Streaming Variational Inference for Bayesian Nonparametric Mixture Models

Alex Tank
University of Washington
Department of Statistics

Nicholas J. Foti
University of Washington
Department of Statistics

Emily B. Fox
University of Washington
Department of Statistics

Abstract

In theory, Bayesian nonparametric (BNP) models are well suited to streaming data scenarios due to their ability to adapt model complexity with the observed data. Unfortunately, such benefits have not been fully realized in practice; existing inference algorithms are either not applicable to streaming applications or not extensible to BNP models. For the special case of Dirichlet processes, streaming inference has been considered. However, there is growing interest in more flexible BNP models building on the class of normalized random measures (NRMs). We work within this general framework and present a streaming variational inference algorithm for NRM mixture models. Our algorithm is based on assumed density filtering (ADF), leading straightforwardly to expectation propagation (EP) for large-scale batch inference as well. We demonstrate the efficacy of the algorithm on clustering documents in large, streaming text corpora.

1 Introduction

Often, data arrive sequentially in time and we are tasked with performing unsupervised learning as the data stream in, without revisiting past data. For example, consider the task of assigning a topic to a news article based on a history of previously assigned documents. The articles arrive daily—or more frequently—with no bound on the total number in the corpus. In clustering such *streaming* data, Bayesian nonparametric (BNP) models are natural as they allow the number of clusters to grow as data arrive. A challenge, however, is that it is infeasible to store the past cluster assignments, and instead inference algorithms must rely solely on summary statistics of these variables.

Stochastic variational inference (SVI) [1] has become a popular method for scaling posterior inference in Bayesian latent variable models. Although SVI has been extended to BNP models, SVI requires specifying the size of the data set a priori, an inappropriate assumption for streaming data. In contrast, streaming variational Bayes (SVB) [2] handles unbounded data sets by exploiting the sequential nature of Bayes theorem to recursively update an approximation of the posterior. Specifically, the variational approximation of the current posterior becomes the prior when considering new observations. While SVB is appropriate for parametric models, it does not directly generalize to the BNP setting that is essential for streaming data.

For BNP models, streaming inference has been limited to algorithms hand-tailored to specific models. For example, a streaming variational inference algorithm for Dirichlet process (DP) mixture models was recently proposed based on heuristic approximations to the Chinese restaurant process (CRP) predictive distribution associated with the DP [3].

We seek a method for streaming inference in BNP models that is more generally extensible. We are motivated by the recent focus on a broader class of BNP priors—*normalized random measures* (NRMs)—that enable greater control of various properties than the DP permits. For example, in clustering tasks, there is interest in having flexibility in the distribution of cluster sizes. Throughout the paper, we focus on the specific case of the normalized generalized gamma process (NGGP), though our methods are more general. Recently, NGGP mixture models have been shown to outperform the DP [4, 5], but inference has relied on Markov chain Monte Carlo (MCMC). Due to the limitations of MCMC, such demonstrations have been limited to small data sets. Importantly, NGGPs and the DP differ mainly in their asymptotic scaling properties and the use of NGGPs may be more appropriate in large data sets where the logarithmic cluster growth

Appearing in Proceedings of the 18^{th} International Conference on Artificial Intelligence and Statistics (AISTATS) 2015, San Diego, CA, USA. JMLR: W&CP volume 38.

rate of the DP is not appropriate.

To address the challenge of streaming inference in NRM mixture models, we develop a variational algorithm based on assumed density filtering (ADF) [6]. Our algorithm uses infinite-dimensional approximations to the mixture model posterior and allows general BNP predictive distributions to be used by leveraging an auxiliary variable representation. As a byproduct of the ADF construction, a multi-pass variant straightforwardly yields an expectation propagation (EP) algorithm for batch inference in BNP models. This provides a new approach to scalable BNP batch inference.

In the special case of DPs, our algorithm reduces to that of [3]. As such, our framework forms a theoretically justified and general-purpose scheme for BNP streaming inference, encompassing previous heuristic and model-specific approaches, and with a structure that enables insight into BNP inference via EP.

We demonstrate our algorithm on clustering documents from text corpora using an NRM mixture model based on the NGGP [5]. After a single pass through a modest-sized data set, our streaming variational inference algorithm achieves performance nearly on par with that of a batch sampling-based algorithm that iterates through the data set hundreds of times. We likewise examine a New York Times corpus of 300,000 documents to which the batch algorithm simply does not scale (nor would it be applicable in a truly streaming setting). In these experiments, we justify the importance of considering the flexible class of NRM-based models. Our work represents the first application of non-DP-based NRMs to such large-scale applications.

2 Background

2.1 Completely Random Measures

A *completely random measure* (CRM) [7] is a distribution over measures G on Θ such that for disjoint $A_k \subset \Theta$, $G(A_k)$ are independent random variables and

$$G = \sum_{k=1}^{\infty} \pi_k \delta_{\theta_k}. \tag{1}$$

The masses π_k and locations θ_k are characterized by a Poisson process on $\Theta \times \mathbb{R}_+$ with Lévy measure $\mu(d\theta, d\pi)$ [7, 8]. We restrict our attention to *homogeneous* CRMs where $\mu(d\theta, d\pi) = H_0(d\theta)\lambda(d\pi)$, a common assumption in the literature [9, 10, 11]. We denote a draw from a homogeneous CRM as

$$G \sim \mathrm{CRM}(\lambda, H_0). \tag{2}$$

The total mass $T = G(\Theta) = \sum_{k=1}^{\infty} \pi_k$ is almost surely finite [12]. However, since $T \neq 1$ in general, CRMs cannot directly be used as priors for mixture models.

2.2 Normalized Random Measures

One can normalize a CRM by its finite total mass to construct a BNP prior for mixture models. Specifically, define the *normalized random measure* (NRM) $P = \sum_{k=1}^{\infty} \frac{\pi_k}{T} \delta_{\theta_k}$. The Dirichlet process (DP) is an NRM which arises from normalizing the masses of a gamma process [9]. However, more flexible NRMs can be constructed by starting with different CRMs.

