Detecting and Adapting to Irregular Distribution Shifts in Bayesian Online Learning

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of online learning in the presence of distribution shifts that occur at an unknown rate and of unknown intensity. We derive a new Bayesian online inference approach to simultaneously infer these distribution shifts and adapt the model to the detected changes by integrating ideas from change point detection, switching dynamical systems, and Bayesian online learning. Using a binary 'change variable,' we construct an informative prior such that–if a change is detected–the model partially erases the information of past model updates by tempering to facilitate adaptation to the new data distribution. Furthermore, the approach uses beam search to track multiple change-point hypotheses and selects the most probable one in hindsight. Our proposed method is model-agnostic, applicable in both supervised and unsupervised learning settings, suitable for an environment of concept drifts or covariate drifts, and yields improvements over state-of-the-art Bayesian online learning approaches.

1 Introduction

Deployed machine learning systems are often faced with the problem of distribution shift, where the new data that the model processes is systematically different from the data the system was trained on Zech et al. [2018], Ovadia et al. [2019]. Furthermore, a shift can happen anytime after deployment, unbeknownst to the users, with dramatic consequences for systems such as self-driving cars, robots, and financial trading algorithms, among many other examples.

Updating a deployed model on new, representative data can help mitigate these issues and improve general performance in most cases. This task is commonly referred to as *online* or *incremental learning*. Such online learning algorithms face a tradeoff between remembering and adapting. If they adapt too fast, their performance will suffer since adaptation usually implies that the model loses memory of previously encountered training data (which may still be relevant to future predictions). On the other hand, if a model remembers too much, it typically has problems adapting to new data distributions due to its finite capacity.

The tradeoff between adapting and remembering can be elegantly formalized in a Bayesian online framework, where a prior distribution is used to keep track of previously learned parameter estimates and their confidences. For instance, variational continual learning (VCL) [Nguyen et al., 2018] is a popular framework that uses a model's previous posterior distribution as the prior for new data. However, the assumption of such continual learning setups is usually that the data distribution is stationary and not subject to change, in which case adaptation is not an issue.

This paper proposes a new Bayesian online learning framework suitable for non-stationary data distributions. It is based on two assumptions: (i) distribution shifts occur irregularly and have to be inferred from the data, and (ii) the model requires not only a good mechanism to aggregate data but

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

also the ability to partially forget information that has become obsolete. To solve both problems, we still use a Bayesian framework for online learning (i.e., letting a previous posterior distribution inform the next prior); however, before combining the previously learned posterior with new data evidence, we introduce an intermediate step. This step allows the model to either broaden the previous posterior's variance to reduce the model's confidence and provide more "room" for new information or remain in the same state (i.e., retain the unchanged, previous posterior as the new prior).

We propose a mechanism for enabling this decision by introducing a discrete "change variable" that indicated the model's best estimate of whether the data in the new batch is compatible with the previous data distribution or not; the outcome then informs the Bayesian prior at the next time step. We further augment the scheme by performing beam search on the change variable. This way, we are integrating change detection and Bayesian online learning in a common framework.

We test our framework on a variety of real-world datasets that show covariate drift, including basketball player trajectories, malware characteristics, sensor data, and electricity prices. We also study sequential versions of SVHN and CIFAR-10, where we simulate distribution shifts in terms of image rotations. Finally, we study word embedding dynamics in an unsupervised learning approach. Our approach leads to a more compact and interpretable latent structure and significantly improved performance in the supervised experiments. Furthermore, it is highly scalable; we demonstrate it on models with hundreds of thousands of parameters and ten thousand feature dimensions.

Our paper is structured as follows: we review related work in Section 2, introduce our methods in Section 3, report our experiments in Section 4, and draw conclusions in Section 5.

2 Related Work

Our paper connects to Bayesian online learning, change detection, and switching dynamical systems.

Bayesian Online and Continual Learning There is a rich literature on Bayesian and continual learning. The main challenge in streaming setups is to reduce the impact of old data on the model which can be done by exponentially decaying old samples [Honkela and Valpola, 2003, Sato, 2001, Graepel et al., 2010] or re-weighting them [McInerney et al., 2015, Theis and Hoffman, 2015]. An alternative approach is to adapt the model posterior between time steps, such as tempering it at a constant rate to accommodate new information [Kulhavỳ and Zarrop, 1993, Kurle et al., 2020]. In contrast, *continual learning* typically assumes a stationary data distribution and simply uses the old posterior as the new prior. A scalable such scheme based on variational inference was proposed by [Nguyen et al., 2018] which was extended my many authors [Farquhar and Gal, 2018, Schwarz et al., 2018]. A related concept is Elastic Weight Consolidation [Kirkpatrick et al., 2017], where new model parameters are regularized towards old ones.

All these approaches need to make assumptions on the expected frequency and strengh of change which are hard-coded in the model parameters (e.g., exponential decay rates, re-weighting terms, prior strengths, or temperature choices). Our approach, in contrast, detects change based on a discrete variable and makes no assumption on its frequency. Other approaches do assume situations where data arrive in irregular time intervals, but are still concerned with static data distributions [Titsias et al., 2019, Lee et al., 2020, Rao et al., 2019].

Change Point Models There is a rich history of models for change detection. A popular class of change point models includes "Product Partition Models" [Barry and Hartigan, 1992] which assume independence of the data distribution across segments. In this regime, Fearnhead [2005] proposed change detection in the context of regression and generalized it to online inference [Fearnhead and Liu, 2007]; Adams and MacKay [2007] described a Bayesian *online* change point detection scheme (BOCD) based on conditional conjugacy assumptions for one-dimensional sequences. Other work generalized change detection algorithms to multivariate time series [Xuan and Murphy, 2007, Xie et al., 2012] and non-conjugate Bayesian inference [Saatçi et al., 2010, Knoblauch and Damoulas, 2018, Turner et al., 2013, Knoblauch et al., 2018].

