Sparse Parallel Training of Hierarchical Dirichlet Process Topic Models

Alexander Terenin
Imperial College London

Måns Magnusson
Uppsala University
and Aalto University

Leif Jonsson
Ericsson AB
and Linköping University

Abstract

To scale non-parametric extensions of probabilistic topic models such as Latent Dirichlet allocation to larger data sets, practitioners rely increasingly on parallel and distributed systems. In this work, we study data-parallel training for the hierarchical Dirichlet process (HDP) topic model. Based upon a representation of certain conditional distributions within an HDP, we propose a doubly sparse data-parallel sampler for the HDP topic model. This sampler utilizes all available sources of sparsity found in natural language—an important way to make computation efficient. We benchmark our method on a well-known corpus (PubMed) with 8m documents and 768m tokens, using a single multi-core machine in under four days.

1 Introduction

Topic models are a widely-used class of methods that allow practitioners to identify latent semantic themes in large bodies of text in an unsupervised manner. They are particularly attractive in areas such as history (Yang et al., 2011; Wang et al., 2012), sociology (DiMaggio et al., 2013), and political science (Roberts et al., 2014), where a desire for careful control of structure and prior information incorporated into the model motivates one to adopt a Bayesian approach to learning. In these areas, large corpora such as newspaper archives are becoming increasingly available (Ehrmann et al., 2020), and models such as latent Dirichlet allocation (LDA) (Blei et al., 2003) and its nonparametric extensions (Teh et al., 2006; Teh, 2006; Hu and Boyd-Graber, 2012; Paisley et al., 2015) are widely used by practitioners. Moreover, these models are emerging as a component of data-efficient language models (Guo et al., 2020). Training topic models efficiently entails two requirements.

1. Expose sufficient parallelism that can be taken advantage of by the hardware.
2. Utilize sparsity found in natural language to control memory requirements and computational complexity.

In this work, we focus on the *hierarchical Dirichlet process* (HDP) topic model of Teh et al. (2006), which we review in Section 2. This model is a simple non-trivial extension of LDA to the nonparametric setting. This parallel implementation provides a blueprint for designing massively parallel training algorithms in more complicated settings, such as nonparametric dynamic topic models (Ahmed and Xing, 2010) and tree-based extensions (Hu and Boyd-Graber, 2012).

Parallel approaches to training HDPs have been previously introduced by a number of authors, including Newman et al. (2009), Wang et al. (2011), Williamson et al. (2013), Chang and Fisher (2014) and Ge et al. (2015). These techniques suit various settings: some are designed to explicitly incorporate sparsity present in natural language and other discrete spaces, while others are intended for HDP-based continuous mixture models. Gal and Ghahramani (2014) have pointed out that some methods can suffer from load-balancing issues, which limit their parallelism and scalability. The largest benchmark of parallel HDP training performed to our awareness is by Chang and Fisher (2014) on the 100m-token NYTIMES corpora. Throughout this work, we focus on Markov chain Monte Carlo (MCMC) methods—empirically, their scalability is comparable to variational methods (Magnusson et al., 2018; Hoffman and Ma, 2019), and, subject to convergence, they yield the correct posterior.

Our contributions are as follows. We propose an augmented representation of the HDP for which the topic indicators can be sampled in parallel over documents. We prove that, under this representation, the global topic distribution Ψ is conditionally conjugate given an auxiliary parameter $\boldsymbol{l}$. We develop

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 2925–2934,
November 16–20, 2020.

Symbol	Description	Symbol	Description
V	Vocabulary size	$\boldsymbol{\Psi} : 1 \times \infty$	Global distribution over topics
D	Total number of documents	$\boldsymbol{\Theta} : D \times \infty$	Document-topic probabilities
N	Total number of tokens	$\boldsymbol{\theta}_d : 1 \times \infty$	Topic probabilities for document d
$v(i)$	Word type for token i	$\mathbf{m} : D \times \infty$	Document-topic sufficient statistic
$d(i)$	Document for token i	$\boldsymbol{\Phi} : \infty \times V$	Topic-word probabilities
$w_{i,d}$	Token i in document d	$\boldsymbol{\phi}_k : 1 \times V$	Word probabilities for topic k
$b_{i,d}$	Global topic draw indicator for $w_{i,d}$	$\mathbf{n} : \infty \times V$	Topic-word sufficient statistic
$z_{i,d}$	Topic indicator for token i in d	$\boldsymbol{l} : 1 \times \infty$	Global topic latent sufficient statistic
K^*	Index for implicitly-represented topics	$\alpha, \boldsymbol{\beta}, \gamma$	Prior concentration for $\boldsymbol{\theta}_d$, $\boldsymbol{\phi}_k$, $\boldsymbol{\Psi}$

Table 1: Notation for the HDP topic model. Sufficient statistics are conditional on the algorithm's current iteration. Bold symbols refer to matrices, bold italics refer to vectors, possibly countably infinite.

fast sampling schemes for $\boldsymbol{\Psi}$ and $\boldsymbol{l}$, and propose a training algorithm with a per-iteration complexity that depends on the minima of two sparsity terms—it takes advantage of both document-topic and topic-word sparsity simultaneously.

2 Partially collapsed Gibbs sampling for hierarchical Dirichlet processes

The hierarchical Dirichlet process topic model (Teh et al., 2006) begins with a global distribution $\boldsymbol{\Psi}$ over topics. Documents are assumed exchangeable—for each document d, the associated topic distribution $\boldsymbol{\theta}_d$ follows a Dirichlet process centered at $\boldsymbol{\Psi}$. Each topic is associated with a distribution of tokens $\boldsymbol{\phi}_k$. Within each document, tokens are assumed exchangeable (bag of words) and assigned to topic indicators $z_{i,d}$. For given data, we observe the tokens $w_{i,d}$.

