Ordered Stick-Breaking Prior for Sequential MCMC Inference of Bayesian Nonparametric Models

Mrinal Das[†] MRINAL@CSA.IISC.ERNET.IN
Trapit Bansal[†] TRAPIT@CSA.IISC.ERNET.IN
Chiranjib Bhattacharyya[†] CHIRU@CSA.IISC.ERNET.IN
[†]Department of Computer Science and Automation,
Indian Institute of Science, Bangalore, India

Abstract

This paper introduces *ordered stick-breaking process* (OSBP), where the atoms in a stick-breaking process (SBP) appear in order. The choice of weights on the atoms of OSBP ensure that; (1) probability of adding new atoms exponentially decrease, and (2) OSBP, though non-exchangeable, admit predictive probability functions (PPFs). In a Bayesian nonparametric (BNP) setting, OSBP serves as a natural prior over sequential *mini-batches*, facilitating exchange of relevant statistical information by sharing the atoms of OSBP. One of the major contributions of this paper is SUMO, an MCMC algorithm, for solving the inference problem arising from applying OSBP to BNP models. SUMO uses the PPFs of OSBP to obtain a Gibbs-sampling based truncation-free algorithm which applies generally to BNP models. For large scale inference problems existing algorithms such as particle filtering (PF) are not practical and variational procedures such as TSVI (Wang & Blei, 2012) are the only alternative. For Dirichlet process mixture model (DPMM), SUMO outperforms TSVI on perplexity by 33% on 3 datasets with million data points, which are beyond the scope of PF, using only 3GB RAM.

1. Introduction

Bayesian nonparametric (BNP) models are powerful tools for understanding probabilistic relationships (Hjort et al., 2010). Inference in BNP models are generally intractable. Markov chain Monte Carlo (MCMC) (Andrieu et al., 2003) based procedures being easy to implement and more accurate than variational inference (Blei & Jordan, 2004; Welling et al., 2012) are often preferred. However MCMC procedures do not scale well to large datasets.

For large datasets one could consider sequentially processing mini-batches of observations. Particle filtering (PF) is a principled technique which sequentially approximates the full posterior using particles (Doucet et al., 2001; Ulker et al., 2010; Andrieu et al., 2010; Fearnhead, 2004; Jun & Coute, 2014). However, due to the nature of recursive dependence structure employed in PF, it needs to maintain multiple configurations of variables, which makes it practical *only in distributed setting* (Canini et al., 2009; Ahmed et al., 2011). Keeping this motivation in mind truncation-free stochastic variational inference (TSVI) was developed (see Section 1, para 2 of Wang & Blei (2012)). Indeed, for pure sequential inference, stochastic variational inference (SVI) has become the state of the art in large scale inference of BNP models (Bryant & Sudderth, 2012; Lin, 2013; Broderick et al., 2013). We aim to develop MCMC procedures which can compete with TSVI on scale and accuracy. In particular we consider the situation where data arrives in *mini-batches* and keeping in mind a true Bayesian spirit we wish to endow the mini-batches with a suitable prior.

Stick-breaking process (SBP) (Ishwaran & James, 2001) gives a constructive definition for designing *atomic* probability measures which can serve as priors for BNP models. Indeed popular priors such as Dirichlet process (DP) (Ferguson, 1973), and Pitman-Yor process (PYP) (Pitman & Yor, 1997) are special cases of SBP. In this paper we study an interesting variation of SBP, where the atoms *appear in order*. Appearing in order has been noted by Pitman (1995), however we did not find any literature in the area which models it in a prior and subsequently applies it to Bayesian nonparametrics. The goal of this paper is to explore *SBP with atoms appearing in order* as a prior over mini-batches and develop sequential MCMC inference that could compete with TSVI on scale and PF on accuracy.

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37.

Contributions. Our contribution is two-fold in this paper. The main technical contribution is to propose a novel BNP prior, **ordered stick-breaking process** (OSBP), where atoms in the stick-breaking framework *appear in order*. An interesting property of OSBP is that, probability of new atom can decrease exponentially as more and more atoms arrive (see Theorem 2). Predictive probability functions (PPFs) are useful tools for designing MCMC based truncation-free inference algorithms. In general, SBP based priors do not admit PPFs except in special cases like DP and PYP. Surprisingly, despite having parameter setting as general as SBP, one can derive the PPFs of OSBP (see Theorem 3), making it an extremely attractive candidate to be used as a prior for BNP models. Our second contribution is to apply OSBP to exchange *relevant statistical information in mini-batches* by sharing atoms. We describe SeqUential MCMC inference through OSBP (SUMO), an MCMC inference algorithm, which can approximate the full posterior distribution for a general class of BNP models by sharing atoms of OSBP across mini-batches. Memory overhead of SUMO is low as the memory scales with the number of atoms that needs to be shared and due to OSBP only a small number of atoms need to be shared (see Theorem 2). SUMO thus marks a significant progress on developing sequential MCMC algorithms for massive datasets. In our experiments with three publicly available large scale corpora namely New York Times (100M tokens), PubMed abstracts (730M tokens) and Wikipedia English subset (296M tokens), using only 3GB RAM, SUMO with Dirichlet process mixture models (DPMM) outperforms TSVI (Wang & Blei, 2012) in terms of perplexity by 33% with competitive run-time memory usage. Solving such large scale inference problem in sequential setting is beyond the scope of MCMC and PF.

Structure of the paper. In Section 2, we propose OSBP and derive PPF. In Section 3, we apply OSBP on BNP models and derive the sequential inference procedure SUMO. Discussing relevant work in Section 4, we provide the empirical study in Section 5 on four real life datasets comparing SUMO with MCMC, PF and TSVI.

Notation. We will use following notations throughout the paper. Γ is a *diffuse* probability measure over a suitable measurable space $(\Omega, \mathcal{B})$, more precisely for any $y \in \Omega$, $\Gamma(y) = 0$. δ_y will denote an atomic probability measure, the entire probability mass being concentrated at y. $\mathbb{E}[\mathrm{X}]$ is the expectation of random variable X. A set of variables $\{x_1, x_2, \ldots, x_n\}$ will be denoted by $x_{1:n}$. $\{x_j\}$ will denote an infinite set, and (x_j) will denote an infinite sequence, j specifying the order. The set of integers $\{1, \ldots, k\}$ will be denoted by $[k]$. $\mathbb{I}[.]$ denotes the indicator function, and $|.|$ means cardinality. $\mathbf{N}$ is the set of all positive integers. If P and Q are two measures we will use $\mathrm{P} = \mathrm{Q}$ to denote that P and Q are same i.e. $\forall \mathrm{B} \in \mathcal{B}$, $\mathrm{P}(\mathrm{B}) = \mathrm{Q}(\mathrm{B})$.

2. Ordered stick-breaking process: Stick breaking with atoms appearing in order

In this section we propose the *ordered stick-breaking process* (OSBP). We begin by recollecting some relevant preliminaries for defining OSBP.

2.1. Preliminaries

Stick-breaking process. Any almost sure (a.s.) discrete probability measure G is a stick-breaking process (SBP) (Ishwaran & James, 2001) if it can be represented as

$$\mathrm{G} = \textstyle\sum_{j=1}^{\infty} \theta_j \delta_{\beta_j}, \theta_1 = v_1,\ \theta_j = v_j \prod_{l=1}^{j-1}(1 - v_l)$$
$$a_j, b_j > 0,\ v_j \sim Beta(a_j, b_j),\ \ \beta_j \sim \mathrm{H} \qquad (1)$$

H is a *diffuse measure* over a measurable space $(\Omega, \mathcal{B})$ and $\{a_j, b_j\}$ are set of parameters.