Our approach relies on jointly inferring changes in the data distribution while carrying out Bayesian parameter updates for adaptation. To this end, we detect change in the high-dimensional space of model (e.g., neural network) parameters, as opposed to directly in the data space. Furthermore, a detected change only *partially* resets the model parameters, as opposed to triggering a complete reset.

Titsias et al. [2020] proposed change detection to detect distribution shifts in sequences based on low-dimensional summary statistics such as a loss function; however, the proposed framework does not use an informative prior but requires complete retraining.

Switching Linear Dynamical Systems Since our approach integrates a discrete change variable, it loosely connects to switching linear dynamic systems. Linderman et al. [2017] considered *recurrent* switching linear dynamical systems, relying on Bayesian conjugacy and closed-form message passing updates. Becker-Ehmck et al. [2019] proposed a variational Bayes filtering framework for switching linear dynamical systems. Murphy [2012] and Barber [2012] developed an inference method using a Gaussian sum filter. Differently, we focus on inferring the full history of discrete latent variable values instead of just the most recent one.

Bracegirdle and Barber [2011] introduce a *reset* variable that sets the continuous latent variable to an unconditional prior. It is similar to our work, but relying on low-dimensional, tractable models. Our tempering prior can be seen as a partial reset, augmented with beam search. We also extend the scope of switching dynamical systems by integrating them into a supervised learning framework.

3 Methods

Overview Section 3.1 introduces the setup and the novel model structure under consideration. Section 3.2 introduces an exact inference scheme based on beam search. Finally, we introduce the variational inference extension for intractable likelihood models in Section 3.3.

3.1 Problem Assumptions and Structure

We consider a stream of data that arrives in batches $\mathbf{x}_t$ at discrete times t.[1] For supervised setups, we consider pairs of features and targets $(\mathbf{x}_t, \mathbf{y}_t)$, where the task is to model $p(\mathbf{y}_t|\mathbf{x}_t)$. An example model could be a Bayesian neural network, and the parameters $\mathbf{z}_t$ could be the network weights. For notational simplicity we focus on the unsupervised case, where the task is to model $p(\mathbf{x}_t)$ using a model $p(\mathbf{x}_t|\mathbf{z}_t)$ with parameters $\mathbf{z}_t$ that we would like to tune to each new batch.[2] We then measure the prediction error either on one-step-ahead samples or a held-out test set.

We furthermore assume that while the $\mathbf{x}_t$ are i.i.d. within batches, they are not necessarily i.i.d. across batches as they come from a time-varying distribution $p_t(\mathbf{x}_t)$ (or $p_t(\mathbf{x}_t, \mathbf{y}_t)$ in the supervised cases) which is subject to distribution shifts. We do not assume whether these distribution shifts occur instantaneously or gradually. The challenge is to optimally adapt the parameters $\mathbf{z}_t$ to each new batch while borrowing statistical strength from previous batches.

As follows, we will construct a Bayesian online learning scheme that accounts for changes in the data distribution. For every new batch of data, our scheme tests whether the new batch is compatible with the old data distribution, or more plausible under the assumption of a change. To this end, we employ a binary "change variable" s_t, with $s_t = 0$ for no detected change, and with $s_t = 1$ in case a change is detected. Our model's joint distribution factorizes as follows:

$$p(\mathbf{x}_{1:T}, \mathbf{z}_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(\mathbf{x}_t|\mathbf{z}_t)p(\mathbf{z}_t|s_t, \boldsymbol{\tau}_t)p(s_t). \tag{1}$$

We assumed a factorized Bernoulli prior $\prod_t p(s_t)$ over the change variable: an assumption that will simplify the inference, but which can be relaxed. As a result, our model is fully-factorized over time, however, the model can still capture temporal dependencies through the informative prior $p(\mathbf{z}_t|\boldsymbol{\tau}_t, s_t)$. Temporal information enters this prior through certain *sufficient statistics* $\boldsymbol{\tau}_t$ that depend on properties on the previous time-step's approximate posterior.

In more detail, $\boldsymbol{\tau}_t$ is a *functional* on the previous time step's approximate posterior, $\boldsymbol{\tau}_t = \mathcal{F}[p(\mathbf{z}_{t-1}|\mathbf{x}_{1:t-1}, s_{1:t-1})]$. Throughout this paper, we will use a specific form of $\boldsymbol{\tau}_t$, namely capturing

[1] In an extreme case, it is possible for a batch to include only a single data point.

[2] In supervised setups, we consider a conditional model $p(\mathbf{y}_t|\mathbf{z}_t, \mathbf{x}_t)$ with features $\mathbf{x}_t$ and targets $\mathbf{y}_t$.

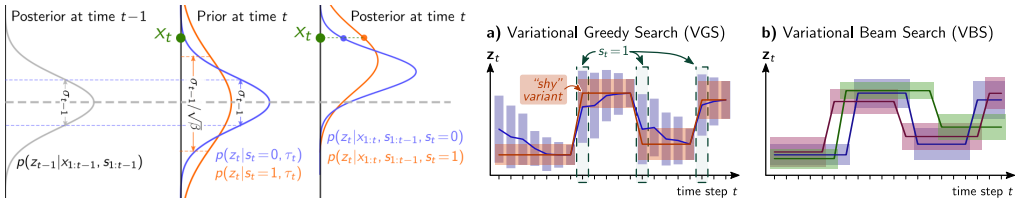

Figure 1: **Left plot:** A single inference step for a Gaussian posterior in 1D. Starting from the previous posterior (left), we consider both its broadened and un-broadened version (middle) and finally compare data point $\mathbf{x}_t$'s marginal likelihoods of both hypotheses (right) to determine the more likely one. **Right plot:** Sparse inference via greedy search (**a**) and variational beam search (**b**). Right: solid lines indicate fitted mean μ_t over time steps t with boxes representing $\pm 1\sigma$ error bars. Please see more information about the pictured "shy" variant in Supplement C.