In the mixture model setting, we observe data $\{x_i \in \mathbb{R}^d\}$ with x_i generated from mixture component θ^{z_i}. Here, we assume the assignment variables, z_i, are 1-of-K coded so that $\sum_k z_{ik} = 1$ and $z_{ik} = 1$ implies that observation i is assigned to component θ_k via θ^{z_i}. The resulting NRM mixture model can be written as:

$$\begin{aligned} G \mid \lambda, H_0 &\sim \mathrm{CRM}(\lambda, H_0) \\ z_i \mid G &\sim \sum_{k=1}^{\infty} \frac{\pi_k}{T} \delta_k \\ x_i \mid z_i, \theta &\sim F(x_i | \theta^{z_i}), \end{aligned} \tag{3}$$

where $F(\cdot|\cdot)$ is an observation model.

For our running example of the *normalized generalized gamma process* (NGGP), the GGP Lévy measure is

$$\lambda(d\pi) = \frac{a}{\Gamma(1-\sigma)} \pi^{-\sigma-1} e^{-\tau\pi} d\pi, \tag{4}$$

where $\tau \in [0, \infty)$, $a \in (0, \infty)$, and $\sigma \in [0, 1)$. Notable special cases of the NGGP are $\sigma = 0$, where we obtain the DP, and $\sigma = 0.5$, where we obtain the normalized inverse-Gaussian (IG) process. The NGGP with $\sigma \neq 0$ provides greater control over model properties, such as the distribution of cluster sizes [4].

For any NRM mixture model, by introducing an auxiliary variable $U_n \sim \Gamma(n, T)$, we can integrate out the NRM P and define a partial urn scheme [5, 11]. In the case of the NGGP we have:

$$p(z_{n+1} = k | U_n, z_{1:n}) \propto \begin{cases} n_k - \sigma, & k \leq K \\ a(U_n + \tau)^{\sigma}, & k = K + 1, \end{cases} \tag{5}$$

where K is the number of instantiated clusters in $z_{1:n}$. When $\sigma = 0$, Eq. (5) reduces to the well known Chinese restaurant process (CRP) corresponding to the DP. The posterior distribution of U_n is given by [11]:

$$p(U_n | z_{1:n}) \propto \frac{U_n^n}{(U_n + \tau)^{n - aK}} e^{-\frac{a}{\sigma}(U_n + \tau)^{\sigma}}. \tag{6}$$

Together, Eqs. (5) and (6) can be used to define MCMC samplers for NGGP mixture models [5, 13]; our streaming algorithm also exploits the use of U_n.

2.3 Assumed Density Filtering

Assumed density filtering (ADF) was first developed as a sequential procedure for inference in dynamic

models that iteratively projects an intractable distribution onto a simpler family of distributions. Let $z_{1:n} = (z_1, z_2, \ldots, z_n)$ be a sequence of random variables with joint distribution $p_n(z_{1:n})$. We can write the joint distribution as a product of factors,

$$p_n(z_{1:n}) \propto \prod_{i=1}^{n} f_i(z_{1:i}). \tag{7}$$

ADF approximates the sequence of distributions $p_n(z_{1:n})$ with a sequence $\hat{q}_n(z_{1:n}) \in \mathcal{Q}_n$, where $\mathcal{Q}_n$ is a family of simpler distributions. Based on the current $\hat{q}_n(z_{1:n})$, the approximation to $p_{n+1}(z_{1:n+1})$ is formed as follows. The $(n+1)$st factor is incorporated to form $\hat{p}_{n+1}(z_{1:n}) \overset{\triangle}{\propto} f_{n+1}(z_{1:n+1})\hat{q}_n(z_{1:n})$, which is then projected onto $\mathcal{Q}_{n+1}$ by minimizing the KL divergence:

$$\hat{q}_{n+1}(z_{1:n+1}) = \underset{q_{n+1}\in\mathcal{Q}_{n+1}}{\arg\min}\ \mathrm{KL}\Big(\hat{p}_{n+1}(z_{1:n+1})||q_{n+1}(z_{1:n+1})\Big). \tag{8}$$

When $\mathcal{Q}_n$ factorizes as $q_n(z_{1:n}) = \prod_{i=1}^{n} q_n(z_i)$, the optimal distribution for each factor is given by the marginal distribution, $\hat{q}_{n+1}(z_i) \propto \int f_{n+1}(z_{1:n+1})\hat{q}_n(z_{1:n})dz_{\setminus i}$, where $z_{\setminus i}$ denotes the set $\{z_j, j \neq i\}$. The tractability of this integral for certain families of factors f_n and $\hat{q}_n$ motivates ADF, and in particular, the recursive projection onto $\{\mathcal{Q}_n\}$.

2.4 Expectation Propagation

ADF can be generalized to perform batch inference in static models resulting in the well known expectation propagation (EP) algorithm [6]. In EP, one approximates an intractable, factorized distribution over a fixed set of model parameters, θ, with a tractable distribution, $q \in \mathcal{Q}$. In place of Eq. (7), we have

$$p(\theta) \propto \prod_{i=1}^{n} f_i(\theta). \tag{9}$$

An EP iteration begins with both a posterior approximation, $\hat{q}(\theta)$, and stored local contributions, $\bar{q}_j(\theta)$, associated with each factor $f_j(\theta)$. To refine the posterior approximation, a local contribution is removed to form a normalized approximation to the remaining $n-1$ factors, $\hat{q}_{\setminus j}(\theta) \propto \frac{q_\theta)}{\bar{q}_j(\theta)}$. As in ADF, the jth factor is then appended to the approximation $\hat{q}_{\setminus j}$ and projected back onto $\mathcal{Q}$ to obtain a refined $\hat{q}(\theta)$:

$$\hat{q}(\theta) = \underset{q\in\mathcal{Q}}{\arg\min}\,\mathrm{KL}\Big(\hat{p}(\theta) \propto f_j(\theta)\hat{q}_{\setminus j}(\theta)\Big|\Big|q(\theta)\Big). \tag{10}$$

The jth local contribution is then updated to

$$\bar{q}_j(\theta) \propto \frac{\hat{q}(\theta)}{\hat{q}_{\setminus j}(\theta)}. \tag{11}$$

When $\hat{q}, \bar{q}_j, \hat{q}_{\setminus j}$ are in the exponential family with the same type of sufficient statistics, $\hat{\nu}, \bar{\nu}_j, \hat{\nu}_{\setminus j} \in \mathbb{R}^m$, respectively, then $\bar{\nu}_j = \hat{\nu} - \hat{\nu}_{\setminus j}$. This process of removing local statistics from the approximation, adding in the respective factor, and re-projecting onto $\mathcal{Q}$ is repeated for all factors until convergence.