Order of the atoms in SBP. The constructive definition of SBP in Eq. (1) allows us to define an *order* among the atoms. Let $(\mathrm{Y}_1, \mathrm{Y}_2, \ldots, \mathrm{Y}_n)$ denotes n random samples drawn from G in Eq. (1). If $j < l$, then $p(\mathrm{Y}_i = \beta_j | v_1, \ldots, v_j, \ldots, v_l, \ldots) = p(\mathrm{Y}_i = \beta_j | v_1, \ldots, v_{j-1}, v_j)$. Therefore, we get a strict ordering among the atoms $\{\beta_j\}$ which is defined by their indices j in Eq. (1).

Appearance in order (Pitman, 1995). Let $(\mathrm{Y}_i)_1^t$ be sequence of random variables with values in some measurable space $(\Omega, \mathcal{B})$, and $\{\bar{\mathrm{Y}}_1, \bar{\mathrm{Y}}_2, \ldots, \bar{\mathrm{Y}}_{k_t}\}$ denotes the set of unique values among $(\mathrm{Y}_i)_1^t$. Define $\mathrm{B}_j = \{i | \mathrm{Y}_i = \bar{\mathrm{Y}}_j, i \in [t]\}$. The set $(\mathrm{Y}_i)_1^t$ is said to *appear in order* if $\mathrm{B}_1 \cup \mathrm{B}_2 \cup \ldots \cup \mathrm{B}_{k_t} = [t]$ where $1 \in \mathrm{B}_1$ and for $2 \leq j \leq k_t$, $\mathrm{B}_j \subseteq [t]$ is such that the least element of $[t] \setminus \cup_{l=1}^{j-1} \mathrm{B}_l$ belongs to B_j. $(\bar{\mathrm{Y}}_j)$ are also *in order* i.e if $\mathrm{Y}_i = \bar{\mathrm{Y}}_j$ and $\mathrm{Y}_m \neq \bar{\mathrm{Y}}_j$, $\forall m < i$ then $\nexists l < i$ such that $\mathrm{Y}_l = \bar{\mathrm{Y}}_{j+r}$ where $r \geq 0$. See that $\mathrm{B}_i \cap \mathrm{B}_j = \emptyset$ and hence B_j's form a partition and is known as an *ordered partition*.

See §S.3 in the supplementary material for an example. Note that, any appearing in order sequence of random variables are not exchangeable. But any non-exchangeable sequence of random variables do not follow appearance in order. The notion of ordered partition is important.

2.2. Ordered stick-breaking process

We define ordered stick-breaking process (OSBP) here for atoms appearing in order. We then discuss properties of OSBP in Lemma 1, Theorems 1, 2 and 3.

Let Γ be a *diffuse* probability measure over random measures, and $\boldsymbol{\mu}, \boldsymbol{\nu}$ denote the set of scalar hyper-parameters $\{\mu_j\}$ and $\{\nu_j\}$ respectively such that $0 < \mu_j < 1$, $\nu_j > 0$, $\forall j$. $(\mathrm{G}_1, \mathrm{G}_2, \ldots)$ is an *appearing in order* sequence of random measures. $(\mathrm{Q}_1, \ldots, \mathrm{Q}_{k_{t-1}})$ is the set of k_{t-1} unique values among $\mathrm{G}_{1:t-1}$. We define, $\mathrm{G}_1, \mathrm{G}_2, \ldots \sim$ OSBP$(\boldsymbol{\mu}, \boldsymbol{\nu}, \Gamma)$ if $\mathrm{G}_1 \sim \Gamma$ and for any $t \geq 2$,

the following holds:

$$\begin{aligned} \mathrm{G}_t\,|\mathrm{G}_{1:t-1},(\rho_j),\Gamma \;&\sim \textstyle\sum_{j=1}^{k_{t-1}}\rho_j\delta_{\mathrm{Q}_j}+\alpha_{k_{t-1}}\Gamma \\ \rho_1=v_1,\;\; \forall j>1,\;\rho_j&=v_j\textstyle\prod_{l=1}^{j-1}(1-v_l) \\ v_j|\mu_j,\nu_j &\sim Beta\,(\mu_j\nu_j,(1-\mu_j)\nu_j) \\ \alpha_{k_{t-1}}&=1-\textstyle\sum_{j=1}^{k_{t-1}}\rho_j \end{aligned} \tag{2}$$

Note that, $\mathrm{Q}_1 = \mathrm{G}_1$, and (Q_j) appear in the order given by the index j. Notice that, implicitly $\forall t, \alpha_{k_t} \geq 0$ and $\forall j, \rho_j \geq 0$, as well as $\sum_{j=1}^{k_{t-1}} \rho_j + \alpha_{k_{t-1}} = 1$. G_t can re-use existing Q_j with probability ρ_j when $k_t = k_{t-1}$. G_t can use a newly sampled value from base measure Γ with *innovation probability* $\alpha_{k_{t-1}}$, and in that case, $k_t = k_{t-1}+1$ and $\mathrm{Q}_{k_t} = \mathrm{G}_t$. Following preliminaries, we define

$$\mathrm{B}_j = \{t|\mathrm{G}_t = \mathrm{Q}_j\},\; z_t = j \text{ iff } \mathrm{G}_t = \mathrm{Q}_j \tag{3}$$

Notice that $(\mathrm{B}_1, \mathrm{B}_2, \ldots, \mathrm{B}_{k_t})$ is an *ordered partition* and $z_t \in [k_{t-1}+1]$ a.s. Γ being a diffuse measure, Q_{k_t} is a.s. distinct from $\mathrm{Q}_{1:k_{t-1}}$. To be definite, Γ can be DP.

2.3. Properties of OSBP

As the atoms in OSBP appear in order, OSBP forms a dynamic system that evolves with t, when

$$k_t \geq k_{t-1},\; \alpha_{k_t} \leq \alpha_{k_{t-1}} \tag{4}$$

This property of OSBP, can be seen directly from the definition and leads to an interesting result below regarding the asymptotic behavior of OSBP.

Theorem 1. *If* $\mathrm{P}_1 = \Gamma$, $\mathrm{P}_t = \sum_{j=1}^{k_{t-1}} \rho_j\delta_{\mathrm{Q}_j} + \alpha_{k_{t-1}}\Gamma$ *for* $t > 1$ *and* $\mathrm{P}^* = \sum_{j=1}^{\infty}\rho_j\delta_{\mathrm{Q}_j}$ *such that* $\sum_{j=1}^{\infty}\rho_j = 1$, *where* (ρ_j), (Q_j), α_{k_t} *and* Γ *as defined in Eq.* (2) *with parameter* $\boldsymbol{\mu}, \boldsymbol{\nu}$, *then* $\lim_{t\to\infty} \mathrm{P}_t = \mathrm{P}^*$ *a.s.*

Proof. See §S.4.1 in the supplementary material. □

Theorem 1 says that, OSBP asymptotically obtains a non-evolving probability measure a.s. when no new atom appears. Moreover, even with infinite number of atoms, OSBP gives a valid probability measure. This asymptotic property is significant to understand the behavior of OSBP based BNP models, for example P^* behaves like SBP and leads to independent and identically distributed or iid (hence exchangeable and marginally invariant) samples.

It also becomes important to understand how quickly P_t stops evolving, which depends on the probability masses over the atoms (ρ_j) and α_{k_t}. Notice that, (ρ_j) and α_{k_t} are defined with random variables (v_j) that introduces the parameters $(\boldsymbol{\mu}, \boldsymbol{\nu})$. From properties of Beta distribution, $\mathbb{E}[v_j] = \mu_j$ and $Var(v_j) = \frac{\mu_j(1-\mu_j)}{1+\nu_j}$. One can set expected value of v_j with μ_j, where precision is governed by ν_j. (ρ_j) follows a distribution as noted below.

