Distributed, partially collapsed MCMC for Bayesian nonparametrics

Avinava Dubey* Google Research
Michael M. Zhang* Princeton University
Eric P. Xing Carnegie Mellon Univ.
Sinead A. Williamson Univ. of Texas Austin

Abstract

Bayesian nonparametric (BNP) models provide elegant methods for discovering underlying latent features within a data set, but inference in such models can be slow. We exploit the fact that completely random measures, which commonly-used models like the Dirichlet process and the beta-Bernoulli process can be expressed using, are decomposable into independent sub-measures. We use this decomposition to partition the latent measure into a finite measure containing only instantiated components, and an infinite measure containing all other components. We then select different inference algorithms for the two components: uncollapsed samplers mix well on the finite measure, while collapsed samplers mix well on the infinite, sparsely occupied tail. The resulting hybrid algorithm can be applied to a wide class of models, and can be easily distributed to allow scalable inference without sacrificing asymptotic convergence guarantees.

1 INTRODUCTION

Bayesian nonparametric (BNP) models are a flexible class of models whose complexity adapts to the data under consideration. BNP models place priors on infinite-dimensional objects, such as partitions with infinitely many blocks; matrices with infinitely many columns; or discrete measures with infinitely many atoms. A finite set of observations is assumed to be generated from a finite—but random—subset of these components, allowing flexibility in the underlying dimensionality and providing the ability to incorporate previously unseen properties as our dataset grows.

While the flexibility of these models is a good fit for large, complex data sets, distributing existing inference algorithms across multiple machines is challenging. If we explicitly represent subsets of the underlying infinite-dimensional object—for example, using a slice sampler—we can face high memory requirements and slow convergence. Conversely, if we integrate out the infinite-dimensional object, we run into problems due to induced global dependencies.

Moreover, a key goal of distributed algorithms is to minimize communication between agents. This can be achieved by breaking the algorithm into independent sub-algorithms, which can be run independently on different agents. In practice, we usually cannot split an MCMC sampler on a Bayesian hierarchical model into entirely independent sub-algorithms since there are typically some global dependencies implied by the hierarchy. Instead, we make use of conditional independencies to temporarily partition our algorithm.

Contributions: In this paper, we propose a distributable sampler for models derived from completely random measures, which unifies exact parallel inference for a wide class of Bayesian nonparametric priors, including the popularly used Dirichlet process (Ferguson, 1973) and the beta-Bernoulli process (Griffiths and Ghahramani, 2011). After reviewing the appropriate background material, we first introduce general recipes for (non-distributed) partially collapsed samplers appropriate for a wide range of BNP models, focusing on the beta-Bernoulli process and the Dirichlet process as exemplars. We then demonstrate that these methods can be easily extended to a distributed setting. Finally we provide experimental results for our hybrid and distributed sampler on DP and BB inference.

2 RELATED WORK

Completely random measures (CRMs, Kingman, 1967) are random measures that assign independent masses to disjoint subsets of a space. For example, the gamma process assigns a gamma-distributed mass to each sub-

* denotes equal contribution

Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108.

Table 1: Comparison of various parallel and distributed inference algorithm proposed for BNPs. Other CRMs include gamma-Poisson process, beta-negative binomial process etc.

Methods	Data Size Millions	Exact	Parallel	Distributed	BN Processes				
					Beta-Bernoulli Process	Other CRMs	DP	HDP	Pitman-Yor Process
Smyth et al. 2009	1M	×	✓	✓	×	×	✓	✓	×
Doshi-Velez and Ghahramani 2009	.01M	✓	×	×	✓	×	×	×	×
Doshi-Velez et al. 2009b	.1M	×	✓	✓	✓	×	×	×	×
Williamson et al. 2013	1M	✓	✓	×	×	×	✓	✓	×
Chang and Fisher 2013	0.3M	✓	✓	×	×	×	✓	✓	×
Dubey et al. 2014	10M	✓	✓	×	×	×	×	×	✓
Ge et al. 2015	0.1M	✓	✓	✓	×	×	✓	✓	×
Yerebakan and Dundar 2017	0.05M	✓	✓	✓	×	×	✓	×	×
This paper	1M	✓	✓	✓	✓	✓	✓	✓	✓

set. Other examples include the beta process (Hjort, 1990) and the Poisson process. The distribution of a CRM is completely determined by its Lévy measure, which controls the size and location of atoms.

Many nonparametric distributions can be expressed in terms of CRMs. For example, if we sample $B = \sum_{i=1}^{\infty} \mu_i \delta_{\theta_i}$ from a (homogeneous) beta process, and generate a sequence of subsets Z_i where $\theta_k \in Z_i$ w.p. μ_k, then we obtain an exchangeable distribution over sequences of subsets known as the beta-Bernoulli process (Thibaux and Jordan, 2007), which is related to the Indian buffet process (IBP, Ghahramani and Griffiths, 2005). If we sample G from a gamma process on Ω with base measure αH, then $D(\cdot) = G(\cdot)/G(\Omega)$ is distributed according to a Dirichlet process with concentration parameter α and base measure H.

Inference in such models tend to fall into three categories: uncollapsed samplers that alternate between sampling the latent measure and the assignments (Ishwaran and Zarepour, 2002; Paisley and Carin, 2009; Zhou et al., 2009; Walker, 2007; Teh et al., 2007); collapsed samplers where the latent measure is integrated out (Ishwaran and James, 2011; Neal, 2000; Ghahramani and Griffiths, 2005); and optimization-based methods that work with approximating distributions where the parameters are assumed to have a mean-field distribution (Blei and Jordan, 2006; Doshi-Velez et al., 2009a).

Collapsed methods often mix slowly due to the dependency between assignments, while blocked updates mean uncollapsed methods typically have good mixing properties at convergence (Ishwaran and James, 2011). Uncollapsed methods are often slow to incorporate new components, since they typically rely on sampling unoccupied components from the prior. In high dimensions, such components are unlikely to be close to the data. Conversely, collapsed methods can make use of the data when introducing new points, which tends to lead to faster convergence (Neal, 2000).

Other methods incorporate both uncollapsed and collapsed sampling, resulting in a "hybrid", partially collapsed sampler, although such approaches have been restricted to specific models. Doshi-Velez and Ghahramani (2009) introduced a linear time accelerated Gibbs sampler for conjugate IBPs that effectively marginalized over the latent factors, while more recently Yerebakan and Dundar (2017) developed a sampler by partially marginalizing latent random measure for DPs. These methods can be seen as special cases of our hybrid framework (Section 3), but do not generalize to the distributed setting.

Several inference algorithms allow computation to be distributed across multiple machines—although again, such algorithms are specific to a single model. The approximate collapsed algorithm of Smyth et al. (2009) is only developed for Dirichlet process-based models, and lacks asymptotic convergence guarantees. Distributed split-merge methods have been developed for Dirichlet process-based models, but not extended to more general nonparametric models (Chang and Fisher, 2013, 2014). Partition-based algorithms based on properties of CRMs have been developed for Dirichlet process- and Pitman-Yor process-based models (Williamson et al., 2013; Dubey et al., 2014), but it is unclear how to extend to other model families. A low-communication, distributed-memory slice sampler has been developed for the Dirichlet process, but since it is based on an uncollapsed method it will tend to perform poorly in high dimensions (Ge et al., 2015). Doshi-Velez et al. (2009b) developed an approximate distributed inference algorithm for the Indian buffet process which is superficially similar to our distributed beta-Bernoulli sampler. However, their approach allows all processors to add new features, which will lead to overestimating the number of features. We contrast

these methods in Table 1.

