification. First, we define the multinomial logistic regression using the probabilistic notations as shown in Appendix 2. Once we construct the probabilistic model we optimize each objective following reparameterization VI while sampling only once to compute the gradients. We employ standard Gaussian distribution as the priors to the models (these will be initial priors to the SSVB and BB-SVB).

Phase 1 - Classification with Standard Multiclass Datasets

First, we analyze the performance of the classification models using three standard multiclass datasets. One of which (*20News*[1]) is a text classification task with high dimensional sparse features and the other two (*MNIST*[2] and *Otto product*[3]) are respectively image and general classification tasks. These datasets are selected considering their diversity in the properties such as number of dimensions, type of features (spares vs dense, continues vs discrete) and the performed task. We have tabulated the properties of each dataset in table 1.

All the objectives are updated using sequential data that are arriving one-by-one ($M = 1$) in order to simulate the standard streaming settings. We use ADAM optimizer with the learning rate ρ of 0.01 for all the datasets except for *20News* dataset, where we set ρ to 0.05. PVI demands to configure an additional

[1] http://qwone.com/ jason/20Newsgroups/
[2] http://yann.lecun.com/exdb/mnist/
[3] https://www.kaggle.com/c/otto-group-product-classification-challenge

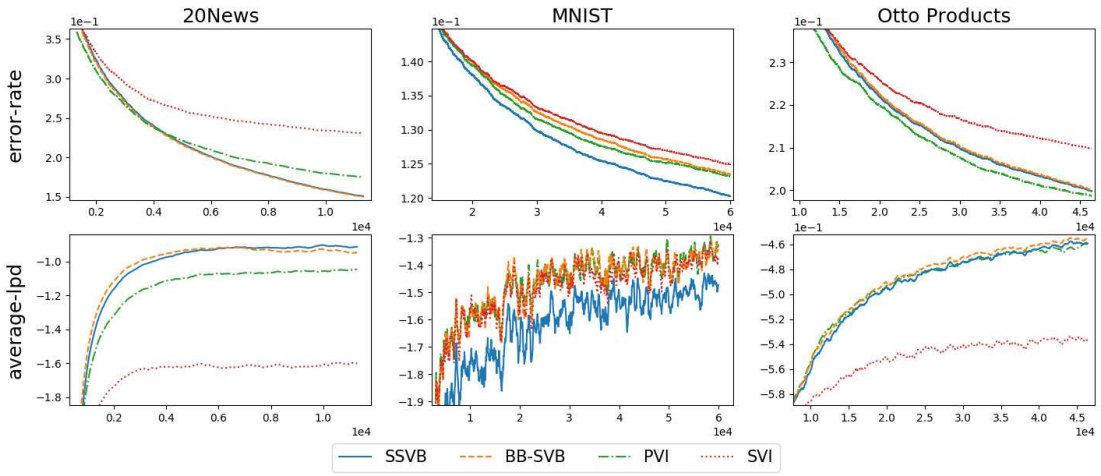

Figure 2. Error rate and average log-predictive density for multiclass classification

	20News	MNIST	Otto Products
SSVB	$0.1509 \pm 0.0019^{\ddagger}$	$0.1202 \pm 0.0011^{\dagger}$	$0.1998 \pm 0.0012^{\ddagger}$
BB-SVB	$0.1502 \pm 0.0018^{\dagger}$	$0.1234 \pm 0.0006^{\#}$	$0.2002 \pm 0.0010^{\#}$
PVI	$0.1750 \pm 0.0008^{\#}$	$0.1231 \pm 0.0012^{\ddagger}$	$0.1987 \pm 0.0006^{\dagger}$
SVI	0.2308 ± 0.0010	0.1249 ± 0.0012	0.2098 ± 0.0012
AROW	-	0.1383 ± 0.0034	0.2102 ± 0.0023
PA	0.2741 ± 0.0027	0.1506 ± 0.0007	0.2040 ± 0.0013
SGD	0.3106 ± 0.0010	0.1480 ± 0.0008	0.2057 ± 0.0009

Table 2. Means and stds of classification error rates for multiclass classification[4]

parameter α, we used the first 10% of full dataset to find a suitable value for α minimizing the error rate. The optimal values found for α are 1e-5, 1e-6 and 1e-6 for *20News*, *MNIST* and *Otto Products*, respectively. We use both the average log-predictive density (lpd) aka average log-likelihood and error rate to evaluate the fit of the models. For both *20News* and *MNIST*, we compute the lpd considering the standard test split as the holdout set, whereas a random split with 25% of the dataset is treated as the holdout dataset for *Otto Products*. Moreover, the error rate is computed using equation 17 following the standard prequential evaluation, where each observation is first used to test the model and then used to train the model. These datasets are not specifically designed for streaming settings, thus the ordering of the data may affect the fairness of the experiments. Therefore, we run the experiment 5 times for each dataset with different random permutations of the data to conduct a fair comparison. Table 2 presents the mean and the standard deviations of the final error rates for those 5 experiments, and figure 2 illustrates the convergence of the error rate and average lpd w.r.t the number of samples observed.

$$error\ rate = \frac{number\ of\ incorrect\ predictions}{total\ number\ of\ predicted\ instances} \tag{17}$$

Let us first consider the final error rates for each approach in table 2. SSVB and BB-SVB achieve significantly higher accuracy compared to PVI and SVI with *20News* dataset. Even though SSVB gains the lowest error rate with *MNIST* surpassing PVI, the PVI marginally outperforms both SSVB and BB-SVI with *Otto Products* dataset. However, when we consider both mean and standard deviation of the error rates, the difference between the error rates of SSVB and PVI with *Otto Products* dataset is not statistically significant, Thus, we can establish that SSVB achieves the best overall accuracy with standard classification datasets as opposed to BB-SVB, PVI and SVI. On the other hand, SVI exhibits the worst

[4]notation †, ‡ and # denote the best three approaches based on mean error out of all the techniques

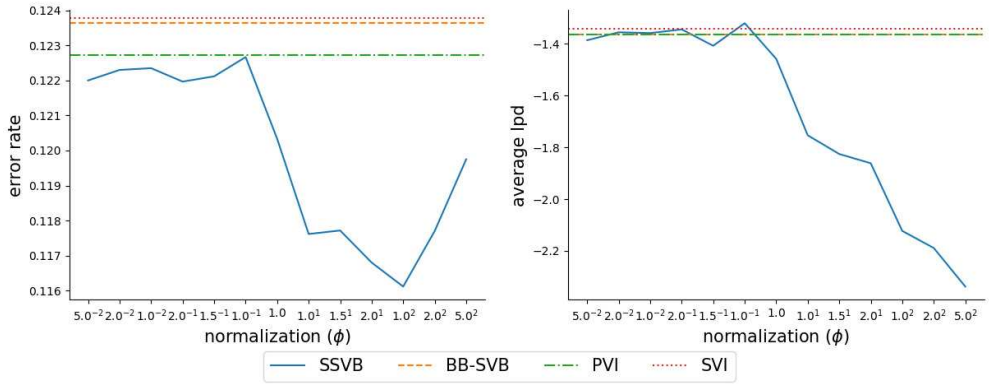

Figure 3. Error rate and average-lpd with different normalization factors for S_b

performance for all three datasets. Notice that the final f1 scores presented in Appendix 4 also support our conclusions.

