Scalable Collapsed Inference for High-Dimensional Topic Models

Rashidul Islam and **James Foulds**
Department of Information Systems
University of Maryland, Baltimore County
{islam.rashidul,jfoulds}@umbc.edu

Abstract

The bigger the corpus, the more topics it can potentially support. To truly make full use of massive text corpora, a topic model inference algorithm must therefore scale efficiently in 1) documents and 2) topics, while 3) achieving accurate inference. Previous methods have achieved two out of three of these criteria simultaneously, but never all three at once. In this paper, we develop an online inference algorithm for topic models which leverages *stochasticity* to scale well in the number of documents, *sparsity* to scale well in the number of topics, and which operates in the *collapsed representation* of the topic model for improved accuracy and run-time performance. We use a Monte Carlo inner loop in the online setting to approximate the collapsed variational Bayes updates in a sparse and efficient way, which we accomplish via the Metropolis-Hastings Walker method. We showcase our algorithm on LDA and the recently proposed mixed membership skip-gram topic model. Our method requires only amortized $O(k_d)$ computation per word token instead of $O(K)$ operations, where the number of topics occurring for a particular document $k_d \ll$ the total number of topics in the corpus K, to converge to a high-quality solution.

1 Introduction

Topic models are powerful tools for analyzing today's massive, constantly expanding digital text information by representing high-dimensional data in a low-dimensional subspace. We can recover the main themes of a corpus by using topic models such as latent Dirichlet allocation (LDA) to organize, understand, search, and explore the documents (Blei et al., 2003).

Traditional LDA inference techniques such as variational Bayes and collapsed Gibbs sampling do not readily scale to corpora containing millions of documents. To scale up inference, the main approaches are distributed algorithms (Newman et al., 2008) and stochastic algorithms (Hoffman et al., 2010, 2013). Stochastic algorithms, such as stochastic variational inference (SVI), operate in an online fashion, and hence do not need to see all of the documents before updating the topics, so they can be applied to corpora of any size, without expensive distributed hardware (Hoffman et al., 2010). The "collapsed" representation of topic models is also frequently important, as it leads to faster convergence, efficient updates, and lower variance in estimation (Griffiths and Steyvers, 2004). The stochastic collapsed variational Bayesian inference (SCVB0) algorithm, proposed by (Foulds et al., 2013), combines the benefits of stochastic and collapsed inference.

Larger corpora typically support more topics, which brings the additional efficiency challenge of training a larger model (Mimno et al., 2012). This challenge has been addressed by exploiting sparsity to perform updates in time sublinear in the number of topics. A sparse variant of the SVI algorithm for LDA, SSVI, proposed by (Mimno et al., 2012), is scalable to large numbers of topics, but does not fully exploit the collapsed representation of LDA, which is important for faster convergence and improved inference accuracy, due to a better variational bound (Teh et al., 2007). The Metropolis Hastings Walker (MHW) method (Li et al., 2014) scales well in the number of topics, and uses a collapsed inference algorithm, but it operates in the batch setting, so it is not scalable to large corpora. LightLDA (Yuan et al., 2015)

Proceedings of NAACL-HLT 2019, pages 2836–2845
Minneapolis, Minnesota, June 2 - June 7, 2019.

is a distributed approach to the MHW method which adopts a data-and-model-parallel strategy to maximize memory and CPU efficiency. However, it is not an online approach, and furthermore requires multiple expensive computer clusters to converge faster. Tensor methods are another approach to speeding up topic models (Anandkumar et al., 2014; Arora et al., 2012), which theoretically guarantee the recovery of the true parameters by overcoming the problem of local optima. These techniques use the method of moments instead of maximum likelihood estimation or Bayesian inference, which leads to lower data efficiency, and sometimes unreliable performance.

In this work, we propose a highly efficient and scalable inference algorithm for topic models. We develop an online algorithm which leverages *stochasticity* to scale well in the number of documents, *sparsity* to scale well in the number of topics, and which operates in the *collapsed* representation of topic models. We thereby combine the individual benefits of SVI, SSVI, SCVB0, and MHW into a single algorithm. Our approach is to develop a sparse version of SCVB0. Inspired by SSVI, we use a Monte Carlo inner loop to approximate the SCVB0 variational distribution updates in a sparse and efficient way, which we accomplish via MHW method. To show the generality of our algorithm, we explore the benefits of our inference method for LDA and another recently proposed topic model, MMSGTM, with experiments on both small and large-scale datasets.

2 Background

To build the foundation for our proposed method, in this section we provide the necessary background on LDA and MMSGTM topic models and their associated inference algorithms. This is followed by a description of the MHW sampler for reducing topic model sampling complexity.

2.1 Latent Dirichlet Allocation and SCVB0

Probabilistic topic models such as LDA (Blei et al., 2003) use latent variables to encode co-occurrence patterns between words in text corpora and other bag-of-words represented data. In LDA, we assume that the D documents in a corpus are each from mixture distributions of K individual topics ϕ_k, $k \in \{1, ..., K\}$, each of which are discrete distributions over words. For a document j of length N_j, the local (document-level) variables θ_j are a distribution over topics drawn from a Dirichlet prior with parameters α_k and for each token, global variables (corpus-level) ϕ_k are drawn from a Dirichlet prior with parameters β_w. Due to conjugacy, we can marginalize out topics $\boldsymbol{\Theta}$ and distributions over topics $\boldsymbol{\Phi}$, and perform inference only on the topic assignments $\boldsymbol{Z}$ in the collapsed representation of LDA (Griffiths and Steyvers, 2004). For scalable and accurate inference, Foulds et al. (2013) proposed a stochastic collapsed variational inference algorithm, SCVB0. The SCVB0 approach computes a variational discrete distribution γ_{ij} over the K topic assignment probabilities for each word i in each document j, but does not maintain the γ variables that increase the memory requirement of original batch CVB0 algorithm (Asuncion et al., 2009). SCVB0 iteratively updates each γ_{ij} using

