PeerJ Preprints NOT PEER-REVIEWED

Streaming stochastic variational Bayes: An improved approach for inference with concept drifting data streams

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 overconfident posteriors, where posterior variance becomes too inadequate to adapt to new changes to the posterior with concept drifting data streams. 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 to control the posterior variance 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 demonstrate the superior 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 emphasize the significant accuracy gain with SSVB based inference against conventional online learning models for each task.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

PeerJ Preprints NOT PEER-REVIEWED

Streaming stochastic variational Bayes: An improved approach for inference with concept drifting data streams

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

[1]**CTO office, WSO2, Colombo, Sri Lanka**

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 overconfident posteriors, where posterior variance becomes too inadequate to adapt to new changes to the posterior with concept drifting data streams. 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 to control the posterior variance 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 demonstrate the superior 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 emphasize 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 such massive data streams in memory, which is infeasible for some cases. With these critical limitations, online learning has become an essential tool, 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.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

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 overconfident 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 Evaluation section, overconfident 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 overconfident 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, 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 overconfident 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. SSVB effectively fuses stochastic gradient descent and Bayesian updating framework avoiding any additional hyperparameters to control the posterior variance in streaming settings as opposed to Hoffman et al. (2013, 2010); Wang et al. (2011); McInerney et al. (2015). On the other hand, SSVB modifies traditional Bayesian updating framework preventing overconfident posteriors encountered with Broderick et al. (2013); Nguyen et al. (2017) under concept drifts. Therefore, SSVB can successfully accommodate concept drifting data streams without undermining the accuracy of the posteriors. Moreover, SSVB enables black-box inference for a wide range of models by replacing manually derived gradient estimators in Hoffman et al. (2013, 2010); Wang et al. (2011); McInerney et al. (2015) with stochastic backpropagation. Accordingly, SSVB is easily adoptable by practitioners or researchers with endless data streams that change over time, to fabricate a wide array of models avoiding tedious model-specific derivations demanded by existing state-of-the-art online inference techniques.

In this paper, we first introduce two modifications to the traditional Bayesian updating framework obtaining a streaming updating rule that is useful to implement black-box inference for handling concept drifting data streams. We discuss the properties of the proposed streaming Bayesian updating using two simple conjugate models while demonstrating the drawbacks of traditional Bayesian updating framework. Following the proposed Bayesian updating approach, we then derive the novel black-box inference technique SSVB for online settings. 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 parameter α. 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 evaluation 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 ceaselessly 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 $p(\mathrm{z}|\mathrm{c}_b \ldots \mathrm{c}_1)$ given the first b batches of data $\mathrm{c}_1 \ldots \mathrm{c}_b$ 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 $p(\mathrm{z}|\mathrm{c}_b \ldots \mathrm{c}_1)$ as below.

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

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

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

Bayesian updating framework illustrated in equation 1 is ill-suited for data that does not hold i.i.d assumptions.

Overconfident Posterior Overconfident posterior is the phenomenon of underestimating the posterior uncertainty when applying traditional Bayesian updating with the data streams that change over time (McInerney et al., 2015). With traditional Bayesian updating, the posteriors continuously shrink with incoming data even after each drift-point assuming that the data are identically distributed. Such posteriors undermine the ability to accommodate changes to the posteriors due to the overestimated posterior confidence, thus resulting in extremely poor accuracies after each drift point.

Streaming Bayesian Updating

This section elucidates the modifications to the traditional Bayesian updating framework to improve its ability to accommodate changes in streaming settings. 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. Any change to the underlying distributions is 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 $p(\mathrm{z}| < \mathrm{c}_1 \dots c_b >)$.

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 propagate only 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.,

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

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 while 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 conceptually, 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 will suggest posteriors shifted from our beliefs that are embedded via priors $p(\mathrm{z})^*$. Such disagreements between likelihood and priors will result in higher variance in posteriors allowing them to shift towards posteriors. Accordingly, the proposed Bayesian updating is capable of accommodating new concepts forgetting outdated observations, whereas traditional Bayesian updating simply encodes each observation to the estimated posteriors irrespective of their order.

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

PeerJ Preprints NOT PEER-REVIEWED

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, the traditional Bayesian updating framework fails to handle real-world data streams with concept drifts diminishing its utility to implement black-box online inference. We have tailor-made the proposed Bayesian updating framework to handle such streaming data while adapting drifting patterns more efficiently when optimized via stochastic gradient descent framework.

STREAMING STOCHASTIC VARIATIONAL BAYES

We now derive an online inference objective fusing Bayesian updating with the traditional variational objective. Such an objective cannot still efficiently accommodate the changes with conventional Bayesian updating. Therefore, we then extend our initial objective with two amendments enabling the ability to adopt drifting patterns in data streams as suggested by the streaming Bayesian updating framework. Lastly, we outline the black-box inference for both streaming variation objectives based on stochastic gradient updates formulating BB-SVB and SSVB frameworks for online inference.

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, thus continuously generating i.i.d. data. Based the traditional Bayesian updates, 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 [1][2].

[1] proof in Appendix 1

[2] appendices can be found with supplemental files

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

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 the improved streaming variational objective can be formulated as [3],

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

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 location of the posterior . 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)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 $\mathcal{S}_b$ instead of the number of batches b to scale the likelihood term in the variational objective in order to introduce additional parameters to the scale the likelihood of data. We define S_b as,

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

where n_b is the total number of data-points used during all the updates including the current update and M is the size of a mini-batch. As an additional parameter, we have introduced a normalization constant ϕ to adjust the regularization to the proposed objective to avoid overfitting. Typically, the variational objective achieves the desired regularization by adjusting the priors. Given that we are now utilizing expectations and the variances of the priors respectively to propagate formerly learned information and to embed the fixed initial uncertainty, the normalization constant ϕ is essential to tweak the regularization of the proposed variational objective. We could also consider normalization constant as a refinement of the net amount of information observed by the model to the control variance of the posteriors. However, in our experiments, we have considered $\phi = 1$ unless specified otherwise; we achieve the state-of-the-art performance with SSVB by employing the default settings $S_b = b$ 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 bth batch inclusively. McInerney et al.