3 HYBRID INFERENCE FOR CRM-BASED MODELS

By definition, completely random measures can be decomposed into independent random measures. If the CRM has been transformed in some manner we can often still decompose the resulting random measure into independent random measures – for example, a normalized random measure can be decomposed as a mixture of normalized random measures. Such representations allow us to decompose our inference algorithms, and use different inference techniques on the constituent measures.

As discussed in Section 2, collapsed and uncollapsed methods both have advantages and disadvantages. Loosely, collapsed methods are good at adding new components and exploring the tail of the distribution, while uncollapsed methods offer better mixing in established clusters and easy parallelization. We make use of the decomposition properties of CRMs to partition our model into two components: One containing (finitely many) components currently associated with multiple observations, and one containing the infinite tail of components.

3.1 Models Constructed Directly From CRMs

Consider a generic hierarchical model,

$$M := \sum_{k=1}^{\infty} \mu_k \delta_{\theta_k} \sim \text{CRM}(\nu(d\mu)H(d\theta))$$
$$Z_{i,k} \sim f(\mu_k), \qquad X_i = \sum_{k=1}^{\infty} g(Z_{i,k}, \theta_k) + \epsilon_i \tag{1}$$

where ν is a measure on $\mathbb{R}_+$; H is a measure on the space of parameters Θ; $g(\cdot,\cdot)$ is some deterministic transformation such that $g(0,\theta) = 0$ for all $\theta \in \Theta$; ϵ_i is a noise term; and $f(\cdot)$ is a likelihood that forms a conjugate pair with M, i.e. the posterior distribution $P(M^*|Z)$ is a CRM in the same family. The indices $i = 1,\dots,n$ refer to the observations and the indices $k = 1, 2, \dots$ refer to the features. This framework includes exchangeable feature allocation models such as the beta-Bernoulli process (Ghahramani and Griffiths, 2005; Thibaux and Jordan, 2007), the infinite gamma-Poisson feature model (Titsias, 2008), and the beta negative binomial process (Zhou et al., 2012; Broderick et al., 2014). We assume, as is the case in these examples, that both collapsed and uncollapsed posterior inference algorithms can be described. We also assume for simplicity that the prior contains no fixed-location atoms, although this assumption could be relaxed (see Broderick et al., 2018).

Algorithm 1 Hybrid Beta-Bernoulli Sampler

while not converged **do**
  Select J
  Sample $\mu_k \sim \text{Beta}(m_k, n - m_k + c)$, $\forall\ k \le J$
  Sample $\theta_k \sim p(\theta_k|H, Z, X)$, $\forall\ k \le J$
  for $i = 1,\dots,N$ **do**
    Sample $\{Z_{i,k}\}_{i=1,k\le J}^{n}$ according to Eq. 4
    Sample $\{Z_{i,k}\}_{i=1,k>J}^{n}$ according to Eq. 5
    Metropolis-Hastings sample $Z_{i,k'}$ for $k' \in \{k : Z_{i,k} = 1, \sum_{j\neq i} Z_{j,k} = 0\}$.
  end for
end while

Lemma 1 (Broderick et al. 2018). *If $M \sim CRM(\nu(d\mu)H(d\theta))$ and $Z_{i,k} \sim f(\mu_k)$ for $i = 1,\dots,n$, and if M and f form a conjugate pair, then the posterior $P(M^*|\{Z_i\})$ can be decomposed into two CRMs, each with known distribution. If K is the number of features for which $\sum_i Z_{i,k} > 0$, the first, $M^*_{\le K} = \sum_{k=1}^{K} \mu^*_k \delta_{\theta_k}$, is a finite measure with fixed-location atoms at locations $\theta_k : \sum_i Z_{i,k} > 0$. The distribution over the corresponding weights is proportional to $\nu(d\mu)\prod_{i=1}^{n} f(Z_{i,k}|\mu)$. The second, with infinitely many random-location atoms, has Lévy measure $\nu(d\mu)H(d\theta)\,(f(0|\mu))^n$.*

Based on Lemma 1, we partition M^* into a finite CRM $M^*_{\le J} = \sum_{k=1}^{J} \mu^*_k \delta_{\theta_k}$ for some $J \le K$, that contains all, or a subset of, the fixed-location atoms; and an infinite CRM $M^*_{>J}$ that contains the remaining atoms. We use an uncollapsed sampler to sample $M^*_{\le J}|\{Z_{i,k}\}_{k\le K}$, and then sample $\{Z_{i,k}\}_{k\le K}|X, M^*_{\text{fixed}}$. Then, we use a collapsed sampler to sample the allocations $Z_{i,k} : k > J$. The size J should be changed periodically to make sure $J \le K$ to avoid explicitly instantiating atoms that are not associated with data. In our experiments, we set $J = K$ at the beginning of each iteration.

Example 1: Beta-Bernoulli Process. As a specific example, consider the beta-Bernoulli process. Let $B := \sum_{k=1}^{\infty} \mu_k \delta_{\theta_k} \sim \text{BetaP}(\alpha, c, H)$ be a homogeneous beta process (Thibaux and Jordan, 2007), and let $Z_{i,k} \sim \text{Bernoulli}(\mu_k)$. The posterior is given by

$$B^*|Z \sim \text{BetaP}\left(\frac{c\alpha + \sum_k m_k}{c+n}, c+n, \frac{c\alpha H + \sum_k m_k \delta_{\theta_k}}{c\alpha + \sum_k m_k}\right)$$

where $m_k = \sum_{i=1}^{N} Z_{i,k}$. In this case the following lemma helps us in decomposing the posterior distribution:

Lemma 2 (Thibaux and Jordan 2007). *If K is the number of features where $\sum_i Z_{i,k} > 0$ and $J \le K$, we can decompose the posterior distribution of beta-*

Bernoulli process as as $B^* = B^*_{\leq J} + B^*_{>J}$ *where*

$$B^*_{\leq J} \sim BetaP\left(\frac{\sum_{k=1}^{J} m_k}{c+n}, c+n, \frac{\sum_k m_k \delta_{\theta_k}}{\sum_{k=1}^{J} m_k}\right)$$
$$B^*_{>J} \sim BetaP\left(\frac{c\alpha}{c+n}, c+n, \frac{c\alpha H + \sum_{k>J} m_k \delta_{\theta_k}}{c\alpha + \sum_{k>J} m_k}\right) \quad (2)$$

We note that the atom sizes of $B^*_{\leq J}$ are Beta$(m_k, n - m_k + c)$ random variables. This allows us to split the beta-Bernoulli process into two independent feature selection mechanisms: one with a finite number of currently instantiated features, and one with an unbounded number of features.

The likelihood of a given data point, given the latent variables and other data points (written as $P(X_i|-)$ for space reasons), can be written as

$$p(X_i|-) = \int p(X_i|Z_i, \theta_1, \ldots, \theta_J, \ldots, \theta_K) \\ p(\theta_{J+1}, \ldots, \theta_K|Z_{-i}, X_{-i}) d\theta_{J+1} \cdots d\theta_K. \quad (3)$$

When working with conjugate likelihoods, we can typically evaluate the integral term in Equation 3 analytically (see Ghahramani and Griffiths (2005) for the case of Gaussian prior and linear Gaussian likelihood for $J = 0$). If this is not possible, we can sample θ_k for $J \geq k \geq K$ as auxiliary variables (Doshi-Velez and Ghahramani, 2009).