Moreover, according to figure 2, we observe that lpd to be corresponding with the error rates for *20News* and *Otto Products* datasets. Surprisingly, lpd indicates a poor fit for SSVB with *MNIST* dataset though SSVB has notably outperformed the other techniques in terms of the error rate for the same dataset. Moreover, all the approaches undergo frequent fluctuations in log-predictive density with *MNIST*, which may an indication of sudden changes due to noisy labels. SSVB seems to overcompensate its posterior uncertainty considering such noisy behaviours as drifting patterns leading to poor log-likelihood. Even though such behaviours do not affect the overall accuracy of SSVB if needed, we can mitigate such shortcomings of SSVB by fine-tuning the normalization ϕ of the scaling function S_b.

To understand the effect of normalizing the scaling function, we analyze SSVB by setting different values for the normalization ϕ with *MNIST* dataset. Figure 3 presents the final error-rate and the average log predictive densities w.r.t the different ϕ employed by S_b during our analysis. The horizontal lines are corresponding to the final average log predictive densities for the rest of the approaches. Since the average log-predictive densities exhibit sudden fluctuations with *MNIST* dataset as already seen with the figure 2, we have considered the mean of average log predictive density measure during last 10 updates to conduct a much accurate comparison.

Interestingly, SSVB has outperformed the rest of the techniques for each normalization applied to the scaling function in terms of the error rate. Especially, for $\phi > 1$ SSVB achieves significantly lower error rate compared to the other inference approaches. However, the average log-predictive density of SSVB is considerably poor than that of the PVI, SVI and BB-SVB for those cases. For the rest of the cases, SSVB exhibits improved or comparable average log-predictive density against PVI. Hence, ϕ governs the trade-off of optimizing the error rate and the log-predicting density. Since the scaling function S_b controls the regularization to the posteriors, using different values for ϕ to alter the amount of regularization. Adequate regularization is important to maintain sufficient robustness to handle sudden changes due to noisy labels (Crammer et al., 2009), thus setting ϕ to greater than 1.0 increases the regularization to the posterior means than the usual scaling function S_b resulting in higher accuracy. On the other hand, increasing ϕ also enhances regularization to the posterior variance, thus forcing posteriors to overestimate their uncertainty misinterpreting noisy labels as sudden drifts.

Accordingly, SSVB achieves the overall best performance with data streams that are not subjected to concepts drifts. Even though PVI also achieves comparable accuracy against SSVB for most of the cases, the additional effort required to tune α has made redundant with SSVB. However, SSVB can be further improved by tuning the normalization term ϕ of the scaling function S_b to better handle the noisy streams trading log-predictive density for better accuracy, and vice versa.

Phase 2 - Classification with Real-World Data Streams

We extend our experiments with two massive real-world data streams; *airline* [5] and *poker-hand* [6] datasets. Unlike the three datasets considered in the previous section, *airline* and *poker-hand* datasets are extracted

[5] https://kt.ijs.si/elena_ikonomovska/data.html
[6] https://archive.ics.uci.edu/ml/datasets/Poker+Hand

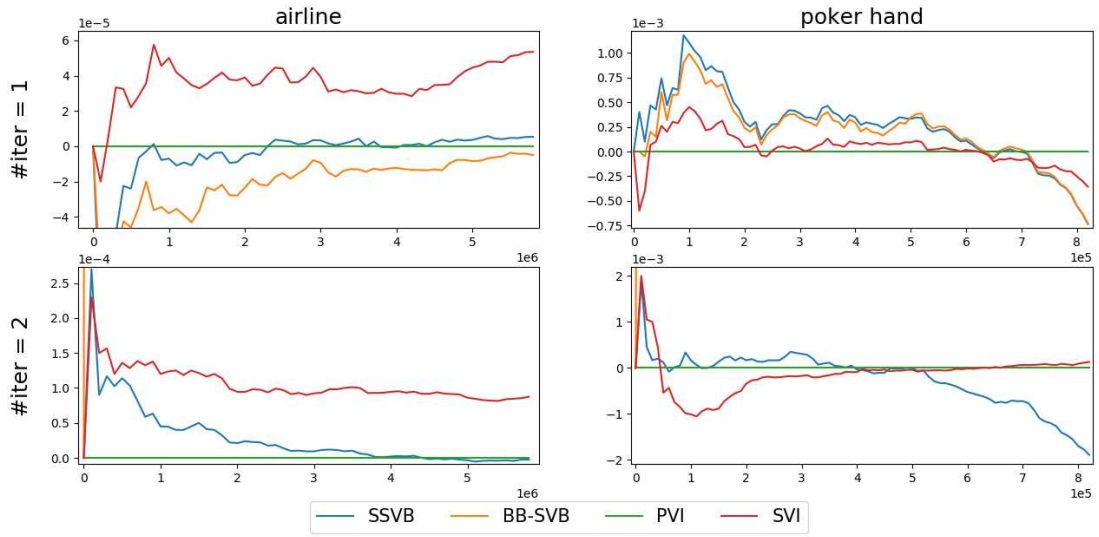

Figure 4. Classification error rate considering PVI as the ground accuracy

	#iterations = 1		#iterations = 2	
	airline	**poker**	**airline**	**poker**
SSVB	0.307257$^{\#}$	0.275782$^{\#}$	0.310322†	0.277216‡
BB-SVB	0.307246†	0.275763‡	0.413883	0.460659
PVI	0.307251‡	0.276675	0.310325‡	0.279263$^{\#}$
SVI	0.307306	0.276221	0.310412$^{\#}$	0.279427
AROW	0.333015	0.429212	0.332917	0.435478
PA	0.376963	0.224140†	0.376963	0.224140†
SGD	0.370204	0.269822	0.370204	0.269822

Table 3. Classification error rates with drifting patterns

from real-world streams with concept drift, thus those datasets present more realistic challenges to the model in testing their online classification ability. The properties of these data streams are also included in table 1.

Analogous to the previous analysis, we feed exactly one data point for each update. However, we investigate both single-pass and multi-pass updates. For the multi-pass scenarios, we perform exactly two passes per each update. We use ADAM optimizer with $\rho = 0.01$ for both datasets. Similar to the previous section, we optimize α using the initial 10% of the complete data stream minimizing the error rate. The optimal α found for *airline* and *poker-hand* datasets are respectively 1e8 and 1e5 with single-pass updates, whereas multi-pass updates required setting α to 1e9 and 1e7 to achieve the optimal settings. We preserve the original ordering of the data and conduct prequential evaluations to compute the error rates shown in equation 17. Table 3 presents the final error rates observed. The '#iterations' in table 3 indicates the number of updates performed using each data-points (i.e. single-pass vs multi-pass updates). Moreover, figure 4 illustrates the convergence of the error rates for SSVB, BB-SVB and SVI considering PVI as the ground accuracy (i.e. we compute the difference the between error rates for each technique and PVI) w.r.t the number of data samples used to update the models. We have excluded BB-SVB from the plots corresponding to multi-pass updates because the error rate of BB-SVB drastically increases concealing the variations among the rest of the techniques.