The link between ADF and EP, comparing Eqs. (8) and (10), allows us to extend our streaming BNP algorithm of Sec. 3 to EP for batch inference (Sec. 3.4). EP is easily parallelized [14], allowing these methods to scale to massive batch data sets, though we leave the parallel extension of our method to future work.

3 Streaming Variational Inference for BNP Mixture Models

We now turn to deriving a streaming inference algorithm for the NRM mixture model of Eq. (3). Here, our goal is joint inference of the growing set of local cluster indicators, $z_{1:n}$, and the static set of global cluster parameters, $\theta = \{\theta_k\}_{k=1}^{\infty}$. The method is derived from the ADF algorithm of Sec. 2.3 and boils down to: (1) a local update of cluster soft assignments for the current data point and (2) a global update of cluster variational parameters. The local update follows directly from ADF. Embedded in this step is computing the NRM predictive probability on cluster assignments, for which we use the auxiliary variable representation of Eq. (5) combined with an additional variational approximation to compute an intractable integral. For computational tractability, the global step uses an approximation similar to that proposed in [15], though an exact ADF update is possible.

To start, note that the posterior for the first n assignments, $z_{1:n}$, and cluster parameters, θ, factorizes as:

$$\begin{aligned} p_n(z_{1:n}, \theta|x_{1:n}) &\propto p(x_n|z_n, \theta)p(z_n|z_{1:n-1}) \\ &\quad \times p(z_{1:n-1}, \theta|x_{1:n-1}) \end{aligned} \tag{12}$$

$$\propto p(\theta)\prod_{i=1}^{n} p(x_i|z_i, \theta)p(z_i|z_{1:i-1}). \tag{13}$$

Eq. (12) emphasizes the sequential decomposition of the posterior while Eq. (13) concretely links our derivation with ADF. We set the first factor to $p(x_1|z_1, \theta)p(z_1)\prod_{k=1}^{\infty} p(\theta_k)$, where $p(z_{11} = 1) = 1$ so that $p(x_1|z_1, \theta)p(z_1) = p(x_1|\theta_1)p(z_1)$. We then define $p(x_i|z_i, \theta)p(z_i|z_{1:i-1})$ as the ith factor. We apply ADF to Eq. (13) to obtain a sequence of factorized variational approximations of the form $\hat{q}_n(z_{1:n}, \theta) = \prod_{k=1}^{\infty}\hat{q}_n(\theta_k)\prod_{i=1}^{n}\hat{q}_n(z_i)$ for the first n factors. Since the first factor takes this factorized form, we have $\hat{q}_1(z_1, \theta) \propto p(z_1)p(x_1|\theta_1)p(\theta_1)\prod_{k=2}^{\infty} p(\theta_k)$, so algorithmically we only update $\hat{q}_1(z_1)$ and $\hat{q}_1(\theta_1)$. For subsequent factors, assume the posterior $p(z_{1:n-1}, \theta|x_{1:n-1})$

is approximated by a factorized $\hat{q}_{n-1}(z_{1:n-1}, \theta)$. To obtain the next approximate posterior after the nth observation, we use Eq. (8):

$$\hat{p}_n(z_{1:n}, \theta | x_{1:n}) \overset{\triangle}{\propto} p(x_n|z_n, \theta) p(z_n|z_{1:n-1}) \hat{q}_{n-1}(z_{1:n-1}, \theta)$$
$$\hat{q}_n(z_{1:n}, \theta) = \arg\min_{q_n \in \mathcal{Q}_n} \mathrm{KL}\Big(\hat{p}_n(z_{1:n}, \theta | x_{1:n}) || q_n(z_{1:n}, \theta)\Big). \tag{14}$$

Given our mean field assumption, the optimal distributions for the local variables, $z_{1:n}$, are given by the marginal distributions:

$$\hat{q}_n(z_i) \propto \sum_{z_{\setminus i}} \int p(x_n|z_n, \theta) p(z_n|z_{1:n-1}) \hat{q}_{n-1}(z_{1:n-1}, \theta) d\theta. \tag{15}$$

For $i < n$, Eq. (15) indicates that the optimal variational distributions for the assignments are retained, i.e. $\hat{q}_n(z_i) = \hat{q}_{n-1}(z_i)$. The optimal distribution for the new observation's assignment, z_n, is given by:

$$\hat{q}_n(z_{nk}) \propto q^{\mathrm{pr}}(z_{nk}) \int p(x_n|z_{nk}, \theta) \hat{q}_{n-1}(\theta_k) d\theta_k, \tag{16}$$

where $q^{\mathrm{pr}}(z_n) = \sum_{z_{1:n-1}} p(z_n|z_{1:n-1}) \prod_{i=1}^{n-1} \hat{q}_{n-1}(z_i)$ is an approximate prior probability for the cluster assignment of x_n, mirroring the role of the predictive rule, $p(z_n|z_{1:n-1})$, when assignments are fully observed. We consider conjugate exponential family models so that $\hat{q}_{n-1}(\theta_k)$ is in the same family as $p(\theta_k)$, allowing the integral in Eq. (16) to be given in closed form. As such, our focus is on studying q^{pr} for NRMs.

The update in Eq. (16) has appeared previously in both batch [15] and streaming [3] inference algorithms for DP mixtures (without being derived from the ADF framework). In the batch case, $q^{\mathrm{pr}}(z_n)$ was evaluated by sampling, and in the latter case a heuristic approximation was used. We instead use a principled variational approximation to evaluate Eq. (16) which extends to a large class of NRMs.