the previous posterior's mean and variance.[3] More formally,

$$\tau_t \equiv \{\mathbb{E}, \mathrm{Var}\}[\mathbf{z}_{t-1}|\mathbf{x}_{1:t-1}, s_{1:t-1}] \equiv \{\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}\}. \tag{2}$$

Based on this choice, we define the conditional prior as follows:

$$p(\mathbf{z}_t|s_t, \boldsymbol{\tau}_t) = \begin{cases} \mathcal{N}(\mathbf{z}_t; \boldsymbol{\mu}_{t-1}, \Sigma_{t-1}) & \text{for } s_t = 0 \\ \mathcal{N}(\mathbf{z}_t; \boldsymbol{\mu}_{t-1}, \beta^{-1}\Sigma_{t-1}) & \text{for } s_t = 1 \end{cases} \tag{3}$$

where $\beta > 0$ is referred to as an *inverse temperature variable*. If no change is detected (i.e., $s_t = 0$), our prior becomes a Gaussian distribution centered around the previous posterior's mean and variance. As long as no changes are detected from batch to batch, the variance of the prior will shrink as the model gets more confident, and statistical strength is transmitted from one batch to the next. If a change is detected ($s_t = 1$), an overconfident prior would be harmful for learning as the model needs to adapt to the new data distribution. We therefore weaken the prior through *tempering*. Given a temperature β, we raise the previous posterior's Gaussian approximation to the power β, renormalize it, and then use it as a prior for the current time step.

The process of tempering the Gaussian posterior approximation can be understood as removing equal amounts of information from all latent coordinates. To see this, consider a diagonal Gaussian posterior with variance σ_i^2 in dimension z_i[4]. Then tempering removes an equal amount of information in each dimension, $H_i = \frac{1}{2}\log(2\pi e\sigma_i^2/\beta) = \frac{1}{2}\log(2\pi e\sigma_i^2) - \frac{1}{2}\log\beta$, erasing learned information to free-up model capacity to adjust to the new data distribution. See Supplement B for more details.

Connection to Sequence Modeling Our model assumptions have a resemblance to time series modeling: if we replaced $\boldsymbol{\tau}_t$ with $\mathbf{z}_{t-1}$, we would condition on previous latent states rather than posterior summary statistics. In contrast, our model still factorizes over time and therefore makes weaker assumptions on temporal continuity. Rather than imposing temporal continuity on a data *instance* level, we instead assume temporal continuity at a *distribution* level.

Connection to changepoint modeling. We also share similar assumptions with the changepoint literature [Barry and Hartigan, 1992]. However, in most cases, these models don't assume an informative prior, effectively not taking into account any sufficient statistics $\boldsymbol{\tau}_t$. This forces these models to re-learn model parameters from scratch after a detected change, whereas our approach allows for some transfer of information before and after the distribution shift.

3.2 Exact Inference

Before presenting a scalable variational inference scheme in our model, we describe an exact inference scheme when everything is tractable, i.e., in linear Gaussian models.

According to our assumptions, the distribution shifts occur at discrete times and are unobserved. Therefore, both $\mathbf{z}_t$ and $s_{1:t}$ have to be inferred from data. Suppose we moved from time $t-1$ to

[3] In later sections, we will use a Gaussian approximation to the posterior, but here it is enough to assume that these quantities are computable.

[4] People can always transform $\mathbf{z} \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ into a diagonal form $P^\top \mathbf{z}$ where P contains eigenvectors of Σ.

time t and obtained the informative priors in Eq. 3. We first consider the conditional posterior of $\mathbf{z}_t$ given $s_{1:t}$. According to the model assumption (Eq. 1), given s_t and the informative prior for $\mathbf{z}_t$, the posterior of $\mathbf{z}_t$ is

$$p(\mathbf{z}_t|\mathbf{x}_{1:t}, s_{1:t}) \propto p(\mathbf{x}_t|\mathbf{z}_t)p(\mathbf{z}_t|s_t, \boldsymbol{\tau}_t). \tag{4}$$

This process is illustrated in Fig. 1 (left) for a 1D case, where we show how a new observation changes the two priors. We then move to the inference over $s_{1:t}$, for which we define the recursion

$$p(s_{1:t}|\mathbf{x}_{1:t}) = p(s_t|s_{1:t-1}, \mathbf{x}_{1:t})p(s_{1:t-1}|\mathbf{x}_{1:t-1}). \tag{5}$$

Again, the conditional posterior $p(s_t|s_{1:t-1}, \mathbf{x}_{1:t})$ is of interest. We can obtain it through the marginal likelihood over data, for which we define

$$p(\mathbf{x}_t|s_{1:t}, \mathbf{x}_{1:t-1}) = \int p(\mathbf{x}_t|\mathbf{z}_t)p(\mathbf{z}_t|s_t, \boldsymbol{\tau}_t)d\mathbf{z}_t, \tag{6}$$

where the dependence on $\mathbf{x}_{1:t-1}$ comes from the summary statistics $\boldsymbol{\tau}_t$. Given Eq. 6, the exact posterior over $s_t|s_{1:t-1}$ is again a Bernoulli $p(s_t|s_{1:t-1}, \mathbf{x}_{1:t}) = \mathrm{Bern}(s_t; m)$ with parameter

$$m = \sigma\left(\log \frac{p(\mathbf{x}_t|s_t{=}1, s_{1:t-1}, \mathbf{x}_{1:t-1})p(s_t{=}1)}{p(\mathbf{x}_t|s_t{=}0, s_{1:t-1}, \mathbf{x}_{1:t-1})p(s_t{=}0)}\right) = \sigma\left(\log \frac{p(\mathbf{x}_t|s_t{=}1, s_{1:t-1}, \mathbf{x}_{1:t-1})}{p(\mathbf{x}_t|s_t{=}0, s_{1:t-1}, \mathbf{x}_{1:t-1})} + \xi_0\right) \tag{7}$$

where σ is the sigmoid function and $\xi_0 = \log p(s_t = 1) - \log p(s_t = 0)$ are the log-odds of the prior $p(s_t)$. This specifies the posterior over $s_t|s_{1:t-1}$. Plugging in Eq. 5 completes the recursion.

