Stochastic Collapsed Variational Bayesian Inference for Latent Dirichlet Allocation

James Foulds
Dept. of Computer Science
University of California, Irvine
jfoulds@ics.uci.edu

Levi Boyles
Dept. of Computer Science
University of California, Irvine
lboyles@ics.uci.edu

Christopher DuBois
Dept. of Statistics
University of California, Irvine
duboisc@ics.uci.edu

Padhraic Smyth
Dept. of Computer Science
University of California, Irvine
smyth@ics.uci.edu

Max Welling
Informatics Institute
University of Amsterdam
M.Welling@uva.nl

ABSTRACT

There has been an explosion in the amount of digital text information available in recent years, leading to challenges of scale for traditional inference algorithms for topic models. Recent advances in stochastic variational inference algorithms for latent Dirichlet allocation (LDA) have made it feasible to learn topic models on very large-scale corpora, but these methods do not currently take full advantage of the collapsed representation of the model. We propose a stochastic algorithm for collapsed variational Bayesian inference for LDA, which is simpler and more efficient than the state of the art method. In experiments on large-scale text corpora, the algorithm was found to converge faster and often to a better solution than previous methods. Human-subject experiments also demonstrated that the method can learn coherent topics in seconds on small corpora, facilitating the use of topic models in interactive document analysis software.

Categories and Subject Descriptors

I.5.1 [**Models**]: Statistical; I.2.7 [**Natural Language Processing**]: Text analysis

General Terms

Algorithms, Experimentation, Performance

Keywords

Topic models, variational inference, stochastic learning

KDD'13, August 11–14, 2013, Chicago, Illinois, USA.

ACM 978-1-4503-2174-7/13/08 ...$15.00.

1. INTRODUCTION

Topic models such as latent Dirichlet allocation (LDA) [7] are now widely used in modern machine learning. Inference algorithms for topic models provide a low-dimensional representation of text corpora that is typically semantically meaningful, despite being completely unsupervised. Their use has spread beyond machine learning to become a standard analysis tool for researchers in many fields [15, 4, 8]. In the internet era, there is a need for tools to learn topic models at the "web scale", especially in an industrial setting. For example, news aggregators such as Yahoo! News publish a continually updated stream of online articles. These applications need to analyze candidate articles for topical diversity and relevance to current trends, which can be facilitated by topic models [1].

In this context it would be useful to have the tools to build topic models that scale to such large corpora, taking advantage of the large amounts of available data to create models that are both more complex (e.g. have more topics) and more accurate. Traditional inference techniques such as Gibbs sampling and variational inference do not readily scale to corpora containing millions of documents or more. In such cases it is very time-consuming to run even a single iteration of the standard collapsed Gibbs sampling [12] or variational Bayesian inference algorithms [7], let alone run them until convergence. For these algorithms, the first few passes through the data are inhibited by randomly initialized values of the parameters and latent variables which misinform the updates, so multiple such expensive iterations are required to learn the topics.

A significant recent advance was made by Hoffman et al. [13], who proposed a stochastic variational inference algorithm for LDA topic models. Because the algorithm does not need to see all of the documents before updating the topics, this method can often learn good topics before a single iteration of the traditional batch inference algorithms would be completed. The algorithm processes documents in an online fashion, so it can be applied to corpora of any size, or even to never-ending streams of documents. A more scalable variant of this algorithm was proposed by Mimno et al. [16], which approximates the gradient updates in a sparse way in order to improve performance for larger vocabularies and greater numbers of topics.

A complementary direction that has been useful for improving inference in LDA is to take advantage of its "collapsed" representation, where parameters are marginalized out, leaving only latent variables. It is possible to perform inference in the collapsed space and recover estimates of the parameters afterwards. For inference techniques that operate in a batch setting, the algorithms that operate in the collapsed space are more efficient at improving held-out log probability than their uncollapsed counterparts, both per iteration and in wall-clock time per iteration [12, 24, 3]. Reasons for this advantage include the per-token updates which propagate updated information sooner, simpler update equations, fewer parameters to update, no expensive calls to the digamma function, and the avoidance of tightly coupled pairs of parameters which inhibit mixing for Gibbs sampling [10, 3, 24]. For variational inference, perhaps the most important advantage of the collapsed representation is that the variational bound is strictly better than that for the uncollapsed representation, leading to the potential for collapsed variational algorithms to learn more accurate topic models than uncollapsed variational algorithms [24]. Existing online inference algorithms for LDA do not fully take advantage of the collapsed representation. An exception is the sparse online LDA algorithm of Mimno et al. [16] which collapses out per-document parameters θ, however the topics themselves are not collapsed out.

In this work, we develop a stochastic algorithm for LDA that operates fully in the collapsed space, thus transferring the aforementioned advantages of collapsed inference to the online setting. This facilitates learning topic models both more accurately and more quickly on large datasets. The proposed algorithm is also very simple to implement, requiring only basic arithmetic operations. In addition to experiments on large datasets, we also explore the benefit of our method on small problems, showing that it is feasible to learn human-interpretable topics in seconds.

2. BACKGROUND

Probabilistic topic models such as LDA [7] use latent variables to encode co-occurrence patterns between words in text corpora and other bag-of-words data. In the LDA model, there are K topics $\phi_k, k \in \{1, \ldots, K\}$, each of which are discrete distributions over words (see Table 1 for relevant notation). For example, a topic on baseball might give high probabilities to words such as "pitcher," "bat" and "base". The assumed generative process for the LDA model is

Generate each topic $\phi_k \sim \text{Dirichlet}(\eta), k \in \{1, \ldots, K\}$
For each document j
  Generate a distribution over topics $\theta_j \sim \text{Dirichlet}(\alpha)$
  For each word i in document j
    Sample a topic $z_{ij} \sim \text{Discrete}(\theta_j)$
    Sample the word $w_{ij} \sim \text{Discrete}(\phi_{z_{ij}})$

To scale LDA inference to very large datasets, a stochastic variational inference algorithm was proposed by Hoffman et al. [13]. We will discuss its more general form [14], which applies to graphical models whose parameters can be split into "global" parameters G and "local" parameters L_j pertaining to each data point x_j, and whose complete conditional distributions for each variable are exponential family distributions. The algorithm examines one data point at a time to learn that data point's local variational parameters, such as θ_j in LDA. It then updates global variational parameters, such as topics ϕ_k, via a stochastic natural gradient update. Their general scheme is given in Algorithm 1.