The combinatorial sum over $z_{1:n-1}$ embedded in evaluating $q^{\mathrm{pr}}(z_n)$ appears to be a daunting barrier to efficient streaming inference. However, as we show in Sec. 3.1, for the models we consider the resulting q^{pr} can be written in terms of sums of local soft assignments, $\sum_{i=1}^{n-1} \hat{q}_{n-1}(z_i)$. Since these past soft assignments remain unchanged, the sum—instead of past assignment histories—can be stored as a sufficient statistic. Furthermore, since $p(z_n|z_{1:n-1})$ places mass on z_n taking a previously unseen component, the approximation $q^{\mathrm{pr}}(z_n)$ inherits this ability and allows our algorithm to introduce new components when needed. This is a crucial feature of our approach that enables our approximate inference scheme to maintain the benefits of nonparametric modeling, and is in contrast to approaches based on truncations to the underlying NRM or on heuristics for creating new clusters.

As in EP [6], the optimal update for the global parameters, θ_k, is also proportional to the marginal:

$$\hat{q}_n(\theta_k) \propto \sum_{z_{1:n}} \int p(x_n|z_n, \theta) p(z_n|z_{1:n-1}) \hat{q}_{n-1}(z_{1:n-1}, \theta) d\theta_{\setminus k}. \tag{17}$$

Eq. (17) is often intractable so we use the conjugate variational Bayes update for θ_k as in [15]:

$$\log q(\theta_k) \approx \mathbb{E}_{\theta_{\setminus k}, z_n} \log[p(x_n|z_n, \theta) \hat{q}_{n-1}(\theta)] + C, \tag{18}$$

where C is a constant. See the Supplement for details. The expectation is taken with respect to the optimal distributions $\hat{q}_n(z_n)$ and $\hat{q}_n(\theta_{\setminus k}) = \prod_{j \neq k} \hat{q}_n(\theta_j)$. This implies that

$$\log q(\theta_k) \approx \hat{q}_n(z_{nk}) \log p(x_n|z_n, \theta) + \log \hat{q}_{n-1}(\theta_k) + C'. \tag{19}$$

For the conjugate models we consider, Eq. (19) leads to tractable updates. Our streaming algorithm, which we refer to as *ADF-NRM*, proceeds at each step by first computing the local update in Eq. (16), and then the global update in Eq. (19). See Alg. 1.

3.1 Predictive Rule for NGGPs

A key part of the streaming algorithm is efficiently computing $q^{\mathrm{pr}}(z_n)$. When a DP prior is used, $q^{\mathrm{pr}}(z_n)$ admits a simple form similar to the CRP:

$$q^{\mathrm{pr}}(z_{nk}) \propto \begin{cases} \sum_{i=1}^{n-1} \hat{q}_i(z_{ik}), & k \leq K_{n-1} \\ a, & k = K_{n-1} + 1, \end{cases} \tag{20}$$

where K_{n-1} is the number of considered components in $x_{1:n-1}$ (see Sec. 3.3). Unfortunately, NRMs do not admit such a straightforward expression for $q^{\mathrm{pr}}(z_n)$ since in general $p(z_n|z_{1:n-1})$ is not known in closed form and for NGGPs it is given by a computationally demanding and numerically unstable expression [16] unsuitable for large, streaming data.

Instead, as in Eq. (5), we can introduce an auxiliary variable, U_n, to obtain a tractable variational approximation for NRMs, as detailed in the Supplement. We focus on the popular case of the NGGP here.

We rewrite $q^{\mathrm{pr}}(z_n)$ in terms of U_{n-1} and the unnormalized masses, π, and integrate over these variables:

$$q^{\mathrm{pr}}(z_n) = \sum_{z_{1:n-1}} \iint \Bigg[p(z_n|\pi) p(\pi|U_{n-1}, z_{1:n-1}) \times p(U_{n-1}|z_{1:n-1}) \prod_{i=1}^{n-1} \hat{q}_{n-1}(z_i) \Bigg] dU_{n-1} d\pi. \tag{21}$$

The term $p(\pi|U_n, z_{1:n-1})$ is stated in the Supplement and $p(U_{n-1}|z_{1:n-1})$ is shown in Eq. (6). The random measure π consists of a set of instantiated atoms, $\pi_1, \ldots, \pi_K$, and a Poisson process π^* representing the remaining mass. Since the integral in Eq. (21) is intractable, we introduce a partially factorized approximation: $p(\pi|U_{n-1}, z_{1:n-1})p(U_{n-1}|z_{1:n-1}) \approx q(\pi|U_{n-1})q(U_{n-1}) \in \mathcal{Q}_{\pi\times U}$ and solve

$$\arg\min_{q\in\mathcal{Q}_{\pi\times U}} \mathrm{KL}\Big(q(\pi|U_{n-1})q(U_{n-1})\hat{q}(z_{1:n-1})|| \; p(\pi|U_{n-1}, z_{1:n-1})p(U_{n-1}|z_{1:n-1})\hat{q}(z_{1:n-1})\Big). \tag{22}$$

The optimal distributions are given by:

$$q(U_{n-1}) \propto e^{-\frac{a}{\sigma}(U_{n-1}+\tau)^\sigma} \frac{U_{n-1}^{n-1}}{(U_{n-1}+\tau)^{n-1-a\mathbb{E}_{\hat{q}}[K_{n-1}]}} \tag{23}$$

$$q(\pi_k|U_{n-1}) \propto \pi_k^{\mathbb{E}_{\hat{q}}[n_k]} e^{-U_{n-1}\pi_k}\lambda(d\pi_k), \tag{24}$$

where $\mathbb{E}_{\hat{q}_{n-1}}[K_{n-1}]$ is the expected number of clusters observed so far, which can be recursively computed as described in the Supplement, and $\mathbb{E}_{\hat{q}_{n-1}}[n_k] = \sum_{i=1}^{n-1}\hat{q}_{n-1}(z_{ik})$ is the expected number of assignments to component k. The variational distribution of π^* is a Poisson process with tilted Lévy measure $e^{U_{n-1}\pi}\lambda(d\pi)$. As detailed in the Supplement, using these variational approximations in Eq. (21) combined with a delta function approximation to $q(U_{m-1})$ yields:

$$q^{\mathrm{pr}}(z_{nk}) \propto \begin{cases} \max\left(\sum_{i=1}^{n-1}\hat{q}_i(z_{ik}) - \sigma, 0\right), & k \le K_{n-1} \\ a(\hat{U}_{n-1}+\tau)^\sigma, & k = K_{n-1}+1, \end{cases} \tag{25}$$

where $\hat{U}_{n-1} = \arg\max q(U_{n-1})$. For the DP ($\sigma = 0$), Eq. (25) reduces to Eq. (20) and the resulting algorithm reduces to that of [3]. Note the differences between Eqs. (25) and (5) and between Eqs. (23) and (6). In both cases, hard assignments are replaced by soft assignments. As previously noted, the sum of these past soft assignments serve as sufficient statistics, and since they do not change between iterations, can be stored in place of individual assignments. Furthermore, the recursive computation of $\mathbb{E}_{\hat{q}_{n-1}}[K_{n-1}]$ in Eq. (23) allows past assignments to be discarded.