$$\gamma_{ijk} :\propto \frac{N^{\Phi}_{w_{ij}k} + \beta_{w_{ij}}}{N^{Z}_{k} + \sum_w \beta_w}(N^{\Theta}_{jk} + \alpha_k) \qquad (1)$$

for each topic k, with j^{th} document's i^{th} word w_{ij}. The $\boldsymbol{N^Z}$, $\boldsymbol{N^\Theta}$, and $\boldsymbol{N^\Phi}$ are referred to as CVB0 statistics, where $\boldsymbol{N^Z}$ is the vector of expected number of words assigned to each topic, each entry j, k of matrix $\boldsymbol{N^\Theta}$, and each entry w, k of matrix $\boldsymbol{N^\Phi}$ are the expected number of times document j, and word w are assigned to topic k, respectively, across the corpus. To do stochastic updates of these variables, one sequence of step-sizes ρ^Φ for $\boldsymbol{N^\Phi}$ and $\boldsymbol{N^Z}$ and another sequence ρ^Θ for $\boldsymbol{N^\Theta}$ are maintained. The update of $\boldsymbol{N^\Theta_j}$ for every token i of document j with an online average of the current value and its expected value is

$$\boldsymbol{N^\Theta_j} := (1 - \rho^\Theta_t)\boldsymbol{N^\Theta_j} + \rho^\Theta_t C_j \gamma_{ij} \qquad (2)$$

where C_j is the document length. In practice, it is too expensive to update $\boldsymbol{N^\Phi}$ after every token. This leads to the use of minibatch updates with the average of the M per-token estimates of the form $\boldsymbol{Y_{ij}}$, which is a $W \times K$ matrix with the w_{ij}th row being γ_{ij} and with zeros in the other entries:

$$\boldsymbol{N^\Phi} := (1 - \rho^\Phi_t)\boldsymbol{N^\Phi} + \rho^\Phi_t \hat{\boldsymbol{N}}^{\boldsymbol{\Phi}} \qquad (3)$$

$$\boldsymbol{N^Z} := (1 - \rho^\Phi_t)\boldsymbol{N^Z} + \rho^\Phi_t \hat{\boldsymbol{N}}^{\boldsymbol{Z}}, \qquad (4)$$

where $\hat{\boldsymbol{N}}^{\boldsymbol{\Phi}} = \frac{C}{|M|}\sum_{ij\in M}\boldsymbol{Y_{ij}}$, $\hat{\boldsymbol{N}}^{\boldsymbol{Z}} = \frac{C}{|M|}\sum_{ij\in M}\gamma_{ij}$, and C is the number of words in the corpus. The SCVB0 algorithm outperforms stochastic VB (Hoffman et al., 2010) on large corpora by converging faster and often to a better solution (Foulds et al., 2013). However, the SCVB0 algorithm does not leverage sparsity, and hence requires $O(K)$ operations per word token.

2.2 MMSG Topic Model

To show the generality of our approach to topic models other than LDA, we will also apply our method to a recent model called the Mixed Membership Skip-gram Topic Model (MMSGTM) (Foulds, 2018), which combines ideas from topic models and word embeddings (cf. also (Das et al., 2015; Liu et al., 2015)). MMSGTM's generative model for words and their surrounding context is:

- For each word w_i in the corpus
  - Sample a topic $z_i \sim \text{Discrete}(\theta_{w_i})$
  - For each word $w_c \in context(i)$
    - Sample a context word $w_c \sim \text{Discrete}(\phi_{z_i})$.

The inferred model can then be used to train embeddings for topics and words, although we do not consider this here. The MMSGTM admits a collapsed Gibbs sampler (CGS) which efficiently resolves the cluster assignments. With Dirichlet priors on the parameters, the CGS update is

$$p(z_i = k|.) :\propto (N_{w_i k}^{(\Phi)-i} + \alpha_k) \times \prod_{c=1}^{|context(i)|} \frac{N_{w_c k}^{(\Phi)-i} + \beta_{w_c} + N_{w_c}^{(i,c)}}{N_k^{(Z)-i} + \sum_w \beta_w + c - 1}, \quad (5)$$

where α and β are parameter vectors for Dirichlet priors over the topic and word distributions, $\boldsymbol{N}_{\boldsymbol{w_i}}^{\boldsymbol{\Phi}}$ and $\boldsymbol{N}_{\boldsymbol{w_c}}^{\boldsymbol{\Phi}}$ are input and output word-topic counts (excluding the current word), N^Z is the total topic counts in output word-topic counts, and $N_{w_c}^{(i,c)}$ is the number of occurrences of word w_c before the c^{th} word in the i^{th} context. MMSGTM exploits the MHW algorithm, which scales sub-linearly in K, but not in the number of training documents.

2.3 Metropolis-Hastings-Walker Sampler

The MHW method (Li et al., 2014), which is a key component of our approach, uses a data structure called an alias table which allows sampling from a discrete distribution in amortized $O(1)$ time. Assuming initial probabilities $p_0, p_1, ..., p_{l-1}$ of a distribution over l outcomes and average of probabilities $a = \frac{1}{l}$, the alias table A can be formed as follows (Marsaglia et al., 2004):

- Initialize: for i from 0 to $l-1$
  - $A_{alias}[i] = i$ and $A_{prob} = (i+1)a$
- Do the following steps $n-1$ times
  - Find smallest p_i and largest p_j
  - Set $A_{alias}[i] = j$ and $A_{prob} = i \times a + p_i$
  - $p_j := p_j - (a - p_i)$ and $p_i := a$.

Then, to sample from p using the alias table:

- Roll l-sided fair die to choose element i of A
- If $Rand(1) < A_{prob}[i]$ return i, else return $A_{alias}[i]$.