Eq. 7 has a simple interpretation as a likelihood ratio test: a change is more or less likely depending on whether or not the observations $\mathbf{x}_t$ are better explained under the assumption of the tempered prior distribution in Eq. 3, in other words, a partial reset of previously learned information.

We thereby finish the recursion Eq. 5 by plugging in $p(s_t|s_{1:t-1}, \mathbf{x}_{1:t})$ parameterized by Eq. 7. Finally, combining $p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})$ and $p(s_{1:t}|\mathbf{x}_{1:t})$, we can achieve any marginal distribution of $\mathbf{z}_t$, such as $p(\mathbf{z}_t|\mathbf{x}_{1:t}) = \sum_{s_{1:t}} p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})p(s_{1:t}|\mathbf{x}_{1:t})$.

Exponential Branching We note that while we had originally started with a posterior $p(\mathbf{z}_{t-1}|\mathbf{x}_{1:t-1}, s_{1:t-1})$ at the previous time, our inference scheme resulted in $p(\mathbf{z}_t|\mathbf{x}_{1:t}, s_{1:t-1})$ being a mixture of two components as it branches over two possible states.[5] When we iterate, we encounter an exponential branching of possibilities, or *hypotheses* $s_{1:t}$ over possible sequences of regime shifts. To still be able to carry out the filtering scheme efficiently, we need a truncation scheme, e.g., approximate the bimodal marginal distribution by a unimodal one.

Greedy Search In the simplest "greedy" setup, we train the model in an online fashion by iterating over time steps t. For each t, we update a *truncated* distribution via the following three steps:

1. Compute the conditional prior $p(\mathbf{z}_t|s_t, \boldsymbol{\tau}_t)$ (Eq. 3) based on $p(\mathbf{z}_{t-1}|\mathbf{x}_{1:t-1}, s_{1:t-1})$ and infer $p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})$ for both $s_t = 0$ and $s_t = 1$.
2. Compute the binary mixture weights m and $(1{-}m)$ using Eq. 7 and update $p(s_{1:t}|\mathbf{x}_{1:t})$ using Eq. 5.
3. Select $s_{1:t}$ that has larger probability and its corresponding hypothesis $p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})$ (i.e., make a "hard" decision over s_t with a threshold of $\frac{1}{2}$).

The above filtering algorithm iteratively updates the posterior distribution over $\mathbf{z}_t$ each time it observes new data $\mathbf{x}_t$. In the version of greedy search discussed above, the approach decides immediately, i.e., before observing subsequent data points, whether a change in $\mathbf{z}_t$ has occurred or not in step 3. (Please note the decision is irrelevant to history, as opposed to the beam search described below.) Greedy search is illustrated in Fig. 1 (a) where VGS is the variational inference counterpart.

Beam Search A greedy search is prone to missing change points in data sets with a low signal/noise ratio per time step because it cannot accumulate evidence for a change point over a series of time steps. On the other end of the spectrum, the most expressive algorithm would involve assessing all possible sequences of $\mathbf{s}_t \in \{0, 1\}^t$ to determine the most likely set of shifts, and thus the most appropriate $p(\mathbf{z}_t|\mathbf{s}_t, \mathbf{x}_{1:t})$; however, this is not feasible due to scaling in $O(2^t)$ for both compute

[5] See also Fig. 1 in Supplement C.

and memory. The most obvious improvement over greedy search that has the ability to accumulate evidence for a change point is beam search. Rather than deciding greedily whether a change occurred or not at each time step, beam search considers both cases in parallel as it delays the decision of which one is more likely (see Fig. 1 (b) and Fig. 2 (left) for illustration). The algorithm keeps track of a fixed number $K > 1$ of possible sequences of change points. For each sequence, it iteratively updates the posterior distribution as greedy search. At time step t, every potential continuation of the K sequences is considered, with $s_t \in \{0, 1\}$, thus doubling the number of histories of which the algorithm has to keep track of. To keep the computational requirements bounded, beam search thus discards half of the sequences based on an exploration-exploitation trade-off.

As follows, we describe how beam search proceeds from time $t-1$ to time t. Suppose at time $t-1$, we will have a fixed set $\mathbb{B}_{t-1}$ of K different "histories", i.e., a set of possible realizations of the change variables $s_{1:t-1}$ to which the model has assigned high probabilities in previous iterations. These probabilities are given by the posteriors over the change variables $p(\mathbf{s}_{t-1}|\mathbf{x}_{1:t-1})$ (Eq. 5).

Similar to greedy search, for each history $\mathbf{s}_{t-1} \in \mathbb{B}_{t-1}$, we iterate the above step 1 and step 2, resulting in $2K$ possible sequences to choose from. This is pared down to K histories again by choosing the sequences resulting in the highest values of the un-normalized distributions $p(s_{1:t}|\mathbf{x}_{1:t})$ which we then normalize as $p(s_{1:t}|\mathbf{x}_{1:t}) = \frac{p(s_{1:t}|\mathbf{x}_{1:t})}{\sum_{\mathbf{s}_t \in \mathbb{B}_t} p(s_{1:t}|\mathbf{x}_{1:t})}$. Those remaining K sequences form $\mathbb{B}_t$. This concludes the recursion. This search strategy scales linearly in both compute and memory to K.