[3] proof in Appendix 1

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

PeerJ Preprints NOT PEER-REVIEWED

(2015) also 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 S_b (e.g. setting $\mathrm{S}_b = 1$) we can refresh the posteriors by forgetting irrelevant information. Such a resetting can be also useful for an occasional re-calibration of the posterior uncertainty. Therefore, resetting of S_b can be triggered by monitoring the log predictive densities (lpd) of new data-points estimated before using them to update the models (i.e. prequential evaluation), because a constantly decreasing lpd can be an indication of deteriorating posteriors.

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 expectation and variance, separately. Even though most of the distributions are not explicitly parameterized as expectation and variance, we can still obtain the distributions having any given expectation and variance > 0. For example, we can easily obtain the Gamma prior with given expectation and variance by defining the shape and rate parameters of the Gamma distribution in terms of the expectation and variance. Alternatively, we could handle such constraints by using appropriate distributions for the priors that allow explicit parameterization of expectation and variance. 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 objectives adopted to streaming settings. In the next section, we will outline the implementation of 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$^{\text{th}}$ 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 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 during stochastic backpropagation (Paisley et al., 2012; Ranganath et al., 2014). Nevertheless, the gradient estimated using Monte 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.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

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

```
Inputs : $c_1 \dots c_b$, $\theta_0$
Initialize : $\theta$
foreach $c_i \in c_1 \dots c_b$ do
    $\bar{\theta} \leftarrow \theta_{i-1}$
    for $t \in 1 : T$ do
        $g \leftarrow \nabla_\theta \mathcal{L}(\theta; c_i, \bar{\theta})$ (Eq. 11)
        $\theta_i \leftarrow$ Update parameters using gradients $g$ (Eq. 14 with ADAM)
    end
end
return $\theta$
```

Algorithm 2: Streaming Stochastic Variational Bayes - SSVB

```
Inputs : $c_1 \dots c_b$, $\mu_0$, $\sigma_0^2$
Initialize : $\theta$
foreach $c_i \in c_1 \dots c_b$ do
    $\bar{\mu} \leftarrow \mu_{i-1}$
    $S_i \leftarrow S_{i-1} + 1$
    for $t \in 1 : T$ do
        $g \leftarrow \nabla_\theta \mathcal{L}(\theta; c_i, S_i, \bar{\mu})$ (Eq. 12)
        $\theta_i \leftarrow$ Update parameters using gradients $g$ (Eq. 14 with ADAM)
    end
end
return $\theta$
```

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 to take a single gradient step. Each update repeats T such passes to approximate the posterior. 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. Each algorithm iterates over each batch of data ($c_1 \dots c_b$) in the order they arrive. First, the priors to the streaming variational objectives are updated using the posteriors estimated with the previous batch following a Bayesian updating framework. BB-SVB employs traditional Bayesian updating similar to Broderick et al. (2013), whereas SSVB follows the proposed Bayesian updating framework in equation 3. Once the priors are updated, a sequence of gradient descent steps is carried out in the direction of the gradient $g(\theta)$ computed following equation 13 to approximate the posteriors.

PeerJ Preprints NOT PEER-REVIEWED

On the other hand, it could be argued that SSVB as optimizing a modified variational objective analogous to SVI Hoffman et al. (2013) or PVI McInerney et al. (2015). However, unlike SVI and PVI, SSVB employs a continuously evolving objective by encoding information regarding the previously observed data by propagating the expectation of the posteriors. As a result, SSVB collectively utilizes gradient descent and Bayesian updating to incrementally approximate the posteriors, whereas SVI and PVI entirely depend on stochastic gradient descent to accommodate the changes to the posterior. Moreover, SSVB scales the likelihood by a factor of the number of data-points observed by the models, whereas PVI and SVI simply employ a fixed scale. Therefore, the fact that PVI and SVI are able to improve the posterior variance is merely an effect of gradient descent steps.

Single Pass Updates to BB-SSVB 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-SSVB 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 the current point. As a result, the KL divergence of most of the commonly adopted families of distributions (e.g. Gaussian and Gamma) has zero gradients during single-pass updates with BB-SVB. Accordingly, single-pass updates with BB-SVB becomes equivalent to likelihood maximization of the model variables.

EVALUATION

This section presents the evaluation of the proposed streaming Bayesian updating framework and SSVB. First, we will analyse the properties of the streaming Bayesian updating with two simple conjugate models. Subsequently, we will conduct an extensive empirical evaluation of SSVB with real-world and generated data streams in comparison with the state-of-the-art black-box inference techniques.

Streaming Bayesian Updating

In this section, we will analyze two simple conjugate models that are derived using the traditional Bayesian updating and the proposed streaming Bayesian updating for streaming data.