3.2 Computational Complexity

Due to the streaming nature of the ADF-NRM algorithm, we analyze the per-observation computational complexity. As seen in Alg. 1, for each observation we compute a finite dimensional probability vector with $K_n + 1$ elements, which is $O(K_n)$. Additionally, we need to compute $\hat{U}_n$ via numerical optimization of $q(U_n)$, which is a univariate and unimodal function so can be maximized efficiently with complexity denoted $O(\mathcal{U})$. Thus, the per-iteration computational complexity of ADF-NRM is $O(K_n + \mathcal{U})$. We note that in practice the runtime is dominated by the $O(K_n)$ term due to the NGGP introducing many clusters; the optimization of $\hat{U}_n$ terminates in a few iterations (independent of K_n) and so does not limit the scalability of the algorithm. It is known that $\mathbb{E}[K_n] \simeq a\log n$ for the DP and follows a power-law with index $\sigma \in (0,1)$ for the NGGP [17]. This implies that for large n the complexity of ADF-NRM with a NGGP is larger than that with a DP, but is sub-linear in n and so remains computationally feasible. Of course, a posteriori K_n can grow much more slowly in practice when the data has a compact representation.

Algorithm 1 ADF for NRM mixture models

Initialize: $K = 1, S_1 = 1$
$\hat{q}_1(\theta_1) \propto p(x_1|\theta_1)p(\theta_1)$, $\hat{q}_1(z_{11}) = 1$
for $n = 1$ to ∞ **do**
  $\hat{U}_n = \arg\max q(U_n)$ with $q(U_n)$ in Eq. (23)
  for $k = 1$ to K **do**
    $q^{\mathrm{pr}}(z_{nk}) \propto \max(S_k - \sigma, 0)$
    $\hat{q}_n(z_{nk}) \propto q^{\mathrm{pr}}(z_{nk}) \int p(x_n|z_{nk}, \theta_k)\hat{q}_{n-1}(\theta_k)d\theta_k$
  end for
  $q^{\mathrm{pr}}(z_{n,K+1}) \propto a(\hat{U}_n + \tau)^\sigma$
  $\hat{q}_n(z_{n,K+1}) \propto q^{\mathrm{pr}}(z_{n,K+1}) \times \int p(x_n|z_{n,K+1}, \theta)p(\theta_{K+1})d\theta_{K+1}$
  normalize $\hat{q}_n(z_{n,1:K+1})$
  if $\hat{q}_n(z_{n,K+1}) > \epsilon$ **then**
    $S_{K+1} = 0$, $\hat{q}_{n-1}(\theta_{K+1}) = p(\theta_{K+1}), K = K+1$
  else
    normalize $\hat{q}_n(z_{n,1:K})$
  end if
  for $k = 1$ to K **do**
    $\hat{q}_n(\theta_k) \propto p(x_n|z_{nk}, \theta_k)^{\hat{q}_n(z_{nk})}\hat{q}_{n-1}(\theta_k)$
    $S_k = S_k + \hat{q}_n(z_{nk})$
  end for
end for

3.3 Efficiently Coping with New Clusters

While the probability that a data point belongs to a new cluster, $\hat{q}_n(z_{n,K+1})$, is always greater than zero, it is computationally infeasible to introduce a new component at each iteration since the per iteration complexity of ADF-NRM is $O(K_n)$. In practice, new components are added only if $\hat{q}_n(z_{n,K_n+1}) > \epsilon$ for $\epsilon \ge \sigma$ a threshold. The restriction $\epsilon \ge \sigma$ is natural: if $\hat{q}_n(z_{n,K_n+1}) < \sigma$ then $K_n + 1$ will be assigned zero prior probability at step $n+1$ in Eq. (25) and will be effectively removed. The threshold parameter explicitly controls the trade off between accuracy and speed; a larger threshold introduces fewer clusters leading to a worse variational approximation but faster run times.

One can view our thresholding as an adaptive truncation of the posterior, in contrast to the common approach of truncating the component prior.

During execution of ADF-NRM and EP-NRM of Sec. 3.4, redundant clusters can be created due to the order of observations processed. As in [3], we introduce merge steps to combine distinct clusters that explain similar observations. Since a benefit of the NGGP over the DP is the addition of many small but important clusters (see Sec. 4), we found that frequent merging degrades predictive performance of NGGP models by prematurely removing these clusters. In our experiments, we only merge clusters whose similarity exceeds a conservatively large merge threshold.