Beam Search Diversification Empirically, we find that the naive beam search procedure does not reveal its full potential. As commonly encountered in beam search, histories over change points are largely shared among all members of the beam. To encourage diverse beams, we constructed the following simple scheme: Let K be the number of hypotheses in a beam. While transitioning from time $t-1$ to t, every hypothesis splits into two scenarios, one with $s_t = 0$ and one with $s_t = 1$, resulting in $2K$ hypotheses. If two resulting hypotheses only differ in their most recent s_t-value, we say that they come from the same "family." Each member among the $2K$ hypotheses is ranked according to its posterior probability $p(s_{1:t}|\mathbf{x}_{1:t})$ in Eq. 5. In a first step, we discard the bottom $1/3$ of the $2K$ hypotheses, leaving $4/3K$ hypotheses (we always take integer multiples of 3 for K). To truncate the beam size from $4/3K$ down to K, we rank the remaining hypotheses according to their posterior probability and pick the top K ones while *also* ensuring that we pick a member from every remaining family. The diversification scheme ensures that underperforming families can survive, leading to a more diverse set of hypotheses. We found this beam diversification scheme to work robustly across a variety of experiments.

3.3 Variational Inference

Unfortunately, the marginal in Eq. 6 is not always available, leaving Eq. 7 intractable to evaluate. However, we can follow a structured variational inference approach [Wainwright and Jordan, 2008, Hoffman and Blei, 2015, Zhang et al., 2018], defining a joint variational distribution $q(\mathbf{z}_t, s_t|s_{1:t-1}) = q(s_t|s_{1:t-1})q(\mathbf{z}_t|s_{1:t})$ to approximate $p(\mathbf{z}_t, s_t|s_{1:t-1}, \mathbf{x}_{1:t}) = p(s_t|s_{1:t-1}, \mathbf{x}_{1:t})p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})$.

By definition, the conditional distributions $q(\mathbf{z}_t|s_t)$ approximate $p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})$. We thus minimize the KL divergence between $q(\mathbf{z}_t|s_t)$ and $p(\mathbf{z}_t|s_{1:t}, \mathbf{x}_{1:t})$ (Eq. 4) and produce the variational inference

$$q^*(\mathbf{z}_t|s_{1:t}) = \underset{q(\mathbf{z}_t|s_{1:t}) \in Q}{\arg\max} \; \mathcal{L}(q|s_{1:t}), \tag{8}$$

$$\mathcal{L}(q|s_{1:t}) := \mathbb{E}_q[\log p(\mathbf{x}_t|\mathbf{z}_t)] - \mathrm{KL}(q(\mathbf{z}_t|s_{1:t})||p(\mathbf{z}_t|s_t, \boldsymbol{\tau}_t)),$$

where Q is the family of potential approximate posterior distributions (i.e. factorized normal distributions), and we term $\mathcal{L}(q|s_{1:t})$ conditional Evidence LowerBOund (ELBO). One may wonder how the exact inference scheme for s_t is modified in the structured variational inference scenario. In Supplement A, we derive the following closed-form update equation for $q(s_t|s_{1:t-1})$ that bears strong similarities to Eq. 7. Since the conditional ELBO in Eq. 8 is a lower bound to the logarithm of the marginal likelihood in Eq. 6, we can use the former as a proxy of the latter, resulting in the update

$$q^*(s_t|s_{1:t-1}) = \mathrm{Bern}(s_t; m); \quad m = \sigma\left(\tfrac{1}{T}\mathcal{L}(q|s_t = 1) - \tfrac{1}{T}\mathcal{L}(q|s_t = 0) + \xi_0\right). \tag{9}$$

Above, we introduced a parameter $T \geq 1$ (not to be confused with β) to optionally downweigh the data evidence relative to the prior (see Experiments Section 4).

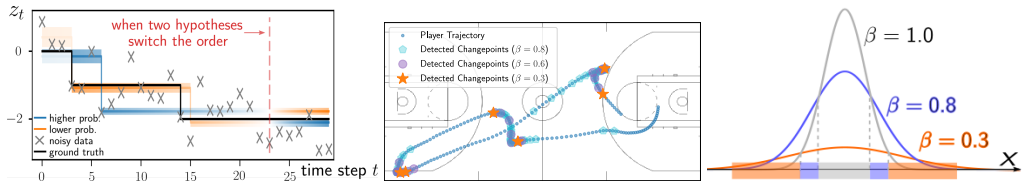

Figure 2: (**Left**) Inferring the mean (black line) of a time-varying data distribution (black samples) with VBS. The initially unlikely hypothesis begins dominating over the other at step 23. (**Middle**) Basketball player tracking: ablation study over β for VBS while fixing other parameters. We used greedy search (K=1) and run the model under different β values. Increasing β leads to more sensitivity to changes in data, leading to more detected changepoints. (**Right**) Small β incurs sparsity in detected change points. Small $\beta = 0.3$ produces a good prior for new observations only if the data is either in the orange region or is more extreme, hence is suitable for abrupt changes. As β increases to 0.8, purple regions are involved to account for small changes in data space.

Different truncations schemes are also available to variational inference. We thus name the corresponding version *variational greedy search* (VGS, VBS (K=1)) and *variational beam search* (VBS) (Fig. 1 right plot).

4 Experiments

Overview The objective of our experiments is to show that compared to other methods, variational beam search (1) better reacts to different distribution shifts – *concept drifts* and *covariate drifts*, while (2) revealing interpretable and temporally sparse latent structure. We experiment on artificial data (Section 4.1), evaluate online linear regression on three datasets with concept shifts (Section 4.3), visualize the detected change points on basketball player movements (Section 4.3), study Bayesian deep learning approaches on sequences of transformed images with covariate shifts (Section 4.4), and study the dynamics of word embeddings on historical text corpora (Section 4.5). Unstated experimental details are in Supplement G.