We thus arrive at the GEM representation of a HDP, given by equation (19) of Teh et al. (2006) as

$$\boldsymbol{\Psi} \sim \mathrm{GEM}(\gamma) \tag{1}$$
$$\boldsymbol{\theta}_d \mid \boldsymbol{\Psi} \sim \mathrm{DP}(\alpha, \boldsymbol{\Psi}) \tag{2}$$
$$\boldsymbol{\phi}_k \sim \mathrm{Dir}(\boldsymbol{\beta}) \tag{3}$$
$$z_{i,d} \mid \boldsymbol{\theta}_d \sim \mathrm{Discrete}(\boldsymbol{\theta}_d) \tag{4}$$
$$w_{i,d} \mid z_{i,d}, \boldsymbol{\Phi} \sim \mathrm{Discrete}(\boldsymbol{\phi}_{z_{i,d}}) \tag{5}$$

where $\alpha, \boldsymbol{\beta}, \gamma$ are prior hyperparameters.

2.1 Intuition and augmented representation

At a high level, our strategy for constructing a scalable sampler is as follows. Conditional on $\boldsymbol{\Psi}$, the likelihood in equations (1)–(5) is the same as that of LDA. Using this observation, the Gibbs step for $\boldsymbol{z}$, which is the largest component of the model, can be handled efficiently by leveraging insights on sparse parallel sampling from the well-studied LDA literature (Yao et al., 2009; Li et al., 2014; Magnusson et al., 2018; Terenin et al., 2019). For this strategy to succeed, we need to ensure that all Gibbs steps involved in the HDP under this representation are analytically tractable and can be computed efficiently. For this, the representation needs to be modified.

To begin, we integrate each $\boldsymbol{\theta}_d$ out of the model, which by conjugacy (Blackwell and MacQueen, 1973) yields a Pólya sequence for each $\boldsymbol{z}_d$. By definition, given in Appendix A, this sequence is a mixture distribution with respect to a set of Bernoulli random variables $\boldsymbol{b}_d$, each representing whether $z_{i,d}$ was drawn from $\boldsymbol{\Psi}$ or from a repeated draw in the Pólya urn. Thus, the HDP can be written

$$\boldsymbol{\Psi} \sim \mathrm{GEM}(\gamma) \tag{6}$$
$$b_{i,d} \sim \mathrm{Ber}\left(\tfrac{\alpha}{i-1+\alpha}\right) \tag{7}$$
$$\boldsymbol{\phi}_k \sim \mathrm{Dir}(\boldsymbol{\beta}) \tag{8}$$
$$\boldsymbol{z}_d \mid \boldsymbol{b}_d, \boldsymbol{\Psi} \sim \mathrm{PS}(\boldsymbol{\Psi}, \boldsymbol{b}_d) \tag{9}$$
$$w_{i,d} \mid z_{i,d} \sim \mathrm{Discrete}(\boldsymbol{\phi}_{z_{i,d}}) \tag{10}$$

where $\mathrm{PS}(\boldsymbol{\Psi}, \boldsymbol{b}_d)$ is a Pólya sequence, defined in Appendix A. This representation defines a posterior distribution over $\boldsymbol{z}, \boldsymbol{\Phi}, \boldsymbol{\Psi}, \boldsymbol{b}$ for the HDP. To derive a Gibbs sampler, we calculate its full conditionals.

2.2 Full conditionals for $\boldsymbol{z}$, $\boldsymbol{\Phi}$, and $\boldsymbol{b}$

The full conditionals $\boldsymbol{z} \mid \boldsymbol{\Phi}, \boldsymbol{\Psi}$ and $\boldsymbol{\Phi} \mid \boldsymbol{z}, \boldsymbol{\Psi}$, with $\boldsymbol{b}$ marginalized out, are essentially those in partially collapsed LDA (Magnusson et al., 2018; Terenin et al., 2019). They are

$$\mathbb{P}(z_{i,d} = k \mid \boldsymbol{z}_{-i,d}, \boldsymbol{\Phi}, \boldsymbol{\Psi}) \tag{11}$$
$$\propto \phi_{k,v(i)} \left[\alpha \Psi_k + m_{d,k}^{-i}\right] \tag{12}$$

where $v(i)$ is the word type for word token i, and

$$\boldsymbol{\phi}_k \mid \boldsymbol{z} \sim \mathrm{Dir}(\boldsymbol{\beta} + \boldsymbol{n}_k) \tag{13}$$

where $m_{d,k}^{-i}$ denotes the document-topic sufficient statistic with index i removed, and $\boldsymbol{n}_k$ is the topic-word sufficient statistic. Note the number of possible topics and full conditionals $\boldsymbol{\phi}_k \mid \boldsymbol{z}$ here is countably infinite. The full conditional for each $b_{i,d}$ is

$$\mathbb{P}(b_{i,d} = 1 \mid \boldsymbol{z}_d, \boldsymbol{\Psi}, \boldsymbol{b}_{-i,d}) \tag{14}$$

$$= \frac{\alpha \Psi_{z_{i,d}}}{\alpha \Psi_{z_{i,d}} + \sum_{j=1}^{i} \mathbb{1}_{z_{j,d}}(z_{i,d})}. \tag{15}$$

The derivation, based on a direct application of Bayes' Rule with respect to the probability mass function of the Pólya sequence, is in Appendix A.