This formulation of the likelihood, combined with the partitioning of the Bernoulli process described in Equation 2, gives us the hybrid sampler, which we summarize in Algorithm 1. For each data point X_i, we sample $Z_{i,k}$ in a three step manner. For $k \leq J$,

$$P(Z_{i,k} = z|-) \propto \\ \begin{cases} \mu_k p(X_i|Z_{i,k} = 1, -) & z = 1 \\ (1 - \mu_k) p(X_i|Z_{i,k} = 0, -) & z = 0. \end{cases} \quad (4)$$

where $p(X_i|Z_{i,k} = 1, -)$ is given by Equation 3 with $Z_{i,k}$ set to 1. For $k > J$ and $m_k > 0$, we have

$$P(Z_{i,k} = z|Z_{-(i,k)}, X_i) \propto \\ \begin{cases} m_k p(X_i|Z_{i,k} = 1, -) & z = 1 \\ (n - m_k) p(X_i|Z_{i,k} = 0, -) & z = 0. \end{cases} \quad (5)$$

where $p(X_i|Z_{i,k} = 1, -)$ is given by Equation 3 with $Z_{i,k}$ set to 0. Finally, we propose adding Poisson(α/n) new features, accepting using a Metropolis-Hastings step. Once we have sampled the $Z_{i,k}$, for every instantiated feature $k \leq J$, we sample $\mu_k \sim$ Beta$(m_k, n - m_k + c)$ and its corresponding parameters $\theta_k \sim p(\theta_k|H, Z, X)$.

We note that similar algorithms can be easily derived for other nonparmetric latent feature models such as those based on the infinite gamma-Poisson process (Thibaux and Jordan, 2007) and the beta-negative Binomial process (Zhou et al., 2012; Broderick et al., 2018).

Algorithm 2 Hybrid DPMM Sampler

1: **while** not converged **do**
2: Select J
3: Sample $B^* \sim$ Beta$(\tilde{n}, n - \tilde{n} + \alpha)$
4: Sample $(\pi_1, \ldots, \pi_J) \sim$ Dir$(m_1, \ldots, m_J)$
5: For $k \leq J$, sample $\theta_k \sim p(\theta_k|H, \{X_i : Z_i = k\})$
6: Sample Z_i, $\forall i = 1, \ldots, n$, using Equation 7
7: **end while**

3.2 Models Based on Transformations of Random Measures

While applying transformations to CRMs means the posterior is no longer directly decomposable, in some cases we can still apply the above general ideas.

Example 2: Dirichlet Process. As noted in Section 2, the Dirichlet process with concentration parameter α and base measure H can be constructed by normalizing a gamma process with base measure αH. If the Dirichlet process is used as the prior in a mixture model (DPMM), the posterior distribution conditioned on the cluster allocations $Z_1, \ldots, Z_n$, having K unique clusters is again a Dirichlet process:

$$D^*|Z_1, \ldots Z_n \sim \text{DP}(\alpha + n, \frac{\alpha H + \sum_{k<K} m_k \delta_{\theta_k}}{n + \alpha}) \quad (6)$$

where $m_k = \sum_i \delta(Z_i, k)$ and K is the number of clusters with $m_k > 0$. In this case also the following lemma helps us in decomposing the posterior.

Lemma 3. *Assuming* J, *and* $\tilde{n} = \sum_{k \leq J} m_k$, *we can decompose the posterior of DP as*

$$D^*|Z_1, \ldots, Z_n = B^* D^*_{\leq J} + (1 - B^*) D^*_{>J}$$

where

$$D^*_{\leq J}|Z_1, \ldots, Z_n \sim DP(\tilde{n}, \frac{\sum_{k \leq J} m_k \delta_{\theta_k}}{\tilde{n}})$$
$$D^*_{>J}|Z_1, \ldots, Z_n \sim DP(\alpha + n - \tilde{n}, \frac{\alpha H + \sum_{k>J} m_k \delta_{\theta_K}}{\alpha + n - \tilde{n}})$$
$$B^* \sim Beta(\tilde{n}, n - \tilde{n} + \alpha)$$

Proof. This is a direct extension of the fact that the Dirichlet process has Dirichlet-distributed marginals (Ferguson, 1973). See Chapter 3 of Ghosh and Ramamoorthi (2003) for a detailed analysis. □

We note that the posterior atom weights $(\pi_1, \ldots, \pi_J)$ for the finite component are distributed according to Dirichlet$(m_1, \ldots, m_J)$, and can easily be sampled as part of an uncollapsed sampler. Conditioned on $\{\pi_k, \theta_k : k \leq J\}$ and B^* we can sample the cluster

allocation, Z_i of point X_i as

$$P(Z_i = k|-) \propto \begin{cases} B^*\pi_k f(x_n;\theta_k) & k \leq J \\ \frac{(1-B^*)m_k}{\sum_{j>K} m_j+\alpha} f_k(x_n) & J < k \leq K \\ \frac{(1-B^*)\alpha}{\sum_{j>K} m_j+\alpha} f_H(x_n) & k = K+1 \end{cases} \tag{7}$$

where $f(X_i;\theta_k)$ is the likelihood for each mixing component; $f_k(X_i) = \int_\Theta f(X_i;\theta)p(\theta|\{X_j : Z_j = k, j \neq i\})d\theta$ is the conditional probability of x_i given other members of the kth cluster; and $f_H(x_i) = \int_\Theta f(x_i;\theta)H(d\theta)$. This procedure is summarized in Algorithm 2.

Example 3: Pitman-Yor processes The Pitman-Yor process (Perman et al., 1992; Pitman and Yor, 1997) is a distribution over probability measures, parameterized by $0 \leq \sigma < 1$ and $\alpha > -\sigma$, that is obtained from a σ-stable CRM via a change of measure and normalization. Provided $\alpha > 0$, it can also be represented as a Dirichlet process mixture of normalized σ-stable CRMs (Lemma 22, Pitman and Yor, 1997). This representation allows us to decompose the posterior distribution into a beta mixture of a finite-dimensional Pitman-Yor process and an infinite-dimensional Pitman-Yor process. We provide more details in the supplementary section A.

Example 4: Hierarchical Dirichlet processes We can decompose the hierarchical Dirichlet process (HDP, Teh et al., 2006) in a manner comparable to the Dirichlet process, allowing our hybrid sampler to be used on the HDP. For space reasons, we defer discussion to the supplementary section B.

4 DISTRIBUTED INFERENCE FOR CRM-BASED MODELS

The sampling algorithms in Section 3 can easily be adapted to a distributed setting, where data are partitioned across several P machines and communication is limited. In this setting, we set $J = K$ (i.e. the number of currently instantiated features) after every communication step. We instantiate the finite measure ($M^*_{\leq J}$ in the case of CRMs, $D^*_{>J}$ for DPMMs), with globally shared atom sizes and locations, on all processors.

We then randomly select one out of P processors by sampling $P^* \sim \text{Uniform}(1,\ldots,P)$. On the remaining $P-1$ processors, we sample the allocations Z_i using restricted Gibbs sampling (Neal, 2000), enforcing $Z_{i,k} = 0$ for $k > J$. When working with DPMMs, this means that we only need to calculate cluster probabilities that depend on the instantiated measure $D^*_{\leq J}$. In the CRM case, it means that we avoid the integral in Equation 3, and hence avoid any dependence on Z_{-i} or X_{-i} when conditioning on $M^*_{\leq J}$. In both cases, this means that we do not need knowledge of the feature allocations on other processors, and can sample the Z_i independent of each other.