Lemma 1. *For any* $t \in \mathbf{N}$, $\mathrm{R}_t = (\rho_1, \rho_2, \ldots, \rho_{k_{t-1}}, \alpha_{k_{t-1}})$ *as defined in Eq.* (2) *is distributed as generalized Dirichlet distribution (Connor & Mosimann, 1969). Furthermore, if* $(1-\mu_{j-1})\nu_{j-1} = \nu_j$ *for* $j,\; 2 \leq j \leq k_{t-1}$, *then* $\mathrm{R}_t \sim Dirichlet(\mu_1\nu_1, \mu_2\nu_2, \ldots, \mu_{k_{t-1}}\nu_{k_{t-1}}, (1-\mu_{k_{t-1}})\nu_{k_{t-1}})$.

Proof. See §S.4.2 in the supplementary material. □

Next we state one important result which says probability of adding new atoms can decrease exponentially with time.

Theorem 2. *For* α_{k_t} *as defined in Eq.* (2) *with parameters* $\boldsymbol{\mu}, \boldsymbol{\nu}$, *and any* $\epsilon \in (0,1)$, *if* $\mu_j > 1/2$ *for all* j, *then* $\alpha_k \leq \epsilon$ *whenever* $k \geq \frac{2}{\log 2}\log\frac{1}{\epsilon}$ *with probability more than* $1-\epsilon$.

Proof. See §S.4.3 in the supplementary material. □

Theorem 2 allows one to ensure that k_t does not increase by too much. Precisely, this is an extremely useful property of OSBP which has a direct bearing on the memory footprint of the MCMC algorithm developed later in Section 3.2. Next we derive PPFs of OSBP.

2.4. Predictive probability functions (PPFs) for OSBP

PPFs are useful tools to design MCMC inference algorithms for BNP models. SBP in general does not admit PPFs except in special cases such as DP and PYP (Ishwaran & James, 2001). Although OSBP has parameter setting as general as SBP, in this section we demonstrate that OSBP has easy to evaluate PPFs which will be exploited later to design a truncation-free MCMC inference procedure.

Let z_t and B_j be defined as in Eq. (3) and $g_j = |\mathrm{B}_j|$, $h_j = \sum_{l>j} g_l$. PPF $(\pi_j, j \in [k_{t-1}]$ and $\sigma_{k_{t-1}})$ are defined by Pitman (1996) as

$$\begin{aligned} \pi_j &= p(z_t = j|z_{1:t-1}, \Theta),\; j \in [k_{t-1}], \\ \sigma_{k_{t-1}} &= p(z_t = k_{t-1}+1|z_{1:t-1}, \Theta) \end{aligned} \tag{5}$$

where Θ denotes the set of hyper-parameters. In words, π_j is the probability of next sample G_t from OSBP to be same as Q_j, and with probability $\sigma_{k_{t-1}}$, $\mathrm{G}_t = \mathrm{Q}_{k_{t-1}+1}$, a new sample from base measure Γ. Notice that, implicitly $\sum_{j=1}^{k_{t-1}} \pi_j + \sigma_{k_{t-1}} = 1$. We state an intermediate lemma useful for deriving the PPF of OSBP.

Lemma 2. *Let,* (v_j) *be defined as in Eq.* (2), *and* $\mathrm{G}_{1:t-1}|\boldsymbol{\mu}, \boldsymbol{\nu}, \Gamma \sim OSBP(\boldsymbol{\mu}, \boldsymbol{\nu}, \Gamma)$. *Then* $\forall j$, $v_j|z_{1:t-1}, \mu_j, \nu_j \sim Beta(\mu_j\nu_j + g_j - 1, (1-\mu_j)\nu_j + h_j)$.

Proof. See §S.5.1 in the supplementary material. □

We are now ready to state the main result of this section.

Theorem 3. *Let* (π_j), $\sigma_{k_{t-1}}$ *be defined in Eq.* (5), *and* $\mathrm{G}_{1:t-1}|\boldsymbol{\mu}, \boldsymbol{\nu}, \Gamma \sim OSBP(\boldsymbol{\mu}, \boldsymbol{\nu}, \Gamma)$. *Then, we have:*

$$\begin{aligned} \pi_j &= \tfrac{\mu_j\nu_j+g_j-1}{\nu_j+g_j+h_j-1}\textstyle\prod_{l=1}^{j-1}\tfrac{(1-\mu_l)\nu_l+h_l}{\nu_l+g_l+h_l-1},\; j \in [k_{t-1}], \\ \sigma_{k_{t-1}} &= \textstyle\prod_{l=1}^{k_{t-1}}\tfrac{(1-\mu_l)\nu_l+h_l}{\nu_l+g_l+h_l-1} \end{aligned} \tag{6}$$

Proof. See §S.5.2 in the supplementary material. □

It is easy to see that, $1-\sum_{l=1}^{k_{t-1}}\pi_l = \prod_{l=1}^{k_{t-1}}\frac{(1-\mu_l)\nu_l+h_l}{\nu_l+g_l+h_l-1}$, therefore $\sum_{j=1}^{k_{t-1}}\pi_j+\sigma_{k_{t-1}}=1$. Thus we get the PPFs of OSBP in (6). See §S.3 in the supplementary material for more discussion on OSBP and PPFs of OSBP.

3. Sequential MCMC inference through OSBP (SUMO)

In this section, we develop **S**eq**U**ential **M**CMC inference through **O**SBP (SUMO). After discussing the relevant background, we propose SUMO and discuss its properties followed by specifying SUMO for text datasets.

3.1. Background

Global and local variables in BNP models. Many hierarchical BNP models can be described generally as

$$p(\varphi,\phi_{1:n},x_{1:n}) = p(\varphi)\prod_{i=1}^{n}p(x_i|\phi_i)p(\phi_i|\varphi) \quad (7)$$

n is the number of observations, $\{\phi_i\}_1^n$ are the *local* latent variables, one for each *observation* x_i, and φ denotes the *global* variables common to the entire dataset. We consider Dirichlet process mixture model (DPMM) (Escobar & West, 1995) as an example of Eq. (7) to describe our approach and show few more examples on PYP (Pitman, 1996), SBP (Ishwaran & James, 2001) and hier-archical DP (Teh et al., 2006) in §S.7 of the supplementary.

Dirichlet process mixture model (DPMM). Using $v_s \sim Beta(1,\gamma)$ for some $\gamma>0$ in Eq. (1) one obtains Dirichlet process $DP(\gamma,\mathrm{H})$, and we write $\mathrm{G}\sim DP(\gamma,\mathrm{H})$ (Sethuraman, 1994). DPMM can be described as

$$\forall i,\;\; x_i\sim f(\phi_i),\;\; \phi_i|\mathrm{G}\sim\mathrm{G},\;\; \mathrm{G}\sim DP(\gamma,\mathrm{H}) \quad (8)$$

Recall that, G can be expressed as $\mathrm{G}=\sum_{s=1}^{\infty}\theta_s\delta_{\beta_s}$. The atoms $\beta_s\sim\mathrm{H}$ and $(\theta_s)\sim GEM(\gamma)$. $\{\theta_s,\beta_s\}$ are the *global* variables and $\{\phi_i\}$ are the *local* variables.