Gaussian Distribution with Known Variance

Consider a scenario where a Gaussian distribution with an unknown mean $\hat{\mu}$ and a known variance $\hat{\sigma}^2$ is used to generate a series of batches $c_1 \dots c_b$, each with M data-points. The natural conjugate prior for the mean $\hat{\mu}$ is another Gaussian distribution with mean μ and variance σ. Following the steps illustrated in Gelman et al. (2004), it can be shown that the posterior of $\hat{\mu}$ after observing c_b is,

$$\hat{\mu}|c_1 \dots c_b \sim \mathcal{N}(\mu_b, \sigma_b^2) \tag{15}$$

$$\mu_b = \frac{\frac{1}{\sigma_{b-1}^2}\mu_{b-1} + \frac{M}{\hat{\sigma}^2}\bar{c}_b}{\frac{1}{\sigma_{b-1}^2} + \frac{M}{\hat{\sigma}^2}} = \frac{\frac{1}{\sigma_0^2}\mu_0 + \frac{b \times M}{\hat{\sigma}^2}\overline{c_1 \dots c_b}}{\frac{1}{\sigma_0^2} + \frac{b \times M}{\hat{\sigma}^2}} \tag{16}$$

$$\sigma_b^2 = \frac{1}{\frac{1}{\sigma_{b-1}^2} + \frac{M}{\hat{\sigma}^2}} = \frac{1}{\frac{1}{\sigma_0^2} + \frac{b \times M}{\hat{\sigma}^2}} \tag{17}$$

During incremental updates, we update the posterior parameters μ_b and σ_b^2 following equations 16 and 17. Here $\overline{c_1 \dots c_b}$ denotes the expected value of the b batches $c_1 \dots c_b$. Let us now emphasize the shortcomings of traditional Bayesian updating using the derived model updates.

Consider the incremental updates to the posterior variance σ_b^2 as illustrated in equation 17. Notice that $\sigma_b^2 < \sigma_{b-1}^2$ thus resulting in an exponential decay of the posterior variance with the number of batches observed by the model, irrespective of the magnitude of the data values. Therefore, with continuously arriving indefinite data streams $\lim_{b \to \infty} \sigma_b^2 = 0$. This is supposed because the posterior uncertainty should eventually disappear with infinite data streams. However, as a result, $\mu_b \to \mu_{b-1}$ disregarding the drifts to the incoming data when b approaches infinity. Accordingly, the posterior estimated following the

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

traditional Bayesian updating is independent of the order of the batches and fails adapt to changing data streams.

Let us now derive the posterior of $\hat{\mu}$ following the proposed streaming Bayesian updating rule illustrated in equation 3. We approximate the posterior of $\hat{\mu}$ given the sequence $< c_1 \ldots c_{b-1} >$ as,

$$\hat{\mu} | < c_1 \ldots c_b > \sim \mathcal{N}(\mu_b, \sigma_b^2) \tag{18}$$

$$\mu_b = \frac{\frac{1}{\sigma_0^2}\mu_{b-1} + \frac{b \times M}{\hat{\sigma}^2}\bar{c}_b}{\frac{1}{\sigma_0^2} + \frac{b \times M}{\hat{\sigma}^2}} = \prod_{i=1}^{b}\left[\frac{\frac{1}{\sigma_0^2}}{\frac{1}{\sigma_0^2} + \frac{i \times M}{\hat{\sigma}^2}}\right]\mu_0 + \frac{M\sigma_0^2}{\hat{\sigma}^2}\sum_{i=1}^{b}\left(\prod_{j=i}^{b}\left[\frac{\frac{1}{\sigma_0^2} \times i}{\frac{1}{\sigma_0^2} + \frac{i \times M}{\hat{\sigma}^2}}\right]\bar{c}_i\right) \tag{19}$$

$$\sigma_b^2 = \frac{1}{\frac{1}{\sigma_0^2} + \frac{b \times M}{\hat{\sigma}^2}} \tag{20}$$

With proposed Bayesian updating, the expectation of each batch $\bar{c}_i$ is weighted by an additional factor $\left(1/\sigma_0^2 + (i-1) \times M/\hat{\sigma^2}\right)/\left((i-1)/\sigma_0^2\right) > 1$ with respect to its preceding batch, preserving the sequential nature of data. Therefore, the posterior updates with streaming Bayesian updating allow duly updating the posteriors, while gradually forgetting the already learned concepts with time.

Updates to the posterior variance following the proposed Bayesian updating is identical to the updates obtained following traditional Bayesian updating. This is mainly due to the fixed and known variance $\hat{\sigma^2}$ of the likelihood of the Gaussian model. However, since the posterior is adjusted in accordance with the drifting patterns in data, updates to the posterior variance using equation 20 does not lead to overconfident posteriors with streaming Bayesian updating.

Poisson Model with Gamma Conjugate Priors

Consider a Poisson distribution with an unknown rate λ and let $c_1 \ldots c_b$ be b batches each with M observations sampled from the Poisson distribution. The natural conjugate prior for λ is a Gamma distribution parameterized by mean μ and variance σ^2. Then the posterior of λ is given by following (Gelman et al., 2004),

$$\lambda | c_1 \ldots c_b \sim \Gamma(\mu, \sigma^2) \tag{21}$$

$$\mu_b = \frac{\mu_{b-1}^2 + \sigma_{b-1}^2 M\bar{c}_b}{\mu_{b-1} + \sigma_{b-1}^2 M} = \frac{\mu_0^2 + b\sigma_0^2 M\overline{c_1 \ldots c_b}}{\mu_0 + b\sigma_0^2 M} \tag{22}$$

$$\sigma_b^2 = \sigma_{b-1}^2 \frac{\mu_{b-1}^2 + \sigma_{b-1}^2 M\bar{c}_b}{\left(\mu_{b-1} + \sigma_{b-1}^2 M\right)^2} = \sigma_0^2 \frac{\mu_b}{\left(\mu_0 + b\sigma_0^2 M\right)} \tag{23}$$

It can be shown that $\lim_{b \to \infty} \sigma_b^2 = 0$ and $\lim_{\sigma_b^2 \to 0} \mu_b = \mu_{b-1}$. Therefore, analogous to the previous model, the Poisson model fails to accommodate changes in data to the posteriors due to overconfident posteriors. Let us now analyse the approximated posterior for λ following the streaming Bayesian updating.