Li et al. (2014) cache alias table samples, avoiding the need to store the table. Once the supply of samples is exhausted they compute a new alias table. They draw samples from the Gibbs sampling update, analogous to γ_{ij} in Equation 1, in amortized $O(k_d)$ time by decomposing the update into

$$p(z_{ij} = k|.) \propto N_{jk}^{\Theta} \frac{N_{w_{ij}k}^{\Phi} + \beta_{w_{ij}}}{N_k^Z + \sum_w \beta_w} + \alpha_k \frac{N_{w_{ij}k}^{\Phi} + \beta_{w_{ij}}}{N_k^Z + \sum_w \beta_w} \quad (6)$$

where the first term, sparse in k_d, admits sampling in $O(k_d)$ time, and the second term is dense but slow changing. A Metropolis-Hastings (M-H) update is used to correct for approximating the CGS update with a proposal distribution $q(k)$ based on the stale alias samples. Foulds et al. (2018) propose to apply simulated annealing to optimize instead of sample, and which improves mixing for the MMSGTM. This is achieved by raising the model part of the M-H acceptance ratio for a new sample $z_i^{(new)} \sim q(k)$ to the power of $\frac{1}{T_j}$ at iteration j:

$$p(accept\ z_i^{(new)}|.) = min(1, (\frac{p(z_i^{(new)})}{p(z_i^{(old)})})^{\frac{1}{T_j}} \frac{q(z_i^{(old)})}{q(z_i^{(new)})}) . \quad (7)$$

3 Sparse Stochastic CVB0

In this section, we introduce our approach, a sparse version of SCVB0, which combines the individual benefits of the SVI, SSVI, SCVB0 and MHW algorithms, to scale well not only in the

SparseSCVB0			Original SCVB0			SVI		
units	neurons	support	data	matrix	eeg	word	cells	circuit
unit	model	kernel	clustering	pca	time	words	cell	current
net	synaptic	margin	cluster	linear	data	character	cortex	circuits
hidden	input	function	clusters	principal	brain	recognition	response	figure
output	neuron	vector	algorithm	eigenvectors	activity	characters	firing	input
neural	response	svm	model	eigenvalues	signal	trained	cortical	analog
networks	cell	machines	problem	eigenvalue	analysis	input	inhibitory	filter

Table 1: Randomly selected example topics, while models trained on the NIPS corpus for $K = 500$.

Algorithm 1 SparseSCVB0 for TM Inference

Randomly initialize $\boldsymbol{N^G}$, $\boldsymbol{N^L}$; $\boldsymbol{N^Z} := \sum_w \boldsymbol{N^G_w}$;
$doSparse = true$ or $false$
for each minibatch M **do**
  $\hat{\boldsymbol{N}}^{\boldsymbol{G}} := 0$; $\hat{\boldsymbol{N}}^{\boldsymbol{Z}} := 0$
  for each document j in M **do**
    for each token i in j **do**
      $\gamma_{ij}^{(pseudo)} := 0$
      for each sample s in S **do**
        draw $z_i^{(new)} \sim q(k)$
          //via efficient sampling or cached alias samples
        accept or reject $z_i^{(new)}$ via Eq. 7
        if (accept), $z_i := z_i^{(new)}$
        $\gamma_{ij}^{(pseudo)}[z_i] := \gamma_{ij}^{(pseudo)}[z_i] + \frac{1}{S}$
      end for
      $\boldsymbol{N_j^L} := (1 - \rho_t^L)\boldsymbol{N_j^L} + \rho_t^L C_j \gamma_{ij}^{(pseudo)}$
      if (not burn-in pass),
          //update estimates for i or for each context word of i:
        $\hat{\boldsymbol{N}}^{\boldsymbol{G}}_{\boldsymbol{w_{ij}}} := \hat{\boldsymbol{N}}^{\boldsymbol{G}}_{\boldsymbol{w_{ij}}} + \frac{C}{|M|}\gamma_{ij}^{(pseudo)}$
        $\hat{\boldsymbol{N}}^{\boldsymbol{Z}} := \hat{\boldsymbol{N}}^{\boldsymbol{Z}} + \frac{C}{|M|}\gamma_{ij}^{(pseudo)}$
      elseif$(doSparse)$,$\boldsymbol{N_j^L}[k] < \tau\rho^L_{t_{C_j}} C_j := 0$
    end for
  end for
  update $\boldsymbol{N^G} := (1 - \rho_t^G)\boldsymbol{N^G} + \rho_t^G \hat{\boldsymbol{N}}^{\boldsymbol{G}}$
  update $\boldsymbol{N^Z} := (1 - \rho_t^G)\boldsymbol{N^Z} + \rho_t^G \hat{\boldsymbol{N}}^{\boldsymbol{Z}}$
end for

number of documents but also in the number of topics, while gaining the benefits of collapsed inference. We refer to our method as the sparse stochastic collapsed variational Bayesian inference (SparseSCVB0) algorithm.