MCMC is not scalable. MCMC inference of the DPMM model involves integrating out G and computing $p(\phi_i|\phi_{1:n}^{-i},x_{1:n})\propto p(x_i|\phi_i)p(\phi_i|\phi_{1:n}^{-i})$ for all $i=1,\ldots,n$ iterating multiple times. For large scale datasets when n is very high, it is not possible to maintain all the variables making the MCMC inference infeasible.

Sequence of mini-batches. To infer from large datasets, one can split the observations $\{x_i\}_1^n$ into *mini-batches* $\{\mathrm{X}_t\}_1^{\bar{d}}$. In the tth mini-batch, $\mathrm{X}_t=\{x_i\}_{i=\bar{n}(t-1)+1}^{\bar{n}t}$, a collection of $\bar{n}$ data points need to be processed[1] which is feasible. The mini-batches (X_t) are *sequentially* processed to approximate the posterior $p(\Phi_{1:\bar{d}},\varphi|\mathrm{X}_{1:\bar{d}})$ recursively, where $\Phi_t=\{\phi_i\}_{i=\bar{n}(t-1)+1}^{\bar{n}t}$.

State of the art sequential algorithms. PF (Fearnhead, 2004; Canini et al., 2009) and TSVI (Wang & Blei, 2012) form the state of the art in sequential inference of BNP models. They follow two different strategies. PF builds a recursive dependence over local variables (Φ_t) by integrating out global variables φ, that makes them to store multiple configurations of $O(n)$ making them feasible only in distributed setup. Whereas, TSVI by employing SVI develops recursive dependence only on global variables φ and reduces memory requirement successfully.

The challenge with MCMC to apply on large datasets is that, no suitable analog of SVI is known in MCMC family that will scale well. We propose our solution below.

3.2. SUMO by applying OSBP on BNP models

Utilizing the appearance in order property, we build sequential dependence across mini-batches for BNP models using OSBP. The attendant MCMC inference is SUMO.

3.2.1. OSBP PRIOR ON BNP MODELS

We apply OSBP to exchange relevant statistical information across mini-batches for DPMM as below.

$$\begin{aligned}
&\mathrm{G}_1=\mathrm{Q}_1;\;\; \forall j,\;\; \mathrm{Q}_j\sim DP(\gamma_j,\mathrm{H})\\
\forall t>1,\;\; &\mathrm{G}_t|\mathrm{G}_{1:t-1},\mathrm{H}\sim\textstyle\sum_{j=1}^{k_{t-1}}\rho_j\delta_{\mathrm{Q}_j}+\alpha_{k_{t-1}}\delta_{\mathrm{Q}_{k_{t-1}+1}}\\
&\forall i,\;\; x_{ti}|\phi_{ti}\sim f(\phi_{ti}),\quad \phi_{ti}|\mathrm{G}_t\sim\mathrm{G}_t
\end{aligned} \quad (9)$$

(ρ_j) and $\alpha_{k_{t-1}}$ are as defined in Eq. (2) of OSBP. $(\mathrm{Q}_1,\ldots,\mathrm{Q}_{k_{t-1}})$ is the set of unique values among $\mathrm{G}_{1:t-1}$ and $\mathrm{Q}_{k_{t-1}+1}$ is pre-sampled. The second line in Eq. (9) models DPMM with G_t similar to Eq. (8).

Hyper-parameter settings. The hyper-parameters are $\boldsymbol{\mu},\boldsymbol{\nu}$ due to OSBP, and (γ_j) for DP. We set $\forall j,\ \mu_j=\mu$ for some $0.5<\mu<1$. $\nu_j=(1-\mu)\nu_{j-1}$ and $\nu_1=\gamma$ ($\gamma>0$ as in Eq. (8)). So $\nu_j=(1-\mu)^{j-1}\gamma$. We use, $\gamma_j=\mu\nu_j$ and hence $\gamma_j=\mu(1-\mu)^{j-1}\gamma$. Thus, we have only two hyper-parameters μ and γ.

Equivalence with DPMM. Note that, formulation Eq. (9) trivially becomes equivalent to DPMM (Eq. (8)) in batch mode. Moreover, we can state an important result as below.

Theorem 4. *For any $t\in\mathbf{N}$, each x_{ti} sampled using model Eq. (9) has marginal distribution same as x_i sampled with DPMM in Eq. (8) with $\mathrm{G}\sim DP(c_t,\mathrm{H})$, where $c_t=\sum_{j=1}^{k_{t-1}}\gamma_j+(1-\mu)^{k_{t-1}}\gamma$. Furthermore, for any $\epsilon>0$ and $t>0$, with probability greater than $1-\epsilon$, each x_{ti} in Eq. (9) has marginal distribution same as x_i in Eq. (8) with $\mathrm{G}\sim DP(\sum_{j=1}^{k}\gamma_j,\mathrm{H})$, when $k_t\geq k\geq\frac{2}{\log 2}\log\frac{1}{\epsilon}$. Also, for $t\to\infty$, each x_{ti} in Eq. (9) has marginal distribution same as x_i in Eq. (8) with $\mathrm{G}\sim DP(\gamma,\mathrm{H})$.*

Proof. See §S.6.1 in the supplementary material. □

Theorem 4 signifies that at every mini-batch all data points are equivalently sampled from a DPMM model which dif-

[1] for simplicity we have assumed $n=\bar{n}\bar{d}$.

Algorithm 1 SUMO. **S**eq**U**ential **M**CMC Inference through **O**SBP.

Require: (X_t)
1: **for** $t = 1, 2, \ldots$ **do**
2: Initialize sufficient statistics $\mathbf{S}$
3: **for** $iter = 1$ to I **do**
4: $\{$*total user-defined number of iterations*$\}$
5: **for** $i = 1$ to $\bar{n}$ **do**
6: Compute $p(\phi_{ti}|\Phi_t^{-i}, \mathrm{X}_t, \mathbf{S})$ $\{$*local variable inference*$\}$
7: **end for**
8: Compute $p(\mathrm{G}_t, \rho_{1:k_t}, \mathrm{H}, |\Phi_t, \mathrm{X}_t, \mathrm{G}_{1:t-1}, \mathbf{S})$ $\{$*global variable inference*$\}$
9: **end for**
10: Update sufficient statistics $\mathbf{S}$
11: Discard local variables $\{\mathrm{X}_t, \Phi_t\}$.
12: **end for**

fers with DPMM in Eq. (8) only in the scale parameter of DP. The scale parameter stops varying with high probability after initial stage and converges to the actual scale parameter γ asymptotically, when Eq. (9) makes loss-less approximation of DPMM in Eq. (8).

3.2.2. SUMO ALGORITHM

The attendant inference algorithm of Eq. (9) gives us the MCMC scheme, SUMO. We describe the SUMO approach in Algorithm 1. Step 4 to 6 corresponds to MCMC local to mini-batch t for inferring local variables Φ_t. Step 7 updates the global variables based on local variables Φ_t and sufficient statistics $\mathbf{S}$. Due to conditional independence of $\{\mathrm{X}_t, \Phi_t\}$ and $\{\mathrm{X}_{t+1}, \Phi_{t+1}\}$ given (G_t) and $\mathbf{S}$, SUMO at step 9 deletes local variables $\{\mathrm{X}_t, \Phi_t\}$ after updating $\mathbf{S}$ at step 8. Step 4 to 6 is similar to MCMC. SUMO differs with MCMC at step 7 which also incurs slightly additional time complexity, which is tolerable given the benefit in memory reduction to apply on large datasets. We defer the details of inference considering a data model to Section 3.4, and discuss the properties below.

