Dirichlet process mixture models for non-stationary data streams

1st Ioar Casado
Basque Center for Applied Mathematics
Bilbao, Spain
icasado@bcamath.org

2nd Aritz Pérez
Basque Center for Applied Mathematics
Bilbao, Spain
aperez@bcamath.org

***Abstract*—In recent years we have seen a handful of work on inference algorithms over non-stationary data streams. Given their flexibility, Bayesian non-parametric models are a good candidate for these scenarios. However, reliable streaming inference under the concept drift phenomenon is still an open problem for these models. In this work, we propose a variational inference algorithm for Dirichlet process mixture models. Our proposal deals with the concept drift by including an exponential forgetting over the prior global parameters. Our algorithm allows to adapt the learned model to the concept drifts automatically. We perform experiments in both synthetic and real data, showing that the proposed model outperforms state-of-the-art variational methods in density estimation, clustering and parameter tracking.**

***Index Terms*—Dirichlet process mixtures, variational inference, streaming data, concept drift, exponential forgetting**

I. Introduction

Bayesian non-parametric (BNP) models have become a successful approach for dealing with increasingly complex data, and when it comes to density estimation and clustering, Dirichlet process mixture (DPM) models are the best known BNP models [1], [2]. In contrast with finite mixture models or standard clustering methods, in DPMs the number of mixture components (or clusters) adjusts to the complexity of available data. Apart from avoiding model selection problems, this property makes them specially suited for working with data streams, where data batches arrive sequentially and models need to adapt to the characteristics of the new data.

Given the ubiquity of non-stationary phenomena in real life data streams, concept drift adaptation has seen great progress in the last decade [3]. However, advances in that area have been rarely combined with BNP models, hindering their real life applications [4]. Even if there are effective streaming inference algorithms for DPMs, the majority of them implicitly assumes that the data stream is stationary. In order to fill this gap, we propose a new streaming variational inference (VI) algorithm for DPMs that can deal with concept drift.

Contributions

In this work, we propose a streaming VI algorithm for DPM models, which, for the first time, extends Bayesian parametric forgetting methods [5] to the non-parametric case. This approach combines flexible adaptation to drifts of different magnitudes with the data-driven model complexity of BNPs.

We perform experiments on both synthetic and real data streams. The experiments evaluate the learned Dirichlet process Gaussian mixtures from the density estimation and clustering points of view. We also analyze our model's ability to track the underlying parameters. The experimental results show that our model outperforms the state-of-the-art variational algorithms, especially in non-stationary environments.

This work has been partially supported by the Basque Government through the BERC 2022-2025 program and the Basque Modelling Task Force project; and by the Ministry of Science, Innovation and Universities: BCAM Severo Ochoa accreditation SEV-2017-0718. We are grateful to Andrés Masegosa for his helpful commentaries and suggestions.

II. Related work

The main challenge when working with DPMs and BNP models is to find efficient learning methods. Markov chain Monte Carlo (MCMC) methods have been the basic approach for inference in DPMs [6], but they have scalability problems for big datasets [7], [8]. In [9] and [10] the authors introduced VI algorithms, which conceived posterior inference as an optimization problem [11], providing faster approximate inference. This framework was adapted for DPMs by [12]. Since then, streaming versions of VI have been widely studied.

Two main paradigms exist to tackle the problem of streaming VI: Streaming variational inference (SVB) [13] and stochastic variational inference (SVI) [14]. SVB updates the priors for batch t with the posterior obtained from batch $t-1$. By initializing priors with the previous variational distribution, SVB implicitly assumes data interchangeability and is not adequate for non-stationary streams. The same limitation hinders the performance of more recent SVB methods such as [15]. SVI, on the other hand, extends gradient based optimization to VI. It is not exactly a streaming algorithm, but assumes we can access a fixed data set in an online fashion using minibatches. This requires to know the size of the dataset, N, beforehand, which is not feasible in a streaming scenario. However, this problem can be partially circumvented by manually selecting a value for N. More recently, sampling-based inference methods have been proposed for DPMs, which can deal with non-stationary data streams [16], but there is still room for improvement among VI methods.

In order to obtain an effective VI algorithm for non-stationary DPMs, we propose a version of SVB with a forgetting mechanism. The global prior distribution for batch t is a combination of i) the initial uniformative prior and ii) the

global variational distribution obtained after observing batch $t-1$. In the proposed procedure, the forgetting parameter controls how much we forget or retain from previous batches. This forgetting parameter is automatically learned in a Bayesian manner with the hierarchical power priors method [5]. We propose a flexible learning approach with a hierarchical power prior for each component of the mixture model. By doing so, each component in the mixture has its own unique dynamic.