In SparseSCVB0, we obtain sparsity by substituting sparse Monte Carlo approximations $\gamma_{ij}^{(pseudo)}$ for the original SCVB0 variational distributions γ_{ij}. The justification for this procedure, also used by (Mimno et al., 2012), is that the expected value of an average over one-hot samples from a distribution is equal to that distribution:

$$E_{s_i \sim p(\mathbf{s})}\Big[\frac{\sum_{i=1}^S \delta_k(s_i)}{S}\Big] = \frac{1}{S}\sum_{i=1}^S E_{s_i \sim p(\mathbf{s})}\Big[\delta_k(s_i)\Big]$$

$$= \frac{1}{S}\sum_{i=1}^S \sum_{k'} \delta_k(s_i = k')p(s_i = k') = p(s = k)\,.$$

Thus, the overall procedure is still a valid stochastic optimization algorithm. We approximate the inner loop sampler in time sublinear in K, constructing $\gamma^{(pseudo)}$ by generating S samples from γ_{ij} using the MHW method. To describe the general form of our algorithm, we introduce SparseSCVB0 statistics: local (e.g. document-level) expected counts $\boldsymbol{N^L}$, global (corpus-level) expected counts $\boldsymbol{N^G}$, and total expected topic counts $\boldsymbol{N^Z}$. We approximate local sufficient statistics $\boldsymbol{N^L}$ for each token i in document j via:

$$N^L_{jk} \approx C_j E_{\gamma_{ij}^{(pseudo)}}\big[\frac{\sum_{s \in S}\delta_{z_{ij}^s = k}}{S}\big]\,.$$

Since $\gamma^{(pseudo)}$ is sparse, we can efficiently update these statistics using only its non-zero entries. This approach allows us to learn high-dimensional topic models efficiently on very large corpora. Before updating global parameters in a similar fashion, it may also be beneficial to perform a small number of burn-in passes to learn the local parameters $\boldsymbol{N^L}$ (Foulds et al., 2013). For large-scale datasets (e.g. Wikipedia), SparseSCVB0 operates in a "mini-epoch" approach where we process a large subset of the corpus (e.g. $5,000$ documents) several (e.g. 3) times, before discarding and processing the next subset, and so on. This allows a "warm start" of $\boldsymbol{N^L}$ in repeating iterations, with a small memory overhead. Pseudo-code of SparseSCVB0 for a mini-epoch is provided in Algorithm 1, which we discuss more in the next two sections, including model-specific aspects.

SparseSCVB0			Original SCVB0			SVI		
infection	human	liver	decreased	dna	medication	mutations	hormone	intensive
infected	humans	hcv	decrease	replication	patients	mutation	invasive	icu
infections	chimpanzee	hbv	increased	chromatin	medications	mutated	androgen	delirium
host	macaque	hepatic	increase	origins	prescribed	crosses	hormones	occurrences
infectious	monkey	hepatitis	decreases	ssdna	patient	missense	testosterone	sedation
antiviral	primates	hepatocytes	decreasing	primase	prescription	mut	chr	haloperidol
inoculation	chimpanzees	cirrhosis	dramatically	xenopus	pharmacy	hpf	hormonal	psychotropic

Table 2: Randomly selected example topics, while models trained on the PubMed corpus with $K = 1,000$.

SparseSCVB0			Original SCVB0			SVI		
band	town	blood	army	data	gold	president	japanese	engine
music	city	treatment	war	system	silver	political	china	test
released	road	disease	division	software	diamond	court	chinese	center
show	south	patients	battle	computer	golden	senate	japan	small
song	river	medical	forces	systems	bronze	constitution	people	gas
live	village	health	corps	version	diamonds	politics	countries	engines
records	school	effects	infantry	bit	pit	elected	asia	base

Table 3: Randomly selected example topics, while models trained on the Wikipedia corpus for $K = 1,000$.

4 SparseSCVB0 for LDA

To deploy SparseSCVB0 for LDA, we use the M-H proposal distribution from (Li et al., 2014) which involves drawing samples exactly from document-specific sparse terms or approximately using cached samples from the alias table:

$$q(k) := \frac{P_L}{P_L + Q_G} p_L(k) + \frac{Q_G}{P_L + Q_G} q_G(k) \quad (8)$$

where $p_L(k)$ and $q_G(k)$ represent the sparse and dense part, respectively from Equation 6 and $P_L = \sum_k p_L(k)$, $Q_G = \sum_k q_G(k)$. When $\frac{P_L}{P_L+Q_G} > RandUnif(1)$, we sample from the sparse part in $O(k_d)$ time depending on only the non-zero entries of $\boldsymbol{N_j^L}$ (analogous to $\boldsymbol{N_j^\Theta}$), as $\boldsymbol{N_j^L}$ is sparse in the LDA setting. Otherwise, we sample from the dense part in amortized $O(1)$ time using the alias method. Unfortunately, due to the stochastic update, any entry of $\boldsymbol{N_j^L}$ never becomes exactly zero, however it may maintain a very small value. To address this, we apply a sparsification heuristic, where we threshold $\boldsymbol{N_j^L}$ after burn-in passes for each document iteration. We parameterize the threshold as $\tau \rho^L_{t_{C_j}} C_j$; where C_j is the length of the j document, $\rho^L_{t_{C_j}}$ is the step size for the last token of this document, and constant $0 < \tau \leq 1$ controls the sparsity. In our preliminary experiments, we found that, somewhat counter-intuitively, the Monte Carlo and sparsification approximations actually improve convergence in early iterations. We believe that this is because they help SCVB0 escape the initial high-entropy regime, during which convergence of variational algorithms is poor (Salakhutdinov et al., 2003). This property makes the benefit of annealing insignificant, so we do not use simulated annealing for LDA inference, fixing $T_j = 1$.

An additional optimization of SparseSCVB0 for LDA inference can be performed by "clumping" (Teh et al., 2007; Foulds et al., 2013), where one update of the local parameters is performed for each distinct word type in each document. This is performed by fixing the variational distribution, and scaling the update by number of copies of the distinct word type in the document. If we observe the distinct word type w_{aj}, which occurs m_{aj} times in document j, the update is

$$\boldsymbol{N_j^L} := (1 - \rho_t^L)^{m_{aj}} \boldsymbol{N_j^L} + (1 - (1 - \rho_t^L)^{m_{aj}}) C_j \gamma_{aj}^{(pseudo)} . \quad (9)$$

5 SparseSCVB0 for MMSGTM

The main contribution of our approach for the MMSGTM algorithm is to scale this algorithm in number of documents with online inference, as MMSGTM already scales sublinearly in K using MHW. Foulds et al. (2018) use an MHW proposal which approximates the CGS update, interpreted as a product of experts (Hinton, 2002) in which each word in the context is an "expert" which weighs in multiplicatively on the update, with a mixture of experts. In the proposal, they draw a