$$\lambda | < c_1 \ldots c_b > \sim \Gamma(\mu, \sigma^2) \tag{24}$$

$$\mu_b = \frac{\mu_{b-1}^2 + b\sigma_0^2 M\bar{c}_b}{\mu_{b-1} + b\sigma_0^2 M} \tag{25}$$

$$\sigma_b^2 = \sigma_0^2 \frac{\mu_{b-1}^2 + b\sigma_0^2 M\bar{c}_b}{\left(\mu_{b-1} + b\sigma_0^2 M\right)^2} = \sigma_0^2 \frac{\mu_b}{\left(\mu_{b-1} + b\sigma_0^2 M\right)} \tag{26}$$

Similar to the previous study, the streaming Bayesian updating continuously updates the posterior expectation with each batch forgetting previously learned information. Moreover, the posterior variance is estimated similar to the traditional Bayesian updating, yet substituting the initial expectation μ_0 term in the denominator of equation 23 with the expectation of the previous posterior μ_{b-1} to enforce the sequential dependencies in the data. By propagating a fixed uncertainty we have able to forget the outdated

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

PeerJ Preprints NOT PEER-REVIEWED

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

information while controlling the posterior uncertainty appropriately by scaling the likelihood with the quantity of the data observed by the models. Accordingly, the proposed Bayesian updating is more suitable to implement black-box inference for the data streams that undergo concept drifts.

Streaming Stochastic Variational Bayes

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), SVI (Hoffman et al., 2013) and lastly BB-SVB, which we derived. To conduct a fair comparison, we have derived black-box inference of both SVI and PVI following an identical approach to the black-box inference of SSVB[4]. We have deliberately omitted SVB (Broderick et al., 2013) from our analysis against SSVB due to the deficiencies in extending the SVB to the black-box inference [5]. In our experiments, we consider BB-SVB as the black-box inference equivalent of SVB.

We conduct two experiments, evaluating the properties of the SSVB and BB-SVB using two supervised non-conjugate probabilistic models: multinomial logistic regression and linear mixed effect model. Both of these models require approximation inference and necessary tools in stream analytics. Moreover, classification models such as multinomial logistic regression can be evaluated extensively in online settings due to the availability of a rich set of datasets collected from real-world applications. Availability of a wide range of conventional online classifiers allows directly appraising the usefulness of SSVB to the practitioners. On the other hand, the linear mixed effect model is a more complex and yet quite useful model for streaming settings. Linear mixed-effect model allows evaluating ability of each approach to accommodate concept drift in both latitudinal and longitudinal data. Furthermore, both of the above models employ Gaussian posteriors, which have a suitable parameterization to easily update the priors with SSVB.

As the first experiment, we conduct an extensive evaluation of the performance of SSVB compared to PVI, SVI and BB-SVB using multinomial logistic regression. The first experiment 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 first experiment. As the second 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 known drift-points, we attempt to generalize the competitive accuracy observed with SSVB with the previous experiment to a wide-range of probabilistic inference tasks. Furthermore, we emphasize the shortcomings with PVI, SVI and BB-SVB with multi-pass updates as opposed to SSVB.

Experiment 1 - Linear Classification

We implement the multinomial logistic regression by simply optimizing the streaming variational objectives that is written considering its standard probabilistic notation [6]. We optimize the objectives following *reparameterization VI* while sampling only once to compute the gradients of the random variables. We employ standard Gaussian distribution as the priors to those random variables (these will be initial priors

[4] derivation in Appendix 2

[5] as demonstrated in Appendix 3

[6] see Appendix 4

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

PeerJ Preprints NOT PEER-REVIEWED

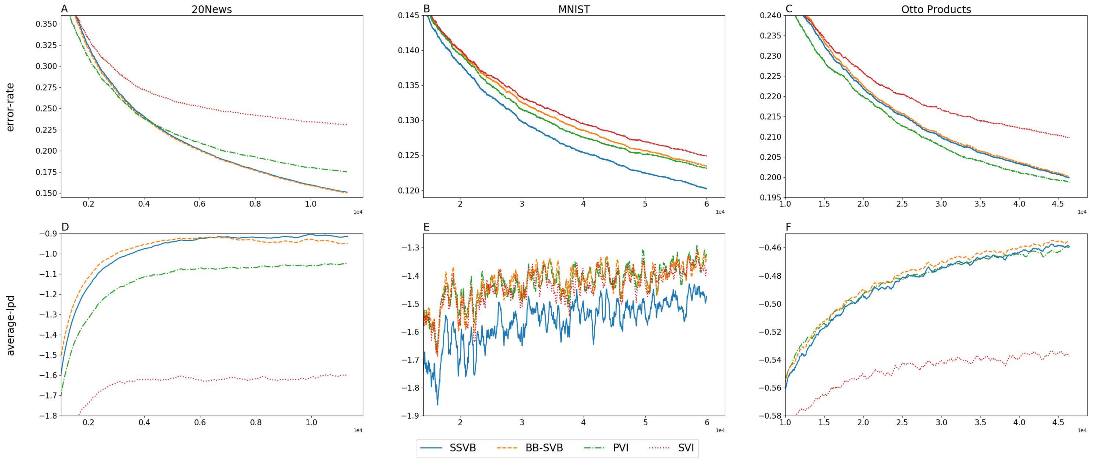

Figure 1. Error rate and average log-predictive density for multiclass classification

	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[7]

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*) is a text classification task with high dimensional sparse features and the other two (*MNIST* and *Otto product*) 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 parameter α, which we tuned using the first 10% of the full dataset minimizing the error. 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 (aka average log-likelihood) and error rate (i.e. ratio between the number of incorrect predictions and the total predictions) 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 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 the all permutations [8], and figure 1 illustrates the convergence of the error rate and average lpd w.r.t the number