If we consider only the final error rates with single-pass updates illustrated in table 3, we do not observe a considerable difference in the accuracies of SSVB and BB-SVB compared to PVI for *airlines* dataset. However, SSVB and BB-SVB have shown a moderate improvement over PVI and SVI with *poker-hand* dataset. We could expect BB-SVB to perform poorly under the concept drift due to the overconfidence posteriors nevertheless, BB-SVB has achieved the best overall accuracy. It should be noticed that under

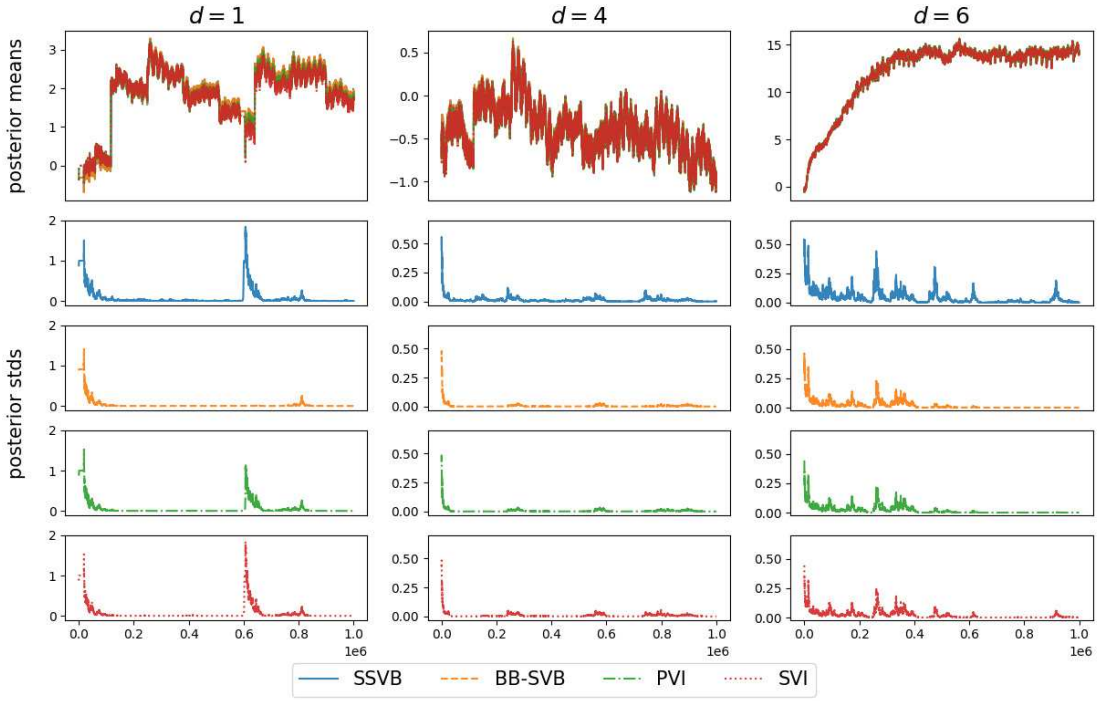

Figure 5. Mean and Std estimated by each approach with first 1e6 data samples from *airline* dataset under single-pass updates

single-pass updates BB-SVB completely ignores the KL-divergence term in the variational objective, thus diminishing the resistance to the changes due to overconfidence posteriors. Therefore, BB-SVB obtains a higher accuracy with single-pass updates by acting as likelihood maximization of the random variables.

One can argue that single-pass updates are insufficient to estimate the intermediate posteriors during Bayesian updating with SSVB and BB-SVB, which could ultimately lead to poor convergence. Nevertheless, the experiment results shown in table 3 prove otherwise. The multi-pass updates have caused a considerable reduction in accuracy contrary to single-pass updates for all the tested scenarios. For most of the cases, this is due to the overfitting which is a phenomenon that could affect any machine learning technique. Furthermore, we observe a substantial drop in the accuracy of SSVB under multi-pass updates though BB-SVB has attained the lowest error for both datasets with single-pass updates. Such poor performance is mainly due to the overconfidence priors, which restrains BB-SVB from accommodating the drifting patterns in the data. On the other hand, SSVB is not affected by overconfidence posteriors even with multi-pass updates instead, SSVB outperforms other approaches for both datasets. Moreover, SSVB does not require optimizing α or knowing the size of the data stream.

Figure 3 reveals an interesting behaviour when analyzing the convergence of SSVB relative to that of PVI. For most of the cases, initially, PVI outperforms SSVB. However, SSVB gradually recovers this accuracy gap with more and more data observed outperforming PVI in the long run. We observe similar behaviour in figure 2 when considering both error rate and average log-predictive density. Irrespective of the initial accuracy, SSVB demonstrates much faster convergence compared to PVI for most of the cases. Moreover, BB-SVB with single-pass updates also resembles the above behaviour when compared with PVI. We can explain such conduct using the different scaling mechanisms employed by each technique to govern the regularization to the posteriors.

It should be emphasized that different scaling mechanism influence the regularization of posterior mean differently, presumably resulting in considerably diverse posterior means after certain drift points for each approach (see Appendix 5). Proper regularization is essential in the online settings to prevent overfitting of the model parameters, thus helping them to recover when a change occurs (Kivinen et al., 2004). Moreover, amply regularizing the posterior variance is essential to avoid overconfidence posteriors (McInerney et al., 2015) with endless data streams. Since PVI uses first 10% of the data stream to find the optimal scale α to adjust its regularization, we can expect PVI to yield higher initial performance

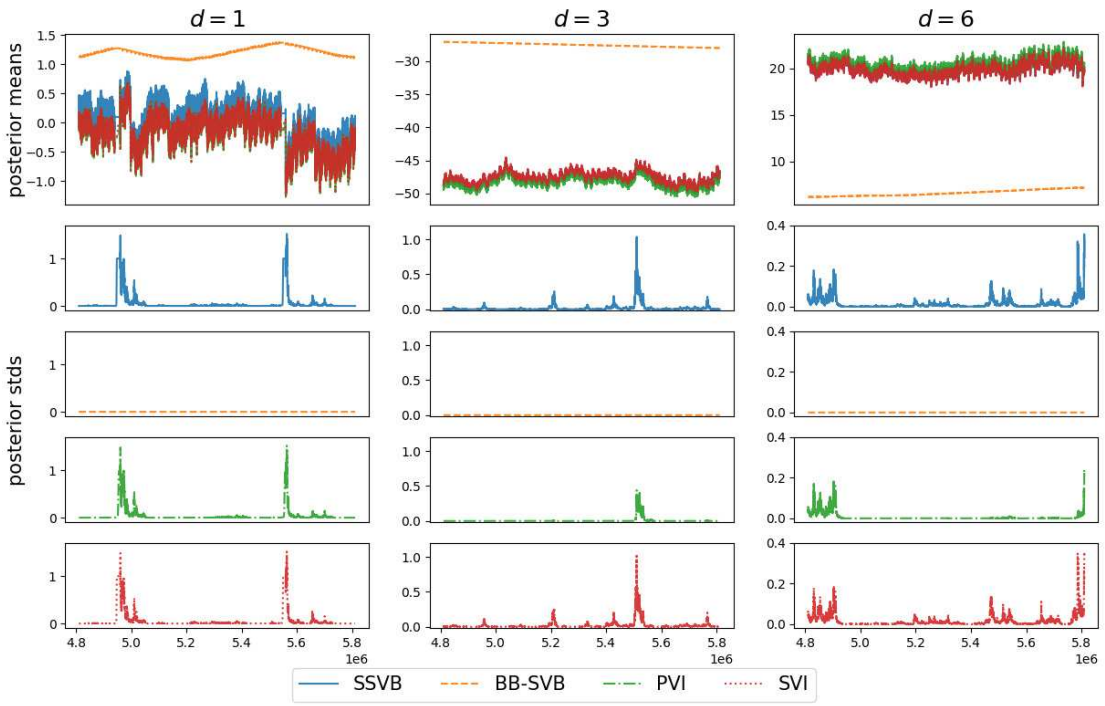

Figure 6. Mean and Std estimated by each approach with last 1e6 data samples from *airline* dataset under multi-pass updates

compared to the technique such as SSVB that does not exploit such optimization. However, the optimal α may expire eventually once the α becomes inadequate to scale the likelihood sufficiently moderating the excess effect of the KL divergence term. Unlike PVI, SSVB dynamically improves its regularization capabilities based on the partially updated priors and the scaling function S_b, thus outperforming the PVI in the long-run.