		Input word = learning Top words in top 2 topics for the input word
NIPS	SparseSCVB0	algorithms algorithm reinforcement problem problems | learning gradient descent rate weight machine
	Original SCVB0	learning networks network algorithm neural | propagation function reinforcement gradient algorithms
Wikipedia	SparseSCVB0	university school college education students research | public center science schools article information
	Original SCVB0	students school education university college schools | center year program degree systems information

Table 4: Top words in the top 2 topics for an input word using original SCVB0 and SparseSCVB0 for MMSGTM.

context word w_c uniformly from the context of the current word, $w_c \sim Uniform(|context(w_i)|)$, and then sample a word based on the chosen context word's contribution to the update:

$$q(k) :\propto \frac{N^G_{w_c k} + \beta_{w_c}}{N^Z_k + \sum_w \beta_w} \tag{10}$$

where $\boldsymbol{N^G_{w_c}}$ is analogous to the output context word-topic counts $\boldsymbol{N^{\Phi}_{w_c}}$ of original MMSGTM model. The proposal samples via the alias method in amortized $O(1)$ time, instead of $O(k_d)$ time, since it does not involve the sparse term. We use this proposal to approximate the CVB0 update for the model, which is a deterministic version of Equation 5, neglecting to exclude the current assignment of z_i. We update $\boldsymbol{N^G_{w_i}}$ (analogous to $\boldsymbol{N^{\Phi}_{w_c}}$) for each current word w_i locally, but update $\boldsymbol{N^G_{w_c}}$ and $\boldsymbol{N^z}$ via minibatch counts $\hat{\boldsymbol{N}}^{\boldsymbol{G}}_{\boldsymbol{w_c}}$ and $\hat{\boldsymbol{N}}^{\boldsymbol{Z}}$, respectively, for each context word w_c of current word w_i. Unlike for LDA, C_j in the local updates represents the total number of input word j in the corpus. As we draw multiple output words from each topic assignment, the effective temperature of the MMSGTM model is much lower than for standard LDA which may cause problems with mixing and leads it to get stuck in the initial regime. Following Foulds et al. (2018), we perform simulated annealing which varies the M-H acceptance ratio to improve mixing. We parameterize the temperature schedule as $T_j = T_0 + \lambda \kappa^{\frac{\mu j}{D}}$, where T_0 is the target final temperature, $\kappa \leq 1$, constant μ controls the amount of temperature reduction after each document iteration j and λ controls the initial temperature.

6 Experiments

In this section we study the performance of our SparseSCVB0 [1] algorithm, on small as well as large corpora to validate the proposed method for topic models such as LDA and MMSGTM, and to compare with other state-of-the-art algorithms.

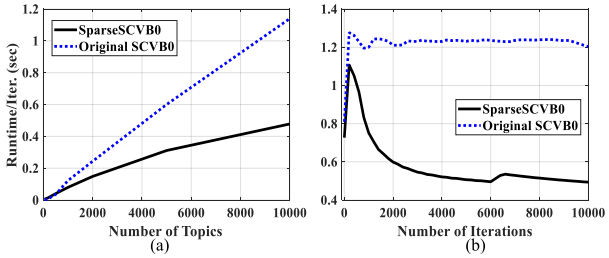

Figure 1: Comparison of runtime per iteration for LDA in terms of: (a) number of topics K and (b) number of iterations when $K = 10,000$. Original SCVB0 is linear in K, while SparseSCVB0 is sublinear in K.

6.1 Experimental Environment and Datasets

We compared SparseSCVB0 to SCVB0 and SVI. For a fair comparison, we implemented all of them in the fast high-level language Julia V0.6.2 (Bezanson et al., 2017). We conducted all experiments on a computer with 64GB memory and an Intel Xeon E5-2623 V4 processor with 2.60 GHz clock rate, 8×256KB L2 Cache and 10MB L3 Cache. As we only use one single thread for sampling across all experiments, only one CPU core is active throughout the experiment with only 256KB available L2 Cache.[2]

We used *NIPS*, *Reuters-150*, *PubMed Central*, and *Wikipedia* as representative very small, small, medium, and large-scale datasets, respectively. The *NIPS* corpus has 1740 scientific articles from years 1987-1999 with 2.3M tokens, due to Sam Roweis. The newswire corpus *Reuters-150* contains $15,500$ articles with dictionary size of $8,350$ words. *PubMed Central* has $320M$ tokens across $165,000$ scientific articles and a vocabulary size of around $38,500$ words. The *Wikipedia* corpus contained 4.6 million articles from the online

[1] Code implementing SparseSCVB0 can be found at https://github.com/dr97531/SparseSCVB0 .

[2] Since SSVI relies on multiple complex implementation details, we were unable to develop a fair implementation, nor were we able to obtain soure code for a previous implementation. We expect that its accuracy would be similar to SVI, with a speed-up at or below that bestowed by a MHW-based inner loop. (Mimno et al., 2012) apply it with only 200 topics for most of their experiments, and at most 1000 topics.