III. Preliminaries

A. Dirichlet process mixtures

Dirichlet processes (DP) are distributions over probability measures, hence draws from a DP are random distributions. Let G_0 be a distribution over the sample space Θ and let α be a positive scalar. A random distribution G with the same support as G_0 is distributed according to a DP with the *concentration parameter* α and the *base distribution* G_0, i.e., $G \sim \text{DP}(\alpha, G_0)$, if for any finite measurable partition $\{B_1, \dots, B_k\}$ of Θ,

$$(G(B_1), \dots, G(B_k)) \sim \text{Dir}(\alpha G_0(B_1), \dots, \alpha G_0(B_k)). \quad (1)$$

DPs were introduced by Ferguson in [1]. G_0 is known as base distribution because, for any measurable $B \subseteq \Theta$ and any $G \sim \text{DP}(\alpha, G_0)$, we have $\mathbb{E}[G(B)] = G_0(B)$. The concentration parameter α controls the probability mass around the mean, as $G \to G_0$ pointwise when $\alpha \to \infty$.

The discreteness and clustering properties of any $G \sim \text{DP}(\alpha, G_0)$ uphold the Dirichlet process as a non-parametric prior for the global parameters of infinite mixture models [2], [17]. The stick-breaking construction of DPs given by [18] takes this intuition further. For $k \geq 1$, we define

$$\begin{aligned} \beta_k &\sim \mathcal{B}(\cdot|1, \alpha), & \theta_k &\sim G_0, \\ \pi_k &= \beta_k \prod_{t=1}^{k-1}(1-\beta_t), & G &= \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}, \end{aligned} \quad (2)$$

where $\mathcal{B}(\cdot|1, \alpha)$ is a Beta distribution with parameters 1 and $\alpha > 0$. Then $G \sim \text{DP}(\alpha, G_0)$. We can understand this with the stick-breaking metaphor: we break a stick of length 1 in two parts, β_1 and $1 - \beta_1$. We define π_1 with β_1 as in (2) and continue breaking $1 - \beta_1$ to obtain $\beta_2, \beta_3, \dots$ and $\pi_2, \pi_3, \dots$. This upholds the interpretation of G as an infinite mixture of point masses with normalized weights π_k.

Now, assume we have data $\boldsymbol{x} = \{x_1, \dots, x_N\}$ drawn from some unknown distribution. We conceive the unknown distribution as a mixture model so that each x_i has distribution $p(\cdot|\theta_i)$, where the mixing distribution over the θ_i is $G \sim \text{DP}(\alpha, G_0)$. Formally, the resulting mixture model has the following hierarchical form:

$$\begin{aligned} G &\sim \text{DP}(\alpha, G_0) \\ \theta_i &\sim G \\ x_i &\sim p(\cdot|\theta_i). \end{aligned}$$

If we introduce a new latent variable z_n that indicates the mixture component to which x_n belongs, our model can now be described by the following generative process:

$$\begin{aligned} &\text{Draw } \beta_k \sim \mathcal{B}(\cdot|1, \alpha) \text{ for } k = 1, 2, \dots \\ &\text{Draw } \theta_k \sim G_0 \text{ for } k = 1, 2, \dots \\ &\text{For the } n\text{-th data point:} \\ &\qquad \text{Draw } z_n \sim \text{Mult}(\boldsymbol{\pi}) \\ &\qquad \text{Draw } x_n \sim p(\cdot|\theta_{z_n}), \end{aligned} \quad (3)$$

where z_n takes value i with probability π_i and $\boldsymbol{\pi} = (\pi_i)_{i=1}^{\infty}$ is computed using $\boldsymbol{\beta} = (\beta_i)_{i=1}^{\infty}$ as in (2). The joint probability density of the DPM model is then

$$p(\boldsymbol{x}, \boldsymbol{\beta}, \boldsymbol{z}, \boldsymbol{\theta}) = \prod_{n=1}^{N} p(x_n|\theta_{z_n}) p(z_n|\boldsymbol{\pi}) \prod_{k=1}^{\infty} G_0(\theta_k) \mathcal{B}(\beta_k|1, \alpha). \quad (4)$$

The DP prior over mixture parameters leads to an infinite mixture model. However, since π_k decreases exponentially as k increases, only a finite number of clusters, K, are actually involved when we deal with finite datasets. This solves the problem of determining the number of components of the mixture model, as we let the DPM infer it from the data.

B. Variational inference

From now on, we assume that all distributions considered are conditionally exponential, and we consider only conjugate priors as in [14].