Therefore, SSVB appears to be more suited with endless data stream due to its comparable performance to PVI, even without tuning any additional hyperparameters as with PVI. Although PVI is not sensitive to the size of the entire dataset, the α being estimated is sensitive to properties of the data-points (e.g. the number of data points, drifting patterns etc) that are used optimize α. Therefore, PVI may require re-estimating the α after a while to avoid any accuracy drop due to the outdated α. Since. SSVB adjust its scaling function dynamically based on the number of data-points observed, it is highly unlikely to expire. Therefore, SSVB is much useful to handle never-ending drifting data streams than PVI.

Phase 3 - Analyzing the Estimated Posterior Uncertainty

Posterior uncertainty is crucial in estimating the predictive uncertainty, which ultimately drives effective decision making. The standard deviation of Gaussian posterior directly correlates to the posterior uncertainty. Therefore, we analyze the posterior means and standard deviations that are estimated by each approach to comprehend their ability to adjust posterior uncertainty under drifting patterns in the data. Figures 5 and 6 illustrate the estimated means and standard-deviations using *airline* dataset for some selected coefficients (d) under single-pass and multi-pass updates, respectively. Moreover, figure 5 presents only the first 1e6 data-points, whereas 6 considers the last 1e6 samples, covering both ends of the experiment.

As expected, BB-SVB leads to overconfidence posteriors, resulting in near zero variance for both cases considered. Especially in figure 6, BB-SVB does not reflect any changes to the posterior uncertainty and is struggling to accommodate the necessary changes to the mean of the posteriors. However, BB-SVB seems to estimate the mean as expected under the relaxed constraints with single-pass updates; where it is equivalent to likelihood maximization of probability distributions. We still do not recommend using BB-SVB as an online inference technique, since it fails to indicate any drifting patterns by adjusting the posterior uncertainty. On the other hand, PVI also seems to underestimate the posterior uncertainty in

contrast to both SSVB and SVI. Let us consider the $d = 6$ from figure 5 and $d = 3$ from 6. PVI seems to neglect certain drifts that are evidenced by the posterior means, thus maintaining higher confidence compared to SSVB and SVI even under sudden changes to the posterior means.

Therefore, SSVB seems to reflect the posterior uncertainty under drifting patterns better than the rest of the approaches. Even though PVI modulates its posterior variance by fine-tuning α while suppressing certain drifts with the motive of optimizing the error rate or average log-predictive density, it may violate Bayesian in doing so. Even BB-SVB with single pass updates achieves comparable accuracy against PVI and SSVB, notwithstanding BB-SVB is useless according to Bayesian; Bayesian does not recommend sacrificing the posterior uncertainty to achieve better accuracy. On the other hand, the variational objective of SSVB can be modeled using Bayes' rule, thus we can assume the posterior estimated using SSVB follows Bayesian, providing us with a better sense of posterior uncertainty as evidenced by figures 5 and 6.

Phase 4 - Comparison with Other Single Pass Classifiers

We have already established that SSVB to give better accuracy in comparison to existing inference techniques such as PVI and SVI, eliminating the requirements such as optimizing α. However, such claims are useless to the practitioners unless SSVB can achieve similar accuracy compared to conventional online learning techniques. Hence, we compare SSVB and BB-SVI against three non-Bayesian online classifiers; most popular first order linear algorithm Passive Aggressive (PA) (Crammer et al., 2006), one of the state-of-the-art of second-order linear methods AROW (Crammer et al., 2009) and a traditional SGD classifier. It should be stressed that we do not consider non-linear classifiers in our analysis because we can not expect our linear classification model to exceed state-of-the-art non-linear classifiers. We follows the implementation proposed with LIBOL (Hoi et al., 2014) to extend AROW for multiclass classification. However, our implementation of AROW fails to scale with the number of features due to the large memory necessary to store the covariance matrices, thus we were unable to report the accuracy of AROW with *20News* dataset (which requires performing operations on top of 100000×100000 matrices).

Considering table 2, all the four inference approaches significantly outperforms three conventional classifiers, except with Otto Product dataset. For Otto Product dataset, SVI has slightly lower accuracy relative to PA and SGD. Moreover, we observe remarkable improvement in accuracy with all four inference techniques with *airline* dataset. We can consider the multinomial logistic regression based on SSVB and BB-SVI as a second-order classification since the underlying implementation of those algorithms updates the regression coefficient based on the gradients evaluated using the mean and the variance of those coefficients. This is similar to the concept of confident weighted linear classification (Dredze et al., 2008; Crammer et al., 2009), which is proven to be effective with online classifiers. Moreover, the online inference approaches estimate the full posterior densities not just confident weighed coefficients, therefore we can expect them to have superior performance even compared to the conventional second-order classifiers such as AROW etc.

Interestingly, PA and SGD have considerably outperformed the online inference approaches with *poker-hand* dataset. Moreover, AROW has approximately twice the error rate of PA with *poker-hand* dataset. It seems that *poker-hand* may have certain properties that lead to inconsistent uncertainties, which ultimately affect the accuracy of the second-order classifiers. However, we may require further analysis to express the exact cause for this behaviour.

Accordingly, SSVB demonstrates superior accuracy even against the conventional classifiers. Therefore, adopting SSVB benefits practitioners in two aspects; SSVB improves the accuracy of the model and SSVB provides predictive uncertainty to support decision making.

Experiment 3 - Linear Mixed Effects Regression

Linear Mixed Effect (LME) models are used to analyze data with both fixed and random effects. LME models have shown great potential with longitudinal data from different domains, where the same information is gathered on several subjects (e.g. multiple sensors or different users etc) at multiple points in the time. Even though there are many applications to LME model with data streams, we have not seen such technique that can tackle data streams with mixed effects. The existing techniques are rather focused on offline learning. Therefore, we introduce an LME model for online learning based on both SSVB and BB-SVI (see Appendix 3).

We use an artificially generated data stream to appraise the performance of LME model updated based on each objective. We generate a standard mixed effect stream with 100 dimensions (D) and 1000 subjects