3.3. Properties of SUMO

SUMO approximates full posterior sequentially. Following the dependency structure in Eq. (9), we can write the full posterior $p(\mathrm{G}_{1:\bar{d}}, \rho_{1:k_{\bar{d}}}, \mathrm{H}, \Phi_{1:\bar{d}}|\mathrm{X}_{1:\bar{d}})$ as

$$\prod_{t=1}^{\bar{d}} p(\Phi_t|\mathrm{G}_t, \mathrm{X}_t) p(\mathrm{G}_t|\Phi_t, \mathrm{X}_t, \mathrm{G}_{1:t-1}, \rho_{1:k_t}, \mathrm{H}) \quad (10)$$
$$p(\rho_{1:k_t}, \mathrm{H}|\Phi_t, \mathrm{G}_{1:t}, \mathrm{X}_t)$$
$$p(\mathrm{G}_{1:t-1}, \rho_{1:k_{t-1}}, \mathrm{H}, \Phi_{1:t-1}|\mathrm{X}_{1:t-1}) \quad (11)$$

This shows how we can move from posterior at time $t-1$, $p(\mathrm{G}_{1:t-1}, \Phi_{1:t-1}, \rho_{1:k_{t-1}}, \mathrm{H}|\mathrm{X}_{1:t-1})$ to the posterior at time t, $p(\mathrm{G}_{1:t}, \Phi_{1:t}, \rho_{1:k_t}, \mathrm{H}|\mathrm{X}_{1:t})$ *recursively* in our sequential inference scheme. Posterior in each time stamp t is approximated using MCMC or Gibbs sampling leading to accurate approximation. Using Eq. (11) we represent the posterior of the global and local variables at time t which are used in Eq. (10). Due to Theorem 4 SUMO approximates the posterior of DPMM ignoring initial period.

Difference with PF. Generally MCMC performs better than PF to approximate full posterior if the dataset is not inherently sequential as noted by Fearnhead (2004). PF allows mini-batch processing of large datasets as an alternative to MCMC. But, PF integrates out φ and at time t, approximates $p(\Phi_t|\mathrm{X}_t)$ using particles and by utilizing a recursive dependence of $p(\Phi_{t+1}|\Phi_{1:t})$ moves on to the next mini-batch. Due to recursive dependence over local variables using particles, PF needs to maintain multiple configurations of local variables which makes it practical only in *distributed setting* (Canini et al., 2009; Ahmed et al., 2011; Williamson et al., 2013).

Reduction in memory requirement. As noted earlier, due to conditional independence of local variables $\{\mathrm{X}_t, \Phi_t\}$ and $\{\mathrm{X}_{t+1}, \Phi_{t+1}\}$ given (G_t) and $\mathbf{S}$, posterior at time t using Eq. (10) does not involve $\Phi_{1:t-1}, \mathrm{X}_{1:t-1}$. Discarding the local variables at step 8 in Algorithm 1, SUMO requires memory for mini-batch specific local variables Φ_t, X_t and global variables for $\mathrm{G}_{1:t}, \rho_{1:k_t}, \mathbf{S}$, which is $O(\bar{n} + k_t \times \mathrm{K}_t)$ (K_t is number of parameters for H). This is a much smaller quantity than $O(n)$, complexity of MCMC and PF. Moreover, $\bar{n}$ can be set conveniently. $\bar{n} + k_t \times \mathrm{K}_t$ can be in hundreds whereas n is in million (e.g. PubMed dataset). Memory footprint of SUMO grows with k_t, that stops increasing after few initial mini-batches (see Theorem 2). Thus, SUMO by reducing memory usage can process large datasets which are beyond the scope of MCMC and PF.

Comparison with TSVI. TSVI in a hybrid approach, approximates $p(\Phi_t|\mathrm{X}_t)$ (collapsing φ) using MCMC *locally inside a mini-batch*, then approximates $p(\varphi|\Phi_{1:t}, \mathrm{X}_{1:t})$ recursively following SVI. Due to the recursive dependence only on global variables, TSVI reduces memory requirement successfully. TSVI, due to the use of local MCMC, is the closest approach to SUMO.

3.4. SUMO instantiated on DPMM for texts

For texts, we assume each data point x_{ti} is a document with $\{x_{til}\}$ words. The data model (second line in Eq. (9)) is

$$\forall t,\ \forall l,\ x_{til}|\phi_{ti} \sim multinomial(\phi_{ti}),\ \forall i,\ \phi_{ti}|\mathrm{G}_t \sim \mathrm{G}_t \quad (12)$$

G_t is sampled from OSBP (see Eq. (9)). For conjugacy, ϕ_{ti} has Dirichlet prior. The task is to compute details of step 5, and 7 in Algorithm 1. We outline the inference here and defer the details to §S.8 in the supplementary material.

Random variables to infer. From Eq. (9), we can say $\mathrm{Q}_j = \sum_{r=1}^{\infty} \zeta_{jr} \delta_{\psi_{jr}}$, where $(\zeta_{jr}) \sim GEM(\gamma_j)$ and $\psi_{jr} \sim \mathrm{H}$. Let, $\mathrm{H} \sim DP(\lambda, Dirichlet(\eta))$, then $\mathrm{H} = \sum_{s=1}^{\infty} \theta_s \delta_{\beta_s}$,

where $\beta_s \sim Dirichlet(\eta)$ and $(\theta_s) \sim GEM(\lambda)$. Thus, $\psi_{jr} \in \{\beta_s\}$ ensures same components across t without invoking ad-hoc merging of components (Newman et al., 2009). Given this setup, we introduce alternative variables to speed up the mixing of the Markov chain following standard approach. Recall that, $z_t = j$ if $\mathrm{G}_t = \mathrm{Q}_j$ as defined in Eq. (3). Let, $a_{ti} = r$ if $\phi_{ti} = \psi_{jr}$ and $z_t = j$. If s is the index of global mixture component represented by ψ_{jr} in Q_j, then we define $b_{jr} = s$ if $\psi_{jr} = \beta_s$. Let, $y_{ti} = s$ if $z_t = j$ and $b_{jr} = s$. y_{ti} is the index of the component assigned to x_{ti}. Due to this representation, the equivalent random quantities are $\mathrm{A}_{1:t} = \{\{a_{li}\}_{i=1}^{\bar{n}}\}_{l=1}^{t}$, $\mathrm{B}_{1:k_t} = \{b_{jr}\}_{j=1}^{k_t}$, and $\mathrm{Y}_{1:t} = \{\{y_{li}\}_{i=1}^{\bar{n}}\}_{l=1}^{t}$. We integrate out (Q_j) and H following Chinese restaurant process (CRP), (ρ_j) following Theorem 3, and $\{\beta_s\}$ following Dirichlet multinomial conjugacy. So, we need to infer A_t, B, and z_t at time t.

Notation. Superscript with hyphen denotes set minus, e.g. $\mathrm{X}_t^{-i} = \mathrm{X}_t \backslash x_{ti}$, and $\mathrm{X}_t^{-r} = \mathrm{X}_t \backslash \mathrm{X}_{tr}$, where $\mathrm{X}_{tr} = \{x_{ti} | a_{ti} = r\}$. $\mathrm{X}_{1:t}^{-tr} = \mathrm{X}_{1:t} \backslash \mathrm{X}_{tr}$, and $\mathrm{X}_{1:t}^{-ti} = \mathrm{X}_{1:t} \backslash x_{ti}$. $\mathrm{A}_{1:t}^{-ti} = \mathrm{A}_{1:t} \backslash a_{ti}$. $\mathrm{B}_{z_t}^{-r} = \mathrm{B}_{z_t} \backslash b_{z_t r}$. $\mathrm{L}_s(x_{ti})$ and $\mathrm{L}_s(\mathrm{X}_{tr})$ are the likelihood of x_{ti} and X_{tr} respectively for mixture component s. Computation of $\mathrm{L}_s(x_{ti})$ and $\mathrm{L}_s(\mathrm{X}_{tr})$ is standard following Dirichlet multinomial cojugacy. Next we describe inference steps, see the supplementary material for details and explanation.