VI has been the fundamental learning procedure for DPMs since the seminal paper [12]. Given observed data $\boldsymbol{x} = \{x_1, \dots, x_N\}$ and a model with global variables $\boldsymbol{\eta} = \{\eta_1, \dots, \eta_K\}$ and local variables $\boldsymbol{z} = \{z_1, \dots, z_N\}$, VI conceives the approximation of the intractable posterior $p(\boldsymbol{\eta}, \boldsymbol{z}|\boldsymbol{x})$ as a continuous optimization problem [11]. More precisely, VI indirectly solves

$$\arg\min_{q \in \mathcal{Q}} \text{KL}\big[q(\boldsymbol{\eta}, \boldsymbol{z}) || p(\boldsymbol{\eta}, \boldsymbol{z}|\boldsymbol{x})\big] \quad (5)$$

by solving the equivalent

$$\arg\max_{q \in \mathcal{Q}} \mathcal{L}(q),$$

where $\mathcal{L}(q)$ is called Evidence Lower BOund (ELBO) and takes the form

$$\mathcal{L}(q) := \int_{\boldsymbol{\eta}, \boldsymbol{z}} q(\boldsymbol{\eta}, \boldsymbol{z}) \log\Big(\frac{p(\boldsymbol{x}, \boldsymbol{\eta}, \boldsymbol{z})}{q(\boldsymbol{\eta}, \boldsymbol{z})}\Big) d\boldsymbol{\eta} d\boldsymbol{z}. \quad (6)$$

In this paper, we consider *mean-field* VI, where the *variational distributions* $q(\boldsymbol{\eta}, \boldsymbol{z}) \in \mathcal{Q}$ factorize as follows:

$$q(\boldsymbol{\eta}, \boldsymbol{z}|\boldsymbol{\phi}, \boldsymbol{\lambda}) = \prod_{n=1}^{N} q(z_n|\phi_n) \prod_{k=1}^{K} q(\eta_k|\lambda_k), \quad (7)$$

where $\{\phi_1, \dots, \phi_N, \lambda_1 \dots, \lambda_K\}$ are the *variational parameters*. We refer to [8] for a survey of VI methods.

In our case, after marginalizing the mixture weights in (4) following [19], we obtain the following update equations, where $\boldsymbol{\eta} = \boldsymbol{\theta}$:

$$q^*(\theta_k|\lambda_k) \propto p(\theta_k) \exp\Big(\sum_{n=1}^{N} q(z_n = k) \log p(x_n|\theta_k) \Big), \tag{8}$$

$$\begin{aligned} q^*(z_n|\phi_k) \propto & \exp\Big(\mathbb{E}_{q_{z_n}}[\log p(z_n|\boldsymbol{z}_{-n})]\Big) \\ & \times \exp\Big(\mathbb{E}_{q_{\theta_{z_n}}}[\log p(x_n|\theta_{z_n})]\Big), \end{aligned} \tag{9}$$

where we write q_{z_n} to denote $q(z_n)$ and so on.

The solutions (8) and (9) are updated iteratively using a coordinate ascent algorithm [11] to obtain the solution to (5). From now on, we write the ELBO as $\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\phi}|\boldsymbol{x}, \boldsymbol{\lambda}_0)$ to emphasize its dependency on the variational parameters and the data; $\boldsymbol{\lambda}_0$ refers to the natural parameters of G_0.

IV. Streaming VI for non-stationary DPMs

A data stream can be represented as a sequence of batches of points $\boldsymbol{x}_t \in \mathbb{R}^{d \times N}$ for $t > 0$, where t corresponds to the time stamp of the batch, d is the dimensionality of each point and N is the size of every batch. We say that a concept drift occurs when the underlying distribution of data changes.

Streaming variational Bayes (SVB) is the best known adaptation of VI to the streaming scenario [13]. At time $t \geq 1$ we receive the data batch $\boldsymbol{x}_t$, and we have to solve

$$\arg\max_{\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t} \mathcal{L}(\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t|\boldsymbol{x}_t, \boldsymbol{\lambda}_{t-1}), \tag{10}$$

where $\boldsymbol{\lambda}_{t-1}$ are the variational global parameters inferred in the previous batch. Thus, the global posterior for batch $t-1$, $q(\cdot|\boldsymbol{\lambda}_{t-1})$, is used as a prior for batch t. This approach assumes data interchangeability and it is not appropriate for non-stationary data streams.

A. SVB with power priors

In this work, following [20] and [21], we propose as a prior for batch t the combination of the uninformative prior $G_0(\boldsymbol{\theta}_t)$ and $q(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_{t-1})$ using an exponential forgetting mechanism:

$$\hat{p}(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_{t-1}, \rho_t) \propto q(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_{t-1})^{\rho_t} G_0(\boldsymbol{\theta}_t)^{1-\rho_t}, \tag{11}$$

where $\rho_t \in [0, 1]$ is the forgetting parameter for batch t. Hence when $\rho_t = 1$, we recover standard SVB in (10) and when $\rho_t = 0$ we simply carry out batchwise VI. Intermediate values of ρ_t emphasize either preserving previous information or reseting the prior. By taking $G_0(\boldsymbol{\theta}_t)$ from the same exponential family as $q(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_{t-1})$, we have that $\hat{p}(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_{t-1}, \rho_t)$ remains in that family, with natural parameters $\rho_t\boldsymbol{\lambda}_{t-1} + (1-\rho_t)\boldsymbol{\lambda}_0$, where $\boldsymbol{\lambda}_0$ is the natural parameter of the prior $G_0(\boldsymbol{\theta})$ [22].