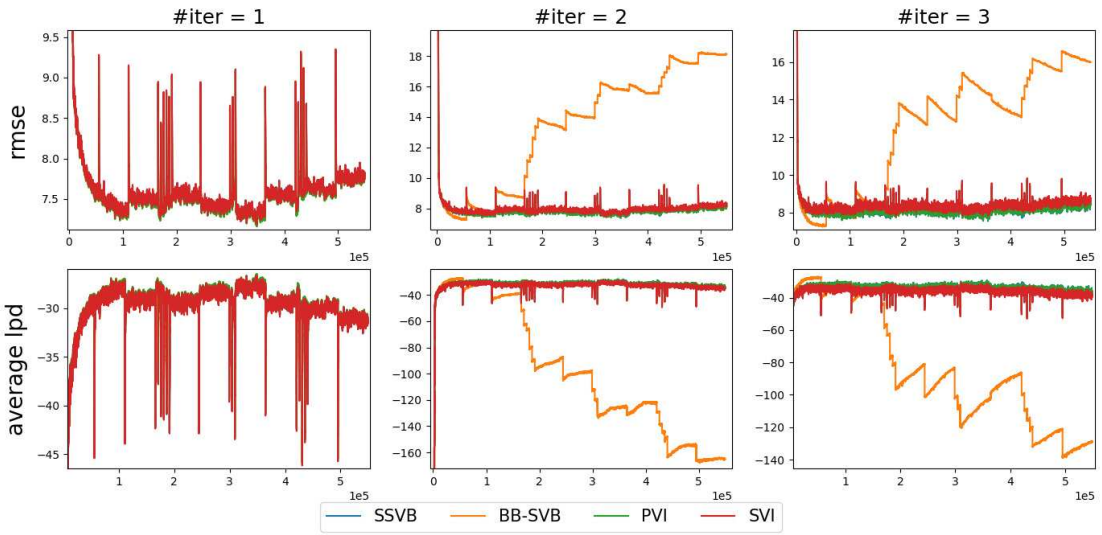

Figure 7. RMSE and lpd for LME models

	#iter	SSVB	BB-SVB	PVI	SVI
rmse	1	7.5552 ± 0.204	7.5439 ± 0.204	7.5441 ± 0.204	7.5648 ± 0.201
	2	7.8475 ± 0.195	16.9170 ± 1.494	7.8400 ± 0.193	7.9585 ± 0.190
	3	8.1600 ± 0.206	14.7463 ± 1.153	8.1844 ± 0.205	8.3896 ± 0.204
mae	1	5.8092 ± 0.150	5.7994 ± 0.150	5.7995 ± 0.150	5.8171 ± 0.149
	2	6.0638 ± 0.141	13.4496 ± 1.168	6.0565 ± 0.140	6.1561 ± 0.141
	3	6.3313 ± 0.153	11.6953 ± 0.921	6.3518 ± 0.153	6.5189 ± 0.157
lpd	1	-29.8617 ± 4.205	-29.7711 ± 4.190	-29.7807 ± 4.196	-29.9355 ± 4.198
	2	-31.6086 ± 1.755	-102.8580 ± 52.927	-31.5454 ± 1.745	-32.4117 ± 1.696
	3	-33.9901 ± 1.783	-83.3946 ± 36.280	-34.1499 ± 1.772	-35.7295 ± 1.774

Table 4. Average RMSE, average MAE and average Log-predictive density for mixed-effect regression

(C) following the equation 22 in Appendix 3. Moreover, we introduce random drifts to both fixed and random effects to simulate more realistic conditions. Typically, it is difficult to identify a single holdout set for a data stream with drifts. A holdout set selected at a particular instance may expire after next drift-point. Therefore, in our experiment, we generate a new holdout set after each drift-point reflecting the changes to the training data. Analogous to the previous experiments, we update the LME model using data arriving one-by-one ($M = 1$). We use the standard Gaussian distribution as the priors to fixed effects β and for random effects u we assume standard Multivariate Gaussian priors. ADAM optimizer is employed to update the model by setting ρ to 0.01. We determine the optimal α following the same criteria performed during the previous analysis. Moreover, we analyze each inference technique changing the number of passes from 1 to 3 during each update. We measure the average log-predictive density, root mean squared error (RMSE) and the mean absolute error (MAE) after each update using the hold-out set. Figure 7 illustrates the RMSE and log predictive density after each update to the models. Moreover, for each error metric, we consider the mean and the standard deviation of all the updates as the overall performance metrics capturing the accuracy of all the updates. The overall performance metrics for each approach is tabulated in the table 4. In the figure 7 and table 4, the number of passes carried out during each update is denoted as '#iter'.

All the three performance metrics reported in table 4 are consistent with each other, henceforth we collectively refer to them as the accuracy, bearing in mind that a decrement in error or higher log-predictive density indicates an improvement in the accuracy. SSVB, PVI and SVI show comparable accuracy for all the case. However, SVI possesses the worst accuracy out of those approaches having slightly less accuracy as against the other two techniques. PVI has gain marginally superior accuracy compared to

SSVB except for the case where three passes were performed during each update. Nevertheless, such negligible gain in accuracy for PVI compared to SSVB does not justify the exhausting analysis requires to determine a suitable value for α.

Analogous to our previous observations during experiment 2, each inference technique suffers from moderate decrement in accuracy when increasing the number of iterations, which we believe due to the overfitting of the posterior densities. Moreover, BB-SVB is failing to converge with multi-pass updates although BB-SVB reports the lowest error with single-pass updates. Figure 7 illustrates the RMSE and log predictive density after each update to the models. In figure 7 we observe significant deviations in accuracy after each drift-point due to the inability to adapt with changes to the data.

Since we were unable to find conventional mixed effect regression techniques that are capable of performing online updates, we choose two standard online regression techniques to understand the value of the proposed LME to the practitioners. We appraise the average RMSE and average MAE for PA regressor and SGD regressor. However, we do not report the error values for SGD as SGD fails to converge even to a local optimal. Nonetheless, PA regressor reports an RMSE and an MAE of 28.8775 ± 50.0463 and 24.2292 ± 43.4287, which is also substantially larger error relative to the observed error values with LME optimized using four inference objectives. Even though PA and SGD regressors are capable of modeling linear fixed effects in the generated data their inability to capture the random effects has lead to the observed poor convergence. Therefore, the discussed online LME is much effective and convenient in handling such random effects in streaming data.

RELATED WORK

As discussed in the introduction, VI was introduced by Jordan et al. (1999) as an efficient inference technique in order to handle complex Bayesian models. Coordinate ascent variational inference (CAVI) was widely adopted to solve the objective of VI as an optimization problem. However, CAVI fails to scale with the modern applications of probabilistic models, which often demands analyzing massive data (Blei et al., 2017; Hoffman et al., 2013). Thus, Hoffman et al. (2010); Wang et al. (2011); Hoffman et al. (2013) extend VI to handle large-scale data based on SVI, where they use mini-batches from a massive dataset to iteratively update the approximated posterior based on steepest descent. Nevertheless, the posterior being estimated using mini-batches is targeted for the full dataset with N data points (Hoffman et al., 2013), thus SVI requires knowing N beforehand. Due to the sensitivity of SVI to the N, it is often difficult for the practitioners to decide a suitable value for N (Broderick et al., 2013). Since SVI needs tedious model specific analyses under both offline and online settings, the black-box inference techniques such as Automatic Differentiation VI (ADVI) (Kucukelbir et al., 2017), Black-Box VI (BBVI) (Ranganath et al., 2014) and Reparameterization VI (Kingma and Welling, 2013; Zhang et al., 2018) was introduced to enable the inference of a wide range of models with little additional derivations. Conceptually, these techniques are not intended to estimate the intermediate posteriors given endless data streams and have not been empirically studied with regards to their effectiveness in online learning.

To apply variational approximation to the streaming data, Broderick et al. (2013); Ghahramani and Attias (2000); Honkela and Valpola (2003) proposed performing recursive Bayesian updating using offline approximation inference primitives such as CAVI. They incrementally update the approximated posterior for each mini-batch by considering most recent posterior as the prior to the Bayes rule, thus allowing to estimate the intermediate posterior densities irrespective of the size of the dataset. However, as pointed out by McInerney et al. (2015), Bayesian updating leads to point mass posterior with never-ending data streams, is thus ineffective in accommodating how the stream might change over time. Later, Nguyen et al. (2017) proposed Variational Continual Learning (VCL) framework dissolving Monte Carlo VI with the online variational inference. Their work suggests using a corset (i.e. a set of samples selected from previously observed data following particular criteria) with each Bayesian update to mitigate the phenomenon of catastrophic forgetting. Nevertheless, VCL is also vulnerable to the shortcomings of Bayesian updating when provided with drifting data. Moreover, the corsets may contain data-points generated prior to the recent drift-point, which will force the models to retain the information that should be forgotten to learn new patterns in data.