3.4 Extension to EP

For data sets of fixed size, N, ADF-NRM can be extended to *EP-NRM* for batch inference analogously to Sec. 2.4. Assume we have both an approximation to the batch posterior $\hat{q}(\theta, z_{1:N})$ and local contributions $\bar{q}_j(\theta, z_j)$ for $j = 1, \ldots, N$, both of which can be computed using ADF. In particular, $\hat{q}(\theta, z_{1:N}) = \hat{q}_N(\theta, z_{1:N})$, the final ADF posterior approximation, and $\bar{q}_j(\theta, z_j) \propto \frac{\hat{q}_j(\theta, z_{1:j})}{\hat{q}_{j-1}(\theta, z_{1:(j-1)})}$, the ratio between successive ADF approximations. Now define

$$\hat{q}_{\setminus j}(\theta, z_{\setminus j}) \propto \frac{\hat{q}(\theta, z_{1:N})}{\bar{q}_j(\theta, z_j)} \tag{26}$$

to be the approximate posterior with x_j removed. We refine $\hat{q}(\theta, z_{1:N})$ by projecting $\hat{q}_{\setminus j}(\theta, z_{\setminus j}) f_j(\theta, z_{1:N}) = \hat{q}_{\setminus j}(\theta, z_{\setminus j}) p(x_j|z_j, \theta) p(z_j|z_{\setminus j})$ onto $\mathcal{Q}$ using Eq. (8). Similar to ADF, the updated soft assignment for z_j is given by $\hat{q}(z_{jk}) \propto q^{\text{pr}}_{\setminus j}(z_{jk}) \int p(x_j|z_{jk}, \theta) \hat{q}_{\setminus j}(\theta_k) d\theta_k$ where $q^{\text{pr}}_{\setminus j}$ is the approximate predictive distribution given all other soft assignments. The approximate global update is given by $\hat{q}(\theta_k) \propto p(x_j|z_k, \theta)^{\hat{q}(z_{jk})} \hat{q}_{\setminus j}(\theta_k)$. The jth local contribution is then updated to

$$\bar{q}_j(\theta, z_j) \propto \frac{\hat{q}(\theta, z_{1:N})}{\hat{q}_{\setminus j}(\theta, z_{\setminus j})}. \tag{27}$$

We cycle through the data set repeatedly, at each stage applying the steps above, until convergence.

For conjugate exponential families, the computations required for the global cluster parameters, θ, in Eq. (26) and Eq. (27) reduce to updating sufficient statistics as in Sec. 2.4. $q^{\text{pr}}_{\setminus j}$ for NGGPs may similarly be updated on each round by letting $S_k = \sum_{i=1}^{N} \hat{q}(z_{ik})$ and $S_{k,\setminus j} = S_k - \hat{q}(z_{jk})$, where $\hat{q}(z_{ik})$ are the current soft assignments. Under the same logic as Eq. (25), $q^{\text{pr}}_{\setminus j}$ for instantiated clusters is approximated by

$$q^{\text{pr}}_{\setminus j}(z_{jk}) \propto \max(S_{k,\setminus j} - \sigma, 0), \tag{28}$$

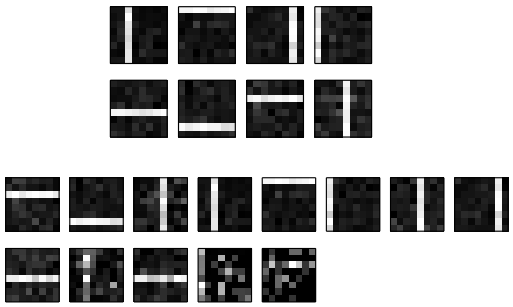

Figure 1: ADF-NRM posterior mean mixture components for the synthetic bars data set with (*top*) and without (*bottom*) merge steps.

and $q^{\text{pr}}_{\setminus j}(z_{j,K+1})$ follows analogously (see Supplement). After computing the refined soft assignment, $\hat{q}(z_{jk})$, we update $S_k = S_{k,\setminus j} + \hat{q}(z_{jk})$. As a consequence of this approach, the total weight on an instantiated cluster k, S_k, can become small upon revisits of the data assignments. In practice, we remove cluster k if $S_k < \epsilon$, where ϵ is as in Sec. 3.3.

4 Experiments

We evaluate ADF-NRM on both real and synthetic data using the task of document clustering. Each document is represented by a vector of word counts, $x_d \in \mathbb{R}^V_+$, where V is the size of the vocabulary, and x_{dw} is the number of occurrences of word w in document d. We then model the corpus as a NGGP mixture of multinomials; that is, our data are generated as in Eq. (3) with $x_d \sim \text{Mult}(N_d, \theta^{z_d})$, where N_d is the number of words in document d and θ_k is a vector of word probabilities in cluster k. We take H_0 to be Dirichlet such that $\theta_k \sim \text{Dir}(\alpha)$. We then use our proposed algorithms to perform inference over $\{z_d\}$ and $\{\theta_k\}$.

We focus on comparing the IG ($\sigma = 0.5$) to the DP ($\sigma = 0$). The choices of α used to set the Dirichlet base measure in our various experiments are discussed in the Supplement. To select the NRM hyperparameters a and τ, we adapt the grid-search method used for the sampling-based batch procedure of [4] to our streaming setting. As detailed in the Supplement, we perform a preliminary analysis on a small subset of the data. Our algorithm is then let loose on the remaining data with these values fixed.

4.1 Synthetic Bars

First, we perform clustering on a synthetic data set of 8×8 images to show that ADF-NRM can recover the correct component distributions. Each image is represented by a vector of positive integer pixel intensities, which we interpret as a document over a vo-

cabulary with 64 terms. The clusters correspond to horizontal and vertical bars with an additive baseline to ensure cluster overlap. Each of 200 images is generated by first choosing a cluster, z_d, and then sampling pixel intensities $x_d \sim \text{Mult}(50, \theta^{z_d})$. Fig. 1 depicts the resulting ADF-NRM posterior mean mixture components under the learned variational distribution, $\mathbb{E}_{\hat{q}_N}[\theta_k]$, based on an IG prior ($\sigma = 0.5$), both with and without merge moves. We see that in both cases the algorithm learns the correct clusters, but merge moves remove redundant and extraneous clusters.

4.2 Synthetic Power-Law Clusters

To explore the benefit of the additional flexibility of IGs over DPs, we generated 10,000 synthetic documents, x_d, from a Pitman-Yor(.75, 1) mixture of multinomials. The Pitman-Yor prior is another commonly used BNP prior famous for its ability to model clusters whose sizes follow certain power-law distributions [18].

We assess the ADF-NRM predictive log-likelihood and inferred number of clusters versus number of observed documents. For each model, we selected hyperparameters based on a randomly selected set of $1,000$ documents. We then continue our algorithm on $7,000$ training documents and use the remaining $2,000$ for evaluation. Mean predictive log-likelihoods, number of clusters, and error estimates were obtained by permuting the order of the training documents 5 times. We compare our ADF-NRM performance to that of a baseline model where the cluster parameters are inferred based on ground-truth-labeled training data. Lastly, after the completion of ADF, we performed 49 additional passes through the data using EP-NRM to obtain refined predictions and number of clusters.