To wrap up, using the power priors method and choosing proper exponential family distributions, we introduce a forgetting parameter in our inference framework. We will solve the following VI problem in each batch, where only the prior differs from (10):

$$\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t = \arg\max_{\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t} \mathcal{L}(\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t|\boldsymbol{x}_t, \rho_t\boldsymbol{\lambda}_{t-1} + (1-\rho_t)\boldsymbol{\lambda}_0). \tag{12}$$

B. SVB with hierarchical power priors

The forgetting parameter ρ_t in (12) is selected by the user and, in practice, its optimal value can be difficult to find. Moreover, ideally, this parameter should change over the time in order to have a quick response to a concept drift.

To overcome these limitations, we automatically learn the value of the forgetting parameter ρ_t with a technique based on [5]: we introduce prior and variational distributions for ρ_t in the variational inference mechanism. This means that our approximation to the optimal ρ_t will automatically change from batch to batch depending on the magnitude of the drift. Thus, this approach allows to detect the drifts by inspecting the values of ρ_t.

We use as a prior a truncated exponential distribution with natural parameter γ:

$$p(\rho_t|\gamma) = \frac{\gamma \exp(-\gamma\rho_t)}{1 - \exp(-\gamma)}. \tag{13}$$

The variational distribution $q(\rho_t|\omega_t)$ will also be a truncated exponential with parameter ω_t, where

$$\mathbb{E}_q[\rho_t] = 1/(1 - e^{-\omega_t}) - 1/\omega_t. \tag{14}$$

The variational parameter ω_t has a natural interpretation in terms of forgetting: if $\omega_t < 0$, then $\mathbb{E}_q[\rho_t] < 0.5$ and the model favours $G_0(\boldsymbol{\theta}_t)$ as a better fit, hence forgetting more past data. Conversely, if $\omega_t > 0$, $\mathbb{E}_q[\rho_t] > 0.5$ and more emphasis is given to past data [23].

Plugging the prior over ρ_t in our collapsed DPM model we obtain an ELBO in which we cannot work directly, because ρ_t breaks the exponential conjugacy conditions for VI. To solve this problem, we work over the surrogate ELBO proposed in [5]. In this case, the update rules for $\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t$ are the same as in (8) and (9). In order to update ω_t, we use the natural gradient [24] of the surrogate ELBO with respect to ω_t. This results in

$$\omega_t^* = \mathrm{KL}\big(q(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_t)||G_0(\boldsymbol{\theta}_t)\big) - \mathrm{KL}\big(q(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_t)||q(\boldsymbol{\theta}_t|\boldsymbol{\lambda}_{t-1})\big) + \gamma. \tag{15}$$

C. Multiple hierarchical power priors for DPMs

The procedure above uses a single forgetting parameter ρ_t. This approach can be extended by considering one independent power prior $\rho_{t,k}$ for each global parameter $\theta_{t,k}$ of the mixture model. This can be easily done by assuming that the $\rho_{t,k}$ are pairwise independent. With this assumption, we implicitly consider that the components of a non-stationary infinite mixture have different dynamics. The update rule for each $\omega_{t,k}$ associated to the parameter of the mixture $\theta_{t,k}$ is:

$$\begin{aligned} \omega_{t,k}^* = & \mathrm{KL}\big(q(\theta_{t,k}|\lambda_{t,k})||G_0(\theta_{t,k})\big) \\ & - \mathrm{KL}\big(q(\theta_{t,k}|\lambda_{t,k})||q(\theta_{t,k}||\lambda_{t-1,k})\big) + \gamma \end{aligned} \tag{16}$$

This extension allows the model to have different forgetting mechanisms for each global parameter, and will be crucial for our DPM model, since the concept drift does not necessarily affect every mixture component equally. We refer to this model as *multiple hierarchical power priors* (MHPP). The inference mechanism of MHPP is shown in Algorithm 1.