2.3 The full conditional for $\boldsymbol{\Psi}$

To derive the full conditional for $\boldsymbol{\Psi}$, we examine the prior and likelihood components of the model. It is shown in Appendix A that the likelihood term $\boldsymbol{z}_d \mid \boldsymbol{b}_d, \boldsymbol{\Psi}$ may be written

$$p(\boldsymbol{z}_d \mid \boldsymbol{b}_d, \boldsymbol{\Psi}) \tag{16}$$

$$= \underbrace{\prod_{\substack{i=1 \\ b_{i,d} \neq 1}}^{N_d} \sum_{j=1}^{i-1} \frac{1}{i-1} \mathbb{1}_{z_{j,d}}(z_{i,d})}_{\text{doesn't enter posterior}} \prod_{\substack{i=1 \\ b_{i,d}=1}}^{N_d} \prod_{k=1}^{\infty} \Psi_k^{\mathbb{1}_k(z_{i,d})}.$$

The first term is a multiplicative constant independent of $\boldsymbol{\Psi}$ and vanishes via normalization. Thus, the full conditional $\boldsymbol{\Psi} \mid \boldsymbol{z}, \boldsymbol{b}$ depends on $\boldsymbol{z}$ and $\boldsymbol{b}$ only through the sufficient statistic $\boldsymbol{l}$ defined by

$$l_k = \sum_{d=1}^{D} \sum_{\substack{i=1 \\ b_{i,d}=1}}^{N_d} \mathbb{1}_{z_{i,d}=k} \tag{17}$$

and so we may suppose without loss of generality that the likelihood term is categorical. Under these conditions, we prove the full conditional for $\boldsymbol{\Psi}$ admits a stick-breaking representation.

Proposition 1. *Without loss of generality, suppose*

$$\boldsymbol{\Psi} \sim \mathrm{GEM}(\gamma) \quad \boldsymbol{x} \mid \boldsymbol{\Psi} \sim \mathrm{Discrete}(\boldsymbol{\Psi}). \tag{18}$$

Then $\boldsymbol{\Psi} \mid \boldsymbol{x}$ is given by

$$\Psi_k = \varsigma_k \prod_{i=1}^{k-1} (1 - \varsigma_i) \quad \varsigma_k \sim \mathrm{Beta}(a_k^{(\boldsymbol{\Psi})}, b_k^{(\boldsymbol{\Psi})}) \tag{19}$$

$$a_k^{(\boldsymbol{\Psi})} = 1 + l_k \qquad b_k^{(\boldsymbol{\Psi})} = \gamma + \sum_{i=k+1}^{\infty} l_i \tag{20}$$

where $\boldsymbol{l}$ are the empirical counts of $\boldsymbol{x}$.

Proof. Appendix B. □

This expression is similar to the stick-breaking representation of a Dirichlet process $\mathrm{DP}(\cdot, F)$—however, it has different weights and does not include random atoms drawn from F as part of its definition—see Appendix B for more details. Putting these ideas together, we define an infinite-dimensional parallel Gibbs sampler.

Algorithm 1. *Repeat until convergence.*

- *Sample $\boldsymbol{\phi}_k \sim \mathrm{Dir}(\boldsymbol{n}_k + \boldsymbol{\beta})$ in parallel over topics for $k = 1, .., \infty$.*
- *Sample $z_{i,d} \propto \phi_{k,v(i)}\, \alpha\, \Psi_k + \phi_{k,v(i)}\, m_{d,k}^{-i}$ in parallel over documents for $d = 1, .., D$.*
- *Sample $b_{i,d}$ according to equation (14) in parallel over documents for $d = 1, .., D$.*
- *Sample $\boldsymbol{\Psi}$ according to equations (19)–(20).*

Algorithm 1 is completely parallel, but cannot be implemented as stated due to the infinite number of full conditionals for $\boldsymbol{\Phi}$, as well as the infinite product used in sampling $\boldsymbol{\Psi}$. We now bypass these issues by introducing an approximate finite-dimensional sampling scheme.

2.4 Finite-dimensional sampling of $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$

By way of assuming $\boldsymbol{\Psi} \sim \mathrm{GEM}(\gamma)$, an HDP assumes an infinite number of topics are present a priori, with the number of tokens per topic decreasing rapidly with the topic's index in a manner controlled by γ. Thus, under the model, a topic with a sufficiently large index should contain no tokens with high probability.

We thus propose to approximate $\boldsymbol{\Psi}$ by projecting its tail onto a single flag topic K^*, which stands for all topics not explicitly represented as part of the computation. This can be done by by deterministically setting $\varsigma_{K^*} = 1$ in equation (19). The resulting finite-dimensional $\boldsymbol{\Psi}$ will be the correct posterior full conditional for the finite-dimensional generalized Dirichlet prior considered previously in Section 2.3. Hence, this finite-dimensional truncation forms a Bayesian model in its own right, which suggests it should perform reasonably well. From an asymptotic perspective, Ishwaran and James (2001) have shown that the approximation is almost surely convergent and, therefore, well-posed.

Once this is done, $\boldsymbol{\Psi}$ becomes a finite vector of length K^*, and only K^* rows of $\boldsymbol{\Phi}$ need to be explicitly instantiated as part of the computation. This instantiation allows the algorithm to be defined on

a fixed finite state space, simplifying bookkeeping and implementation.

From a computational efficiency perspective, the resulting value K^* takes the place of K in partially collapsed LDA. However, it *cannot* be interpreted as the number of topics in the sense of LDA. Indeed, LDA implicitly assumes that $\boldsymbol{\Psi} = \mathrm{Unif}(1, .., K)$ deterministically—i.e., that every topic is assumed a priori to contain the same number of tokens. In contrast, the HDP model learns this distribution from the data by letting $\boldsymbol{\Psi} \sim \mathrm{GEM}(\gamma)$.

If we allow the state space to be resized when topic K^* is sampled, then following Papaspiliopoulos and Roberts (2008), it is possible to develop truncation schemes which introduce no error. Since this results in more complicated bookkeeping which reduces performance, we instead fix K^* and defer such considerations to future work. We recommend setting K^* to be sufficiently large that it does not significantly affect the model's behavior, which can be checked by tracking the number of tokens assigned to the topic K^*.

2.5 Sparse sampling of $\boldsymbol{\Phi}$ and $\boldsymbol{z}$

To be efficient, a topic model needs to utilize the sparsity found in natural language as much as possible. In our case, the two main sources of sparsity are as follows.