K	Number of topics
D	Number of documents
C	Number of words in corpus
C_j	Number of words in document j
z_{ij}	Topic for (i,j), the ith word of the jth document
w_{ij}	Dictionary index for word (i,j)
θ_j	Distribution over topics for document j, $K \times 1$
ϕ_k	Distribution over words for topic k, $W \times 1$
α	Dirichlet prior parameters for θ, $K \times 1$
η	Dirichlet prior parameters for Φ, $W \times 1$
γ_{ij}	Variational distribution for word (i,j), $1 \times K$
$\mathbf{N}^{\Theta}$	Expected topic counts per document, $D \times K$
$\mathbf{N}^{\Phi}$	Expected topic counts per word, $W \times K$
$\mathbf{N}^{Z}$	Expected topic counts overall, $1 \times K$
$\mathbf{Y}^{(ij)}$	Estimate of $\mathbf{N}^{\Phi}$ based only on word (i,j), $W \times K$
M	Minibatch, a set of documents
$\hat{\mathbf{N}}^{\Phi}$	Estimate of $\mathbf{N}^{\Phi}$ from current minibatch, $W \times K$
$\hat{\mathbf{N}}^{Z}$	Estimate of $\mathbf{N}^{Z}$ from current minibatch, $1 \times K$
ρ_t^{Θ}	Step size for $\mathbf{N}^{\Theta}$ at timestep t
ρ_t^{Φ}	Step size for $\mathbf{N}^{\Phi}$ and $\mathbf{N}^{Z}$ at timestep t
w_{aj}	Dictionary index for ath distinct word of j
m_{aj}	Count of ath distinct word of j
γ_{aj}	Variational dist. for ath distinct word of j, $1 \times K$

Table 1: Summary of notation

Algorithm 1 Stochastic Variational Inference (Hoffman et al.)

- Input: Data points $x_1, \ldots, x_D$ (e.g. word count histograms for documents), step sizes $\rho_t, t = 1:m$ (where m is the maximum number of iterations)
- Randomly initialize "global" (e.g. topic) parameters G
- For $t = 1:m$
  - Select a random data point x_j, $j \in \{1, \ldots, D\}$
  - Compute "local" (e.g. document-level) variational parameters $\mathbf{L}_j$
  - $\hat{\mathbf{G}} = D\mathbf{L}_j$
  - $\mathbf{G} := (1-\rho_t)\mathbf{G} + \rho_t\hat{\mathbf{G}}$

For an appropriate local update and sequence of step sizes ρ, this algorithm is guaranteed to converge to the optimal variational solution [14]. In the case of LDA, let λ_k be the parameter vector for a variational Dirichlet distribution on topic ϕ_k. For each document j, the method computes variational distributions for both the topic assignments and the document's distribution over topics using regular VB updates. These values are then used to update the topics. Specifically, for each topic k the algorithm computes $\hat{\lambda}_k$, an estimate of what λ_k would be if all D documents were identical to document j. The algorithm then updates the λ_k's via a natural gradient update, which takes the form

$$\lambda_k := (1-\rho_t)\lambda_k + \rho_t\hat{\lambda}_k \ . \tag{1}$$

In a somewhat broader context, the online EM algorithm of Cappe and Moulines [9] is another general-purpose method for learning latent variable models in an online setting. This EM algorithm alternates between a standard M-step which

maximizes the EM lower bound with respect to parameters θ, and a stochastic expectation step, which updates exponential family sufficient statistics $\mathbf{s}$ with an online average

$$\mathbf{s} := (1-\rho_t)s + \rho_t \hat{\mathbf{s}}(Y_{n+1};\theta) \ , \tag{2}$$

with Y_{n+1} being a new data point, θ being the current parameters, and $\hat{\mathbf{s}}(Y_{n+1};\theta)$ being an estimate of the sufficient statistics based on these values.

In this article, we show how to perform stochastic variational inference in the collapsed representation of LDA, using an algorithm inspired by both the online algorithms of Hoffman et al. and Cappe and Moulines. This new algorithm takes advantage of a fast collapsed inference method called "CVB0" [3] to further improve the efficiency of stochastic LDA inference.

2.1 CVB0

In the collapsed representation of LDA, we marginalize out topics $\mathbf{\Theta}$ and distributions over topics $\mathbf{\Phi}$, and perform inference only on the topic assignments $\mathbf{Z}$. The collapsed variational Bayesian inference (CVB) approach of Teh et al. [24] maintains variational discrete distributions γ_{ij} over the K topic assignment probabilities for each word i in each document j. The coordinate ascent updates to optimize the evidence lower bound with respect to γ are intractable. Nonetheless, Teh et al. showed that an algorithm using approximate updates works well in practice, outperforming the classical VB algorithm in terms of prediction performance. Asuncion et al. [3] later showed that a simpler version of this method called CVB0, based on additional approximations, is much faster while still maintaining the accuracy of CVB. The CVB0 algorithm iteratively updates each γ_{ij} via

$$\gamma_{ijk} :\propto \frac{N^{\Phi\neg ij}_{w_{ij}k} + \eta_{w_{ij}}}{N^{Z\neg ij}_{k} + \sum_w \eta_w}(N^{\Theta\neg ij}_{jk} + \alpha_k) \tag{3}$$

for each topic k, with w_{ij} corresponding to the word index for the jth document's ith word, and where $a :\propto b$ denotes that a is assigned to be proportional to b. The $\mathbf{N}^Z$, $\mathbf{N}^\Theta$ and $\mathbf{N}^\Phi$ variables, henceforth referred to as the CVB0 statistics, are variational expected counts corresponding to their indices, and the $\neg ij$ superscript indicates the exclusion of the current value of γ_{ij}. Specifically, $\mathbf{N}^Z$ is the vector of expected number of words assigned to each topic, $\mathbf{N}^\Theta_j$ is the equivalent vector for document j only, and each entry w,k of matrix $\mathbf{N}^\Phi$ is the expected number of times word w is assigned to topic k across the corpus,

$$N^Z_k \triangleq \sum_{ij} \gamma_{ijk} \qquad N^\Theta_{jk} \triangleq \sum_i \gamma_{ijk} \qquad N^\Phi_{wk} \triangleq \sum_{ij:w_{ij}=w} \gamma_{ijk} \ . \tag{4}$$