We see in Fig. 2 that both the IG and DP models perform similarly for small n, but as the amount of data increases, the IG provides an increasingly better fit in terms of both predictive log-likelihood and number of clusters. This substantiates the importance of our streaming algorithm being able to handle a broad class of NRMs. Furthermore, after a single data pass, ADF-NRM comes close to reaching the baseline model even with the IG/Pitman-Yor model mismatch. It is also evident in Fig. 2 that additional EP iterations both improve predictions and the match between inferred and true number of clusters for both prior specifications.

4.3 KOS Blog Corpus

We also applied ADF-NRM to cluster the KOS corpus of 3,430 blog posts [19]. The fact that the corpus is small enough to use non-streaming (batch) inference algorithms allows us to compare ADF-NRM, EP-NRM, and the collapsed Gibbs sampler for NGGP

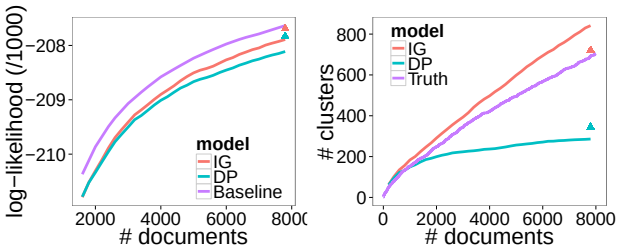

Figure 2: Mean predictive log-likelihood (*left*) and number of clusters (*right*) for the DP (*cyan*) and IG (*red*) priors on the synthetic power-law data set using ADF-NRM. Triangles indicate final values for EP-NRM after 50 epochs. The ground-truth model is shown in purple. Error bars are omitted due to their small size relative to the plot scale.

Table 1: Mean predictive performance and number of clusters ($\pm$ 1 std. err.) for ADF-NRM, EP-NRM, and a collapsed Gibbs sampler on the KOS corpus.

Method	Pred. log-lik	#Clusters	Epochs
ADF-DP	-346023 $\pm$ 165	80 $\pm$ .17	1
ADF-IG	-345588 $\pm$ 159	92 $\pm$ .18	1
EP-DP	-342535 $\pm$ 181	104 $\pm$ 2.4	50
EP-IG	-342195 $\pm$ 161	114 $\pm$ 1.5	50
Gibbs-DP	-342164 $\pm$ 11	119 $\pm$ 0.3	215
Gibbs-IG	-341468 $\pm$ 338	128 $\pm$ 1.3	215

mixture models presented in [5]. Importantly, we only compare to Gibbs, which is not suited to the streaming setting, in an attempt to form a gold standard. (Recall that Gibbs targets the exact posterior in contrast to our variational-based approach, and we do not expect mixing to be an issue in this modest-sized data set.)

We evaluated performance as in Sec. 4.2. Here, we held out 20% of the entire corpus as a test set and trained (given the hyperparameters determined via grid search) on the remaining 80% of documents. The ADF-NRM predictive log-likelihoods for the IG and DP were computed after a single pass through the data set while those for EP-NRM were computed by cycling through the data set 50 times. Error estimates were obtained by permuting the order of the documents 20 times. Predictions for the collapsed Gibbs sampler were computed by running 5 chains for 215 passes through the data and averaging the predictive log-likelihood for the last 50 samples across chains.

The comparisons between all methods are depicted in Table 1. For all algorithms (ADF, EP, and Gibbs) the added flexibility of the IG provides a better fit in terms of predictive log-likelihood. The additional ≈ 10 clusters associated with the IG for all algorithms correspond to small clusters which seem to capture finer-scale latent structure important for prediction. Although performance increases moving from the one-pass ADF-NRM to multi-pass EP-NRM, Fig. 3 displays that the most significant gains occur in the initial

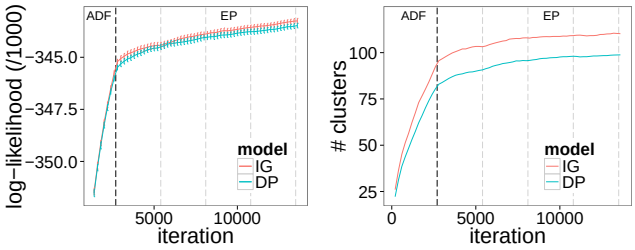

Figure 3: Predictive log-likelihood (*left*) and mean number of clusters (*right*) using EP-NRM on KOS corpus. Vertical lines indicate epochs and error bars ± 1 st. dev..

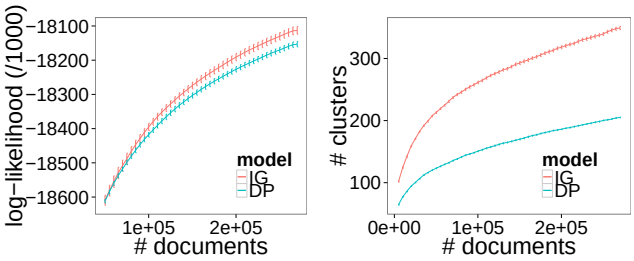

Figure 4: Comparison of (*left*) predictive log-likelihood and (*right*) number of clusters using ADF-NRM on the New York Times corpus for the IG and DP priors.

Topic 1	Topic 2	Topic 3	Topic 4
athletes (.83)	merger (.36)	reform (.31)	quarterback (.45)
weight (.75)	revenue (3.3)	conservative (.26)	yankees (.45)
exercise (.68)	shares (.31)	senator (.24)	scored (.43)
steroid (.55)	cable (.31)	parties (.22)	pitcher (.38)
supplement (.49)	businesses (.29)	supporter (.22)	offense (.37)

Figure 5: Most probable words and their respective contributions (in %) for the 4 most prevalent topics.

epoch. In fact, after a single epoch ADF performs significantly better than a single epoch of Gibbs; it takes about three Gibbs epochs to reach comparative performance (see Supplement). Finally, while the IG Gibbs sampler leads to the best performance, EP-NRM with the IG prior is competitive and reaches similar performance to the DP using Gibbs.