McInerney et al. (2015) introduced PVI where they approximate population posterior by considering each batch as a randomly sampled points from a population posterior. Their results justify using a different value for N instead of the size of the dataset, which they conceive as the number of data-points in the population posterior α. They use α to control the variance of the population posterior avoiding the

overconfidence posteriors.

Our proposed technique adopts recursive Bayesian updating to derive a black-box inference technique for streaming data similar to Broderick et al. (2013); Ghahramani and Attias (2000); Honkela and Valpola (2003); Nguyen et al. (2017). Our initial objective is more similar to the continual learning objective (Nguyen et al., 2017) without corsets. However, the improved objective is significantly different from VCL with the additional modifications to be more suited for concepts drifts. However, our approach does not enforce new hyperparameters to the traditional Bayesian methods as in Population VI. Instead, it controls the posterior variance based on the amount of data that have been observed at a given point.

CONCLUSION

In this paper, we first introduce two modifications to the traditional Bayesian updating framework deriving a novel streaming Bayesian updating approach that is capable of handling data streams with concept drift. Fusing the proposed Bayesian updating approach with reparameterization VI, we then derive a black-box inference technique for online settings; "Streaming and Stochastic Variational Bayes" (SSVB). Unlike existing online inference techniques, SSVB does not suffer from overconfidence posteriors and capable of adequately accommodating drifting patterns in data streams without tuning any additional hyperparameters. We evaluated the performance of SSVB against BB-SVB, and two existing online inference approaches PVI and SVI for two essential models to the online learning; multinomial logistic regression and linear-mixed effects model. SSVB achieves either superior or comparable performance without any additional hyperparameter tuning as against current state-of-the; PVI. In addition, SSVB demonstrates a significant accuracy gain for online classification against the conventional online classifiers such as AROW, PA and SGD classifiers.

Therefore, SSVB can be considered as a more efficient online inference technique in contrast to the current online inferences techniques such as PVI, SVI and SVB. Practitioners can easily adopt SSVB easily to derive wide-range of online inference tasks with real-world streaming data avoiding the discussed drawbacks with existing state-of-the-art techniques. Nevertheless, SSVB demands its prior distributions to follow a specific parameterization allowing embedding only the mean of preceding posterior leaving the variance of the prior unaltered. Regardless of such limitations, SSVB is yet useful with a wide range of models to handle endless data streams with drifting patterns.

REFERENCES

Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American Statistical Association*, 112(518):859–877.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A. C., and Jordan, M. I. (2013). Streaming variational bayes. In Burges, C. J. C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 26*, pages 1727–1735. Curran Associates, Inc.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., and Singer, Y. (2006). Online passive-aggressive algorithms. *J. Mach. Learn. Res.*, 7:551–585.

Crammer, K., Kulesza, A., and Dredze, M. (2009). Adaptive regularization of weight vectors. In *Advances in neural information processing systems*, pages 414–422.

Dredze, M., Crammer, K., and Pereira, F. (2008). Confidence-weighted linear classification. In *International Conference on Machine Learning (ICML)*.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

Figurnov, M., Mohamed, S., and Mnih, A. (2018). Implicit reparameterization gradients. *arXiv preprint arXiv:1805.08498*.

Gal, Y. (2016). *Uncertainty in Deep Learning*. PhD thesis, University of Cambridge.

Gama, J. a., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. (2014). A survey on concept drift adaptation. *ACM Comput. Surv.*, 46(4):44:1–44:37.

Ghahramani, Z. and Attias, H. (2000). Online variational bayesian learning. In *Slides from talk presented at NIPS workshop on Online Learning*.

Hoffman, M., Bach, F. R., and Blei, D. M. (2010). Online learning for latent dirichlet allocation. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 856–864. Curran Associates, Inc.

Hoffman, M. D., Blei, D. M., Wang, C., and Paisley, J. (2013). Stochastic variational inference. *J. Mach. Learn. Res.*, 14(1):1303–1347.

Hoi, S. C., Wang, J., and Zhao, P. (2014). Libol: A library for online learning algorithms. *Journal of Machine Learning Research*, 15:495–499.

Honkela, A. and Valpola, H. (2003). On-line variational bayesian learning. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, pages 803–808.

Jordan, M. I., Ghahramani, Z., and et al. (1999). An introduction to variational methods for graphical models. In *MACHINE LEARNING*, pages 183–233. MIT Press.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Welling, M. (2013). Auto-encoding variational bayes. *CoRR*, abs/1312.6114.

Kivinen, J., Smola, A. J., and Williamson, R. C. (2004). Online learning with kernels. *IEEE transactions on signal processing*, 52(8):2165–2176.

Kucukelbir, A., Tran, D., Ranganath, R., Gelman, A., and Blei, D. M. (2017). Automatic differentiation variational inference. *J. Mach. Learn. Res.*, 18(1):430–474.

McInerney, J., Ranganath, R., and Blei, D. M. (2015). The population posterior and bayesian modeling on streams. In *NIPS*.

Nguyen, C. V., Li, Y., Bui, T. D., and Turner, R. E. (2017). Variational continual learning. *arXiv preprint arXiv:1710.10628*.

Paisley, J. W., Blei, D. M., and Jordan, M. I. (2012). Variational bayesian inference with stochastic search. In *ICML*.

Ranganath, R., Gerrish, S., and Blei, D. (2014). Black Box Variational Inference. In Kaski, S. and Corander, J., editors, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pages 814–822, Reykjavik, Iceland. PMLR.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*.

Wainwright, M. J. and Jordan, M. I. (2008). Graphical models, exponential families, and variational inference. *Found. Trends Mach. Learn.*, 1(1-2):1–305.

Wang, C., Paisley, J., and Blei, D. (2011). Online variational inference for the hierarchical dirichlet process. In Gordon, G., Dunson, D., and Dudík, M., editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 752–760, Fort Lauderdale, FL, USA. PMLR.

Webb, G. I., Hyde, R., Cao, H., Nguyen, H., and Petitjean, F. (2015). Characterizing concept drift. *CoRR*, abs/1511.03816.

Zhang, C., Butepage, J., Kjellstrom, H., and Mandt, S. (2018). Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*.

APPENDIX 1 - PROOF OF STREAMING VARIATIONAL OBJECTIVES

We will outline the derivations of streaming variational objectives presented with equations 7 and 10 respectively starting from the Bayesian updating frameworks in equations 1 and 3, respectively.