1. *Document-topic sparsity*: most documents will only contain a handful of topics.

2. *Topic-word sparsity*: most word types will not be present in most topics.

We thus expect the document-topic sufficient statistic $\mathbf{m}$ and topic-word sufficient statistic $\mathbf{n}$ to contain many zeros. We seek to use this to reduce sampling complexity. Our starting point is the Poisson Pólya Urn sampler of Terenin et al. (2019), which presents a Gibbs sampler for LDA with computational complexity that depends on the minima of two sparsity coefficients representing document-topic and topic-word sparsity—such algorithms are termed *doubly sparse*. The key idea is to approximate the Dirichlet full conditional for $\boldsymbol{\phi}_k$ with a Poisson Pólya Urn (PPU) distribution defined by

$$\phi_{k,v} = \frac{\varphi_{k,v}}{\sum_{v=1}^{V} \varphi_{k,v}} \quad \varphi_{k,v} \sim \mathrm{Pois}(\beta_{k,v} + n_{k,v}) \tag{21}$$

for $v = 1, .., V$. This distribution is discrete, so $\boldsymbol{\Phi}$ becomes a sparse matrix. The approximation is accurate even for small values of $n_{k,v}$, and Terenin et al. (2019) proves that the approximation error will vanish for large data sets in the sense of convergence in distribution.

If $\boldsymbol{\beta}$ is uniform, we can further use sparsity to accelerate sampling $\varphi_{k,v}$. Since a sum of Poisson random variables is Poisson, we can split $\varphi_{k,v} = \varphi^{(\beta)}_{k,v} + \varphi^{(\mathbf{n})}_{k,v}$. We then sample $\varphi^{(\beta)}_{k,v}$ sparsely by introducing a Poisson process and sampling its points uniformly, and sample $\varphi^{(\mathbf{n})}_{k,v}$ sparsely by iterating over nonzero entries of $\mathbf{n}$.

For $\boldsymbol{z}$, the full conditional

$$\begin{aligned} &\mathbb{P}(z_{i,d} = k \mid \boldsymbol{z}_{-i,d}, \boldsymbol{\Phi}, \boldsymbol{\Psi}) && (22)\\ &\propto \phi_{k,v(i)} \left[\alpha \Psi_k + m^{-i}_{d,k} \right] && (23)\\ &\propto \underbrace{\phi_{k,v(i)} \alpha \Psi_k}_{(a)} + \underbrace{\phi_{k,v(i)} m^{-i}_{d,k}}_{(b)} && (24) \end{aligned}$$

is similar to to the one in partially collapsed LDA (Magnusson et al., 2018)—the difference is the presence of Ψ_k. As Ψ_k only enters the expression through component (a) and is identical for all $z_{i,d}$, it can be absorbed at each iteration directly into an alias table (Walker, 1977; Li et al., 2014). Component (b) can be computed efficiently by utilizing sparsity of $\boldsymbol{\Phi}$ and $\mathbf{m}$ and iterating over whichever has fewer non-zero entries.

2.6 Direct sampling of $\boldsymbol{l}$

Rather than sampling $\boldsymbol{b}$, whose size will grow linearly with the number of documents, we introduce a scheme for sampling the sufficient statistic $\boldsymbol{l}$ directly. Observe that

$$l_k = \sum_{d=1}^{D} \sum_{\substack{i=1 \\ b_{i,d}=1}}^{N_d} \mathbb{1}_{z_{n,d}=k} = \sum_{d=1}^{D} \sum_{\substack{i=1 \\ z_{i,d}=k}}^{N_d} \mathbb{1}_{b_{i,d}=1} \tag{25}$$

where the domain of summation and the value of the indicators have been switched. By definition of $b_{i,d}$, we have

$$\sum_{\substack{i=1 \\ z_{i,d}=k}}^{N_d} \mathbb{1}_{b_{i,d}=1} = \sum_{j=1}^{m_{d,k}} b_{j,d,k} \tag{26}$$

where

$$b_{j,d,k} \sim \mathrm{Ber}\left(\frac{\Psi_k \alpha}{\Psi_k \alpha + j - 1} \right). \tag{27}$$

Summing this expression over documents, we obtain the expression

$$l_k = \sum_{j=1}^{\max_d m_{d,k}} c_{j,k} \quad c_{j,k} \sim \mathrm{Bin}\left(D_{k,j}, \frac{\Psi_k \alpha}{\Psi_k \alpha + j - 1} \right) \tag{28}$$

where $D_{k,j}$ is the total number of documents with $m_{d,k} \geq j$. Since $m_{d,k} = 0$ for all topics k without any tokens assigned, we only need to sample $\boldsymbol{l}$ for topics that have tokens assigned to them. This idea can also be straightforwardly applied to other HDP samplers (Chang and Fisher, 2014; Ge et al., 2015), by allowing one to derive alternative full conditionals in lieu of the *Stirling distribution* (Antoniak, 1974). The complexity of sampling $\boldsymbol{l}$ directly is constant with respect to the number of documents, and depends instead on the maximum number of tokens per document.

To handle the bookkeeping necessary for computing $D_{k,j}$, we introduce a sparse matrix $\mathbf{d}$ of size $K \times \max_d N_d$ whose entries $d_{k,p}$ are the number of documents for topic k that have a total of p topic indicators assigned to them. We increment $\mathbf{d}$ once $\boldsymbol{z}_d$ been sampled by iterating over non-zero elements in $\boldsymbol{m}_d$. We then compute $D_{k,j}$ as the reverse cumulative sum of the rows of $\mathbf{d}$.

2.7 Poisson Pólya urn partially collapsed Gibbs sampling

Putting all of these ideas together, we obtain the following algorithm.