In summary, ADF-NRM provides competitive performance with only a single pass through the data; more refined approximations nearly matching the computationally intensive sampling-based approaches can be computed via EP-NRM if it is feasible to both save and cycle through the data.

4.4 New York Times Corpus

We performed streaming inference on a corpus of 300,000 New York Times articles [19]. We first identified a vocabulary of $7,841$ unique words by removing words occurring in fewer than 20 and more than 90% of documents, as well as terms resulting from obvious errors in data acquisition. Then, we removed documents containing fewer than 20 words in our vocabulary, resulting in a corpus of 266,000 documents. The corpus is too large for batch algorithms, so we focus on ADF-NRM comparing the DP and IG priors.

We determined hyperparameters as before and held out $5,000$ documents as a test set, evaluating the predictive log-likelihood and number of clusters after every $5,000$ training documents were processed. See Fig. 4. As before, the IG obtains superior predictive log-likelihood and introduces many additional small clusters compared to the DP, suggesting that the IG may be able to capture nuanced latent structure in the corpus that the DP cannot (see the Supplement for more details). Reassuringly, the recovered clusters with highest weights correspond to interpretable topics (Fig. 5). Yet again, we see the benefits of being able to consider NRMs beyond the DP, which to date has been the most widely considered BNP prior largely due to the computational tools developed for it.

5 Discussion

We introduced the ADF-NRM algorithm, a variational approach to streaming approximate posterior inference in NRM-based mixture models. Our algorithm leverages the efficient sequential updates of ADF while importantly maintaining the infinite-dimensional nature of the BNP model. The key to tractability is focusing on approximating a partial-urn characterization of the NRM predictive distribution of cluster assignments. We also showed how to adapt the single-pass ADF-NRM algorithm to a multiple-pass EP-NRM variant for batch inference. Our empirical results demonstrated the effectiveness of our algorithms, and the importance of considering NRMs beyond the DP.

A potential drawback of the EP-NRM scheme is that each observation needs to store its variational distribution over cluster assignments. An interesting question is whether the local distributions can be grouped and memoized [20] to both save computation and perform data-driven split-merge moves. This combined with simple parallel EP schemes [14, 21] would scale EP-NRM to massive data sets.

Instead of examining predictive distributions and exploiting the NRM partial-urn scheme, a natural question is whether similar algorithms can be developed that do not integrate out the underlying measure. Such algorithms would be directly applicable to hierarchical BNP models such as topic models and hidden Markov models [22].

Acknowledgements: This work was supported in part by DARPA Grant FA9550-12-1-0406 negotiated by AFOSR, ONR Grant N00014-10-1-0746, and the TerraSwarm Research Center sponsored by MARCO and DARPA. AT was partially funded by an IGERT fellowship.

References

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

[2] T. Broderick, N. Boyd, A. Wibisono, A. C. Wilson, and M. I. Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, 2013.

[3] D. Lin. Online learning of nonparametric mixture models via sequential variational approximation. In *Advances in Neural Information Processing Systems*. 2013.

[4] E. Barrios, A. Lijoi, L. E. Nieto-Barajas, and I. Prunster. Modeling with normalized random measure mixture models. *Statistical Science*, 28(3):313–334, 08 2013.

[5] S. Favaro and Y. W. Teh. MCMC for normalized random measure mixture models. *Statistical Science*, 28(3):335–359, August 2013.

[6] T. Minka. Expectation propagation for approximate Bayesian inference. In *Advances in Neural Information Processing Systems*, 2001.

[7] J. F. C. Kingman. Completely random measures. *Pacific Journal of Mathematics*, 21(1):59–78, 1967.

[8] J. F. C. Kingman. *Poisson Processes*. Oxford University Press, 1993.

[9] T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *Annals of Statistics*, 1(2):209–230, 1973.

[10] N. L. Hjort. Nonparametric Bayes estimators based on beta processes in models for life history data. *Annals of Statistics*, 18(3):1259–1294, 09 1990.

[11] L. F. James, A. Lijoi, and I. Prunster. Posterior analysis for normalized random measures with independent increments. *Scandinavian Journal of Statistics*, 36(1):76–97, 2009.

[12] E. Regazzini, A. Lijoi, and I. Prunster. Distributional results for means of normalized random measures with independent increments. *Annals of Statistics*, 31(2):560–585, 04 2003.

[13] J. E. Griffin and S. G. Walker. Posterior simulation of normalized random measure mixtures. *Journal of Computational and Graphical Statistics*, 20(1):241–259, 2011.

[14] A. Gelman, A. Vehtari, P. Jylänki, C. Robert, N. Chopin, and J. P. Cunningham. Expectation propagation as a way of life. *ArXiv e-prints*, December 2014.

[15] C. Wang and D. M. Blei. Truncation-free online variational inference for Bayesian nonparametric models. In *Advances in Neural Information Processing Systems*. 2012.

[16] Antonio Lijoi, Ramsés H. Mena, and Igor Prünster. Controlling the reinforcement in bayesian non-parametric mixture models. *Journal of the Royal Statistical Society: Series B*, 69(4):715–740, 2007.

[17] S. Favaro, A. Lijoi, and I. Prunster. Asymptotics for a bayesian nonparametric estimator of species variety. *Bernoulli*, 18(4):1267–1283, 11 2012.

[18] J. Pitman and M. Yor. The two-parameter poisson-dirichlet distribution derived from a stable subordinator. *Annals of Probability*, 25(2):855–900, 04 1997.

[19] K. Bache and M. Lichman. UCI machine learning repository, 2013.

[20] M. C Hughes and E. Sudderth. Memoized online variational inference for Dirichlet process mixture models. In *Advances in Neural Information Processing Systems*. 2013.

[21] M. Xu, Y. W. Teh, J. Zhu, and B. Zhang. Distributed context-aware bayesian posterior sampling via expectation propagation. In *Advances in Neural Information Processing Systems*, 2014.

[22] Y. W. Teh and M. I. Jordan. Hierarchical Bayesian nonparametric models with applications. In N. Hjort, C. Holmes, P. Müller, and S. Walker, editors, *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, 2010.