Algorithm 3 Distributed Beta Bernoulli Sampler

procedure LOCAL($\{X_i, Z_i\}, P$)
  ▷ Global variables J, P^*, $\{\theta_k, \mu_k\}_{k=1}^J, n$
  for $k \leq J$ **do**
    Sample $\{Z_{i,k}\}$ according to (4)
  end for
  if $P = P^*$ **then**
    For $k > J$, sample $Z_{i,k}$ according to (5)
    Sample Poisson(α/n) new features
  end if
end procedure
procedure GLOBAL($\{X_i, Z_i\}$)
  Gather feature counts m_k and parameter sufficient statistics Ψ_k from all processors.
  Let J be the number of instantiated features.
  For $k : m_k > 0$, sample

$$\mu_k \sim \text{Beta}(m_k, n - m_k + c)$$
$$\theta_k \sim p(\theta_k|\Psi_k, H)$$

  Sample $P^* \sim \text{Uniform}(1,\ldots,P)$
end procedure

On the remaining processor P^*, we sample the Z_i using unrestricted Gibbs sampling. Let $\mathcal{P}^*$ be the set of indices of data points on processor P^*. Since we know that $Z_{j,k} = 0$ for all $j \notin \mathcal{P}^*$, then we can calculate the full sufficient statistics for features $k > J$ without knowledge of data or latent features on other processors. While $P(Z_i|-)$ does depend on $\{Z_j, X_j : j \in \mathcal{P}\}$, it is independent of $\{Z_j, X_j : j \notin \mathcal{P}\}$ conditioned on $D^*_{\leq J}$ (or $M^*_{\leq J}$) plus the fact that $Z_{j,k} = 0$ for all $j \notin \mathcal{P}^*$, so can be calculated without further information from the other processors.

At each global step, we gather the sufficient statistics from all instantiated clusters – from both the finite component $M^*_{\leq J}$ / $D^*_{\leq J}$ and the infinite component $M^*_{>J}$ / $D^*_{>J}$ – and sample parameters for those clusters. We then create a new partition, redefining J as the current number of instantiated component parameters. In the case of the DPMM, we also resample $B \sim \text{Beta}(N, \alpha)$. We summarize the distributed algorithm for the special cases of the beta-Bernoulli process and the DPMM in Algorithms 3 and 4 and for PYMM in algorithm 6 in supplementary.

4.1 Warm-start Heuristics

In our distributed approach, only $1/P$ of the data points eligible to start a new cluster or feature, mean-

Algorithm 4 Distributed DPMM Sampler

procedure LOCAL($\{X_i, Z_i\}$)
  ▷ Global variables $J, P^*, \{\theta_k, \pi_k\}_{k=1}^J, B^*$
  if $P = P^*$ **then**
    Sample Z_i according to (7)
  else
    $P(Z_i = k) \propto \pi_k f(X_i; \theta_k)$
  end if
end procedure
procedure GLOBAL($\{X_i, Z_i\}$)
  Gather cluster counts m_k and parameter sufficient statistics Ψ_k from all processors.
  Sample $B^* \sim \text{Beta}(n, \alpha)$
  Let J be the number of instantiated clusters.
  Sample $(\pi_1, \ldots, \pi_J) \sim \text{Dir}(m_1, \ldots, m_J)$
  For $k : m_k > 0$, sample $\theta_k \sim p(\theta_k | \Psi_k, H)$.
  Sample $P^* \sim \text{Uniform}(1, \ldots, P)$
end procedure

ing that the rate at which new clusters/features are added will decrease with the number of processors. This can lead to slow convergence if we start with too few clusters. To avoid this problem, we initialize our algorithm by allowing *all* processors to instantiate new clusters. At each global step, we decrease the number of randomly selected processors eligible to instantiate new clusters, until we end up with a single processor. This approach tend to over-estimate the number of clusters. However, the procedure acts in a manner similar to simulated annealing, by encouraging large moves early in the algorithm but gradually decreasing the excess stochasticity until we are sampling from the correct algorithm. We note that a sampler with multiple processors instantiating new clusters is not a correct sampler until we revert to a single processor proposing new features, because correct MCMC samplers are invariant to its starting position.

5 EXPERIMENTAL EVALUATION

While our primary contribution is in the development of distributed algorithms, we first consider, in Section 5.1, the performance of the hybrid algorithms developed in Section 3 in a non-distributed setting. We show that this performance extends to the distributed setting, and offers impressive scaling, in Section 5.2.

5.1 Evaluating the Hybrid Sampler

We begin by considering the performance of the hybrid samplers introduced in Section 3 in a non-distributed setting. For this, we focus on the Dirichlet process, since there exist a number of collapsed and uncollapsed inference algorithms; we expect similar results under other models.

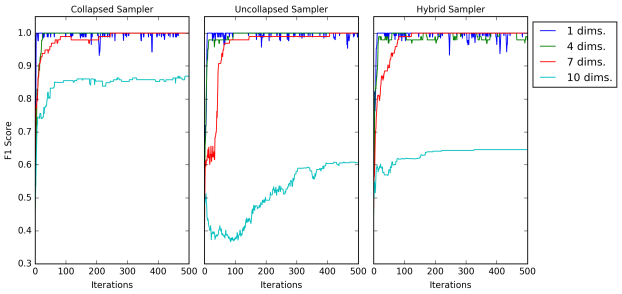

Figure 1: Synthetic data experiments. Comparison of F1 scores over iteration for the collapsed, uncollapsed and hybrid samplers for growing dimensions.

We compare the hybrid sampler of Algorithm 2 with a standard collapsed Gibbs sampler and an uncollapsed sampler based on Algorithm 8 of Neal (2000). Algorithm 8 collapses occupied clusters and instantiates a subset of unoccupied clusters; we modify this to instantiate the atoms associated with unoccupied clusters. Concretely, at each iteration, we sample weights for the K instantiated clusters plus U uninstantiated clusters as $(\pi_1, \ldots, \pi_K, \pi_{K+1}, \ldots, \pi_{K+U}) \sim \text{Dir}\left(m_1, \ldots, m_K, \frac{\alpha}{J}, \ldots, \frac{\alpha}{J}\right)$, and sample locations for the uninstantiated clusters from the base measure H. Note that this method can be distributed easily.

Figure 1 shows convergence plots for the three algorithms. The data set is a D dimensional synthetic data set consisting of 100 observations of Gaussian mixtures with 2 true mixture components centered at $5 \times \{1\}^D$ and $-5 \times \{1\}^D$ with an identity covariance matrix.

While the three algorithms perform comparably on low-dimensional data, as the dimension increases the performance of the uncollapsed sampler degrades much more than the collapsed sampler. This is because in high dimensions, it is unlikely that a proposed parameter will be near our data, so the associated likelihood of any given data point will be low. This is in contrast to the collapsed setting, where we integrate over all possible locations. While the hybrid method performs worse in high dimensions than the collapsed method, it outperforms the uncollapsed method.