Algorithm 2. *Repeat until convergence.*

- *Sample* $\boldsymbol{\phi}_k \sim \mathrm{PPU}(\boldsymbol{n}_k + \boldsymbol{\beta})$ *in parallel over topics for* $k = 1, .., K^*$.
- *Sample* $z_{i,d} \propto \phi_{k,v(i)}\, \alpha\, \Psi_k + \phi_{k,v(i)}\, m_{d,k}^{-i}$ *in parallel over documents for* $d = 1, .., D$.
- *Sample* l_k *according to equation* (28) *in parallel over topics for* $k = 1, .., K^*$.
- *Sample* $\boldsymbol{\Psi}$ *according to equations* (19)–(20)*, except with* $\varsigma_{K^*} = 1$.

Algorithm 2 is sparse, massively parallel, defined on a fixed finite state space, and contains no infinite computations in any of its steps. The Gibbs step for $\boldsymbol{\Phi}$ converges in distribution (Terenin et al., 2019) to the true Gibbs steps as $N \to \infty$, and the Gibbs step for $\boldsymbol{\Psi}$ converges almost surely (Ishwaran and James, 2001) to the true Gibbs step as $K^* \to \infty$.

2.8 Computational complexity

We now examine the per-iteration computational complexity of Algorithm 2. To proceed, we fix K^* and maximum document size $\max_d N_d$, and relate the vocabulary size V with the number N of total words as follows.

Assumption (Heaps' Law)**.** *The number of unique words in a corpus follows Heaps' law (Heaps, 1978)* $V = \xi N^{\zeta}$ *with constants* $\xi > 0$ *and* $\zeta < 1$.

The per-iteration complexity of Algorithm 2 is equal to the sum of the per-iteration complexity of sampling its components. The sampling complexities of $\boldsymbol{\Psi}$ and $\boldsymbol{l}$ are constant with respect to the number of tokens, and the sampling complexity of $\boldsymbol{\Phi}$ has been shown by Magnusson et al. (2018) to be negligible under the given assumptions. Thus, it suffices to consider $\boldsymbol{z}$.

At a given iteration, let $K_{d(i)}^{(\mathbf{m})}$ be the number of existing topics in document d associated with word token i, and let $K_{v(i)}^{(\boldsymbol{\Phi})}$ be the number of nonzero topics in the row of $\boldsymbol{\Phi}$ corresponding to word token i. It follows immediately from the argument given by Terenin et al. (2019) that the per-iteration complexity of sampling each topic indicator z_i is

$$\mathcal{O}\left[\min\left(K_{d(i)}^{(\mathbf{m})}, K_{v(i)}^{(\boldsymbol{\Phi})}\right)\right]. \tag{29}$$

Algorithm 2 is thus a doubly sparse algorithm.

3 Performance results

To study performance of the *partially collapsed* sampler—Algorithm 2—we implemented it in Java using the open-source MALLET[1] (McCallum, 2002) topic modeling framework. We ran it on the AP, CGCBIB, NEURIPS, and PUBMED corpora,[1] which are summarized in Table 2. Prior hyperparameters controlling the degree of sparsity were set to $\alpha = 0.1, \beta = 0.01, \gamma = 1$. We set $K^* = 1000$ and observed no tokens ever allocated to the topic K^*. Data were preprocessed with default Mallet (McCallum, 2002) stop-word removal, minimum document size of 10, and a rare word limit of 10. Following Teh et al. (2006), the algorithm was initialized with one topic. All experiments were repeated five times to assess variability. Total runtime for each experiment is given in Table 2.

To assess Algorithm 2 in a small-scale setting, we compare it to the widely-studied sparse fully collapsed *direct assignment* sampler of Teh et al. (2006), which is not parallel. We ran 100 000

[1] See HTTP://MALLET.CS.UMASS.EDU and HTTPS://GITHUB.COM/LEJON/PARTIALLYCOLLAPSEDLDA. AP and CGCBIB can be found therein. NeurIPS and PubMed can be found at HTTPS://ARCHIVE.ICS.UCI.EDU/ML/DATASETS/BAG+OF+WORDS. Full output of experiments can be found at HTTPS://GITHUB.COM/ATERENIN/PARALLEL-HDP-EXPERIMENTS/.