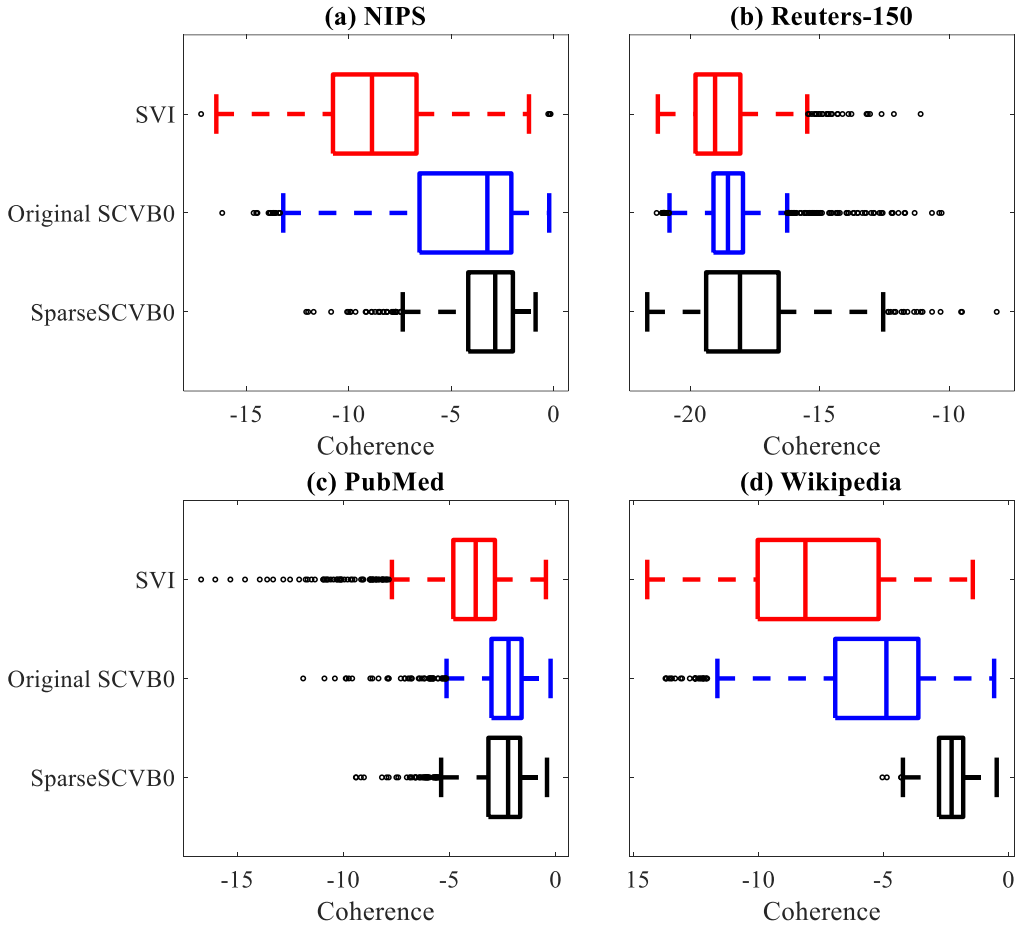

Figure 2: Per-topic coherence for LDA when $K = 1,000$ on (a) NIPS, (b) PubMed, and (c) Wikipedia. SparseSCVB0 completely outperforms other models for large-scale corpus.

network data communication information communicate connection
prison prisoners prisoner imprisoned jail escaped detained guards
dog dogs shepherd hounds bred coat scent instinct eating companion
song sung sing singing sings sang songs recorded melody tune
votes vote cast elections voted candidate parties majority election
wind winds blowing speed blows direction high low blown chill
hour hours noon time daylight minutes midnight morning seconds

Table 5: Randomly selected topics from a $10,000$-topic model trained using SparseSCVB0 on Wikipedia

encyclopedia. We used the dictionary of $7,700$ words which was extracted by Hoffman et al. (Hoffman et al., 2013). There were 811M tokens in the corpus.

6.2 Performance for LDA

We implement SparseSCVB0, original SCVB0 and SVI algorithms using the clumping optimization (Teh et al., 2007) technique. In all LDA experiments, each algorithm was trained using mini-batches of size 20 for the NIPS corpus and 100 for other corpora. For PubMed and Wikipedia, we chose mini-epoch subsets of size $5,000$ documents and processed for 5 passes. We used a step-size schedule of $\frac{scale}{(\eta+t)^\kappa}$ as in original SCVB0 for global parameters, where t is the document iteration with $scale = 100.0$, $\eta = 1000.0$ and $\kappa = 0.9$. For document-level parameters, we used the $scale = 1.0$, $\eta = 10.0$ and $\kappa = 0.9$, with t referring here to the word iteration of the current document. In case of PubMed corpus, we found out that original SCVB0 and SVI tend to stuck in the initial regime for document-level step-size parameter $\eta = 1.0$ which we later fixed by setting $\eta = 10.0$, while SparseSCVB0 didn't suffer from this problem due to the extra randomness from the sparse sampled updates. Finally, we choose hyper-parameters $\alpha = 0.1$ and $\beta = 0.01$ and burn-in pass of 5 for each document in all LDA experiments. For SparseSCVB0, we used sample size $S = 5$ to approximate $\gamma^{(pseudo)}$ and $\tau = 1/K$ for the sparsification heuristic on local parameters.

To study the acceleration benefits of our approach, we evaluated the runtime performance per