Algorithm 1: MHPP-DPM

Input: Data batch $\boldsymbol{x}_t$ and variational posterior $\boldsymbol{\lambda}_{t-1}$
Output: $\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t, \boldsymbol{\omega}_t$
$\boldsymbol{\lambda}_t \leftarrow \boldsymbol{\lambda}_{t-1}$
$\mathbb{E}_q[\rho_{t,k}] = 1/2$ for $1 \leq k \leq K$
Initialize $\boldsymbol{\phi}_t$
repeat
  for $1 \leq k \leq K$:
    Compute $\hat{p}(\cdot|\lambda_{t-1,k}, \mathbb{E}_q[\rho_{t,k}])$, (Eq. 11).
    Compute $q(\theta_{t,k})$, (Eq. 8) with
    $\hat{p}(\theta_{t,k}|\lambda_{t-1,k}, \mathbb{E}_q[\rho_{t,k}])$ instead of $p(\theta_{t,k})$.
    Compute $w_{t,k}$, (Eq. 16).
    Update $\mathbb{E}_q[\rho_{t,k}]$, (Eq. 14).
  for $1 \leq n \leq |\{\boldsymbol{x}_t\}|$:
    Update $q(z_{t,n})$, (Eq. 9).
until convergence
return $\boldsymbol{\lambda}_t, \boldsymbol{\phi}_t, \boldsymbol{\omega}_t$

V. Experiments

In this section we empirically evaluate the three proposed models: PP, where the single forgetting parameter has to been hand-tuned beforehand; HPP, which automatically learns the forgetting parameter; and MHPP-DPM (Algorithm 1), which learns a forgetting parameter for each global parameter of the mixture model. For the PP methods, in each experiment we choose the best forgetting parameter in $\rho \in \{0.9, 0.99\}$. We compare their performance against the following baselines:

- Streaming variational bayes DPM (SVB) of [13].
- Stochastic variational inference (SVI) of [14] as implemented by [25].
- Privileged-DPM (Privileged), a version of SVB-DPM discarding previous information when a drift happens.

SVB and SVI are the state-of-the-art procedures for DPM inference over data streams. Privileged represents the gold standard, however, it can not be used in practice because it requires to know when each concept drift occurs. Our Python implementations of SVB, SVI, Privileged, PP, HPP and MHPP are available online. [1]

In HPP and MHPP, the inference step for ρ_t is simultaneous to that of $\boldsymbol{\theta}$ and $\boldsymbol{z}$, hence the computational complexity of the proposed algorithms will be the same as the standard VI. We evaluate the ability to adapt to drifts in the following tasks:

- Density estimation: we measure the quality of the learned DPMs using the log-likelihood in test data.
- Clustering: we evaluate the obtained clusters using four popular metrics: Silhouette score, Normalized Mutual Information (NMI), Adjusted Rand Index (ARI) and Purity. We have reported the mean values across all the batches of the data stream. All the measures have been computed using the implementations in scikit-learn [26].
- Model parameter tracking: using the synthetic data, we compare the parameters of the mixture model obtained by the different algorithms with respect to the parameters of the true model.

[1] https://github.com/IoarCT/DPM-for-non-stationary-streams

Every algorithm implements collapsed Dirichlet process (isotropic) Gaussian mixtures with truncation K, hence in our case the global parameters are the means and covariances of each component: $\boldsymbol{\theta} = \{\boldsymbol{\mu}_1, \tau_1, \dots \boldsymbol{\mu}_K, \tau_K\}$. We use uninformative conjugate priors $\mathcal{N}(\boldsymbol{\mu}; \mathbf{0}, I)$, Gamma$(\tau; 1, 1)$.

A. Synthetic data

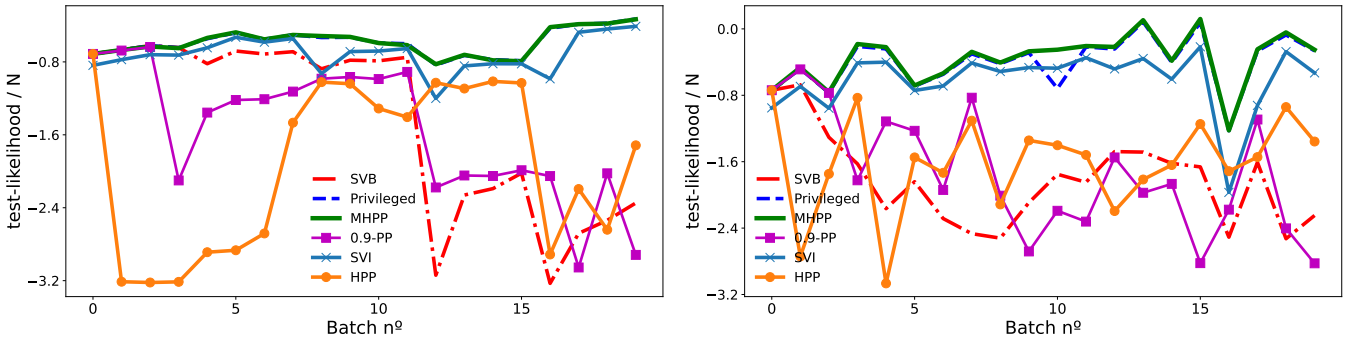
We generate four 2D isotropic Gaussians, and randomly vary their mean and covariance for 20 batches in order to simulate drift. We set $\alpha = 2$, truncation parameter $K = 10$ and run all the algorithms for 100 iterations. We use 1000 training points and 500 test points per batch. In the case of SVI, we work as if each batch was the full dataset and warm-start the model for the next batch. This can bias the results towards SVI. We fix its learning parameters $rhoexp = 0.55$ and $rhodelay = 1$.