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

[8] final f1-scores are presented in appendix 5

PeerJ Preprints NOT PEER-REVIEWED

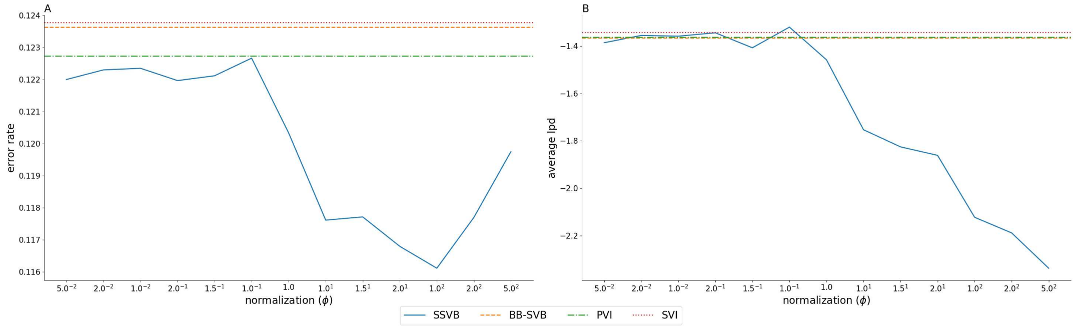

Figure 2. Error rate and average-lpd with different normalization factors for S_b

of samples observed.

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 performance for all three datasets.

Moreover, according to figure 1, 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, each approach has undergone frequent fluctuations in log-predictive density with *MNIST*, which appears to be 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 2 presents the final error-rate and the average lpd w.r.t the different ϕ employed by S_b during our analysis. The horizontal lines are corresponding to the final average lpd for the rest of the approaches. Since the average log-predictive densities exhibit sudden fluctuations with *MNIST* dataset as already seen with figure 1, we have considered the mean of average lpd measured 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 a significantly lower error rate compared to the other inference approaches. However, the average log-predictive density of SSVB is considerably lower than that of the PVI, SVI, and BB-SVB for those cases. For the rest of the cases, SSVB exhibits either 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. It is important to maintain adequate regularization to achieve sufficient robustness when handling sudden changes due to noisy labels (Crammer et al., 2009). Therefore, setting ϕ to a value greater than 1.0 result in higher accuracy due to additional regularization employed to the posterior means as opposed to the usual scaling function S_b. 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.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

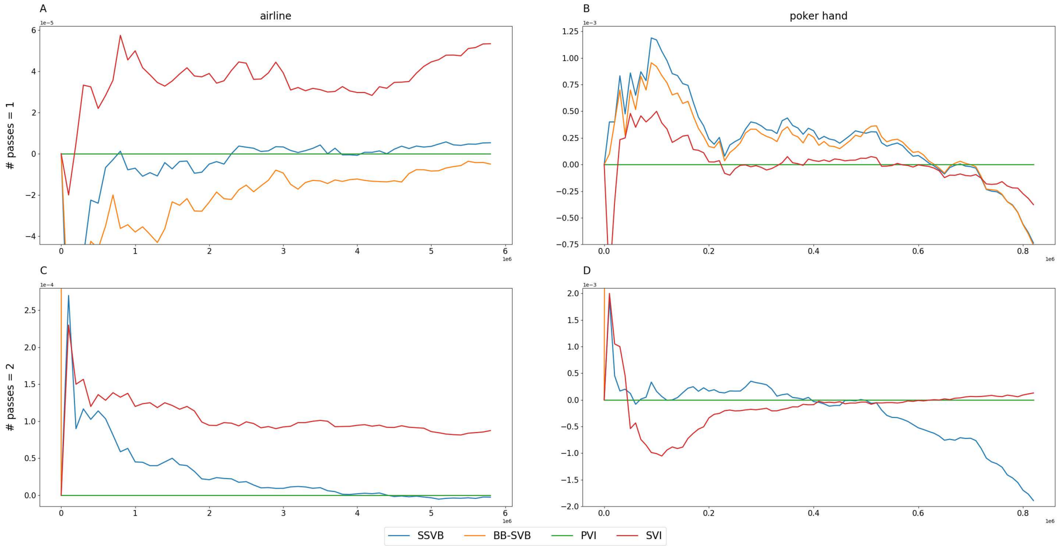

Figure 3. Classification error rate considering PVI as the ground accuracy

	# passes = 1		# passes = 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

Phase 2 - Classification with Real-World Data Streams

We extend our experiments with two massive real-world data streams: *airline* and *poker-hand* datasets. Unlike the three datasets considered in the previous section, *airline* and *poker-hand* datasets are extracted 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 rate. Table 3 presents the final error rates observed. The '# passes' in table 3 indicates the number of updates performed using each data-points (i.e. single-pass vs multi-pass updates). Moreover, figure 3 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

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

PeerJ Preprints NOT PEER-REVIEWED

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 overconfident posteriors. Nevertheless, BB-SVB has achieved the best overall accuracy. It should be noticed that under single-pass updates BB-SVB completely ignores the KL-divergence term in the variational objective, thus diminishing the resistance to the changes due to overconfident 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 overconfident priors, which restrains BB-SVB from accommodating the drifting patterns in the data. On the other hand, SSVB is not affected by overconfident 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. 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 overconfident 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 compared to the technique such as SSVB that does not exploit such optimization. However, the optimal α may expire eventually once 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, α being estimated is sensitive to properties of the data-points (e.g. the number of data points and drifting patterns) that are used optimize α. Therefore, PVI may require re-estimating 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 a crucial factor 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 4 and 5 illustrate the estimated means and standard-deviations using *poker* dataset for some selected coefficients under single-pass and multi-pass updates, respectively.