4.1 An Illustrative Example

We first aim to provide a basic intuition on VBS based on a simple setup involving artificial data. We demonstrated how different hypotheses compete against each other and how a better hypothesis eventually dominates in likelihood over time. To this end, we simulated noisy data points centered around a piecewise-constant step function involving two steps (seen as in black in Fig. 2). We then used VBS with beam size $K = 2$ to fit the same model that generated the data. The ranking among both hypotheses at each moment in time is indicated by their color (blue as "more likely" vs. orange as "less likely"). While hypothesis 1 assumes a single distribution shift (initially blue), the hypothesis 2 (initially orange) assumes two shifts. We see that hypothesis 1 is initially more likely, but gets over-ruled by hypothesis 2 at later times (note the color swap at step 23).

4.2 Baselines

In our supervised experiments (Section 4.3 and Section 4.4), we compared VBS against adaptive methods, established Bayesian online learning baselines and independent batch learning baselines. We formulated a supervised learning version of Bayesian online changepoint detection (BOCD) [Adams and MacKay, 2007][6] and implemented Bayesian forgetting (BF) [Kurle et al., 2020] not only for Bayesian fully connected networks, but for Bayesian convolutional neural networks, thus allowing for proper comparisons. In addition to variational continual learning (VCL, [Nguyen et al., 2018]), we also compared against Laplace propagation (LP) [Smola et al., 2003] (see also [Nguyen et al., 2018] for LP). Finally, we also adopt a trivial baseline: learning independent regressors/classifiers on each batch. Here, we adopt both a Bayesian and a non-Bayesian approach. For VBS and BOCD,

[6] 1) Although BOCD is mostly applied for unsupervised learning, its application in supervised learning and its underlying model's adaptation to change points are seldom investigated. 2) When the model is non-conjugate, such as Bayesian neural network, we approximate the log evidence $\log p(y|x)$ by the evidence lower bound.

Table 1: Evaluation of Different Datasets

MODELS	CIFAR-10 (ACCURACY)↑	SVHN (ACCURACY)↑	MALWARE (MCAE 10^{-2})↓	SENSORDRIFT (MCAE 10^{-2})↓	ELEC2 (MCAE 10^{-2})↓	NBAPLAYER (LOGLIKE 10^{-2})↑
VBS (K=6)*	**69.2±0.9**	**89.6±0.5**	**11.61**	**10.53**	7.28	**29.49±3.12**
VBS (K=3)*	68.9±0.9	89.1±0.5	11.65	10.71	7.28	**29.22±2.63**
VBS (K=1)*	68.2±0.8	88.9±0.5	11.65	10.86	**7.27**	**29.25±2.59**
BOCD (K=6)♯	65.6±0.8	88.2±0.5	12.93	24.34	12.49	22.96±7.42
BOCD (K=3)♯	67.3±0.8	88.8±0.5	12.74	24.31	12.49	20.93±7.83
BF¶	**69.8±0.8**	**89.9±0.5**	11.71	11.40	13.37	24.17±2.29
VCL†	66.7±0.8	88.7±0.5	13.27	24.90	16.59	3.48±25.53
LP‡	62.6±1.0	82.8±0.9	13.27	24.90	16.59	3.48±25.53
IB§	63.7±0.5	85.5±0.7	16.6	27.71	12.48	-44.87±16.88
IB§ (BAYES)	64.5±0.3	87.8±0.1	16.6	27.71	12.48	-44.87±16.88

* PROPOSED, ♯ [ADAMS AND MACKAY, 2007], ¶ [KURLE ET AL., 2020]
† [NGUYEN ET AL., 2018], ‡ [SMOLA ET AL., 2003], § INDEPENDENT BATCH

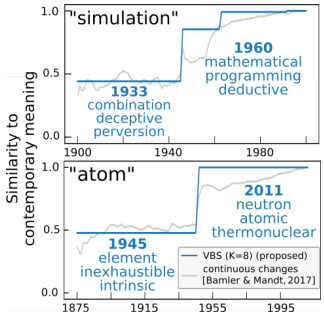

Figure 3: Sparse word meaning changes in "simulation" and "atom".

we always report the dominant hypothesis. In unsupervised learning baselines on dynamic word embeddings, we compared against the online version of dynamic word embeddings with a diffusion prior [Bamler and Mandt, 2017]. Here, our focus on finding interpretable changes in word semantics.

4.3 Bayesian Linear Regression Experiments

As a simple first version of VBS, we tested an online linear regression setup for which the posterior can be computed analytically. The analytical solution removes the approximation error of the variational inference procedure as well as optimization-related artifacts since closed-form updates are available. Detailed derivations are in Supplement D.

Real Datasets with Concept Shifts. We investigated three real-world datasets with *concept shifts*:

- **Malware** This data set is a collection of 100K malignous and benign computer programs, collected over 44 months [Huynh et al., 2017]. Each program has 482 counting features and a real-valued probability $p \in [0, 1]$ of being malware. We linearly predicted the log-odds.
- **SensorDrift** A collection of chemical sensor readings [Vergara et al., 2012]. We focused on the gas *acetaldehyde*, whose 2,926 samples span 36 months. We predicted the gas concentration level given 128 other features.
- **Elec2** The dataset contains the electricity price over three years of two Australian states [Harries and Wales, 1999]. While the original problem was 0-1 binary classification, we reproduced the targets with real-valued probabilities. After processing, we had 45,263 samples and 14 features.

We split every dataset into a train, validation, and test set. We evaluated all methods with one-step ahead absolute error[7] and computed the mean cumulative absolute error (MCAE) for each observation. In Table 1, since there is not optimization noise, we only reported the mean of all MCAEs. VBS outperforms all baselines. Please see data split details and more temporal results in Supplement G.2.