Note that $\mathbf{N}^\Theta_j + \alpha$ is an unnormalized variational estimate of the posterior mean of document j's distribution over topics θ_j, and column k of $\mathbf{N}^\Phi + \eta$ is an unnormalized variational estimate of the posterior mean of topic ϕ_k.

CVB0 is currently the fastest known technique for LDA inference for single-core batch inference in terms of convergence rate [3]. It is also as simple to implement as collapsed Gibbs sampling, and has a very similar update procedure except that the update is deterministic. Sato and Nakagawa [22] showed that the terms in the CVB0 update can be understood as optimizing the α-divergence, with different values of α for each term. The α-divergence is a generalization of the KL-divergence that variational Bayes minimizes, and optimizing it is known as power expectation propagation [17]. A disadvantage of CVB0 is that the memory requirements are large as it needs to store a variational distribution γ for every token in the corpus (although this can be improved slightly by "clumping" every occurrence of a specific word in each document together and storing a single γ for them).

3. STOCHASTIC CVB0

Given the discussion above, a desirable stochastic algorithm would be one that exploits both (a) the efficiency and simplicity of CVB0, and (b) the improved variational bound of the collapsed representation. Such an algorithm should not need to maintain the γ variables, thus circumventing the memory requirements of CVB0. It should also be able to provide an estimate for the topics when only a subset of the data have been visited. Recall that the CVB0 statistics $\mathbf{N}^Z$, $\mathbf{N}^\Theta$ and $\mathbf{N}^\Phi$ are all that are needed to both perform a CVB0 update and to recover estimates of the topics. Given this, we wish to estimate the CVB0 statistics based only on the subset of tokens we have observed.

Suppose we have seen a token w_{ij}, and its associated γ_{ij}. The information this gives us about the statistics depends on how the token was drawn. If the token was drawn uniformly at random from all of the tokens in the corpus, the expected value of $\mathbf{N}^Z$ with respect to the sampling distribution is $C\gamma_{ij}$, where C is the number of words in the corpus. For the same sampling procedure, the expectation of the word-topic expected counts matrix $\mathbf{N}^\Phi$ is $C\mathbf{Y}^{(ij)}$, where $\mathbf{Y}^{(ij)}$ is a $W \times K$ matrix with the w_{ij}th row being γ_{ij} and with zeros in the other entries. Now if the token was drawn uniformly from the tokens in document j, the expected value of $\mathbf{N}^\Theta_j$ is $C_j\gamma_{ij}$, where C_j is the length of document j.[1]

Since we may not maintain the γ's, we cannot perform these sampling procedures directly. However, with a current guess at the CVB0 statistics we can *update* a token's variational distribution, and observe its new value. We can then use this γ_{ij} to improve our estimate of the CVB0 statistics. This suggests an iterative procedure, alternating between a "maximization" step, approximately optimizing the evidence lower bound with respect to a particular γ_{ij} via CVB0, and an "expectation" step, where we update the expected count statistics to take into account the new γ_{ij}. As the algorithm continues, the γ_{ij}'s we observe will change, so we cannot simply average them. Instead, we can follow Cappe and Moulines [9] and perform an online average of these statistics via Equation 2.

In the proposed algorithm, we process the corpus one token at a time, examining the tokens from each document in turn. For each token, we first compute a new γ_{ij}. We do not store the γ's, but compute (updated versions of) them as needed via CVB0. This means we must make a small additional approximation in that we cannot subtract current values of γ_{ij} in Equation 3. With large corpora and large documents this difference is negligible. The update becomes

[1]Other sampling schemes are possible, which would lead to different algorithms. For example, one could sample from the set of tokens with word index w to estimate $\mathbf{N}^\Phi_w$. Our choice leads to an algorithm that is practical in the online setting.

$$\gamma_{ijk} :\propto \frac{N^{\Phi}_{w_{ij}k} + \eta_{w_{ij}}}{N^{Z}_{k} + \sum_w \eta_w}(N^{\Theta}_{jk} + \alpha_k) \ . \quad (5)$$

We then use this to re-estimate our CVB0 statistics. We use one sequence of step-sizes ρ^{Φ} for $\mathbf{N}^{\Phi}$ and $\mathbf{N}^{Z}$, and another sequence ρ^{Θ} for $\mathbf{N}^{\Theta}$. While we are processing randomly ordered tokens i of document j, we are effectively drawing random tokens from it, so the expectation of $\mathbf{N}^{\Theta}_{j}$ is $C_j\gamma_{ij}$. We update $\mathbf{N}^{\Theta}_{j}$ with an online average of the current value and its expected value,

$$\mathbf{N}^{\Theta}_{j} := (1-\rho^{\Theta}_{t})\mathbf{N}^{\Theta}_{j} + \rho^{\Theta}_{t} C_j \gamma_{ij} \ . \quad (6)$$