Inference of a. We infer a as below.

$$p(a_{ti} = r | \mathrm{A}_{1:t}^{-ti}, \mathrm{B}_{1:k_t}, z_{1:t}, \mathrm{X}_{1:t}) \propto \quad (13)$$
$$p(x_{ti} | a_{ti} = r, z_{1:t}, \mathrm{A}_{1:t}^{-i}, \mathrm{B}_{1:k_t}, \mathrm{X}_{1:t}^{-i}) p(a_{ti} = r | \mathrm{A}_{1:t}^{-ti}, z_t)$$

where $p(x_{ti} | a_{ti} = r, z_{1:t}, \mathrm{A}_{1:t}^{-i}, \mathrm{B}_{1:k_t}, \mathrm{X}_{1:t}^{-i})$ is $\mathrm{L}_{b_{z_t r}}(x_{ti})$. $p(a_{ti} = r | \mathrm{A}_{1:t}^{-ti}, z_t)$ comes from CRP as

$$\propto \mathrm{L}_{b_{z_t r}}(x_{ti}) (m_{z_t r}^{-i} + \mathrm{M}_{z_t r})(1 - \iota_r) + \gamma_{b_{z_t}} \mathrm{L}_{b_{z_t r_{new}}}(x_{ti}) \iota_r \quad (14)$$

where $\iota_r = \mathbb{I}[r = r_{new}]$, $m_{z_t r} = \sum_{i=1}^{\bar{n}} \mathbb{I}[a_{ti} = r]$ and $\mathrm{M}_{jr} = \sum_{l=1}^{t-1} \sum_{i=1}^{\bar{n}} \mathbb{I}[z_l = j, a_{li} = r]$. When a new r_{new} is sampled we obtain $b_{z_t r_{new}}$ from $p(b_{z_t r} = s_{new} | z_{1:t}, \mathrm{A}_{1:t}, \mathrm{B}_{1:k_t}, \mathrm{X}_{1:t})$ which is shown later.

Inference of z. Following the dependence structure in Eq. (9), z_t is independent of X_t given Y_t. So, we can infer z from $p(z_t = j | z_{1:t-1}, \mathrm{Y}_t, \mathrm{B}_{1:k_t})$ as

$$\propto [\textstyle\prod_{i=1}^{\bar{n}} p(y_{ti} = s | z_{1:t}, \mathrm{B}_{1:k_t})] \;\; p(z_t = j | z_{1:t-1}) \quad (15)$$

$p(z_t = j | z_{1:t-1})$ comes from Theorem 3. Recall that $y_{ti} = b_{z_t a_{ti}}$. So $p(y_{ti} = s | z_t = j, \mathrm{B}_{1:t}, z_{1:t-1})$ comes from CRP by integrating out G_t and H as

$$\propto \left[\textstyle\prod_{i=1}^{\bar{n}} \mathrm{J}_{js} (1 - \iota_s^j) + \gamma_j \iota_s^j \left(\mathrm{J}_{\cdot s} (1 - \iota_s^0) + \lambda \iota_s^0 \right) \right]$$
$$\pi_j (1 - \iota_j) + \left[\mathrm{J}_{\cdot s} (1 - \iota_s^0) + \lambda \iota_s^0 \right] \sigma_{k_{t-1}} \iota_j \quad (16)$$

where, $\iota_j = \mathbb{I}[z_t = j_{new}]$, $\iota_s^j = \mathbb{I}[z_t = j, s = s_{new}]$, $\iota_s^0 = \prod_{l=1}^{k_t} \iota_s^l$, $\mathrm{J}_{js} = \sum_r \mathbb{I}[b_{jr} = s, z_t = j]$ and $\mathrm{J}_{\cdot s} = \sum_{j=1}^{k_{t-1}} \mathrm{J}_{js}$. π_j and $\sigma_{k_{t-1}}$ are as defined in Eq. (6). ι_s^j, ι_s^0 denote if β_s is present in Q_j, H respectively. J_{js} counts number of times β_s is present among $\{\psi_{jr}\}$.

Inference of b. We infer b as below.

$$p(b_{z_t r} = s | z_{1:t}, \mathrm{A}_{1:t}, \mathrm{B}_{1:k_t}, \mathrm{X}_{1:t}) \propto \quad (17)$$
$$p(\mathrm{X}_{tr} | z_{1:t}, \mathrm{A}_t, \mathrm{B}_{1:k_t}, \mathrm{X}_{1:t}^{-tr}) p(b_{z_t r} = s | \mathrm{B}_{z_t}^{-r}, z_{1:t}, \mathrm{A}_{1:t}, \mathrm{B}_{1:k_t})$$

where $p(\mathrm{X}_{tr} | z_{1:t}, \mathrm{A}_t, \mathrm{B}_{1:k_t}, \mathrm{X}_{1:t}^{-tr})$ is $\mathrm{L}_s(\mathrm{X}_{tr})$ and $p(b_{z_t r} = s | \mathrm{B}_{z_t}^{-r}, z_{1:t}, \mathrm{A}_{1:t}, \mathrm{B}_{1:k_t})$ comes from CRP as

$$\propto \mathrm{L}_s(\mathrm{X}_{tr}) (n_{z_t s}^{-r} + \mathrm{N}_s^{-z_t})(1 - \iota_s) + \lambda \mathrm{L}_{s_{new}}(\mathrm{X}_{tr}) \iota_s \quad (18)$$

where $\iota_s = \mathbb{I}[s = s_{new}]$, $n_{z_t s}^{-r} = \sum_{q \neq r} \mathbb{I}[b_{z_t q} = s]$ and $\mathrm{N}_s^{-z_t} = \sum_{l=1}^{k_{t-1}} \sum_q \mathbb{I}[b_{lq} = s, l \neq z_t]$.

SUMO for DPMM on texts. Using Eq. (13) in step 5, and Eq. (15), Eq. (17) in step 7 of Algorithm 1, we obtain SUMO for text datasets. The algorithm is presented in the supplementary material §S.8. Notice from Eq. (14), Eq. (16) and Eq. (18) that, by maintaining statistics M, J, N and L, we need not store the local variables $(\mathrm{A}, \mathrm{Y}, \mathrm{X})_{1:t-1}$.

4. Related work

Existing stick-breaking priors either assume exchangeability among partitions and hence atoms such as DP, PYP or model spatial dependence among atoms such as πDDP (Griffin & Steel, 2006) and local DP (Chung & Dunson, 2011). This is significantly different from appearance in order phenomenon which is neither exchangeable nor related to spatial distances. BNP priors forming sequential dependency (Lin et al., 2010; Chen et al., 2013) is well known but none of them define dependency on atoms of an SBP, neither modeling appearance in order of atoms.

Although the OSBP based model for DPMM (Eq. (9)) is not a dependent DP (DDP) model (MacEachern, 2000; Caron et al., 2008) but use of multiple DP distributions make it closer to DDP than any other models. The fundamental difference is that, unlike DDP, we do not intend to modify the DP framework to explore novel probabilistic relationships in datasets. Eq. (9) is asymptotically equivalent to DPMM. Furthermore, the existing DDP models neither work in mini-batch setting for sequential inference nor address the memory issue in MCMC inference of DPMM, what we do here. Equivalence of DP and Dirichlet mixture of DPs has been used earlier by Williamson et al. (2013) to make loss-less approximation in distributed setup.