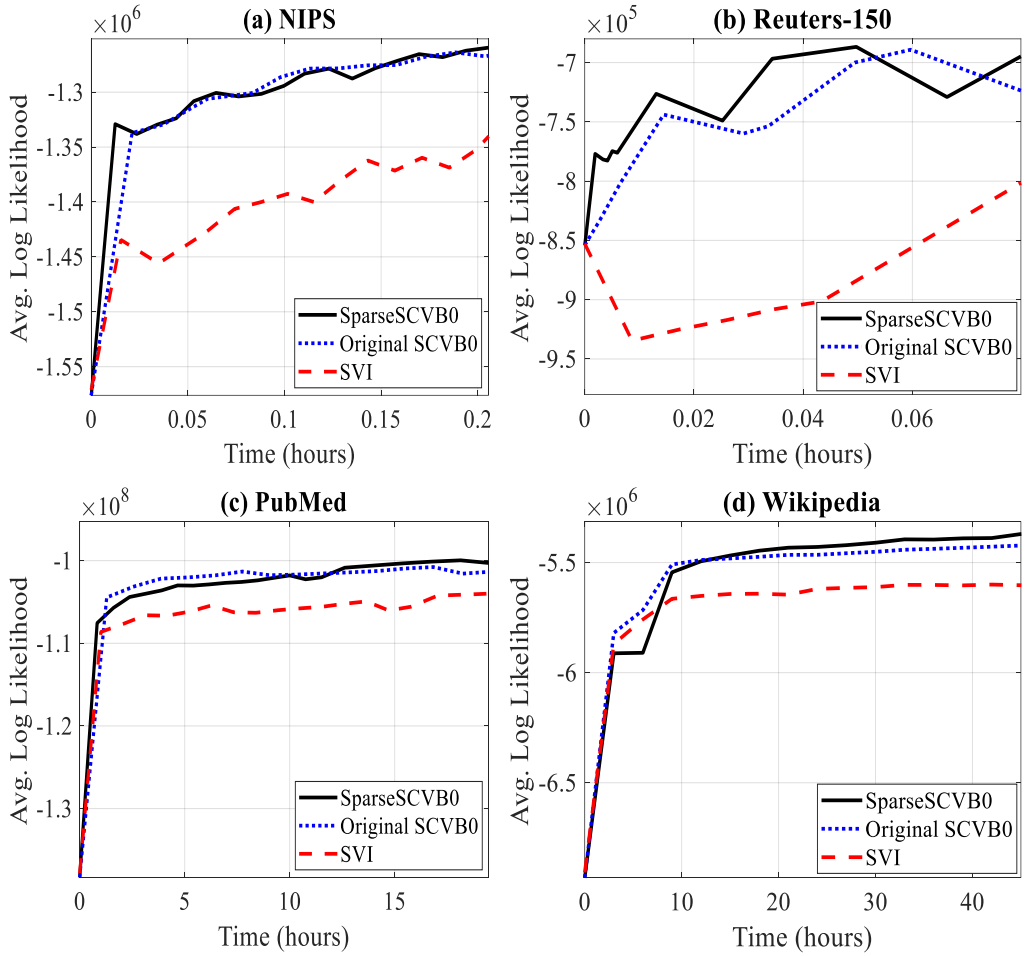

Figure 3: Comparison of average log-likelihood $vs.$ Time for LDA on (a) NIPS, (b) PubMed, and (c) Wikipedia.

iteration on the number topics and the number of iterations. In Figure 1(a), SparseSCVB0 is compared to original SCVB0 in terms of the average runtime per document iteration as a function of the number topics. We see that original SCVB0 requires average linear runtime due to $O(K)$ operations to compute collapsed variational distribution, while the average runtime for SparseSCVB0 grows sublinearly in K, due to $O(k_d)$ operations instead of $O(K)$ operations. SparseSCVB0 starts with approximately $O(K)$ operations in its initial stage of iterations, but it starts getting a benefit from sparsification heuristic after burning in, as shown in Figure 1(b) for $K = 10,000$.

To evaluate the performance in terms of learned topic quality, we start by comparing all of the algorithms in qualitative experiments (see Table 1, Table 2, and Table 3) where we show randomly selected example topics, while all the models were trained on the NIPS, PubMed, and Wikipedia corpus for $K = 500$, $K = 1,000$, and $1,000$, respectively. To get a quantitative insight we evaluated the topics using the per topic coherence metric, which measures the semantic quality of a topic based on the W most probable words for the topics (Mimno et al., 2011), thereby approximating the user viewing experience. In Figure 2, we see that SparseSCVB0 generates better quality topics with higher coherence scores than the other two models for $K = 1000$ with $W = 10$ after running all the models on each corpus for the same amount of time. The coherence performance of SparseSCVB0 increases substantially in the case of the large-scale corpus (Figure 2(c)), since it gets the opportunity to use its runtime advantage and process more documents than the other algorithms.

To investigate model convergence, we measured the held-out log-probability versus wall-clock time for all the algorithms. For each experiment we held-out a set of documents (150 documents for NIPS, 3500 documents for Reuters, and 1000 documents for all other corpora) as test data

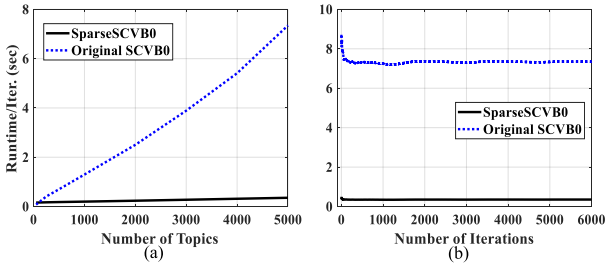

Figure 4: Comparison of runtime per iteration for MMSGTM in terms of: (a) number of topics K and (b) number of iterations when $K = 5,000$. SparseSCVB0 runs in amortized $O(1)$ time, while original SCVB0 is linear in K.

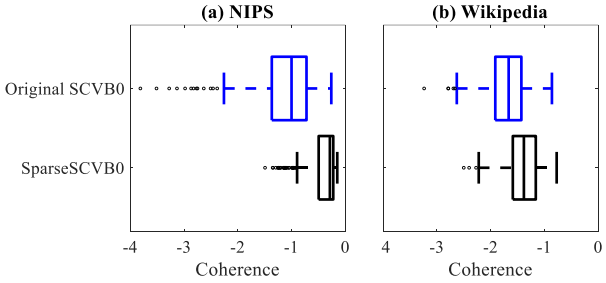

Figure 5: Per-topic coherence for MMSGTM, $K = 500$, on (a) NIPS and (b) Wikipedia. SparseSCVB0 has higher coherence scores for both the small and large corpora.

and trained the model on the rest of the corpus. Then, we split each test document in half, estimated local parameters on first half and finally computed the log-likelihood of the remaining half of the document. Figure 3 shows the comparison of average log-likelihood versus wall-clock time for all four corpora. In terms of log-likelihood, SparseSCVB0 provides an approximately similar result to original SCVB0 for the small corpus, but it converged to a better solution than others in the case of large corpora like Wikipedia (see Figure 3(c)), likely due to its processing a larger number of documents.