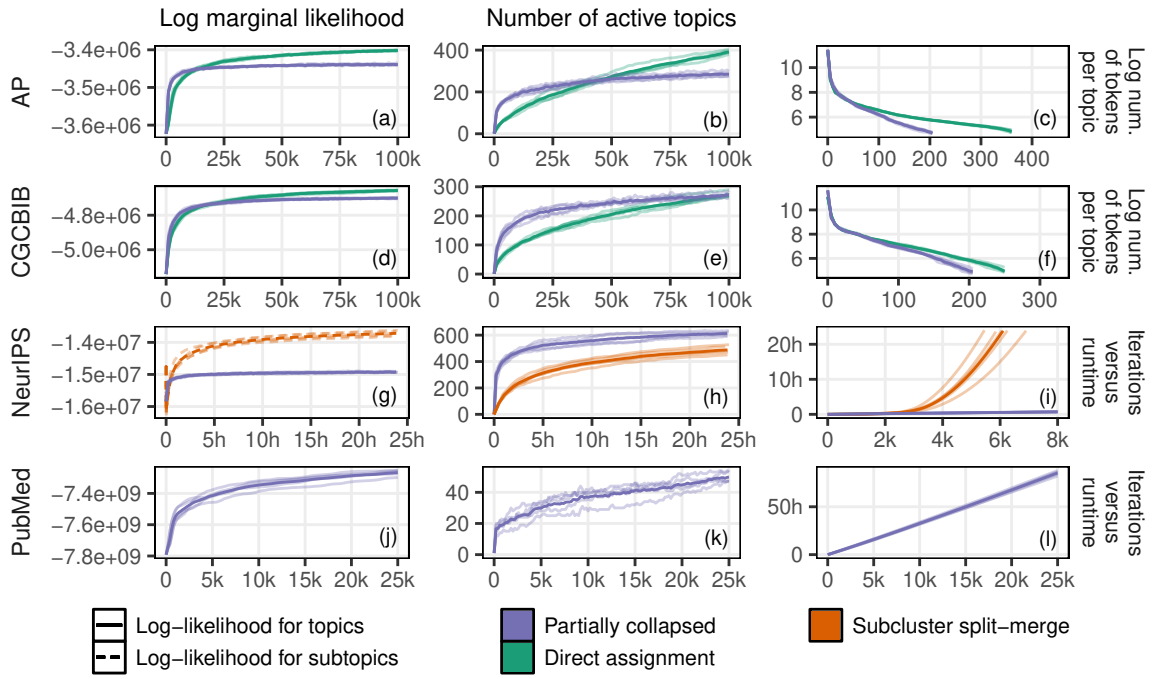

Figure 1: Trace plots for log-likelihood, number of active topics, and additional metrics for CGCBIB, NeurIPS, and PubMed. On the x axis, per-iteration scale is used for AP, CGCBIB and PubMed, and real-time scale is used for NeurIPS. Algorithms used are partially collapsed HDP for all corpora, direct assignment HDP for AP and CGCBIB, and subcluster split-merge HDP for NeurIPS. Individual traces are partially transparent, and their mean is opaque.

iterations of both methods on AP and CGCBIB. We selected these corpora because they were among the larger corpora on which it was feasible to run our direct assignment reference implementation within one week.

Trace plots for the log marginal likelihood for $\boldsymbol{z}$ given $\boldsymbol{\Psi}$ and the number of active topics, i.e., those topics assigned at least one token, can be seen in Figure 1(a,d) and Figure 1(b,e), respectively. The direct assignment algorithm converges slower, but achieves a slightly better local optimum in terms of marginal log-likelihood, compared to our method. This fact indicates that the direct assignment method may stabilize around a different local optimum, and may represent a potential limitation of the partially collapsed sampler in settings where non-parallel methods are practical.

To better understand the distributional differences between the algorithms, we examined the number of tokens per topic, which can be seen in Figure 1(c,f). The partially collapsed sampler is seen to assign more tokens to smaller topics, indicating that it stabilizes around a local optimum with slightly broader semantic themes.

To visualize the effect this has on the topics, we examined the most common words for each topic. Since the algorithms generate too many topics to make full examination practical, we instead compute a quantile summary with five topics per quantile. The quantile is computed by ranking all topics by the number of tokens, choosing the five closest topics to the 100%, 75%, 50%, 25%, and 5% quantiles in the ranking, and computing their top words. This approach gives a representative view of the algorithm's output for large, medium, and small topics. Results may be seen in Appendix D and Appendix C—we find the direct assignment and partially collapsed samplers to be mostly com-

Corpus	V	D	N	Iterations	Threads	Runtime
AP	7 074	2 206	393 567	100 000	8	3.8 hours
CGCBIB	6 079	5 940	570 370	100 000	12	2.7 hours
NeurIPS	12 419	1 499	1 894 051	255 500	8	24 hours
PubMed	89 987	8 199 999	768 434 972	25 000	20	82.4 hours

Table 2: Corpora used in experiments, together with compute configuration.

parable, with substantial overlap in top words for common topics.

Next, we assess Algorithm 2 in a more demanding setting and compare against previous parallel state-of-the-art. There are various scalable samplers available for the HDP. For a fair comparison, we restrict ourselves to those samplers designed for topic models and explicitly incorporate sparsity of natural language in their construction. Among these, we selected the parallel *subcluster split-merge* algorithm of Chang and Fisher (2014) as our baseline because it was used in the largest-scale benchmark of the HDP topic model performed to date to our awareness, and shows comparable performance to other methods (Ge et al., 2015). The subcluster split-merge algorithm is designed to converge with fewer iterations, but is more costly to run per iteration. Thus, we used a fixed computational budget of 24 hours of wall-clock time for both algorithms. Computation was performed on a system with a 4-core 8-thread CPU and 8GB RAM.

Results can be seen in Figure 1(g)—note that the subcluster split-merge algorithm is parametrized using *sub-topic indicators* and *sub-topic probabilities*, so its numerical log-likelihood values are not directly comparable to ours and should be *interpreted purely to assess convergence*. Algorithm 2 stabilizes much faster with respect to both the number of active topics in Figure 1(g), and marginal log-likelihood in Figure 1(h). The subcluster split-merge algorithm adds new topics one-at-a-time, whereas our algorithm can create multiple new topics per iteration—we hypothesize this difference leads to faster convergence for Algorithm 2.

In Figure 1(i), we observe that the amount of computing time per iteration increases substantially for the subcluster split-merge method as it adds more topics. For Algorithm 2, this stays approximately constant for its entire runtime.

To evaluate the topics produced by the algorithms, we again examined the most common words for each topic via a quantile summary, given in Appendix E. We find the subcluster split-merge algorithm appears to generate topics with slightly more semantic overlap compared to Algorithm 2, but otherwise produces comparable output.

Finally, to assess scalability, we ran 25 000 iterations of Algorithm 2 on PubMed, which contains 768m tokens. To our knowledge, this dataset is an order of magnitude larger than any datasets used in previous MCMC-based approaches for the HDP. Computation was performed on a compute node with 2x10-core CPUs with 20 threads and 64GB of RAM. The marginal likelihood and number of active topics are given in Figure 1(j) and Figure 1(k).

To evaluate the topics discovered by the algorithm, we examined their most common words—these may be seen in full in Appendix F. We observe that the semantic themes present in the topics vary according to how many tokens they have: topics with more tokens appear to be broader, whereas topics with fewer tokens appear to be more specific. This behavior illustrates a key difference between the HDP and methods like LDA, which do not contain a learned global topic distribution $\boldsymbol{\Psi}$ in their formulation. We suspect the effect is particularly pronounced on PubMed compared to CGCBIB and NeurIPS due to its large number of tokens.