As expected, BB-SVB leads to overconfident posteriors, resulting in near-zero variance for both cases considered. Especially in figure 5, 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. As a result,

PeerJ Preprints NOT PEER-REVIEWED

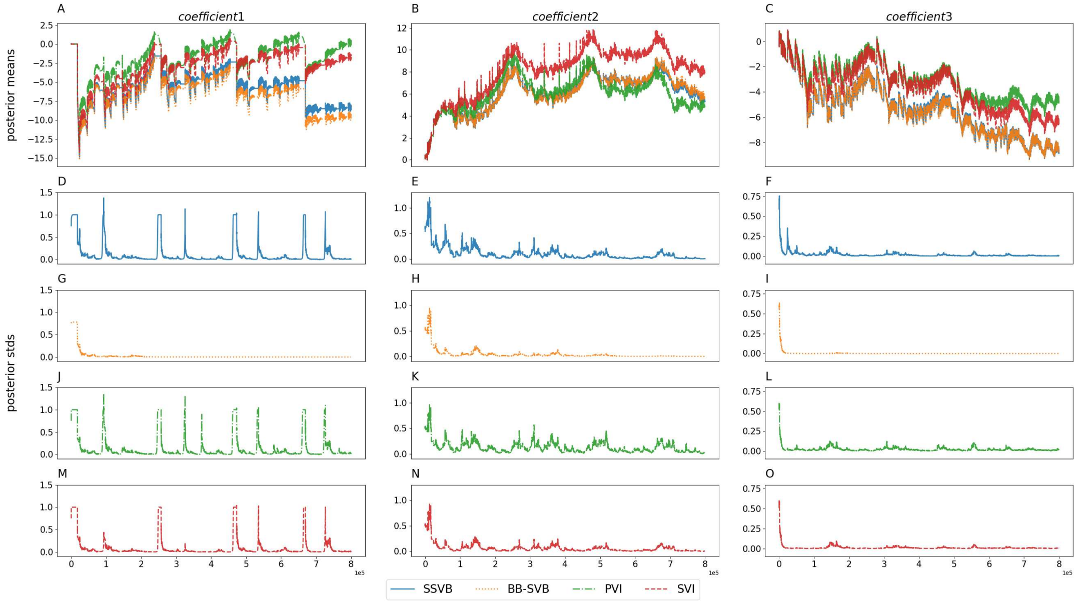

Figure 4. Mean and Std estimated by each approach from *poker* dataset under single-pass updates

BB-SVB appears to demonstrate high resistance against the changes to the posterior means in figure 5. However, BB-SVB seems to estimate the mean as expected under the relaxed constraints with single-pass updates, where it is equivalent to maximization the likelihood of the probabilistic model. We still do not recommend using BB-SVB as an online inference technique even with single-pass updates, since it fails to indicate the drifting patterns by adjusting the posterior uncertainty.

Moreover, PVI also seems to underestimate the posterior uncertainty in contrast to both SSVB and SVI under multi-pass updates. In figure 5, PVI neglects 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. We believe this is due to the inability of PVI to dynamically control the updates to the posterior variance, distinguishing multi-pass gradient steps from updates due to new batches. On the other hand, modulating the posterior variance by fine-tuning α to optimize the error rate or average log-predictive density violates the Bayesian assumptions. Bayesian does not suggest adjusting the posterior variance to achieve a better predictive accuracy; variance of the posterior is considered as a measure of the posterior uncertainty.

SSVB seems to adjust the posterior uncertainty under drifting patterns considerably better than the rest of the approaches. The posterior variance estimated by SSVB does not shrink with time similar to BB-SVB, nor shrink due to multiple passes analogous to PVI as evidence by figures 4 and 5. Instead, SSVB maintains higher posterior uncertainty under multi-pass updates avoiding overfitting. Moreover, SSVB scales the likelihood term following Bayesian assumptions. Therefore, the estimated posterior densities by applying SSVB can be interpreted almost Bayesian.

Phase 4 - Comparison with Other Single Pass Classifiers

We have already established the superiority of SSVB 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

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

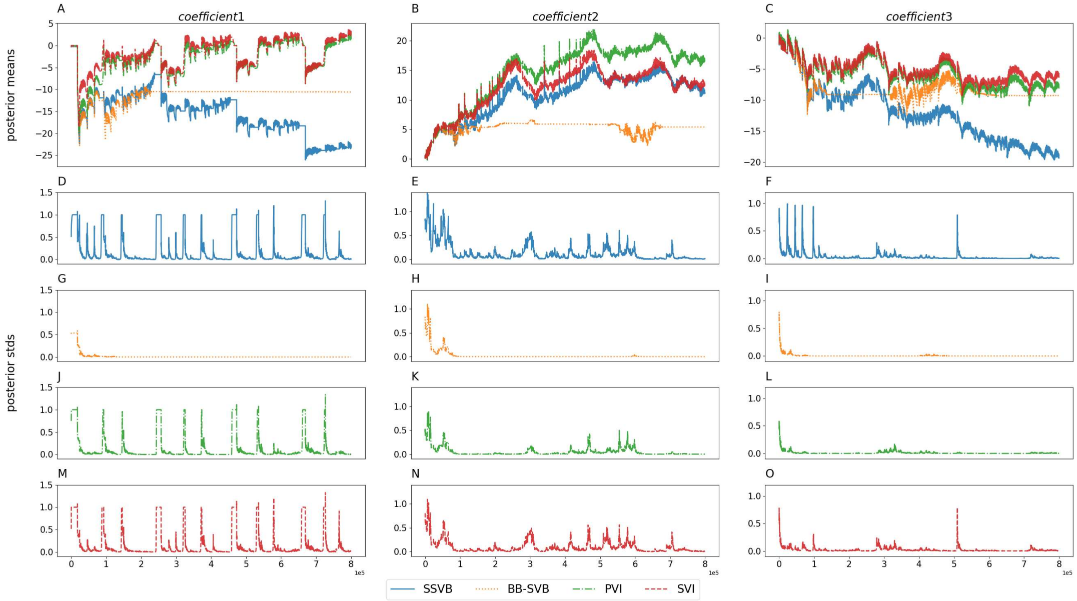

Figure 5. Mean and Std estimated by each approach from *poker* dataset under multi-pass updates