The synthetic data in Figure 1 has fairly low-dimensional structure, so we do not see negative effects due to the poor mixing of the collapsed sampler. Next, we evaluate the algorithms on the CIFAR-100 dataset (Krizhevsky, 2009). We used PCA to reduce the dimension of the data to between 8 and 64, and plot the test set log likelihood over time in Figure 2. Each marker represents a single iteration. We see that the uncollapsed sampler requires more iterations to con-

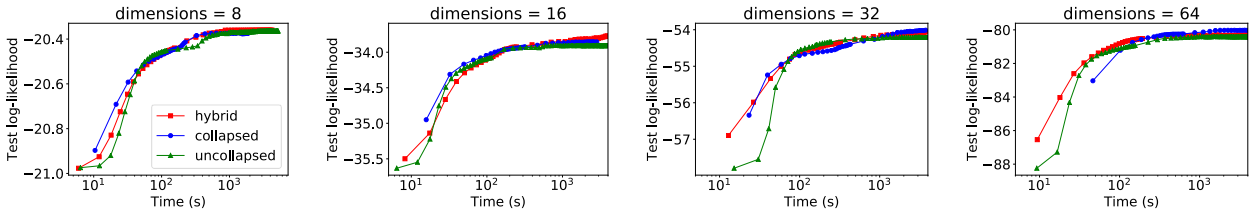

Figure 2: Test log-likelihood with time on CIFAR-100 dataset as the dimensionality increases from 8 to 64.

verge than the collapsed sampler; however since each iteration takes less time, in some cases the wall time to convergence is slower. The hybrid method has comparable iteration time to the collapsed, but, in general, converges faster. We see that, even without taking advantage of parallelization, the hybrid method is a compelling competitor to pure-collapsed and pure-uncollapsed algorithms.

5.2 Evaluating the Distributed Samplers

Here, we show that the distributed inference algorithms introduced in Section 4 allow inference in BNP models to be scaled to large datasets, without sacrificing accuracy. We focus on two cases: the beta-Bernoulli process (Algorithm 3) and the Dirichlet process (Algorithm 4)[1].

5.2.1 Beta-Bernoulli Process

We evaluate the beta-Bernoulli sampler on synthetic data based on the "Cambridge" dataset, used in the original IBP paper (Griffiths and Ghahramani, 2011), where each datapoint is the superposition of a randomly selected subset of four binary features of dimension 36, plus Gaussian noise with standard deviation 0.5.[2] We model this data using a linear Gaussian likelihood, with $Z \sim \text{Beta-Bern}(\alpha, 1)$, $A_k \sim \text{Normal}(0, \sigma_A^2 \mathbf{I})$, $X_n \sim \text{Normal}\left(\sum_k z_{nk} A_k, \sigma_X^2 \mathbf{I}\right)$

We initialized to a single feature, and ran the hybrid sampler for 1,000 total observations with a synchronization step every 5 iterations, distributing over 1, 4, 8, 16, 32, 64 and 128 processors.

We first evaluate the hybrid algorithm under a "cold start", where only one processor is allowed to introduce new features for the entire duration of the sampler. In the top left plot of Figure 3, we see that the cold start results in slow convergence of in the test set log likelihood for large numbers of processors. We can see in the bottom left plot of Figure 3 that the number of features grows very slowly, as only $1/P$ processors are allowed to propose new features in the exact setting.

Next, we explore warm-start initialization, as described in Section 4.1. For the first one-eighth of the total number of MCMC iterations, all processors can propose new features; after this we revert to the standard algorithm. The top central plot of Figure 3 shows predictive log likelihood over time, and the bottom central plot shows number of features. We see that convergence is significantly improved relative to the cold-start experiments. Since we revert to the asymptotically correct sampler, the final number of features is generally close to the true number of features, 4.[3] Additionally, we see that convergence rate increases monotonically in the number of processors.

Next, we allowed all processors to propose new features for the entire duration ("always-hot"). This setting approximately replicates the behavior of the parallel IBP sampler of Doshi-Velez et al. (2009b). In the top right plot of Figure 3, we can see that all experiments roughly converge to the same test log likelihood. However, the number of features introduced (bottom right plot) is much greater than the warm start experiment, grows with the number of processors. Moreover, the difference in convergence rates between processors is not as dramatic as in the warm-start trials.

Next, we demonstrate the scalability of our distributed algorithm on a massive synthetic example, showing it can be used for large-scale latent feature models. We generate one million "Cambridge" synthetic data points, as described for the previous experiments, and distribute the data over 256 processors. This experiment represents the largest experiment ran for a beta-Bernoulli process algorithm (the next largest being 100,000 data points, in Doshi-Velez et al. 2009b). We limited the sampler to run for one day and completed 860 MCMC iterations. In Figure 4, we see in the test set log likelihood traceplot that we can converge to a local mode fairly quickly under a massive distributed

[1] Our code is available online at `https://github.com/michaelzhang01/hybridBNP`

[2] See Figure 6 in the supplement for more details

[3] Note that BNP models are not guaranteed to find the correct number of features in the posterior; see Miller and Harrison (2013)

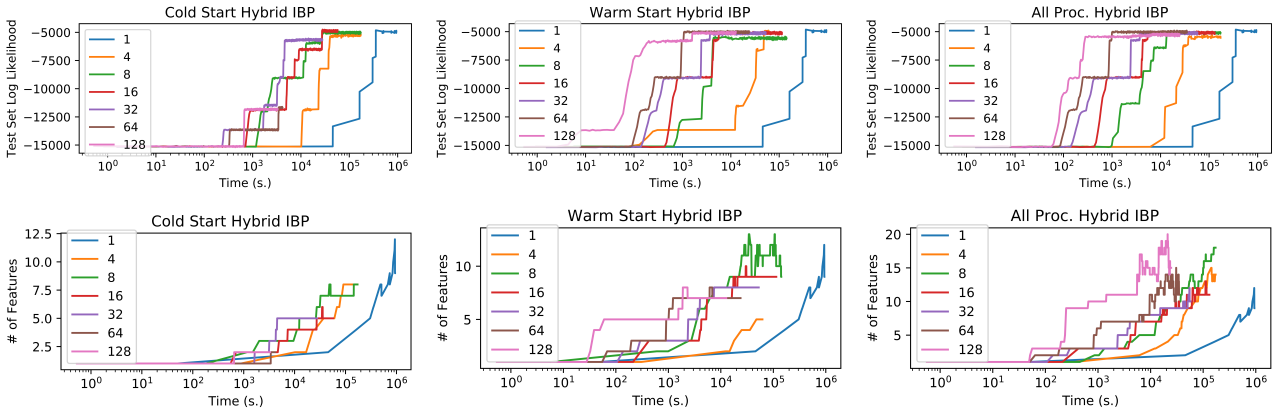

Figure 3: **Top row left to right**: Test set log likelihood (y-axis) on synthetic data without warm-start initialization, with warm start, and with all processors on. The x-axis represents CPU wall time in seconds, on a log scale. **Bottom row left to right** number of features over iteration with cold-start, warm-start, and all processors introducing features. Y-axis represents number of instantiated features.

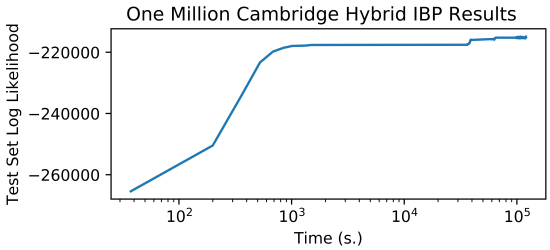

Figure 4: Test set log likelihood trace plot for a million observation "Cambridge" data set.

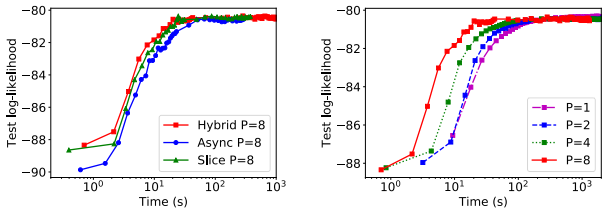

Figure 5: Comparison of CIFAR-100 test log-likelihood with time, with baseline methods (**left**) and with varying number of processor (**right**)

setting.