4 Discussion

In this work, we introduce the parallel partially collapsed Gibbs sampler—Algorithm 1—for the HDP topic model, which converges to the correct target distribution. We propose a doubly sparse approximate sampler—Algorithm 2—which allows the HDP to be implemented with per-token sampling complexity of $\mathcal{O}\left[\min\left(K_{d(i)}^{(\mathbf{m})}, K_{v(i)}^{(\boldsymbol{\Phi})}\right)\right]$ which is the same as that of Pólya Urn LDA (Terenin et al., 2019). Compared to other approaches for the HDP, it offers the following improvements.

1. The algorithm is fully parallel in all steps.
2. The topic indicators $\boldsymbol{z}$ utilize all available sources of sparsity to accelerate sampling.
3. All steps not involving $\boldsymbol{z}$ have constant complexity with respect to data size.
4. The proposed sparse approximate algorithm becomes exact as $N \to \infty$ and $K^* \to \infty$.

These improvements allow us to train the HDP on larger corpora. The data-parallel nature of our approach means that the amount of available parallelism increases with data size. This parallelism avoids load-balancing-related scalability limitations pointed out by Gal and Ghahramani (2014).

Nonparametric topic models are less straightforward to evaluate empirically than ordinary topic models. In particular, we found topic coherence scores (Mimno et al., 2011) to be strongly affected by the number of active topics K, which causes preference for models with fewer topics and more

k	Topic 1	Topic 5	Topic 9	Topic 13	Topic 17
$n_{k,\bullet}$	42 395 289	23 907 517	22 167 377	20 925 933	18 924 590
	care	cancer	protein	protein	cell
	health	tumor	binding	cell	neuron
	patient	patient	membrane	kinase	electron
	medical	cell	acid	expression	brain
	research	carcinoma	activity	receptor	rat
	system	breast	cell	activation	nerve
	clinical	tumour	gel	pathway	fiber
	cost	survival	human	phosphorylati	nucleus

k	Topic 21	Topic 25	Topic 29	Topic 33	Topic 37
$n_{k,\bullet}$	18 033 777	16 308 024	15 128 822	13 562 338	10 819 160
	cell	rat	gene	infection	plant
	growth	day	mutation	strain	strain
	expression	mice	genetic	antibiotic	acid
	factor	liver	chromosome	bacterial	growth
	beta	animal	analysis	isolates	extract
	human	effect	genes	bacteria	activity
	mrna	control	polymorphism	resistance	cell
	endothelial	mg	dna	coli	production

Figure 2: Top 8 words for topics obtained by Algorithm 2 on PubMed, together with topic index k and total number of words $n_{k,\bullet}$ present in the topic. We observe that the topics range from broad to specific: this is a consequence of the hierarchical Dirichlet process prior via the inclusion of the global topic proportions $\boldsymbol{\Psi}$. Topics obtained by Algorithm 2 on all corpora may be seen in Appendix C, Appendix D, Appendix E, and Appendix F.

semantic overlap per topic. We view the development of summary statistics that are K-agnostic and those measuring other aspects of topic quality such as overlap, to be an important direction for future work. We are particularly interested in techniques that can be used to compare algorithms for sampling from the same model defined over fully disjoint state spaces, such as Algorithm 2 and the subcluster split-merge algorithm in Section 3.

Partially collapsed HDP can stabilize around a different local mode than fully collapsed HDP as proposed by Teh et al. (2006). There have been attempts to improve mixing in that sampler (Chang and Fisher, 2014), including the use of Metropolis-Hastings steps for jumping between modes (Jain and Neal, 2004). These techniques are largely complementary to ours and can be explored in combination with the ideas presented here.

The HDP posterior is a heavily multimodal target for which full posterior exploration is known to be difficult (Chang and Fisher, 2014; Gal and Ghahramani, 2014; Buntine and Mishra, 2014), and sampling schemes are generally used more in the spirit of optimization than traditional MCMC. These issues are mirrored in other approaches, such as variational inference. There, restrictive mean-field factorization assumptions are often required, which reduces the quality of discovered topics. We view MAP-based analogs of ideas presented here as a promising direction, since these may allow additional flexibility that may enable faster training.

Many of the ideas in this work, such as the binomial trick, are generic and apply to any topic model structurally similar to the HDP's GEM representation (Teh et al., 2006) given in Section 2. For example, one could consider an informative prior for $\boldsymbol{\Psi}$ in lieu of $\mathrm{GEM}(\gamma)$, potentially improving convergence and topic quality, or developing parallel schemes for other nonparametric topic models such as Pitman-Yor models (Teh, 2006), tree-based models (Hu and Boyd-Graber, 2012; Paisley et al., 2015), embedded topic models (Dieng et al., 2020), as well as nonparametric topic models used within data-efficient language models (Guo et al., 2020) in future work.

Conclusion

We introduce the doubly sparse partially collapsed Gibbs sampler for the hierarchical Dirichlet process topic model. By formulating this algorithm using a representation of the HDP which connects it with the well-studied Latent Dirichlet Allocation model, we obtain a parallel algorithm whose per-token sampling complexity is the minima of two sparsity

terms. The ideas used apply to a large array of topic models which possess the same full conditional for the topic indicators $\boldsymbol{z}$. Our algorithm for the HDP scales to a 768m-token corpus (PubMed) on a single multicore machine in under four days.

The proposed techniques leverage parallelism and sparsity to scale nonparametric topic models to larger datasets than previously considered feasible for MCMC or other methods possessing similar convergence properties. We hope these contributions enable wider use of Bayesian nonparametrics for large collections of text.