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, thereby 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 2 - Linear Mixed Effects Regression

We use an artificially generated data stream to appraise the performance of the LME model updated based on each objective. We generate a standard mixed effect stream with 100 dimensions (D) and 1000 subjects (C) [9]. We introduce random drifts to both fixed and random effects simultaneously simulating a more realistic data stream. Typically, it is difficult to identify a single holdout set for a data stream that evolves over time. A holdout set selected at a particular instance will expire after the 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

[9] following equation 8 in Appendix 6

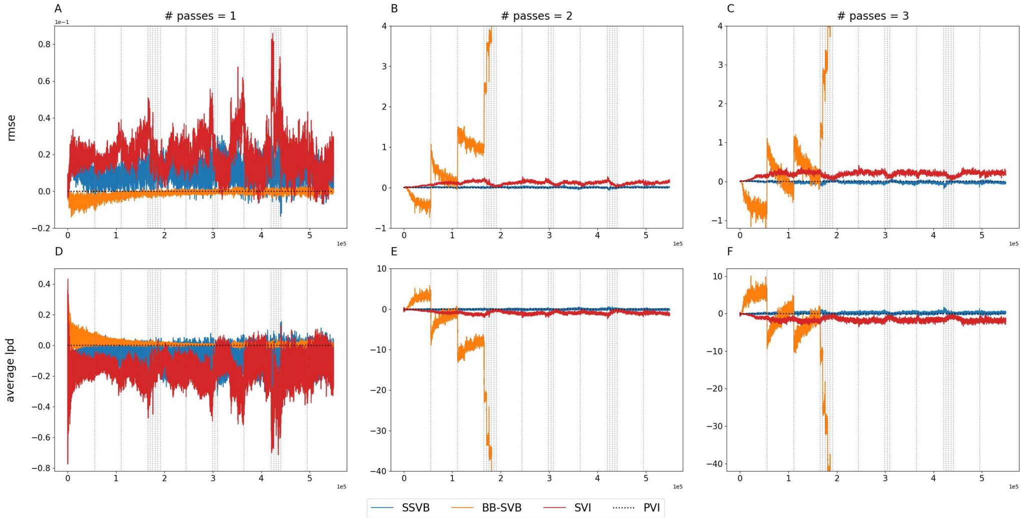

Figure 6. RMSE and lpd for LME models

	# passes	SSVB	BB-SVB	PVI	SVI
rmse	1	7.5941 ± 0.4207	7.5823 ± 0.4196	7.5835 ± 0.4202	7.6039 ± 0.4192
	2	7.8300 ± 0.2137	13.4069 ± 3.7529	7.8219 ± 0.2126	7.9322 ± 0.2041
	3	8.1289 ± 0.2122	12.4684 ± 3.0807	8.1486 ± 0.2105	8.3401 ± 0.2069
	10	9.8371 ± 0.3611	15.2848 ± 4.6595	10.2216 ± 0.3768	10.6604 ± 0.3685
mae	1	5.8538 ± 0.3559	5.8438 ± 0.3550	5.8447 ± 0.3557	5.8627 ± 0.3548
	2	6.0604 ± 0.1749	10.6181 ± 3.0258	6.0537 ± 0.1738	6.1473 ± 0.1664
	3	6.3158 ± 0.1696	9.8638 ± 2.4874	6.3331 ± 0.1677	6.4908 ± 0.1649
	10	7.7456 ± 0.2926	12.1282 ± 3.7428	8.0592 ± 0.3030	8.4219 ± 0.2971
lpd	1	-34.6041 ± 59.3907	-34.5125 ± 59.3824	-34.5227 ± 59.3889	-34.6779 ± 59.3912
	2	-33.9970 ± 40.2957	-99.6289 ± 60.0246	-33.9330 ± 40.2891	-34.7930 ± 40.2466
	3	-35.5450 ± 31.6466	-84.5065 ± 46.6077	-35.7020 ± 31.6277	-37.2687 ± 31.5594
	10	-49.8064 ± 14.1731	-128.1795 ± 67.7404	-53.6353 ± 14.0649	-58.1863 ± 14.1110

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

during the previous analysis. One could argue that single-pass updates are insufficient to estimate the intermediate posteriors with SSVB. Therefore, in addition to the single-pass updates, this experiment is also planned to further investigate the effect of multi-pass updates by conducting 2, 3 and 10 passes per each data-point. 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 6 illustrates the convergence of RMSE and average lpd of SSVB, BB-SVB, and SVI as against PVI. We intentionally avoid absolute error values and have omitted the latter error values of BB-SVB in figure 6 to improve the visibility of the plots. The dotted regions indicate the simulated concept drifts. The mean and the standard deviation of the absolute error values are presented in table 4, as the overall performance metrics. In figure 6 and table 4, '# passes' denotes the number of passes carried out during each update.