Streaming Variational Objective with Traditional Bayesian Updating

Let us first rewrite Bayesian updating (eq. 1) in terms of likelihood, prior and marginal probability of data.

$$\begin{aligned} p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1) &= \prod_{i=1}^{b} \big[p(\mathrm{c}_i|\mathrm{z})\big] p(\mathrm{z}) / p(\mathrm{c}_b \dots \mathrm{c}_1) \\ &= p(\mathrm{c}_b|\mathrm{z}) \prod_{i=1}^{b-1} \big[p(\mathrm{c}_i|\mathrm{z})\big] p(\mathrm{z}) / p(\mathrm{c}_b) p(\mathrm{c}_{b-1} \dots \mathrm{c}_1) \\ &= p(\mathrm{c}_b|\mathrm{z}) p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1) / p(\mathrm{c}_b) \end{aligned} \tag{18}$$

Consider the KL-divergence between an appropriate family of distribution $q_\theta(.)$ and posterior $p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1)$ estimated using Bayesian updating. Recall that $q_\theta(.)$ is parameterized by θ.

$$\begin{aligned} \mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1)] &= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z}) \ln \frac{q_\theta(\mathrm{z})}{p(\mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1)} \mathrm{dz} \\ &= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z}) \left[\ln \frac{q_\theta(\mathrm{z})}{p(\mathrm{c}_b|\mathrm{z}) p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1)} + \ln p(\mathrm{c}_b) \right] \mathrm{dz}, \quad \text{eq. 18} \\ &= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z}) \ln \frac{q_\theta(\mathrm{z})}{p(\mathrm{c}_b|\mathrm{z}) p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1)} \mathrm{dz} + \ln p(\mathrm{c}_b) \\ &= -\mathcal{L}(\theta; \mathrm{c}_b \dots \mathrm{c}_1) + \ln p(\mathrm{c}_b) \end{aligned}$$

We will consider the variational lower bound $\mathcal{L}(\theta; \mathrm{c}_b \dots \mathrm{c}_1)$ for traditional Bayesian updating, separately.

$$\begin{aligned} \mathcal{L}(\theta; \mathrm{c}_b \dots \mathrm{c}_1) &= -\int_{-\infty}^{+\infty} q_\theta(\mathrm{z}) \ln \frac{q_\theta(\mathrm{z})}{p(\mathrm{c}_b|\mathrm{z}) p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1)} \mathrm{dz} \\ &= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z}) \ln p(\mathrm{c}_b|\mathrm{z}) \mathrm{dz} - \int_{-\infty}^{+\infty} q_\theta(\mathrm{z}) \ln \frac{q_\theta(\mathrm{z})}{p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1)} \mathrm{dz} \\ &= \mathrm{E}[\ln p(\mathrm{c}_b|\mathrm{z})] - \mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1)] \end{aligned}$$

$$\begin{aligned} &q_{\theta_b}(\mathrm{z}) \simeq p(\mathrm{z}|\mathrm{c}_{b-1} \dots \mathrm{c}_1) \\ &\therefore \mathcal{L}(\theta; \mathrm{c}_b, \theta_{b-1}) = \mathrm{E}[\ln p(\mathrm{c}_b|\mathrm{z})] - \mathrm{D}_{KL}[q_\theta(\mathrm{z})||q_{\theta_b}(\mathrm{z})], \end{aligned}$$

The derived objective is identical to streaming variational objective illustrated in equation 7.

Streaming Variational Objective with Proposed Bayesian Updating

Let us first rewrite the proposed Bayesian updating (eq. 1) with the scaling function S_b to scale the likelihood of batch c_b instead of simply using the number of batches. Since $S_b \in \mathcal{R}^+$, we substitute the product of likelihoods term with a likelihood raised to the power of S_b.

$$\begin{aligned} p(\mathrm{z}| < \mathrm{c}_1 \dots c_b >) &\simeq \prod_{i=1}^{\mathrm{S}_b} \big[p(\mathrm{c}_b|\mathrm{z})\big] p(\mathrm{z})^* / p(< \mathrm{c}_1 \dots c_b >) \\ &= p(\mathrm{c}_b|\mathrm{z})^{\mathrm{S}_b} p(\mathrm{z})^* / p(< \mathrm{c}_1 \dots c_b >) \end{aligned} \tag{19}$$

Analogous to the previous proof, consider the KL-divergence between an appropriate family of distribution $q_\theta(.)$ and posterior $p(< \mathrm{z}|\mathrm{c}_b \dots \mathrm{c}_1 >)$ estimated using Bayesian updating.

$$\begin{aligned}
\mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z}|<\mathrm{c}_b\ldots\mathrm{c}_1>)] &= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z})\ln\frac{q_\theta(\mathrm{z})}{p(\mathrm{z}|<\mathrm{c}_b\ldots\mathrm{c}_1>)}\mathrm{dz} \\
&= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z})\left[\ln\frac{q_\theta(\mathrm{z})}{p(\mathrm{c}_b|\mathrm{z})^{\mathrm{S}_b}p(\mathrm{z})^*}+\ln p(<\mathrm{c}_1\ldots c_b>)\right]\mathrm{dz}, \quad \text{eq.19} \\
&= \int_{-\infty}^{+\infty} q_\theta(\mathrm{z})\ln\frac{q_\theta(\mathrm{z})}{p(\mathrm{c}_b|\mathrm{z})^{\mathrm{S}_b}p(\mathrm{z})^*}\mathrm{dz}+\ln p(<\mathrm{c}_b\ldots\mathrm{c}_1>) \\
&= -\mathcal{L}(\theta;c_b,\mathrm{S}_b)+\ln p(<\mathrm{c}_b\ldots\mathrm{c}_1>)
\end{aligned}$$

Let us consider the variational lower bound $\mathcal{L}(\theta;c_b,\mathrm{S}_b)$ of proposed Bayesian updating.

$$\begin{aligned}
\mathcal{L}(\theta;c_b,\mathrm{S}_b) &= -\int_{-\infty}^{+\infty} q_\theta(\mathrm{z})\ln\frac{q_\theta(\mathrm{z})}{p(\mathrm{c}_b|\mathrm{z})^{\mathrm{S}_b}p(\mathrm{z})^*}\mathrm{dz} \\
&= \mathrm{S}_b\int_{-\infty}^{+\infty} q_\theta(\mathrm{z})\ln p(\mathrm{c}_b|\mathrm{z})\mathrm{dz}-\int_{-\infty}^{+\infty} q_\theta(\mathrm{z})\ln\frac{q_\theta(\mathrm{z})}{p(\mathrm{z})^*}\mathrm{dz} \\
&= \mathrm{S}_b\times\mathrm{E}[\ln p(\mathrm{c}_b|\mathrm{z})]-\mathrm{D}_{KL}[q_\theta(\mathrm{z})||p(\mathrm{z})^*]
\end{aligned}$$

We have derived the proposed streaming variational objective from considering the KL-divergence between an suitable family of distribution $q(.)$ and the proposed posterior $p(\mathrm{z}|<\mathrm{c}_1\ldots c_b>)$.

APPENDIX 2 - MULTINOMIAL LOGISTIC REGRESSION

Let us consider M data-points $\mathrm{x}=\{x_i\}_{i=1}^M$ where each sample x_i is D-dimensional. The targets $\mathrm{y}=\{y_i\}_{i=1}^M$ consist K-dimensional vectors representing probability of each class given the respective x_i. Then likelihood presented in equation 20 describes the data generated i.i.d., where $h(.)$ denotes the Softmax function.

$$p(\mathrm{y}|\mathrm{x},\mathrm{w})=\prod_{i=1}^{M}\mathcal{C}at(y_i|h(x_i.\mathrm{w})) \tag{20}$$

The inference process is expected to approximate the posterior of the coefficient matrix w that is parameterized by μ and σ^2. Therefore, the prior $p(w_{ij})$ and posterior $q(w_{ij})$ corresponding to the i^{th} predictor and the j^{th} class can be defined as follows.

$$\begin{aligned}
p(w_{ij}) &= \mathcal{N}(\bar{\mu}_{ij},\bar{\sigma}_{ij}^2) \\
q(w_{ij}) &= \mathcal{N}(\mu_{ij},\sigma_{ij}^2)
\end{aligned} \tag{21}$$

This concludes the probabilistic model for multinomial logistic regression. We optimize the probabilistic model based on two derived techniques SSVB and BB-SVB, and the existing state-of-the-art; PVI and SVI.