Basketball Player Tracking. We explored a collection of basketball player movement trajectories.[8] There is a large variation in the data in terms of player locations and velocities. We treated the player trajectories as time series and used a Bayesian autoregressive model to predict the next position $\mathbf{x}_{t+1}$ based on the current position $\mathbf{x}_t$ given a Bayesian transition matrix. This matrix is learned and adapted on the fly for each trajectory.

We first investigated the effect of the temperature parameter β on our approach. To this end, we visualized the detected change points on an example trajectory (Fig. 2 (middle)). We used the greedy search version of VBS (K=1) and compared models with different values of $0 < \beta < 1$. The figure shows that the larger β, the more change points are detected, i.e., beta determines the sensitivity of the model to changes. As β gets smaller, the detected change points get sparser.

[7] We measure the error in the probability space for classification problems (Malware and Elec2) and the error in the data space for regression problems (SensorDrift).

[8] `https://github.com/linouk23/NBA-Player-Movements`

We evaluated all methods with the average predictive log-probability on the test set in Table 1. Our proposed methods yield better performance than the baselines. In Supplement F, we also compared the proposed VBS with different beam sizes and BOCD on change point detection.

4.4 Bayesian Deep Learning Experiments

Our larger-scale experiments involve Bayesian convolutional neural networks trained on sequential batches for image classification using CIFAR-10 [Krizhevsky et al., 2009] and SVHN [Netzer et al., 2011]. Every few batches, we manually introduce *covariate shifts* through transforming all images globally by combining rotations, shifts, and scalings. Each transformation is generated from a fixed, predefined distribution (see Supplement G.3). The experiment involved 100 batches in sequence, where each batch contained a third of the transformed MNIST or SVHN data sets. We set the temperature $\beta = 2/3$ and set the conditional ELBO variable $T = 20,000$ (in Eq. 9) for all supervised experiments.

Table 1 shows the performances of all considered methods and both data sets, averaged across all of the 100 batches. Within their confidence bounds, VBS and BF have comparable performances and outperform the other baselines. We conjecture that the strong performance of BF can be attributed to the fact that our imposed changes are still relativeley evenly spaced and regular. The benefit of beam search in VBS is evident, with larger beam sizes consistently performing better.

4.5 Unsupervised Experiments

Our final experiment focused on unsupervised learning with the intent of demonstrating that VBS helps uncover interpretable latent structure in high-dimensional time series, such as localized changes in the meaning of individual words over long periods, through the use of change points in co-occurrence matrices. Note that this does not rule out the application of BOCD because we can overcome the non-conjugacy by using variational inference. But, considering practical issues, VBS is more favourable in that, unlike BOCD, the detected change points are free from post-process thresholding (see Supplement F). On the other hand, VBS provides completely data-driven change points and multiple plausible hypotheses of change points (Supplement G.4). The beam search also leads to an interesting structure when used for document dating tasks (Supplement G.4).

We selected two example words and their three nearest neighbors in the embedding space in Fig. 3. The plot reveals the most likely hypothesis in blue and the continuous baseline in grey. The term "atom" changes its meaning from "element" to "nuclear" in 1945–the year when two nuclear bombs were detonated. The word "race" changes from the cold-war era "arms"(-race) to its more prevalent meaning after 1991 when the cold war ended. The word "simulation" changes its dominant context from "deception" to "programming" with the advent of computers.

5 Conclusions

We introduced variational beam search: an approximate inference algorithm for Bayesian online learning on non-stationary data with irregular changes. Our approach mediates the tradeoff between a model's ability to memorize past data whilee still being able to adapt to change. It is based on a Bayesian treatment of a given model's parameters and aimed at tuning them towards the most recent learning batch while exploiting prior knowledge from previous batches. To this end, we introduced a sequence of a discrete change variables whose value controlled the way we regularized the model. For no detected change, we regularized the new learning task towards the previously learned solution; for a detected change, we broadened the prior to give room for new data evidence. This procedure is combined with beam search over the discrete change variable. In different experiments, we showed that our proposed model (1) achieved lower error in supervised setups, and (2) revealed a more interpretable and compressible latent structure in unsupervised experiments.

Broader Impacts. As with many machine learning algorithms, there is a danger that more automation could potentially result in unemployment. Yet, more autonomous adaptation to changes will enhance the safety and robustness of deployed machine learning systems, such as self-driving cars.

References

John R Zech, Marcus A Badgeley, Manway Liu, Anthony B Costa, Joseph J Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: a cross-sectional study. *PLoS medicine*, 15(11):e1002683, 2018.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, David Sculley, Sebastian Nowozin, Joshua V Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model's uncertainty? evaluating predictive uncertainty under dataset shift. In *Advances in Neural Information Processing Systems*, 2019.

Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018.

Antti Honkela and Harri Valpola. On-line variational bayesian learning. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation*, 2003.

Masa-Aki Sato. Online model selection based on the variational bayes. *Neural computation*, 13(7): 1649–1681, 2001.

Thore Graepel, Joaquin Quinonero Candela, Thomas Borchert, and Ralf Herbrich. Web-scale bayesian click-through rate prediction for sponsored search advertising in microsoft's bing search engine. In *ICML*, 2010.

James McInerney, Rajesh Ranganath, and David Blei. The population posterior and bayesian modeling on streams. *Advances in neural information processing systems*, 28:1153–1161, 2015.

Lucas Theis and Matt Hoffman. A trust-region method for stochastic variational inference with applications to streaming data. In *International Conference on Machine Learning*, pages 2503–2511. PMLR, 2015.

R Kulhavỳ and Martin B Zarrop. On a general concept of forgetting. *International Journal of Control*, 58(4):905–924, 1993.