Although we process one document at a time, we eventually process all of the words in the corpus. So for the purposes of updating $\mathbf{N}^{\Phi}$ and $\mathbf{N}^{Z}$, in the long-run the algorithm is effectively drawing tokens from the entire corpus. The expected $\mathbf{N}^{\Phi}$ after observing one γ_{ij} is $C\mathbf{Y}^{(ij)}$, and the expected $\mathbf{N}^{Z}$ is $C\gamma_{ij}$. In practice, it is too expensive to update the entire $\mathbf{N}^{\Phi}$ after every token, suggesting the use of minibatch updates. The expected $\mathbf{N}^{\Phi}$ after observing a minibatch M is the average of the per-token estimates, and similarly for $\mathbf{N}^{Z}$, leading to the updates:

$$\mathbf{N}^{\Phi} := (1-\rho^{\Phi}_{t})\mathbf{N}^{\Phi} + \rho^{\Phi}_{t}\hat{\mathbf{N}}^{\Phi} \quad (7)$$

$$\mathbf{N}^{Z} := (1-\rho^{\Phi}_{t})\mathbf{N}^{Z} + \rho^{\Phi}_{t}\hat{\mathbf{N}}^{Z} \quad (8)$$

where $\hat{\mathbf{N}}^{\Phi} = \frac{C}{|M|}\sum_{ij\in M}\mathbf{Y}^{(ij)}$ and $\hat{\mathbf{N}}^{Z} = \frac{C}{|M|}\sum_{ij\in M}\gamma_{ij}$. Depending on the lengths of the documents and the number of topics, it may also be beneficial to perform a small number of extra passes to learn the document statistics before updating the topic statistics. We found empirically that one such burn-in pass was sufficient in all of the datasets we tried in our experiments. Pseudo-code for the algorithm, which we refer to as "Stochastic CVB0" (SCVB0) is given in Algorithm 2.

Algorithm 2 Stochastic CVB0

- Randomly initialize $\mathbf{N}^{\Phi}$, $\mathbf{N}^{\Theta}$; $\mathbf{N}^{Z} := \sum_w \mathbf{N}^{\Phi}_{w}$
- For each minibatch M
  - $\hat{\mathbf{N}}^{\Phi} := \mathbf{0}$; $\hat{\mathbf{N}}^{Z} := \mathbf{0}$
  - For each document j in M
    - For zero or more "burn-in" passes
      - For each token i
        - Update γ_{ij} (Equation 5)
        - Update $\mathbf{N}^{\Theta}_{j}$ (Equation 6)
    - For each token i
      - Update γ_{ij} (Equation 5)
      - Update $\mathbf{N}^{\Theta}_{j}$ (Equation 6)
      - $\hat{\mathbf{N}}_{w_{ij}} := \hat{\mathbf{N}}_{w_{ij}} + C\gamma_{ij}$
      - $\hat{\mathbf{N}}^{Z} := \hat{\mathbf{N}}^{Z} + C\gamma_{ij}$
  - Update $\mathbf{N}^{\Phi}$ (Equation 7)
  - Update $\mathbf{N}^{Z}$ (Equation 8)

An optional additional optimization of the above algorithm is to only perform one update for each distinct token in each document, and scale the update by the number of copies in the document. This process, often called "clumping," is standard practice for fast implementations of all LDA inference algorithms (e.g. see Teh et al. [24] and Jonathan Chang's R package for LDA[2]), though it is only exact for uncollapsed algorithms, where the z_{ij}'s are D-separated by θ_j. Suppose we have observed w_{aj}, which occurs m_{aj} times in document j. Plugging Equation 6 into itself m_{aj} times and noticing that all but one of the resulting terms form a geometric series, we can see that performing m_{aj} updates for $\mathbf{N}^{\Theta}_{j}$ while holding γ_{aj} fixed is equivalent to

$$\mathbf{N}^{\Theta}_{j} := (1-\rho^{\Theta}_{t})^{m_{aj}}\mathbf{N}^{\Theta}_{j} + C_j\gamma_{aj}(1-(1-\rho^{\Theta}_{t})^{m_{aj}}) \ . \quad (9)$$

4. EXPERIMENTS

This section describes an experimental analysis of the proposed SCVB0 algorithm, with direct comparison to the stochastic variational Bayes algorithm of Hoffman et al., hereafter referred to as SVB. As well as performing an analysis on several large-scale problems, we also investigate the effectiveness of the stochastic LDA inference algorithms at learning topics in near real-time on small corpora.

4.1 Large-Scale Experiments

We studied the performance of the algorithms on three large corpora. The corpora are:

- *PubMed Central*: A corpus of full-text scientific articles from the open-access PubMed Central database of scientific literature in the biomedical and life sciences.[3] After processing to remove stopwords and words occurring less than 300 times, the corpus contained approximated 320M tokens across 165,000 articles, with a vocabulary size of around 38,500 words.

- *New York Times*: A corpus containing 1.8 million articles from the New York Times, published between 1987 and 2007. After removing stopwords and words occurring less than 500 times, the corpus had a dictionary of about 50,000 words and contained 475M distinct tokens.

- *Wikipedia*: This collection contains 4.6 million articles from the online encyclopedia Wikipedia. We used the dictionary of 7,700 words extracted by Hoffman et al. for their experiments on an earlier extracted Wikipedia corpus. There were 811M tokens in the corpus.

We explored predictive performance versus wall-clock time for both SCVB0 and SVB. To compare the algorithms fairly, we implemented both of them in the fast high-level language Julia [6]. Our implementation of SVB closely follows the python implementation provided by Hoffman, and has several optimizations not mentioned in the original paper including handling the latent topic assignments z implicitly, "clumping" of like tokens, and sparse updates of the topic matrix. The SCVB0 algorithm was implemented as it is written in Algorithm 2, using the clumping optimization but with no additional algorithmic optimizations. Specifically, neither implementation used the complicated optimizations taking advantage of sparsity that are exploited by the Vowpal Wabbit implementation of SVB[4] and in the variant of

[2] http://cran.r-project.org/web/packages/lda/
[3] http://www.ncbi.nlm.nih.gov/pmc/tools/ftp/
[4] https://github.com/JohnLangford/vowpal_wabbit/wiki

SVB proposed by Mimno [16]. Instead, our implementations represent a "best-effort" attempt to implement each algorithm efficiently yet following the spirit of the original pseudo-code.