5.2.2 Dirichlet Process

Our distributed inference framework can also speed up inference in a DP mixture of Gaussians, using the version described in Algorithm 4. We used a dataset containing the top 64 principle components of the CIFAR-100 dataset, as described in Section 5.1. We compared against two existing distributed inference algorithms for the Dirichlet process mixture model, chosen to represent models based on both uncollapsed and collapsed samplers: 1) A DP variant of the asynchronous sampler of Smyth et al. (2009), an approximate collapsed method; and 2) the distributed slice sampler of Ge et al. (2015), an uncollapsed method.

Figure 5 shows, when distributed over eight processors, our algorithm converges faster than the two comparison methods, showing that the high quality performance seen in Section 5.1 extends to the distributed setting. Further, in Figure 5 we see roughly linear speed-up in convergence as we increase the number of processors from 1 to 8.

6 Conclusion

We have proposed a general inference framework for a wide variety of BNP models. We use the inherent decomposability of the underlying completely random measures to partition the latent random measures into a finite-dimensional component that represents the majority of the data, and an infinite-dimensional component that represents mostly uninstantiated tail. This allows us to take advantage of the inherent parallelizability of the uncollapsed sampler on the finite partition and the better performance of the collapsed sampler for proposing new components. Thus the proposed hybrid inference method can be easily distributed over multiple machines, providing provably correct inference for many BNP models. Experiments show that, for both the DP and the beta-Bernoulli process, our proposed distributed hybrid sampler converges faster than the comparison methods.

References

Blei, D. M. and Jordan, M. I. (2006). Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 1(1):121–143.

Broderick, T., Mackey, L., Paisley, J., and Jordan, M. I. (2014). Combinatorial clustering and the beta negative binomial process. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):290–306.

Broderick, T., Wilson, A. C., and Jordan, M. I. (2018). Posteriors, conjugacy, and exponential families for completely random measures. *Bernoulli*, 24(4B):3181–3221.

Chang, J. and Fisher, J. W. (2013). Parallel sampling of DP mixture models using sub-cluster splits. In *Advances in Neural Information Processing Systems*, pages 620–628.

Chang, J. and Fisher, J. W. (2014). Parallel sampling of HDPs using sub-cluster splits. In *Advances in Neural Information Processing Systems*, pages 235–243.

Doshi-Velez, F. and Ghahramani, Z. (2009). Accelerated sampling for the Indian buffet process. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 273–280. ACM.

Doshi-Velez, F., Miller, K., Van Gael, J., and Teh, Y. W. (2009a). Variational inference for the Indian buffet process. In *Artificial Intelligence and Statistics*, pages 137–144.

Doshi-Velez, F., Mohamed, S., Ghahramani, Z., and Knowles, D. A. (2009b). Large scale nonparametric Bayesian inference: Data parallelisation in the Indian buffet process. In *Advances in Neural Information Processing Systems*, pages 1294–1302.

Dubey, A., Williamson, S., and Xing, E. P. (2014). Parallel Markov chain Monte Carlo for Pitman-Yor mixture models. In *Uncertainty in Artificial Intelligence*, pages 142–151.

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

Ge, H., Chen, Y., Wan, M., and Ghahramani, Z. (2015). Distributed inference for Dirichlet process mixture models. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2276–2284.

Ghahramani, Z. and Griffiths, T. L. (2005). Infinite latent feature models and the Indian buffet process. In *Advances in Neural Information Processing Systems*, pages 475–482.

Ghosh, J. K. and Ramamoorthi, R. V. (2003). Bayesian non-parametrics. In *Springer*.

Griffiths, T. L. and Ghahramani, Z. (2011). The Indian buffet process: An introduction and review. *Journal of Machine Learning Research*, 12:1185–1224.

Hjort, N. L. (1990). Nonparametric Bayes estimators based on beta processes in models for life history data. *Annals of Statistics*, pages 1259–1294.

Ishwaran, H. and James, L. F. (2011). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*.

Ishwaran, H. and Zarepour, M. (2002). Exact and approximate sum representations for the Dirichlet process. *Canadian Journal of Statistics*, 30(2):269–283.

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

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images.

Miller, J. W. and Harrison, M. T. (2013). A simple example of Dirichlet process mixture inconsistency for the number of components. In *Advances in Neural Information Processing Systems*, pages 199–206.

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9(2):249–265.

Paisley, J. and Carin, L. (2009). Nonparametric factor analysis with beta process priors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 777–784. ACM.

Perman, M., Pitman, J., and Yor, M. (1992). Size-biased sampling of Poisson point processes and excursions. *Probability Theory and Related Fields*, 92(1):21–39.

Pitman, J. (1996). Some developments of the Blackwell-MacQueen urn scheme. In *Statistics, Probability and Game Theory*, pages 245–267. Institute of Mathematical Statistics.

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

Smyth, P., Welling, M., and Asuncion, A. U. (2009). Asynchronous distributed learning of topic models. In *Advances in Neural Information Processing Systems*, pages 81–88.

Teh, Y. W., Görür, D., and Ghahramani, Z. (2007). Stick-breaking construction for the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, pages 556–563.

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

Thibaux, R. and Jordan, M. I. (2007). Hierarchical beta processes and the Indian buffet process. In *International Conference on Artificial Intelligence and Statistics*, pages 564–571.

Titsias, M. K. (2008). The infinite gamma-Poisson feature model. In *Advances in Neural Information Processing Systems*, pages 1513–1520.

Walker, S. G. (2007). Sampling the Dirichlet mixture model with slices. *Communications in Statistics—Simulation and Computation*, 36(1):45–54.

Weyrauch, B., Heisele, B., Huang, J., and Blanz, V. (2004). Component-based face recognition with 3D morphable models. In *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 85–85. IEEE.

Williamson, S., Dubey, A., and Xing, E. (2013). Parallel Markov chain Monte Carlo for nonparametric mixture models. In *Proceedings of the 30th International Conference on Machine Learning*, pages 98–106.

Yerebakan, H. Z. and Dundar, M. (2017). Partially collapsed parallel Gibbs sampler for Dirichlet process mixture models. *Pattern Recognition Letters*, 90:22–27.

Zhou, M., Chen, H., Ren, L., Sapiro, G., Carin, L., and Paisley, J. W. (2009). Non-parametric Bayesian dictionary learning for sparse image representations. In *Advances in Neural Information Processing Systems*, pages 2295–2303.

Zhou, M., Hannah, L., Dunson, D., and Carin, L. (2012). Beta-negative binomial process and poisson factor analysis. In *Artificial Intelligence and Statistics*, pages 1462–1471.

Distributed, partially collapsed MCMC for Bayesian Nonparametrics: Supplement

A Hybrid Sampler for the Pitman Yor Mixture Model (PYMM)

In this section, we expand upon Example 3 in Section 3.

Example 3: Pitman-Yor processes The Pitman-Yor process (Perman et al. (1992); Pitman and Yor (1997)) is a distribution over probability measures, parametrized by a discount parameter $0 \leq \sigma < 1$, a concentration parameter $\alpha > -\sigma$, and a base measure H. While the Pitman-Yor process is not a normalized CRM, it can be derived from a σ-stable CRM via a change of measure and normalization. When the discount parameter is zero, we recover the Dirichlet process. As the discount parameter increases, we get increasingly heavy-tailed distributions over the atom sizes in the resulting probability measure.