Richard Kurle, Botond Cseke, Alexej Klushyn, Patrick van der Smagt, and Stephan Günnemann. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations*, 2020.

Sebastian Farquhar and Yarin Gal. A unifying bayesian view of continual learning. *Third workshop on Bayesian Deep Learning (NeurIPS 2018)*, 2018.

Jonathan Schwarz, Daniel Altman, Andrew Dudzik, Oriol Vinyals, Yee Whye Teh, and Razvan Pascanu. Towards a natural benchmark for continual learning. In *NeurIPS Workshop on Continual Learning*, 2018.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114 (13):3521–3526, 2017.

Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2019.

Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. In *International Conference on Learning Representations*, 2020.

Dushyant Rao, Francesco Visin, Andrei Rusu, Razvan Pascanu, Yee Whye Teh, and Raia Hadsell. Continual unsupervised representation learning. In *Advances in Neural Information Processing Systems*, pages 7647–7657, 2019.

Daniel Barry and John A Hartigan. Product partition models for change point problems. *The Annals of Statistics*, pages 260–279, 1992.

Paul Fearnhead. Exact bayesian curve fitting and signal segmentation. *IEEE Transactions on Signal Processing*, 53(6):2160–2166, 2005.

Paul Fearnhead and Zhen Liu. On-line inference for multiple changepoint problems. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):589–605, 2007.

Ryan Prescott Adams and David JC MacKay. Bayesian online changepoint detection. *arXiv preprint arXiv:0710.3742*, 2007.

Xiang Xuan and Kevin Murphy. Modeling changing dependency structure in multivariate time series. In *Proceedings of the 24th international conference on Machine learning*, pages 1055–1062, 2007.

Yao Xie, Jiaji Huang, and Rebecca Willett. Change-point detection for high-dimensional time series with missing data. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):12–27, 2012.

Yunus Saatçi, Ryan D Turner, and Carl Edward Rasmussen. Gaussian process change point models. In *ICML*, 2010.

Jeremias Knoblauch and Theodoros Damoulas. Spatio-temporal bayesian on-line changepoint detection with model selection. In *International Conference on Machine Learning*, pages 2718–2727. PMLR, 2018.

Ryan Turner, Steven Bottone, and Clay Stanek. Online variational approximations to non-exponential family change point models: with application to radar tracking. In *Proceedings of the 26th International Conference on Neural Information Processing Systems-Volume 1*, pages 306–314, 2013.

Jeremias Knoblauch, Jack Jewson, and Theodoros Damoulas. Doubly robust bayesian inference for non-stationary streaming data with β-divergences. *arXiv preprint arXiv:1806.02261*, 2018.

Michalis K Titsias, Jakub Sygnowski, and Yutian Chen. Sequential changepoint detection in neural networks with checkpoints. *arXiv preprint arXiv:2010.03053*, 2020.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian learning and inference in recurrent switching linear dynamical systems. In *Artificial Intelligence and Statistics*, pages 914–922, 2017.

Philip Becker-Ehmck, Jan Peters, and Patrick Van Der Smagt. Switching linear dynamics for variational bayes filtering. In *International Conference on Machine Learning*, pages 553–562, 2019.

Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

Chris Bracegirdle and David Barber. Switch-reset models: Exact and approximate inference. In *Proceedings of The Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 190–198, 2011.

Martin J Wainwright and Michael I Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1-2):1–305, 2008.

Matthew D Hoffman and David M Blei. Structured stochastic variational inference. In *Artificial Intelligence and Statistics*, 2015.

Cheng Zhang, Judith Bütepage, Hedvig Kjellström, and Stephan Mandt. Advances in variational inference. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):2008–2026, 2018.

Alexander J Smola, Vishy Vishwanathan, and Eleazar Eskin. Laplace propagation. In *NIPS*, pages 441–448, 2003.

Robert Bamler and Stephan Mandt. Dynamic word embeddings. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 380–389. JMLR. org, 2017.

Ngoc Anh Huynh, Wee Keong Ng, and Kanishka Ariyapala. A new adaptive learning algorithm and its application to online malware detection. In *International Conference on Discovery Science*, pages 18–32. Springer, 2017.

Alexander Vergara, Shankar Vembu, Tuba Ayhan, Margaret A Ryan, Margie L Homer, and Ramón Huerta. Chemical gas sensor drift compensation using classifier ensembles. *Sensors and Actuators B: Chemical*, 166:320–329, 2012.

Michael Harries and New South Wales. Splice-2 comparative evaluation: Electricity pricing. 1999.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

6 Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?
   Yes. See Section 3 and Section 4.
   (b) Have you read the ethics review guidelines and ensured that your paper conforms to them?
   Yes.
   (c) Did you discuss any potential negative societal impacts of your work?
   Yes. See Section 5.
   (d) Did you describe the limitations of your work?
   Yes. See Supplement G.
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results?
   Yes. See Section 3.1.
   (b) Did you include complete proofs of all theoretical results?
   Yes. See Supplement A.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)?
   No. But we will release the code afterwards.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)?
   Yes. See Supplement G.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)?
   Yes. See Table 1.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)?
   Yes. See Supplement G.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators?
   Yes. See Section 4.
   (b) Did you mention the license of the assets?
   N/A.
   (c) Did you include any new assets either in the supplemental material or as a URL?
   Yes. See the footnotes in Supplement G and Section 4.3.

(d) Did you discuss whether and how consent was obtained from people whose data you're using/curating?
Yes. They are publicly available on website.

(e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content?
N/A. we are using publicly-available data published on websites. Even if some text data may contain person's name in historical facts, we do not report any of that information.

5. If you used crowdsourcing or conducted research with human subjects...

(a) Did you include the full text of instructions given to participants and screenshots, if applicable?
N/A.

(b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable?
N/A.

(c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation?
N/A.