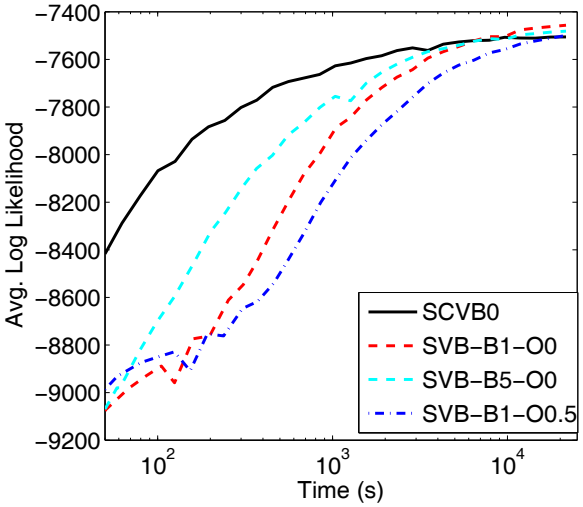
In all experiments, each algorithm was trained using mini-batches of size 100. We used a step-size schedule of $\frac{s}{(\tau+t)^\kappa}$ for document iteration t, with $s = 10$, $\tau = 1000$ and $\kappa = 0.9$. For SCVB0, the document parameters were updated using the same schedule with $s = 1$, $\tau = 10$ and $\kappa = 0.9$, with t referring to the word iteration of the current document. We used LDA hyper-parameters $\alpha = 0.1$ and $\eta = 0.01$ for SCVB0. For SVB, we tried both these same hyperparameter values as well as shifting by 0.5 as recommended by [3] to compensate for the implicit bias in how uncollapsed VB treats hyper-parameters. We used a single pass to learn document parameters for SCVB0, and tried both a single pass and five passes for SVB.

For each experiment we held out 10,000 documents and trained on the remaining documents. We split each test document in half, estimated document parameters on one half and computed the log-probability of the remaining half of the document. Figures 1(a) through 1(c) show held-out log-likelihood versus wall-clock time for each algorithm. In the figures, SVB-Bx-Oy corresponds to running SVB with x "burn-in" passes per document and with hyper-parameters offset from $\alpha = 0.1$ and $\eta = 0.01$ by y.

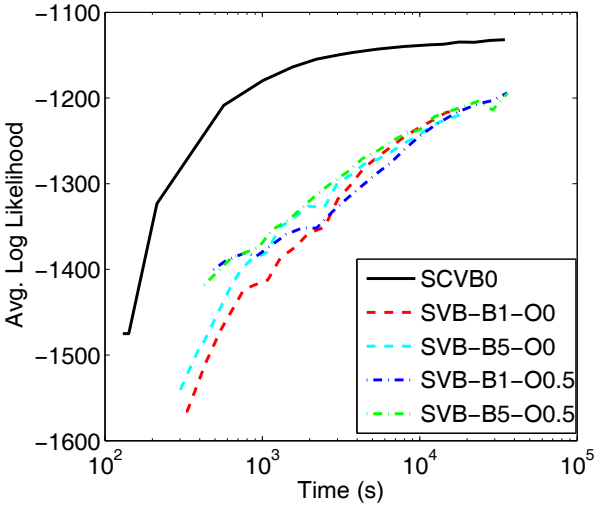
For the PubMed Central data, we found that all algorithms perform similarly after about an hour, but prior to that SCVB0 is better, indicating that SCVB0 makes better use of its time. All algorithms perform similarly per-iteration (see Figure 2), but SCVB0 is able to benefit by processing more documents in the same amount of time. The per-iteration plots for the other datasets were similar.

Our experiments show that SCVB0 shows a more substantial benefit when employed on larger datasets. For both the New York Times and Wikipedia datasets (which are each significantly larger than the PubMed Central dataset in terms of the number of documents), SCVB0 converged to a better solution than SVB for any of its parameter settings. Furthermore, SCVB0 outperforms SVB throughout the run. The superior performance of SCVB0 over the uncollapsed SVB method is consistent with the fact that the variational bound for the collapsed representation is strictly better than the bound for the uncollapsed representation of LDA [24].

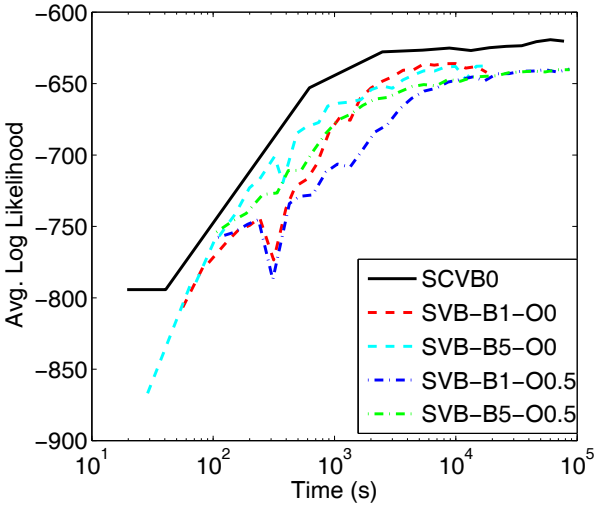
For completeness, we also compared SCVB0 to the batch VB algorithm on the Wikipedia dataset (Figure 3); other standard batch algorithms such as Gibbs sampling tend to perform similarly to VB at convergence, particularly if the hyper-parameters are learned for each algorithm [3]. Note that it was not possible to perform even a single iteration of batch VB on the full dataset in the allotted time of twelve hours. Following Hoffman et al., we show instead the performance of the algorithms on subsets of the data. This facilitates faster convergence, but reduces the quality of the final solution as the algorithms are consequently unable to exploit all of the data. In contrast, the stochastic algorithms are able to make use of large datasets while still converging quickly.



(a) Log-likelihood vs Time for PubMed Central.



(b) Log-likelihood vs Time for New York Times.



(c) Log-likelihood vs Time for Wikipedia.