Acknowledgments

The research was funded by the Academy of Finland (grants 298742, 313122), as well as the Swedish Research Council (grants 201805170, 201806063). Computations were performed using compute resources within the Aalto University School of Science and Department of Computing at Imperial College London. We also acknowledge the support of Ericsson AB.

References

Amr Ahmed and Eric P. Xing. 2010. Timeline: a dynamic hierarchical Dirichlet process model for recovering birth/death and evolution of topics in text stream. In *Uncertainty in Artificial Intelligence*, pages 20–29.

Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. 2005. *Gradient Flows in Metric Spaces and in the Space of Probability Measures*. Birkhäuser.

Charles E. Antoniak. 1974. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 2(6):1152–1174.

David Blackwell and James B. MacQueen. 1973. Ferguson distributions via Pólya urn schemes. *The Annals of Statistics*, 1(2):353–355.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(1):993–1022.

Vladimir I. Bogachev. 2007. *Measure Theory: Volume II*. Springer.

Wray L. Buntine and Swapnil Mishra. 2014. Experiments with non-parametric topic models. In *Knowledge Discovery and Data Mining*, pages 881–890.

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

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

Adji B. Dieng, Francisco J. R. Ruiz, and David M. Blei. 2020. Topic modeling in embedding spaces. *Transactions of the Association for Computational Linguistics*, 8:439–453.

Paul DiMaggio, Manish Nag, and David M. Blei. 2013. Exploiting affinities between topic modeling and the sociological perspective on culture: application to newspaper coverage of US government arts funding. *Poetics*, 41(6):570–606.

Maud Ehrmann, Matteo Romanello, Simon Clematide, Phillip B. Ströbel, and Raphaël Barman. 2020. Language resources for historical newspapers: the Impresso collection. In *Language Resources and Evaluation Conference*, pages 958–968.

Yarin Gal and Zoubin Ghahramani. 2014. Pitfalls in the use of parallel inference for the Dirichlet process. In *International Conference on Machine Learning*, pages 208–216.

Hong Ge, Yutian Chen, Moquan Wan, and Zoubin Ghahramani. 2015. Distributed inference for Dirichlet process mixture models. In *International Conference on Machine Learning*, pages 2276–2284.

Dandan Guo, Bo Chen, Ruiying Lu, and Mingyuan Zhou. 2020. Recurrent hierarchical topic-guided neural language models. In *International Conference on Machine Learning*, pages 10994–11005.

Harold S. Heaps. 1978. *Information Retrieval: Computational and Theoretical Aspects*. Academic Press.

Matthew D. Hoffman and Yian Ma. 2019. Langevin dynamics as nonparametric variational inference. In *Advances in Approximate Bayesian Inference*.

Yuening Hu and Jordan Boyd-Graber. 2012. Efficient tree-based topic modeling. In *Proceedings of the Association for Computational Linguistics*, pages 275–279.

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

Sonia Jain and Radford M. Neal. 2004. A split-merge Markov chain Monte Carlo procedure for the Dirichlet process mixture model. *Journal of Computational and Graphical Statistics*, 13(1):158–182.

Aaron Q. Li, Amr Ahmed, Sujith Ravi, and Alexander J. Smola. 2014. Reducing the sampling complexity of topic models. In *Knowledge Discovery and Data Mining*, pages 891–900.

Måns Magnusson, Leif Jonsson, Mattias Villani, and David Broman. 2018. Sparse partially collapsed MCMC for parallel inference in topic models. *Journal of Computational and Graphical Statistics*, 27(2):449–463.

Andrew K. McCallum. 2002. MALLET: A Machine Learning for Language Toolkit.

David Mimno, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum. 2011. Optimizing semantic coherence in topic models. In *Conference on Empirical Methods in Natural Language Processing*, pages 262–272.

David Newman, Arthur Asuncion, Padhraic Smyth, and Max Welling. 2009. Distributed algorithms for topic models. *Journal of Machine Learning Research*, 10(62):1801–1828.

John Paisley, Chong Wang, David M. Blei, and Michael I. Jordan. 2015. Nested hierarchical Dirichlet processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):256–270.

Omiros Papaspiliopoulos and Gareth O. Roberts. 2008. Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95(1):169–186.

Margaret E. Roberts, Brandon M. Stewart, Dustin Tingley, Christopher Lucas, Jetson Leder-Luis, Shana Kushner Gadarian, Bethany Albertson, and David G. Rand. 2014. Structural topic models for open-ended survey responses. *American Journal of Political Science*, 58(4):1064–1082.

Yee Whye Teh. 2006. A hierarchical Bayesian language model based on Pitman–Yor processes. In *Proceedings of the Association for Computational Linguistics*, pages 985–992.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. 2006. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476):1566–1581.

Alexander Terenin, Måns Magnusson, Leif Jonsson, and David Draper. 2019. Pólya urn latent Dirichlet allocation: a doubly sparse massively parallel sampler. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(7):1709–1719.

Alastair J. Walker. 1977. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 10(8):253–256.

Chong Wang, John Paisley, and David M. Blei. 2011. Online variational inference for the hierarchical Dirichlet process. In *Artificial Intelligence and Statistics*, pages 752–760.

William Yang Wang, Elijah Mayfield, Suresh Naidu, and Jeremiah Dittmar. 2012. Historical analysis of legal opinions with a sparse mixed-effects latent variable model. In *Proceedings of the Association for Computational Linguistics*, volume 1, pages 740–749.

Sinead Williamson, Avinava Dubey, and Eric P. Xing. 2013. Parallel Markov chain Monte Carlo for nonparametric mixture models. In *International Conference on Machine Learning*, pages 98–106.

Tze-I Yang, Andrew J. Torget, and Rada Mihalcea. 2011. Topic modeling on historical newspapers. In *ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 96–104.

Limin Yao, David Mimno, and Andrew K. McCallum. 2009. Efficient methods for topic model inference on streaming document collections. In *Knowledge Discovery and Data Mining*, pages 937–946.