5. Experiments

We experimentally evaluate here the proposed approach SUMO on DPMM for text datasets as given in Section 3.4.

Objective. Our experimental goals are as follows.
(1) To compare with state of the art methods on held-out data *perplexity*, a standard metric in BNP experiments

Table 1. Four real life datasets used. Two datasets have more than million data points representing contemporary large scale data.

Dataset	Documents	Tokens
NIPS	1500	1.9 M
NYT	300 K	100 M
PMA	8.2 M	730 M
WPE	1 M	296 M

Table 2. Comparison of SUMO with MCMC, PF and TSVI on NIPS. SUMO uses much less memory than MCMC and PF with much better perplexity than PF and TSVI.

Methods	Perplexity	Memory
MCMC	2196	480 MB
PF	6432	450 MB
TSVI	3740	70 MB
SUMO	2386	110 MB

which is suitable to measure how well a model learns the training data to generalize over the unseen dataset.
(2) To verify *memory usage* on contemporary large datasets comparing with state of the art.
(3) To evaluate by varying experimental settings such as (i) *order of the mini-batches*, (ii) value of DPMM hyper-parameter, and also (iii) size of the mini-batches.

5.1. Datasets, baselines and settings

Datasets. We have used four real world datasets: NIPS proceedings, New York Times (NYT), PubMed abstracts (PMA) and Wikipedia English (WPE). Table 1 contains the details of these datasets. NYT, PMA and NIPS are available at (Bache & Lichman, 2013), and WPE is available at `dumps.wikimedia.org`.

Baselines. We evaluate the proposed approach SUMO comparing with MCMC, PF (Fearnhead, 2004) and TSVI (Wang & Blei, 2012) using the implementation made available by the authors[2]. PF has been used in the same sequential setting. We have used 10K particles to run PF and removed variables inactive for long time. Number of particles is kept low to reduce memory usage with considerable accuracy. For large datasets, NYT, PMA and WPE, standard MCMC and PF suffer from out of memory issue. TSVI is a state of the art with no Monte Carlo family competitor and is also the closest approach as argued in Section 3.3. We compare with TSVI on large datasets.

Experimental settings. We use $\mu = 0.6$ and $\gamma = 0.5$, $\lambda = 5$ and $\eta = 0.5$. γ and η are kept same for TSVI to make the underlying model same. We do not learn or tune parameters. We set η and γ higher than commonly used in large scale setting as that favors variational inference over MCMC in general (Asuncion et al., 2009). Parameters specific to TSVI are used as in (Wang & Blei, 2012). The mini-batch sizes are as follows: 500 for NYT, 100 for NIPS, and for the much larger datasets of PMA and WPE we use 10,000. We converted all the characters into small case and removed special characters. Except NIPS, we removed stop words and limited vocabulary to 10K, 5K, 10K based on term frequency for NYT, PMA and WPE respectively. We removed documents smaller than 50 tokens.

[2]lists.cs.princeton.edu/pipermail/topic-models/attachments/20140424/8eea8833/attachment-0001.zip

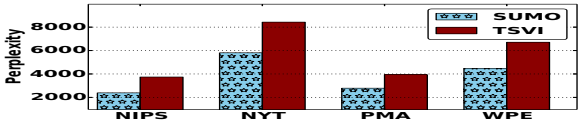

Figure 1. Held-out data perplexity (less is better). SUMO outperforms TSVI on all datasets (average 33%).

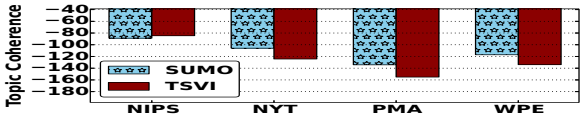

Figure 2. Average topic coherence using 10 most probable words (more is better). SUMO is on average 9% better than TSVI.

Computing system. All experiments are done on a system with *single* processor of 2.66GHz speed and 3GB RAM.

Held-out data for perplexity. 33%, 10%, 33% and 20% of datasets are held out (not used in training) for NYT, PMA, WPE and NIPS respectively to measure perplexity. Data points were held-out uniformly at random and then the training datasets were split into mini-batches. Held-out datasets being small, single batches were used.

5.2. Results

Perplexity. In Table 2 we report results comparing SUMO with MCMC, PF and TSVI. SUMO outperforms PF and TSVI and is very close to MCMC. This shows that SUMO is able to achieve approximation quite close to standard MCMC as argued analytically earlier in Theorem 4. MCMC and PF could not be run due to memory issue on NYT, PMA and WPE, where we compare with TSVI.

Figure 1 shows results, where SUMO performs substantially better than TSVI on all datasets with an average margin of 33%. Relatively low perplexity on PMA for both models is due to the small ratio of held-out test data compared to the training data.

Cluster coherence. We further verify the quality of mixture components inferred (often referred to as topics for texts), using topic coherence with 10 highest probability words per topic. Figure 2 shows that SUMO beats TSVI in most of the cases.

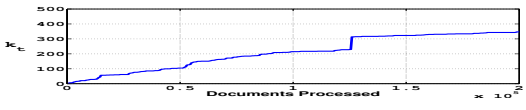

Figure 3. Growth of k_t on NYT. k_t grows fast initially and gradually settles down.

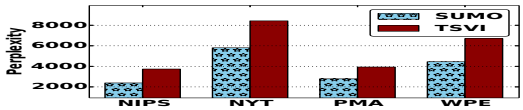

Figure 4. Held-out data perplexity (less is better) by varying order of mini-batches in NIPS training data (solid line is mean, dashed lines are actual values). SUMO is consistently better than TSVI (on average 33%).

5.2.1. Memory usage

In Table 2, we show results on memory usage comparing with MCMC, PF and TSVI for NIPS. Memory usage of SUMO is 4.4 and 4 times less than that of MCMC and PF respectively. If we retain all particles PF consumes additional 200 MB of memory. SUMO uses memory 1.5 times more than that of TSVI. As run-time memory usage depends on the mini-batch size, for the two largest datasets PMA and WPE we have used a large mini-batch size of 10,000 to stress-test the memory consumption. Among all the datasets, WPE is observed to have the largest size in a mini-batch. We observe that on WPE during run-time, SUMO and TSVI consume maximum memory of around 1.8 and 1.1 GB respectively. For other datasets the memory usage is much less for both SUMO and TSVI. Thus, SUMO is not far behind TSVI in run-time memory usage.

Growth of k_t. As memory usage of SUMO increases with k_t, we experimentally note the growth of k_t on NYT dataset in Figure 3. As expected, we see a fast growth initially that slows down gradually, reciprocating Theorem 2.

5.2.2. Varying experimental settings

Varying the order of mini-batches. We study the effect of change in the order of the mini-batches in the training set, keeping the test set fixed on NIPS. Figure 4 shows that both the methods are quite stable against the change in order of the mini-batches, but SUMO beats TSVI consistently with an average difference of 33%.

Varying the hyper-parameters. In Figure 5, we report held-out data perplexity on NIPS by varying γ in $\{0.5, 1, 2, 5\}$. SUMO is more stable across the variation compared to TSVI and on average 25% better.

Varying the size of mini-batches. On NYT dataset we report held-out data perplexity in Figure 6, by varying the size of mini-batches in $\{300, 400, 500, 10000\}$. SUMO is quite consistent and 35% better than TSVI on average.