Figure 1: Log-likelihood vs Time experiments

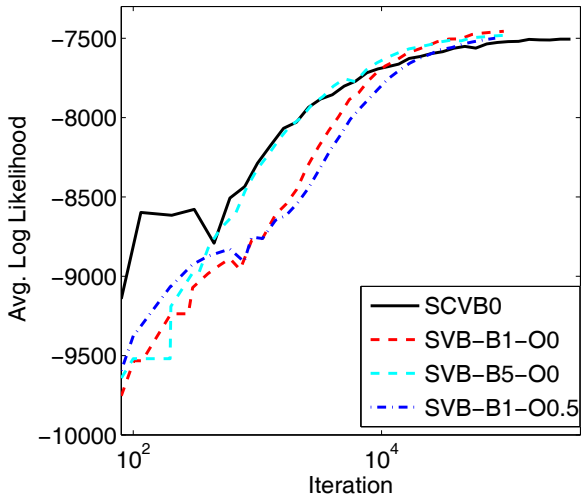


Figure 2: Log-likelihood vs Iteration for the PubMed Central experiments.

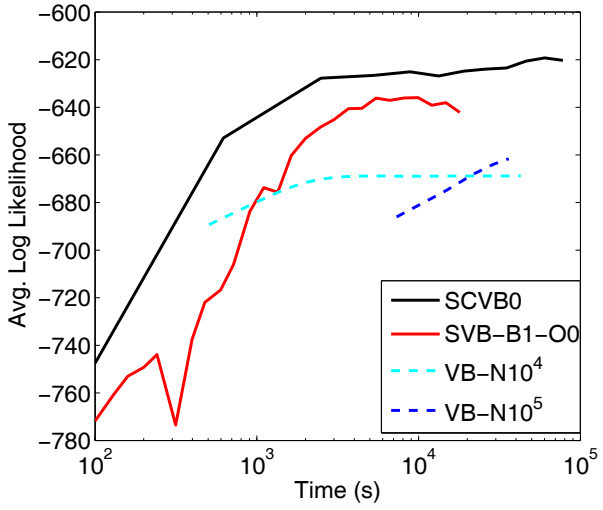


Figure 3: Log-likelihood vs Iteration compared to batch VB for the Wikipedia experiments, where N is the number of documents used for training.

4.2 Small-Scale Experiments

Stochastic algorithms for LDA have previously only been used on large corpora, however they have the potential to be useful for finding topics very quickly on small corpora as well. The ability to learn interpretable topics in a matter of seconds is very beneficial for exploratory data analysis (EDA) applications, with a human in the loop. Near real-time topic modeling opens the way for the use of topic models in interactive software tools for document analysis.

We investigated the performance of the stochastic algorithms in this small-scale scenario using a corpus of 1740 scientific articles from years 1987 – 1999 of the machine learning conference NIPS. We ran the two stochastic inference algorithms for five seconds each, using the parameter settings from the previous experiments but with 20 topics. Each algorithm was run ten times. In the five seconds of training, SCVB0 was typically able to examine 3300 documents, while SVB was typically able to examine around 600 documents.

With the EDA application in mind, we performed a human-subject experiment in the vein of the experiments proposed by Chang and Blei [11]. The sets of topics returned by each run were randomly assigned across seven human subjects. The participants were all machine learning researchers with technical expertise in the subjects of interest to the NIPS community. The subjects did not know which algorithms generated which runs. The top ten words of the topics in each run were shown to the subjects, who were given the following instructions:

> Here are 20 collections of related words. Some words may not seem to "belong" with the other words. Count the total number of words in each collection that don't "belong."

The results provide an estimate of the number of "errors" that a topic model inference algorithm makes, relative to human judgement. It was found that the SCVB0 algorithm had 0.76 errors per topic on average, with a standard deviation of 1.1, while SVB had 1.6 errors per topic on average, with standard deviation 1.2. A one-sided two sample t-test rejected the hypothesis that the means of the errors per topic were equal, with significance level $\alpha = 0.05$. Randomly selected example topics are shown in Table 2. As can be seen from the table, both algorithms successfully learned coherent topics in this relatively short time frame.

We also performed a similar experiment on Amazon Turk using the New York Times corpus. We ran the two stochastic inference algorithms for 60 seconds each using the same parameter settings as above but with 50 topics. Each user was presented with 20 random topics from each algorithm. Again, the subjects did not know which algorithms generated each set of topics. We included two easy questions with obvious answers and removed results from users who did not answer them correctly. This step eliminated 4 users, and the analysis was performed with the data from the remaining 52 participants. Comparing the number of "errors" for SCVB0 to SVB for each user, we find that SCVB0 had 2.1 errors per topic on average, with standard deviation 1.0, and SVB had 4.4 errors on average with standard deviation 2.4. A paired t-test finds these differences significant for the sampled population at the $\alpha = .05$ level, with p-value $< .001$. Example topics selected uniformly at random from a randomly selected run of each algorithm are shown in Table 3, illustrating the relative difference in the coherence of the topics recovered by the two methods in this time period.

5. CONVERGENCE ANALYSIS AND CONNECTIONS TO MAP ESTIMATION

In the SCVB0 algorithm, because the γ's are not maintained we must approximate Equation 3 with Equation 5, neglecting the subtraction of the previous value of γ_{ij} from the CVB0 statistics when updating γ_{ij}. In an extended version of this paper, available on the arXiv,[5], we show that this approximation results in an algorithm which is equivalent to an EM algorithm for MAP estimation, due to Asuncion et

[5] `http://arxiv.org/abs/1305.2452`.

SCVB0			SVB		
receptor	data	learning	model	results	visual
protein	classification	function	set	learning	data
secondary	vector	network	data	distribution	activity
proteins	class	neural	training	information	saliency
transducer	classifier	networks	learning	map	noise
binding	set	time	error	activity	similarity
concentration	algorithm	order	parameters	time	model
odor	feature	error	markov	figure	neural
morphology	space	dynamics	estimate	networks	representations
junction	vectors	point	speech	state	functions

Table 2: Randomly selected example topics after five seconds running time on the NIPS corpus.