1) Density estimation: Figures 1a and 1b compare the algorithms according to the log-likelihood (higher is better) of test data using the obtained Gaussian mixture. In 1a, where concept drift occurs every 4 batches, the performance of MHPP is remarkably similar to that of Privileged, both in response to drifts and in the stationary phase. MHPP outperforms the state-of-the-art procedures.

Figure 1b shows the same experimental framework with drift in every batch. Again, MHPP obtains results remarkably similar to Privileged. This shows that MHPP is able to address density learning in scenarios with very frequent drifts. Conversely, PP, SVB and SVI methods perform worse with frequent drifts. Note that, in both Figures 1a and 1b, HPP and 0.9-PP show high numerical instability, justifying the need for multiple forgetting parameters as in MHPP.

2) Clustering: Table I summarizes the results of different metrics under the two drift frequencies studied. When the drift occurs every 4 batches MHPP and SVB obtain the best results. In the scenario where the drift occurs at every batch, MHPP obtains the best results, and they are remarkably better than the state-of-the-art. This suggests that MHPP is the best algorithm addressing the concept drift in clustering problems.

3) Parameter tracking: To analyze the ability of the algorithms to learn the parameters of the true Gaussian mixture model, we show the evolution of the estimated means and standard deviations of the four most populated clusters in each batch. In order simplify the visualization of the results, we have selected MHPP, SVB and SVI, and we have considered the scenario with drifts every 4 batches. Figures 2a and 2b show the evolution of the standard deviations and means respectively. In Figure 2b the numbers represent the order of drifts. MHPP is able to recover the parameters of the underlying Gaussian mixture adapting to concept drift immediately, while the time of response of SVI and SVB is higher and less accurate. The experiments also show that the forgetting parameters of MHPP tend to 0.5 when their component is not *active* in the DPM, while capturing different dynamics for means and covariances.



(a) Drift every 4 batches

(b) Drift every batch

Fig. 1. log-likelihood per test data point from 2D Gaussians. We consider different drift frequencies.

	Privileged	SVB	SVI	0.9-PP	HPP	**MHPP**
Drift every 4 batches						
Silhouette score	0.83 ± 0.05	$\mathbf{0.81} \pm 0.07$	0.68 ± 0.12	0.55 ± 0.29	0.71 ± 0.15	0.80 ± 0.10
NMI score	1	$\mathbf{0.99} \pm 0.01$	0.83 ± 0.19	0.87 ± 0.16	0.96 ± 0.05	$\mathbf{0.99} \pm 0.01$
ARI score	1	$\mathbf{0.99} \pm 0.01$	0.95 ± 0.07	0.75 ± 0.27	0.81 ± 0.24	$\mathbf{0.99} \pm 0.02$
Purity score	1	**1**	0.98 ± 0.04	0.84 ± 0.19	0.83 ± 0.21	**1**
Drift every batch						
Silhouette score	0.84 ± 0.03	0.74 ± 0.10	0.57 ± 0.23	0.46 ± 0.28	0.78 ± 0.08	$\mathbf{0.82} \pm 0.06$
NMI score	0.99 ± 0.04	0.94 ± 0.07	0.95 ± 0.06	0.79 ± 0.03	0.91 ± 0.09	$\mathbf{0.99} \pm 0.03$
ARI score	0.97 ± 0.09	0.92 ± 0.11	0.94 ± 0.09	0.67 ± 0.27	0.84 ± 0.15	$\mathbf{0.98} \pm 0.06$
Purity score	0.97 ± 0.07	0.96 ± 0.09	0.86 ± 0.6	0.77 ± 0.19	0.86 ± 0.15	$\mathbf{0.99} \pm 0.05$

TABLE I

RESULTS FOR DIFFERENT CLUSTER METRICS IN 2D GAUSSIANS.
WE DO NOT CONSIDER *Privileged* WHEN HIGHLIGHTING THE BEST ALGORITHM