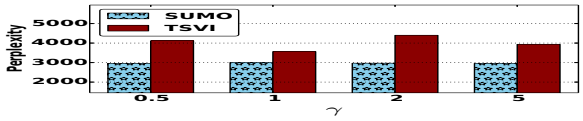

Figure 5. Held-out data perplexity (less is better) for different values of γ on NIPS. SUMO is more stable than TSVI (on average 25% better).

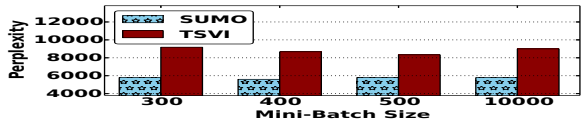

Figure 6. Perplexity for different mini-batch sizes on NYT. SUMO is more consistent than TSVI (on average 35% better).

5.3. Discussion

Significance of our experiments is that we apply SUMO to process three real world large scale corpora using only 3GB of RAM, which is beyond the scope of the existing Monte Carlo methods. This marks a significant improvement in Monte Carlo family for learning with large datasets. SUMO makes it possible as memory usage grows only with k_t which is not very high due to Theorem 2.

Apart from memory advantage, SUMO shows ability to learn from large datasets on perplexity measures outperforming PF (Table 2) and state of the art TSVI (Figure 1). This affirms that reduction in memory does not deteriorate learning ability of SUMO. Efficacy of SUMO can be attributed to the following facts. Although SUMO is sequential in nature, after initial burn-in stage (see Theorem 1) underlying model becomes equivalent to DPMM (see Theorem 4) and SUMO makes accurate approximation of the full posterior globally (as shown in Eq. (10), Eq. (11)). For mini-batches processed in order, it is natural that some mini-batches are statistically similar to each other than the rest, SUMO effectively models this through appearnce in order. Stability across the order and the size of the mini-batches (Figure 4, 6) also establishes SUMO as a valid approach in sequential learning. Additionally, SUMO seems to be less sensitive to the DPMM parameter (Figure 5) justifying complete Bayesian approach adopted in SUMO.

6. Conclusion

This paper introduces the *ordered stick-breaking* process (OSBP), by constraining the atoms in the stick to be *appearing in order*. OSBP can be of independent interest for streaming datasets. Using OSBP, we design a sequential inference based on MCMC (SUMO) for BNP models, that requires memory order of magnitude less than MCMC and PF, and is competitive to TSVI (Wang & Blei, 2012). SUMO is easy to implement, and outperforms PF, and TSVI on real life large scale datasets.

Acknowledgments

We are thankful to all the reviewers for their valuable comments. The authors were partially supported by DST grant (DST/ECA/CB/1101).

References

Ahmed, A., Ho, Q., Teo, C., Eisenstein, J., Smola, A., and Xing, E. Online Inference for the Infinite Topic-Cluster Model: Storylines from Streaming Text. In *In International Conference on Artificial Intelligence and Statistics*, pp. 101–109, 2011.

Andrieu, C., de Freitas, N., Doucet, A., and Jordan, M. An introduction to MCMC for machine learning. *Machine Learning*, 50:5–43, 2003.

Andrieu, C., Doucet, A., and Holenstein, R. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72:269–342, 2010.

Asuncion, A., Welling, M., Smith, P., and Teh, Y. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pp. 27–34, 2009.

Bache, K. and Lichman, M. UCI machine learning repository, 2013. URL http://archive.ics.uci.edu/ml.

Blei, D. and Jordan, M. Variational methods for the Dirichlet process. In *Proceedings of the 21st International Conference on Machine Learning*, 2004.

Broderick, T., Boyd, N., Wibisono, A., Wilson, A., and Jordan, M. Streaming Variational Bayes. In *Advances in Neural Information Processing Systems 26*, pp. 1727–1735, 2013.

Bryant, M. and Sudderth, E. Truly Nonparametric Online Variational Inference for Hierarchical Dirichlet Processes. In *Advances in Neural Information Processing Systems 25*, pp. 2708–2716, 2012.

Canini, K., Shi, L., and Griffiths, T. Online inference of topics with latent Dirichlet allocation. In *International Conference on Artificial Intelligence and Statistics*, pp. 65–72, 2009.

Caron, F., Davy, M., Doucet, A., Duflos, E., and Vanheeghe, P. Bayesian inference for linear dynamic models with Dirichlet process mixtures. *IEEE Transactions on Signal Processing*, 56:71–84, 2008.

Chen, C., Rao, V., Buntine, W., and Teh, Y. Dependent Normalized Random Measures. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 969–977, 2013.

Chung, Y. and Dunson, D. The local Dirichlet process. *Annals of the Institute of Statistical Mathematics*, 63(1): 59–80, 2011.

Connor, R. and Mosimann, J. Concepts of independence for proportions with a generalization of the dirichlet distribution. *Journal of the American Statistical Association*, 64(325):194–206, 1969.

Doucet, A., de Freitas, N., and Gordon, N. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.

Escobar, M. and West, M. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90:577–588, 1995.

Fearnhead, P. Particle filters for mixture models with an unknown number of components. *Journal of Statistics and Computing*, 14:11–21, 2004.

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

Griffin, J. and Steel, M. Order-based dependent Dirichlet processes. *Journal of the American statistical Association*, 101(473):179–194, 2006.

Hjort, N., Holmes, C., Mueller, P., and Walker, S. *Bayesian Nonparametrics: Principles and Practice*. Cambridge University Press, Cambridge, UK, 2010.

Ishwaran, H. and James, L. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 96(453):161–173, 2001.

Jun, S. and Coute, A. Memory (and Time) Efficient Sequential Monte Carlo. In *Proceedings of the 31st International Conference on Machine Learning*, 2014.

Lin, D. Online Learning of Nonparametric Mixture Models via Sequential Variational Approximation. In *Advances in Neural Information Processing Systems 26*, pp. 395–403, 2013.

Lin, D., Grimson, E., and Fisher, J. Construction of Dependent Dirichlet Processes based on Poisson Processes. In *Advances in Neural Information Processing Systems 23*, pp. 1396–1404, 2010.

MacEachern, S. Dependent dirichlet processes. *Unpublished manuscript, Department of Statistics, The Ohio State University*, 2000.

Newman, D., Asuncion, A., and Smyth, P. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10:1801–1828, 2009.

Pitman, J. Exchangeable and partially exchangeable random partitions. *Probability Theory and Related Fields*, 102(2):145–158, 1995.

Pitman, J. Some developments of the Blackwell-MacQueen urn scheme. *Lecture Notes-Monograph Series*, pp. 245–267, 1996.

Pitman, J. and Yor, M. The two-parameter Poisson-Dirichlet distribution derived from a stable subordinator. *The Annals of Probability*, 25(2):855–900, 1997.

Sethuraman, J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4:639–650, 1994.

Teh, Y., Jordan, M., Beal, M., and Blei, D. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581, 2006.

Ulker, Y., Gunsel, B., and Cemgil, A. Sequential Monte Carlo samplers for Dirichlet process mixtures. In *International Conference on Artificial Intelligence and Statistics*, pp. 876–883, 2010.

Wang, C. and Blei, D. Truncation-free online variational inference for Bayesian nonparametric models. In *Advances in Neural Information Processing Systems 25*, pp. 422–430, 2012.

Welling, M., Teh, Y., and Kappen, H. Hybrid variational/Gibbs collapsed inference in topic models. *arXiv preprint arXiv:1206.3297*, 2012.

Williamson, S., Dubey, A., and Xing, E. Parallel Markov Chain Monte Carlo for Nonparametric Mixture Models. In *Proceedings of the 30th International Conference on Machine Learning*, pp. 98–106, 2013.