All 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. For single-pass updates, SSVB, BB-SVB, PVI, and SVI show comparable accuracy. Interestingly, BB-SVB has slightly outperformed each approach avoiding overconfident posteriors because BB-SVB deviates from the traditional Bayesian updating during single-pass updates. Moreover, PVI has achieved marginally superior accuracy compared to SSVB, although such negligible gain in accuracy against SSVB does not justify the exhausting analysis carried out with PVI to determine a suitable value for α.

Similar to the observations from the previous experiment, multi-pass updates have not improved the

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

models. In fact, such updates have degraded the accuracy of models due to overfitted posterior densities. Hence, it is safe to conclude that single-pass updates are sufficient to approximate the intermediate posteriors at-least for the experimented scenarios. However, the accuracy of SSVB is less affected compared to the rest of the techniques when increasing the number of passes. Therefore, it seems that PVI and SVI are more prone to overfitting compared to SSVB. Unlike SSVB, PVI and SVI cannot distinguish between observing a new data-point and multiple passes using the same data-point with the fixed scale (α or N) employed to the likelihood. On the other hand, BB-SVB appears to fail drastically at each drift point (see figure 6) with multi-pass updates. This is solely due to the discussed shortcomings with original Bayesian updating to the online settings.

Therefore, SSVB is a more suitable candidate for online inference compared to BB-SVB, PVI, and SVI. SSVB exhibits superior or comparable performance as against the rest of the approaches without any additional analysis to tune a hyperparameter than controls the posterior variance. Unlike BB-SVB, SSVB avoids overconfident posteriors and less prone to overfitting even compared to PVI and SVI.

Furthermore, we evaluate the implemented LME model against two conventional online regression models, PA regressor and SGD regressor. Even though these models are not designed to handle random effects, we conduct this analysis to emphasize the importance of the proposed LME to practitioners. With PA regressor, we observe an RMSE and an MAE of respectively 28.8775 ± 50.0463 and 24.2292 ± 43.4287, which is a significantly larger error compared to the error of the LME optimized using any of the inference approaches. On the other hand, SGD fails to converge even to a local optimum, illustrating the complexity of the task. Therefore, LME is an essential tool to handle both fixed and random effects with data stream, and SSVB facilitates black-box inference to efficiently develop and evaluate such predictive models in online settings.

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.

Theis and Hoffman (2015) apply SVI to streaming settings by accumulating incoming data points into a database then uniformly sampling from this database to optimize the variational objective. Their approach eliminates the need for knowing N beforehand with SVI at the cost of additional storage capacity. Nevertheless, this approach is infeasible in true online settings, where a storage complexity of $\mathcal{O}(1)$ is assumed to learn from continuously evolving infinite data streams. Theis and Hoffman (2015) also proposed trust regions to update parameters mitigating the local optima found with natural gradients. This innovation can be easily integrated into our approach.

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 as opposed to 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 overconfident posteriors.

Assumed-Density Filtering (ADF) and Expectation Propagate (EP) (Maybeck, 1979; Opper, 1998; Minka, 2001a,b) have fused Bayesian updating and approximate inference taking a different approach to Broderick et al. (2013); Ghahramani and Attias (2000); Honkela and Valpola (2003); Nguyen et al. (2017). These techniques compute the exact posterior considering a single data-point and approximate the posterior to the same family of distribution as the priors. The approximated distributions will be the priors to next posterior estimation. However, unlike the variational methods discussed above, ADF and EP cannot be applied if the true posteriors are intractable. Even though ADF is sensitive to the sequence of data due to the single-pass updates of approximated posteriors (Minka, 2001a), these approaches are not guaranteed to handle data streams that change over time. Conceptually, they still employ traditional Bayesian updating and are susceptible to the phenomenon of overconfident posterior.

The proposed technique extends 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 additional 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 introduced two modifications to the traditional Bayesian updating framework deriving a novel streaming Bayesian updating approach that is capable of efficiently handling data streams with concept drift. We then derived a black-box inference technique for online settings: “Streaming and Stochastic Variational Bayes” (SSVB), by adopting reparameterization VI to approximate the intermediate posteriors with proposed Bayesian updating framework. Unlike the existing online inference approach, SSVB does not suffer from overconfident posterior nor require additional hyperparameters to control the posterior variance compared to its offline counterparts.

We appraised the performance of SSVB against BB-SVB, and two existing online inference approaches PVI and SVI with two essential models to the online learning: multinomial logistic regression and linear-mixed effects model. SSVB demonstrated either superior or comparable performance as opposed to the current state-of-the-art, PVI. Furthermore, SSVB demonstrated a significant gain in the accuracy for online classification compared to the conventional state-of-the-art conventional online classifiers such as AROW, PA and SGD. Accordingly, SSVB can be considered as a much effective online inference framework in contrast to PVI, SVI, and BB-SVB. Moreover, practitioners and researchers can easily adopt SSVB to efficiently build a wide range of models to handle endless streaming data with concept drifts.

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.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004). *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition.

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., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233.

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.

Maybeck, P. S. (1979). Stochastic models, estimation, and control.

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

Minka, T. P. (2001a). Expectation propagation for approximate bayesian inference. In *Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence*, UAI '01, pages 362–369, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Minka, T. P. (2001b). *A Family of Algorithms for Approximate Bayesian Inference*. PhD thesis, Cambridge, MA, USA. AAI0803033.

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

Opper, M. (1998). A bayesian approach to online learning.

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*.

Theis, L. and Hoffman, M. D. (2015). A trust-region method for stochastic variational inference with applications to streaming data. In *ICML*.

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

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019

PeerJ Preprints NOT PEER-REVIEWED

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*.

PeerJ Preprints | https://doi.org/10.7287/peerj.preprints.27790v2 | CC BY 4.0 Open Access | rec: 19 Nov 2019, publ: 19 Nov 2019