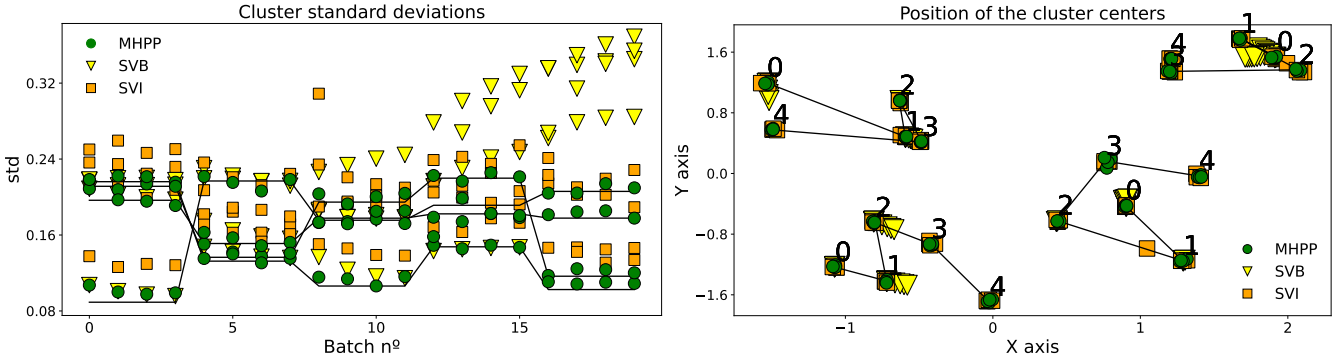


Fig. 2. Global parameter tracking for different algorithms. We represent data of the 4 most populated components. Ground truth is indicated by black lines.

B. Real data

In order to test our algorithm with real data, we use the MNIST [27] and the noisy MNIST (n-MNIST) [28] datasets. The first is the standard digit recognition dataset, while the second includes three datasets, each of them created by adding a different kind of noise to the digits: additive white Gaussian noise, motion blur, and a combination of additive white Gaussian noise and reduced contrast. The transition from MNIST to n-MNIST data and the addition or removal of type of digits will simulate the concept drifts.

The experimental framework is as follows: we consider all four data sources and for each batch we first randomly select the number of digits we consider in a range from 6 to all 9. Then we sample those from one of the data sources randomly. This will create a data stream where the number of cluster varies and the source of those clusters can change from batch to batch. Every dataset is preprocessed with minmax scaling and the 764 (28×28) dimensions are reduced to 50 using PCA. For this experiment we set truncation parameter $K = 30$, $\alpha = 3$, 1000 training points and 500 test points.

1) Density estimation: Figure 3 shows test log-likelihood for different algorithms in the n-MNIST experiment. The performances of SVI, SVB, PP and MHPP are very similar. HPP provides the worst results. Overall, all the likelihoods

	Privileged	SVB	SVI	0.99-PP	HPP	**MHPP**
Silhouette score	0.06 ± 0.03	0.02 ± 0.03	$\mathbf{0.17} \pm 0.04$	0.01 ± 0.04	0.03 ± 0.03	0.05 ± 0.02
NMI score	0.67 ± 0.03	0.64 ± 0.04	0.24 ± 0.08	0.67 ± 0.02	0.67 ± 0.04	$\mathbf{0.69} \pm 0.04$
ARI score	0.45 ± 0.05	0.43 ± 0.05	0.08 ± 0.04	0.45 ± 0.04	0.48 ± 0.10	$\mathbf{0.50} \pm 0.07$
° Purity score	0.78 ± 0.04	0.75 ± 0.06	0.25 ± 0.03	0.76 ± 0.05	0.70 ± 0.06	$\mathbf{0.79} \pm 0.06$

TABLE II
RESULTS FOR DIFFERENT CLUSTER METRICS IN N-MNIST DATA SET.
WE DO NOT CONSIDER *Privileged* WHEN HIGHLIGHTING THE BEST ALGORITHM

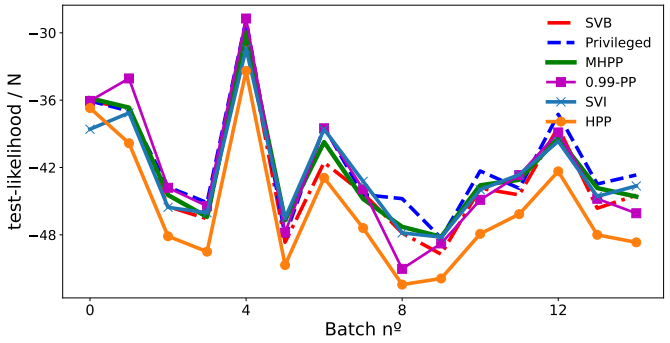


Fig. 3. log-likelihood per data point of different algorithms for held-out data from n-MNIST.

remain comparable. However, we have observed that SVI and SVB requires more components in the mixture model to reach the results of MHPP.

2) Clustering: Table II shows the results of each model in the four clustering metrics considered. MHPP is superior in 3 out of 4 metrics, followed by 0.99-PP. The different performance with respect to density estimation can be explained by the fact that the log-likelihood does not penalize the use of too many mixture components.

Overall, the experimentation upholds MHPP as the most competitive method for DPM density estimation and clustering in non-stationary streaming scenarios.