SCVB			SVB		
county	station	league	president	year	mr
district	company	goals	midshipmen	cantatas	company
village	railway	years	open	edward	mep
north	business	club	forrester	computing	husbands
river	services	clubs	archives	main	net
area	market	season	iraq	years	state
east	line	played	left	area	builder
town	industry	cup	back	withdraw	offense
lake	stations	career	times	households	obscure
west	owned	team	saving	brain	advocacy

Table 3: Randomly selected example topics after sixty seconds running time on the NYT corpus.

al. [3], which operates on an unnormalized parameterization of LDA. Using this interpretation of the algorithm, we can alternatively derive SCVB0 as an adapted version of Cappe and Moulines' online EM algorithm [9], where the algorithm is extended to perform MAP estimation and to handle document-specific parameters. Therefore, the approximate collapsed variational updates of SCVB0 can also be understood as MAP estimation updates. The MAP interpretation of the algorithm implicitly uses adjusted values of the hyperparameters, so this does not contradict the original CVB interpretation, but suggests that there is a close relationship between the optimal solutions of the CVB and MAP estimation problems.

It is difficult to establish the convergence properties of the original CVB0 algorithm, as its updates are approximate. However, the alternative view of the algorithm is more amenable to convergence analysis as the MAP updates are exact. Under the MAP estimation interpretation of SCVB0, it can be shown that the algorithm converges to a stationary point of the MAP objective function, computed as if the prior were modified by increasing the hyper-parameters by one. The proof strategy broadly follows that of Cappe and Moulines. First, the algorithm is written as a Robbins and Monro [20] stochastic approximation (SA) algorithm. Then, it is shown that there exists a Lyapunov function satisfying the conditions of Andreiu et al. [2], which are sufficient to establish convergence for an SA algorithm. In the context of an SA algorithm, a Lyapunov function can be understood as an "objective function" which, in the absence of stochastic noise, the SA would improve monotonically if small enough steps were taken in the direction of the updates. We refer the reader to the extended version of this paper for details.[5]

The MAP estimation interpretation of SCVB0 may also help to explain the improvement in predictive performance relative to SVB. The MAP estimate approximates the posterior distribution by a delta function at its mode, while mean field variational Bayes approximates the posterior by a factorized distribution. As the amount of training data increases, the posterior distribution should become more peaked around the mode, i.e. more similar to the delta function at the MAP. The factorized distribution of mean field, on the other hand, may not be able to accurately represent the posterior distribution in the large data regime. So we conjecture that in many cases, given enough data it may be preferable to perform MAP estimation instead of variational inference. This observation seems particularly relevant in the case where stochastic algorithms are necessary due to the large amount of data available.

6. DISCUSSION / RELATED WORK

Connections can be drawn between SCVB0 and other methods in the literature. The SCVB0 scheme is reminiscent of the online EM algorithm of Cappe and Moulines [9], which also alternates between per data-point parameter updates and online estimates of the expected values of sufficient statistics. Online EM optimizes the EM lower bound on the log-likelihood in the M-step and computes online averages of exponential family sufficient statistics, while SCVB0 (approximately) updates the mean-field evidence lower bound in the M-step and computes online averages of sufficient statistics required for a CVB0 update in the E-step. As discussed in the previous section, when viewed as a MAP estimation algorithm SCVB0 can also be derived as an extension of online EM, applied to LDA.

The SCVB0 algorithm also has a very similar structure to SVB, alternating between passes through a document (the optional "burn-in" passes) to learn document parameters, and updating variables associated with topics. However, SCVB0 is stochastic at the word-level while SVB is stochastic at the document level. In the general framework of Hoffman et al., inference is performed on "local" parameters specific to a data point, which are used to perform a stochastic update on the "global" parameters. For SVB, the document parameters Θ_j are local parameters for document j, and topics are global parameters. For SCVB0, the γ_{ij}'s are local parameters for a word, and both document parameters N^{Θ} and topic parameters N^{Φ} are global parameters. This means that updates to document parameters can be made before processing all of the words in the document.

The incremental algorithm of Banerjee and Basu [5], for MAP inference in LDA, is also closely related to the proposed algorithm. They estimate topic probabilities for each word sequentially, and update MAP estimates of Φ and Θ incrementally, using the expected assignments of words to topics in the current document. SCVB0 can be understood as the collapsed, stochastic variational version of Banerjee and Basu's incremental uncollapsed MAP estimation algorithm. Interpreting SCVB0 as a MAP estimation algorithm, SCVB0 is the online EM algorithm for MAP estimation operating on the unnormalized representation of LDA, while Banerjee and Basu's algorithm is the incremental EM algorithm operating on the usual normalized representation of LDA. A related algorithm is the sequential Monte Carlo (SMC) approach used by Ahmed et al. [1], which sequentially Gibbs samples the topic assignments of each document for each of F importance-weighted particles. This method updates count statistics for each particle incrementally via sampling, while SCVB0 updates count statistics with online-averaged updates via optimization.

Another stochastic algorithm for LDA, due to Mimno et al. [16], operates in a partially collapsed space, placing it in-between SVB and SCVB0 in terms of representation. Their algorithm collapses out $\boldsymbol{\Theta}$ but does not collapse out $\boldsymbol{\Phi}$. Estimates of online natural gradient update directions are computed by performing Gibbs sampling on the topic assignments of the words in each document, and averaging over the samples. The gradient estimate is non-zero only for word-topic pairs which occurred in the samples. When carefully implemented to take advantage of the sparsity, the updates scale sub-linearly in the number of topics, causing large improvements in high-dimensional regimes. For SCVB0, the minibatch updates are sparse in the rows (words), so some performance enhancements along the lines of those used by Mimno et al. are likely to be possible.

There has been a substantial amount of other work on speeding up LDA inference in the literature. Porteous et al. [19] improved the efficiency of the sampling step for the collapsed Gibbs sampler, and Yao, Mimno and McCallum [26] explore a number of alternatives for improving the efficiency of LDA. The Vowpal Wabbit system for fast machine learning,[4] due to John Langford and collaborators, has a version of SVB that has been engineered to be extremely efficient. Parallelization is another approach for improving the efficiency of topic models. Newman et al. [18] introduced an approximate parallel algorithm for LDA where data is distributed across multiple machines, and an exact algorithm for an extension of LDA which takes into account the distributed storage. Smola and Narayanamurthy developed an efficient architecture for parallel LDA inference [23], using a distributed (key, value) storage for synchronizing the state of the sampler between machines. All of these computational improvements are somewhat orthogonal to those proposed in this paper, and it is likely that some of these ideas could be adapted to apply to SCVB0 as well.