Lemma 4. *If $D \sim PY(\alpha, \sigma, H)$ with $\alpha > 0$, and $Z_i \sim D$, then the posterior distribution $P(Y^*|Z_1, \ldots, Z_n)$ is described by*

$$\begin{aligned} D^*_{\leq J} \sim & PY\left(\tilde{n} - J\sigma, 0, \frac{\sum_{k\leq j}(m_k - \sigma)\delta_{\theta_k}}{\tilde{n} - J\sigma}\right) \\ D^*_{>J} \sim & PY\Big(\alpha + n - \tilde{n} + J\sigma, \sigma, \\ & \qquad \frac{(\alpha + Kd)H + \sum_{k>J}(m_k - \sigma)\delta_{\theta_k}}{\alpha + n - \tilde{n} + J\sigma}\Big) \\ B \sim & Beta(\tilde{n} - J\sigma, \alpha + n - \tilde{n} + J\sigma) \\ D^* = & BD^*_{\leq J} + (1 - B)D^*_{>J} \end{aligned} \tag{8}$$

where K is the number of occupied clusters, $J \leq K$, and $\tilde{n} = \sum_{k=1}^{J} m_k$.

Proof. The proof is a direct consequence of Lemma 22 in Pitman and Yor (1997) and Theorem 1 in Dubey et al. (2014). The special case for $J = K$ is presented in Corollary 20 of Pitman (1996). □

We note that the posterior atom weights $(\pi_1, \ldots, \pi_J)$ for the finite component $D^*_{\leq J}$ are distributed according to Dirichlet$(m_1 - \sigma, \ldots, m_J - \sigma)$, and can easily be sampled as part of an uncollapsed sampler. Conditioned on $\{\pi_k, \theta_k : k \leq J\}$ and B^* we can sample the cluster allocation, Z_i of point X_i as

$$P(Z_i = k|-) \propto \begin{cases} B^*\pi_k f(x_n; \theta_k) & k \leq J \\ \frac{(1-B^*)(m_k - \sigma)}{n - \tilde{n} + \alpha + J\sigma} f_k(x_n) & J < k \leq K \\ \frac{(1-B^*)(\alpha + K\sigma)}{n - \tilde{n} + \alpha + J\sigma} f_H(x_n) & k = K + 1 \end{cases} \tag{9}$$

where $f(X_i; \theta_k)$ is the likelihood for each mixing component; $f_k(X_i) = \int_\Theta f(X_i; \theta)p(\theta|\{X_j : Z_j = k, j \neq i\})d\theta$ is the conditional probability of x_i given other members of the kth cluster; and $f_H(x_i) = \int_\Theta f(x_i; \theta)H(d\theta)$. This procedure is summarized in Algorithm 5.

We can similarly derive the distributed sampler for PYMM shown in algorithm 6.

Algorithm 5 Hybrid PYMM Sampler

while not converged **do**
  Select J
  Sample $B^* \sim \text{Beta}(\tilde{n} - J\sigma, n - \tilde{n} + \alpha + J\sigma)$
  Sample $(\pi_1, \ldots, \pi_J) \sim \text{Dir}(m_1 - \sigma, \ldots, m_J - \sigma)$
  For $k \leq J$, sample
$$\theta_k \sim p(\theta_k | H, \{X_i : Z_i = k\})$$
  For each data point X_n sample Z_i according to Equation 9
end while

Algorithm 6 Distributed PYMM Sampler

procedure LOCAL($\{X_i, Z_i\}$)
  ▷ Global variables $J, P^*, \{\theta_k, \pi_k\}_{k=1}^J, B^*$
  if *Processor* $= P*$ **then**
    Sample Z_i according to (9)
  else
    $P(Z_i = k) \propto \pi_k f(X_i; \theta_k)$
  end if
end procedure
procedure GLOBAL($\{X_i, Z_i\}$)
  Gather cluster counts m_k and parameter sufficient statistics Ψ_k from all processors.
  Let J be the number of instantiated clusters.
  Sample $B^* \sim \text{Beta}(n - J\sigma, \alpha + J\sigma)$
  Sample $(\pi_1, \ldots, \pi_J) \sim \text{Dir}(m_1 - \sigma, \ldots, m_J - \sigma)$
  For $k : m_k > 0$, sample
$$\theta_k \sim p(\theta_k | \Psi_k, H)$$
  Sample $P^* \sim \text{Uniform}(1, \ldots, P)$
end procedure

B Hybrid Sampler for Hierarchical Dirichlet Processes

In this section, we expand upon Example 4 in Section 3.

Example 4: Hierarchical Dirichlet Process. Hierarchical Dirichlet processes (HDPs, Teh et al., 2006) extend the DP to model grouped data. The Hierarchical Dirichlet Process is a distribution over probability distributions $D_s, s = 1, \ldots, S$, each of which is conditionally distributed according to a DP. These distributions are coupled using a discrete common base-measure H, itself distributed according to a DP. Each distribution D_s can be used to model a collection of observations $\mathbf{x}_s := \{x_{si}\}_{i=1}^{N_s}$, where

$$\begin{array}{cc} H \sim \text{DP}(\alpha, D_0), & D_s \sim \text{DP}(\gamma, H), \\ \theta_{si} \sim D_s, & x_{si} \sim f(\theta_{si}), \end{array} \tag{10}$$

for $s = 1, \ldots, S$ and $i = 1, \ldots, N_s$.

We consider a Chinese Restaurant Franchise (CRF, Teh et al., 2006) representation of the HDP, where each data point is represented by a customer, each atom in D_s is represented by a table, and each atom location in the support of H is represented by a dish. Let x_{si} represent the ith customer in the sth restaurant; let t_{si} be the table assignment of customer x_{si}; let k_{st} be the dish assigned to table t in restaurant s. Let m_{sk} denote the number of tables in restaurant s serving disk k, and n_{stk} denote the number of customers in restaurant s at table t having dish k.

Lemma 5. *Conditioned on the table/dish assignment counts* $m_{\cdot k} = \sum_s m_{sk}$*, the posterior distribution* $P(H^* | \{t_{si}\}, \{k_{st}\}$ *can be written as*

$$H^* = B^* H^*_{\leq J} + (1 - B^*) H^*_{>J}$$

where

$$H^*_{\leq J}|m_{\cdot 1},\ldots,m_{\cdot K} \sim DP\left(\tilde{m}, \frac{\sum_{k\leq J} m_{\cdot k}\delta_{\phi_k}}{\tilde{m}}\right)$$
$$H^*_{>J}|m_{\cdot 1},\ldots,m_{\cdot K} \sim DP\left(\alpha+m-\tilde{m}, \frac{\alpha H + \sum_{k>J} m_{\cdot k}\delta_{\phi_k}}{\alpha+m-\tilde{m}}\right)$$
$$B^* \sim Beta(\tilde{m}, m-\tilde{m}+\alpha),$$

where K is the total number of instantiated dishes; $J \leq K$; $m = \sum_{k=1}^K m_{\cdot k}$; and $\tilde{m} = \sum_{k=1}^J m_{\cdot k}$

Proof. This is a direct extension of Lemma 3, applied to the top-level Dirichlet process H. □