REFERENCES

[1] T. S. Ferguson, "A Bayesian analysis of some nonparametric problems," *The annals of statistics*, pp. 209–230, 1973.

[2] Y. W. Teh, "Dirichlet process." *Encyclopedia of machine learning*, vol. 1063, pp. 280–287, 2010.

[3] J. Lu, A. Liu, F. Dong, F. Gu, J. Gama, and G. Zhang, "Learning under concept drift: A review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 12, pp. 2346–2363, 2018.

[4] N. L. A. Ghani, I. A. Aziz, and M. Mehat, "Concept drift detection on unlabeled data streams: A systematic literature review," in *2020 IEEE Conference on Big Data and Analytics (ICBDA)*, 2020, pp. 61–65.

[5] A. Masegosa, T. D. Nielsen, H. Langseth, D. Ramos-López, A. Salmerón, and A. L. Madsen, "Bayesian models of data streams with hierarchical power priors," in *International Conference on Machine Learning*. PMLR, 2017, pp. 2334–2343.

[6] R. M. Neal, "Markov chain sampling methods for Dirichlet process mixture models," *Journal of computational and graphical statistics*, vol. 9, no. 2, pp. 249–265, 2000.

[7] Z. Li, "A review of Bayesian posterior distribution based on MCMC methods," in *International Conference on Computing and Data Science*. Springer, 2021, pp. 204–213.

[8] D. M. Blei, A. Kucukelbir, and J. D. McAuliffe, "Variational inference: A review for statisticians," *Journal of the American statistical Association*, vol. 112, no. 518, pp. 859–877, 2017.

[9] H. Attias, "A variational Bayesian framework for graphical models," *Advances in Neural Information Processing Systems*, vol. 12, 1999.

[10] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul, "An introduction to variational methods for graphical models," *Machine learning*, vol. 37, no. 2, pp. 183–233, 1999.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, ser. Information Science and Statistics. Springer New York, 2006, vol. 4.

[12] D. M. Blei and M. I. Jordan, "Variational inference for Dirichlet process mixtures," *Bayesian analysis*, vol. 1, no. 1, pp. 121–143, 2006.

[13] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan, "Streaming variational bayes," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[14] M. D. Hoffman, D. M. Blei, C. Wang, and J. Paisley, "Stochastic variational inference," *Journal of Machine Learning Research*, 2013.

[15] T. Campbell, J. Straub, J. W. Fisher III, and J. P. How, "Streaming, distributed variational inference for Bayesian nonparametrics," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.

[16] O. Dinari and O. Freifeld, "Sampling in Dirichlet process mixture models for clustering streaming data," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2022, pp. 818–835.

[17] M. D. Escobar and M. West, "Bayesian density estimation and inference using mixtures," *Journal of the American statistical association*, vol. 90, no. 430, pp. 577–588, 1995.

[18] J. Sethuraman, "A constructive definition of Dirichlet priors," *Statistica sinica*, pp. 639–650, 1994.

[19] K. Kurihara, M. Welling, and Y. W. Teh, "Collapsed variational Dirichlet process mixture models." in *IJCAI*, vol. 7, 2007, pp. 2796–2801.

[20] M. Kárnỳ, "Approximate Bayesian recursive estimation," *Information Sciences*, vol. 285, pp. 100–111, 2014.

[21] E. Özkan, V. Šmídl, S. Saha, C. Lundquist, and F. Gustafsson, "Marginalized adaptive particle filtering for nonlinear models with unknown time-varying noise parameters," *Automatica*, vol. 49, no. 6, pp. 1566–1575, 2013.

[22] R. Kulhavỳ and F. J. Kraus, "On duality of regularized exponential and linear forgetting," *Automatica*, vol. 32, no. 10, pp. 1403–1415, 1996.

[23] A. R. Masegosa, D. Ramos-López, A. Salmerón, H. Langseth, and T. D. Nielsen, "Variational inference over nonstationary data streams for exponential family models," *Mathematics*, vol. 8, no. 11, p. 1942, 2020.

[24] S.-I. Amari, "Natural gradient works efficiently in learning," *Neural computation*, vol. 10, no. 2, pp. 251–276, 1998.

[25] M. C. Hughes and E. B. Sudderth, "Bnpy: Reliable and scalable variational inference for Bayesian nonparametric models," in *Proceedings of the NIPS Probabilistic Programimming Workshop, Montreal, QC, Canada*, 2014, pp. 8–13.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[28] S. Basu, M. Karki, S. Ganguly, R. DiBiano, S. Mukhopadhyay, S. Gayaka, R. Kannan, and R. Nemani, "Learning sparse feature representations using probabilistic quadtrees and deep belief nets," *Neural Processing Letters*, vol. 45, no. 3, pp. 855–867, 2017.