SparseSCVB0 enables the large-scale computation needed to learn high-dimensional topic models that could not feasibly be trained using previous methods due to their runtime complexity in the number of documents and/or topics. We show randomly selected topics from the LDA model with $K = 10,000$ in Table 5. This big topic model was trained for 36 hours using SparseSCVB0 on Wikipedia. We performed a dense initialization, running original SCVB0 for the first 5 hours, which was found to help avoid local optima.

6.3 Performance for MMSGTM

We also conducted experiments to evaluate the performance of SparseSCVB0 for MMSGTM and compare with original SCVB0. In all MMSGTM experiments, we kept the same step size schedule for global parameters as $scale = 1.0$, $\eta = 5.0$ and $\kappa = 0.9$, but for local parameter updates we maintain a separate step-size schedule of $\frac{scale}{(\eta+t)^\kappa}$ for each input word, with t referring to the number of times we processed this input word, while η and κ values remained the same. For simulated annealing of SparseSCVB0, we used $T_0 = 0.00001$, $\kappa = 0.9$, $\mu = 5$ and $\lambda = |context|$ with a $context$ size of 5. We kept the same number of document burn-in passes as we did for the LDA experiments.

In Figure 4, we show the runtime improvement of SparseSCVB0 over original SCVB0 for MMSGTM in a similar experiment to the one for LDA. For MMSGTM, SparseSCVB0 substantially outperforms original SCVB0 by processing each document in amortized $O(1)$ time. We provide qualitative results in the case of MMSGTM model by showing several top words in the top 2 topics for an input word using original SCVB0 and SparseSCVB0 in Table 4 for $K = 500$. As for LDA, SparseSCVB0 allows us to generate topics with higher coherence scores compared to the original SCVB0 after running for the same amount of time (Figure 5) on both small and large corpora.

7 Conclusions

This paper introduced SparseSCVB0, a sparse version of the SCVB0 inference algorithm which performs fast, scalable high-dimensional topic model inference. SparseSCVB0 leverages stochasticity to scale well in both the corpus size and in the number of topics. It operates in the collapsed representation of topic models which leads to fast convergence while providing an improved variational bound. We show that SparseSCVB0 reduces the operational complexity for the variational Bayes update of online topic models from $O(K)$ to $O(k_d)$ time for LDA and amortized $O(1)$ time for MMSGTM. We evaluated and compared the performance of our approach with state-of-the-art models such as original SCVB0 and SVI to demonstrate that SparseSCVB0 converges much more efficiently, while maintaining high quality topics with a better per-topic coherence score.

References

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. 2014. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832.

Sanjeev Arora, Rong Ge, and Ankur Moitra. 2012. Learning topic models–going beyond SVD. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 1–10. IEEE.

Arthur Asuncion, Max Welling, Padhraic Smyth, and Yee Whye Teh. 2009. On smoothing and inference for topic models. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 27–34. AUAI Press.

Jeff Bezanson, Alan Edelman, Stefan Karpinski, and Viral B Shah. 2017. Julia: A fresh approach to numerical computing. *SIAM review*, 59(1):65–98.

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

Rajarshi Das, Manzil Zaheer, and Chris Dyer. 2015. Gaussian LDA for topic models with word embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 795–804.

J. R. Foulds. 2018. Mixed membership word embeddings for computational social science. *Proceedings of the 21st International Conference on Artificial Intelligence and Statistics (AISTATS)*.

James Foulds, Levi Boyles, Christopher DuBois, Padhraic Smyth, and Max Welling. 2013. Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 446–454. ACM.

Thomas L Griffiths and Mark Steyvers. 2004. Finding scientific topics. *Proceedings of the National Academy of Sciences*, 101(suppl 1):5228–5235.

Geoffrey E Hinton. 2002. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800.

Matthew Hoffman, Francis R Bach, and David M Blei. 2010. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 856–864.

Matthew D Hoffman, David M Blei, Chong Wang, and John Paisley. 2013. Stochastic variational inference. *The Journal of Machine Learning Research*, 14(1):1303–1347.

Aaron Q Li, Amr Ahmed, Sujith Ravi, and Alexander J Smola. 2014. Reducing the sampling complexity of topic models. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 891–900. ACM.

Yang Liu, Zhiyuan Liu, Tat-Seng Chua, and Maosong Sun. 2015. Topical word embeddings. In *AAAI*, pages 2418–2424.

George Marsaglia, Wai Wan Tsang, Jingbo Wang, et al. 2004. Fast generation of discrete random variables. *Journal of Statistical Software*, 11(3):1–11.

David Mimno, Matt Hoffman, and David Blei. 2012. Sparse stochastic inference for latent Dirichlet allocation. *Proceedings of the 29th International Conference on Machine Learning*.

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

David Newman, Padhraic Smyth, Max Welling, and Arthur U Asuncion. 2008. Distributed inference for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1081–1088.

Ruslan Salakhutdinov, Sam T Roweis, and Zoubin Ghahramani. 2003. Optimization with EM and expectation-conjugate-gradient. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 672–679.

Yee W Teh, David Newman, and Max Welling. 2007. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems*, pages 1353–1360.

Jinhui Yuan, Fei Gao, Qirong Ho, Wei Dai, Jinliang Wei, Xun Zheng, Eric Po Xing, Tie-Yan Liu, and Wei-Ying Ma. 2015. LightLDA: Big topic models on modest computer clusters. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1351–1361. International World Wide Web Conferences Steering Committee.