APPENDIX 3 - LINEAR MIXED EFFECT MODEL

Consider the set of M observations $\mathrm{y}=\{y_i\}_{i=1}^M$ corresponding to D-dimensional fixed-effect predictors $\mathrm{X}=\{X_i\}_{i=1}^M$ collected sequentially from $\mathcal{C}$ subjects that is described by the random effects vector u. Assuming that the fixed effect predictors and the observations follow a linear relationship we can denote the i^{th} observation y_i as follows.

$$y_i = X_i\beta+Z_iu+\varepsilon_i \tag{22}$$

In equation 22 the observations y_i are described as a combination of the fixed effects β with random effects u. Fixed effect β is a D dimensional vector that consists of regression coefficient for the D predictors X_i, whereas random effect u is $\mathcal{C}$ dimensional vector corresponding to the random effects for $\mathcal{C}$ subjects. Random effects design vector Z_i is typically a one-hot encoded vector indicating the source of the observation y_i out of $\mathcal{C}$ subjects to assign the corresponding random effect from u. Error term ε_i represents the noise in the each observation y_i.

The likelihood of the observations y can be express as the conditional probability shown in 23 which is assumed to be corrupted by i.i.d. Gaussian noise with unknown variance σ^2.

$$p(\mathrm{y}|\mathrm{x},\beta,b) = \prod_{i=1}^{M} \mathcal{N}(y_i|X_i\beta + Z_i u, \sigma^2) \tag{23}$$

In our implementations of LME, we consider both β and u as random variables, thus coefficients of fixed effect predators and random effects are respectively given Gaussian and Multivariate Gaussian priors as illustrated in equation 24. The respective posteriors are approximated to the same distributions as their priors.

$$\begin{aligned} p(\beta_d) &= \mathcal{N}(\mu_d, \sigma_d^2) \\ p(u) &= \mathcal{N}_C(\mu, \Sigma) \end{aligned} \tag{24}$$

Figure 8 illustrates the inference network implemented for LME model.

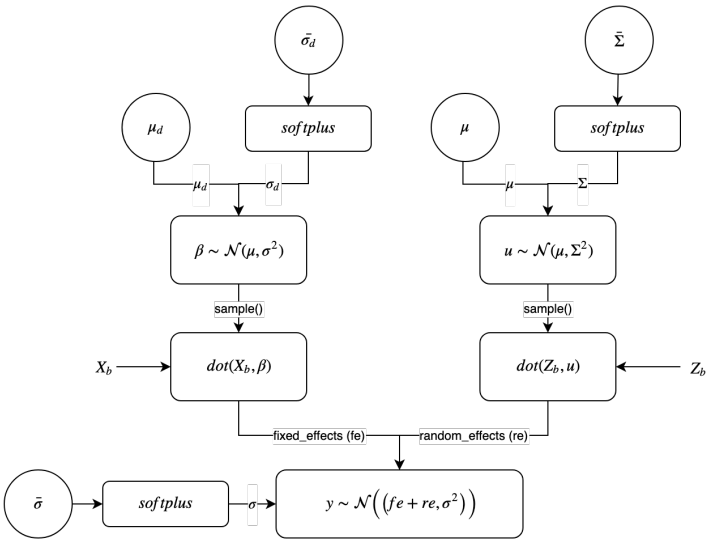

Figure 8. Forward Propagation of the Inference Network Implemented for LME Model

APPENDIX 4 - CLASSIFICATION FINAL F1 SCORES

	20News	MNIST	Otto Products
SSVB	0.8236 ± 0.0014	0.8932 ± 0.0014	0.8226 ± 0.0034
BB-SVB	0.8235 ± 0.0027	0.8874 ± 0.0148	0.8226 ± 0.0034
PVI	0.7749 ± 0.0110	0.8906 ± 0.0066	0.8222 ± 0.0007
SVI	0.7123 ± 0.0148	0.8845 ± 0.0097	0.8041 ± 0.0049
AROW	-	0.8970 ± 0.0014	0.8004 ± 0.0018
PA	0.7965 ± 0.0237	0.8792 ± 0.0084	0.8241 ± 0.0325
SGD	0.7659 ± 0.0278	0.8722 ± 0.0082	0.8268 ± 0.0197

Table 5. Final F1 scores using with-hold set for multi-class classification

Table 5 presents the f1 scores computed using the with-hold set for each multiclass classification dataset. These values are computed once the model is updated using all the datasets in the training set. We have followed the same experiment settings that are used to estimate the values in 2.

APPENDIX 5 - MEANS AND STDS WITH POKER DATASET

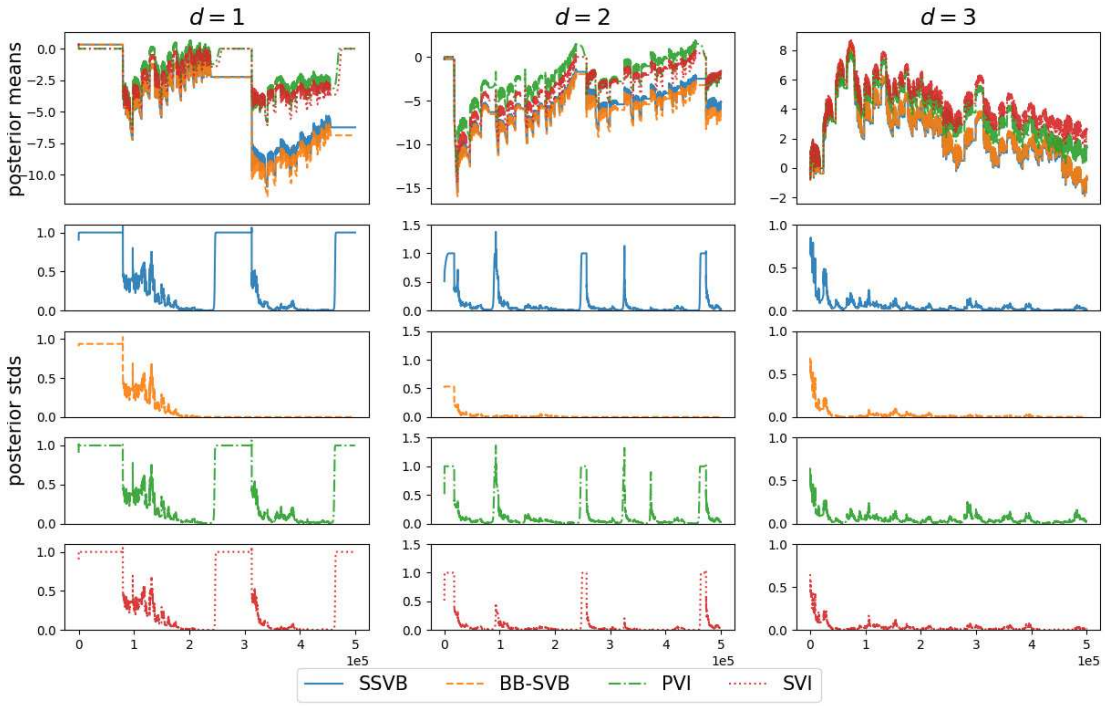

Figure 9. Mean and Std estimated by each approach for poker dataset under single-pass updates

In figure 9, the posterior means of the PVI and SVI are significantly vary from the posterior means of estimated by the SSVB and BB-SVB.