We can therefore construct a hybrid sampler, where $H^*_{\leq J} := \sum_{k=1}^J \beta_k \delta_{\phi_k}$ is represented via $(\beta_1,\ldots,\beta_J) \sim \text{Dir}(m_{\cdot 1},\ldots,m_{\cdot J})$ and corresponding $\phi_k \sim h(\phi_k)\prod_{s,i:k_{st_{si}}=k} f(x_{si}|\phi_k)$, and $H^*_{>J}$ is represented using a Chinese restaurant process. We can then sample the table allocations according to

$$t_{si} = t|t_{-si},\{k_{st}\},- \propto \begin{cases} \frac{n^{-si}_{st\cdot}}{n^{-si}_{s\cdot\cdot}+\gamma} f(x_{si};\phi_{k_{st}}) & \text{if } t \text{ previously used and } k_{st}\leq J \\ B^*\frac{\gamma}{n^{-si}_{s\cdot\cdot}+\gamma}\beta_{k_{st}} f(x_{si};\phi_{k_{st}}) & t = \text{ new table table and } k_{st}\leq J \\ (1-B^*)\frac{\gamma}{n^{-si}_{s\cdot\cdot}+\gamma}\frac{m_{\cdot k_{st}}}{\sum_{k:k>J} m_{\cdot k}+\alpha} f(x_{si}|t_{-si},,-;D_0) & \text{if } t \text{ new and } k_{st}>J \\ (1-B^*)\frac{\gamma}{n^{-si}_{s\cdot\cdot}+\gamma}\frac{\alpha}{\sum_{k:k>J} m_{\cdot k}+\alpha} f(x_{si};D_0) & t \text{ new and } k=K+1 \end{cases} \tag{11}$$

and sample each table according to

$$p(k_{st}=k|\{t_{si}\},k_{-st}) \propto \begin{cases} B^*\beta_k p(\{X_{si}:t_{si}=t\}|\phi_k) & k\leq J \\ (1-B^*)\frac{m_{\cdot k}}{\sum_{k>J} m_{\cdot k}+\alpha} p(\{X_{si}:t_{si}=t\}|D_0,\mathbf{X}) & J<k\leq K \\ (1-B^*)\frac{\alpha}{\sum_{k>J} m_{\cdot k}+\alpha} p(\{X_{si}:t_{si}=t\}|D_0) & k=K+1 \end{cases} \tag{12}$$

We summarize the hybrid sampler shown in Algorithm 7

Algorithm 7 Hybrid HDP Sampler

while not converged **do**
  Select J
  Sample $B^* \sim \text{Beta}(\tilde{m}, m-\tilde{m}+\alpha)$
  Sample $(\beta_1,\ldots,\beta_J) \sim \text{Dir}(m_{\cdot 1},\ldots,m_{\cdot J})$
  For $k\leq J$, sample
$$\phi_k \sim p(\phi_k|D_0,\{X_{si}:k_{st_{si}}=k\})$$
  For each data point X_{si} sample t_{si} according to Equation 11
  Sample the dish k_{st} according to Equation 12
end while

If we ensure that data associated with each "restaurant" or group lies on the same table, we can extend this hybrid algorithm to a distributed setting, as described in Algorithm 8.

C Further IBP Empirical Results

For the "Cambridge" data sets described in the main paper, we generated images based on a superposition of the four features in the top row of Figure 6, and then flattened the image to create a 36-dimensional vector. The bottom row of Figure 6 shows some sample data points.

In addition to synthetic data, we also evaluated the distributed beta-Bernoulli process sampler to a real-world data set, the CBCL MIT face dataset (Weyrauch et al., 2004). This data set consists of 2,429 images of 19×19

Algorithm 8 Distributed HDP Sampler

procedure LOCAL($\{X_{si}, t_{si}, k_{st}\}$)
▷ Global variables $J, P^*, \{\phi_k, \beta_k\}_{k=1}^J, B^*$
  if *Processor* $= P*$ **then**
    Sample t_{si} according to (11)
    Sample k_{st} according to (12)
  else
    Sample tables according to

$$P(t_{si} = t | n_{st.}^{-si}) \propto \begin{cases} n_{st.}^{-si} f(x_{si}; \phi_{k_{st}}) & \text{if } t \text{ occupied} \\ \gamma \beta_{k_{st}} f(x_{si}; \phi_{k_{st}}) & \text{if } t \text{ is new} \end{cases} \tag{13}$$

    Sample dishes according to

$$P(k_{st} = k | \beta_k, \{t_{si}\}, X) \propto \beta_k p(\{X_{si} : t_{si} = t\} | \phi_k) \tag{14}$$

  end if
end procedure
procedure GLOBAL($\{X_{si}, t_{si}, k_{st}\}$)
  Gather cluster counts m_k and parameter sufficient statistics Ψ_k from all processors.
  Sample $B^* \sim \text{Beta}(\tilde{m}, \alpha + m - \tilde{m})$
  Let J be the number of instantiated clusters.
  Sample $(\beta_1, \ldots, \beta_J) \sim \text{Dir}(m_{.1}, \ldots, m_{.J})$
  For $k \leq J$, sample

$$\phi_k \sim p(\phi_k | D_0, \{X_{si} : k_{st_{si}} = k\})$$

  Sample $P^* \sim \text{Uniform}(1, \ldots, P)$
end procedure

dimensional faces. We distributed the data across 32 processors and ran the sampler in parallel for 500 iterations. Figure 8 shows the test set log likelihood of the sampler over time and Figure 7 shows the features learned by the hybrid IBP method and we can clearly see that our method can discover the underlying facial features in this data set.

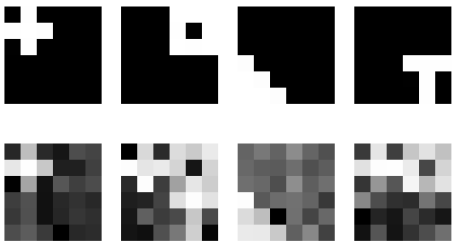

Figure 6: Top: The true features present in the synthetic data set. Bottom: Examples of observations in the synthetic data set.

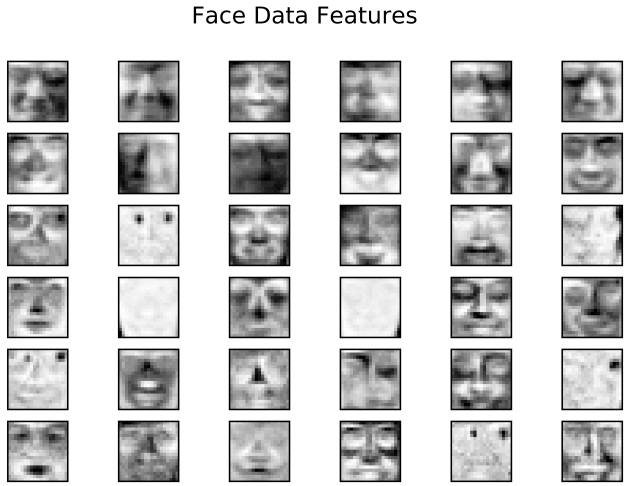

Figure 7: Learned features from the hybrid IBP for the CBCL face data set.

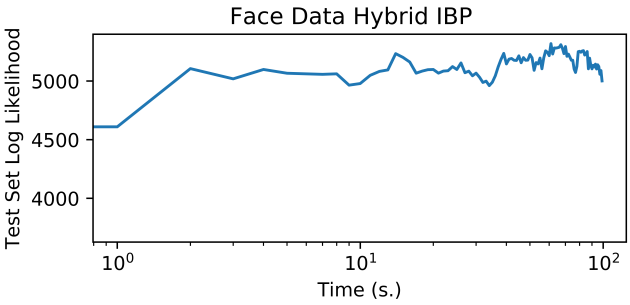

Figure 8: Test set log-likelihood trace plot for CBCL face data set.