7. CONCLUSIONS

This paper introduces SCVB0, an algorithm for performing fast stochastic collapsed variational inference in LDA, and shows that it outperforms stochastic VB on several large document corpora, converging faster and often to a better solution. The algorithm is relatively simple to implement, with intuitive update rules consisting only of basic arithmetic operations. We also found that the algorithm was effective at learning good topics from small corpora in seconds, finding topics that were superior than those of stochastic VB according to human judgement.

There are many directions for future work. The method could potentially be adapted to Teh et al. [25]'s hierarchical Dirichlet process version of LDA, leveraging the work of Sato et al. [21]. The speed of the method could likely be improved by exploiting sparsity, using techniques such as those employed by Mimno et al. [16]. Furthermore, the collapsed representation facilitates the use of the parallelization techniques explored by Newman et al. in [18]. Finally, SCVB0 could be incorporated into an interactive software tool for exploring the topics of document corpora in real-time.

8. ACKNOWLEDGMENTS

JF, CD, and PS were partially supported by the Office of Naval Research under MURI grant N00014-08-1-1015 and by a Google Faculty Research Award. LB was supported by the National Science Foundation under Grant No. 0914783, 0928427, 1018433, 1216045. The authors would also like to thank Arthur Asuncion for many helpful discussions.

9. REFERENCES

[1] A. Ahmed, Q. Ho, C. H. Teo, J. Eisenstein, A. Smola, and E. Xing. Online inference for the infinite topic-cluster model: Storylines from streaming text. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS 2011), Journal of Machine Learning Research Workshop and Conference Proceedings*, volume 15, pages 101–109, 2011.

[2] C. Andrieu, É. Moulines, and P. Priouret. Stability of stochastic approximation under verifiable conditions. *SIAM Journal on control and optimization*, 44(1):283–312, 2005.

[3] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh. On smoothing and inference for topic models. In *Proceedings of the Twenty-Fifth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-09)*, pages 27–34, Corvallis, Oregon, 2009. AUAI Press.

[4] D. C. Atkins, T. N. Rubin, M. Steyvers, M. A. Doeden, B. R. Baucom, and A. Christensen. Topic models: A novel method for modeling couple and family text data. *Journal of Family Psychology*, 6:816–827, 2012.

[5] A. Banerjee and S. Basu. Topic models over text streams: A study of batch and online unsupervised learning. In *SIAM Data Mining*, 2007.

[6] J. Bezanson, S. Karpinski, V. B. Shah, and A. Edelman. Julia: A fast dynamic language for technical computing. *Computing Research Repository*, abs/1209.5145, 2012.

[7] D. Blei, A. Ng, and M. Jordan. Latent Dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.

[8] S. Block and D. Newman. What, where, when and sometimes why: Data mining twenty years of women's history abstracts. *Journal of Women's History*, 23(1):81–109, 2011.

[9] O. Cappé and E. Moulines. On-line expectation–maximization algorithm for latent data models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(3):593–613, 2009.

[10] B. Carpenter. Integrating out multinomial parameters in latent Dirichlet allocation and naive Bayes for collapsed Gibbs sampling. Technical report, LingPipe, 2010.

[11] J. Chang, J. Boyd-Graber, S. Gerrish, C. Wang, and D. Blei. Reading tea leaves: How humans interpret topic models. In *Advances in Neural Information Processing Systems 22*, pages 288–296, 2009.

[12] T. Griffiths and M. Steyvers. Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America*, 101(Suppl 1):5228, 2004.

[13] M. Hoffman, D. Blei, and F. Bach. Online learning for latent Dirichlet allocation. In *Advances in Neural Information Processing Systems 23*, pages 856–864, 2010.

[14] M. Hoffman, D. Blei, C. Wang, and J. Paisley. Stochastic variational inference. *Journal of Machine Learning Research (to appear, see also arXiv preprint arXiv:1206.7051)*, 2013.

[15] D. Mimno. Computational historiography: Data mining in a century of classics journals. *Journal on Computing and Cultural Heritage (JOCCH)*, 5(1):3, 2012.

[16] D. Mimno, M. Hoffman, and D. Blei. Sparse stochastic inference for latent Dirichlet allocation. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 1599–1606, New York, NY, USA, July 2012. Omnipress.

[17] T. Minka. Power EP. Technical Report MSR-TR-2004-149, Microsoft Research, Cambridge, UK, 2004.

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

[19] I. Porteous, D. Newman, A. Ihler, A. Asuncion, P. Smyth, and M. Welling. Fast collapsed Gibbs sampling for latent Dirichlet allocation. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 569–577, 2008.

[20] H. Robbins and S. Monro. A stochastic approximation method. *The Annals of Mathematical Statistics*, pages 400–407, 1951.

[21] I. Sato, K. Kurihara, and H. Nakagawa. Practical collapsed variational Bayes inference for hierarchical Dirichlet process. In *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 105–113. ACM, 2012.

[22] I. Sato and H. Nakagawa. Rethinking collapsed variational Bayes inference for LDA. In J. Langford and J. Pineau, editors, *Proceedings of the 29th International Conference on Machine Learning (ICML-12)*, pages 999–1006, New York, NY, USA, July 2012. Omnipress.

[23] A. Smola and S. Narayanamurthy. An architecture for parallel topic models. *Proceedings of the VLDB Endowment, 36th International Conference on Very Large Data Bases*, 3(1-2):703–710, 2010.

[24] Y. Teh, D. Newman, and M. Welling. A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. *Advances in Neural Information Processing Systems*, 19:1353, 2007.

[25] Y. W. Teh, M. I. Jordan, M. J. Beal, and D. M. Blei. Hierarchical Dirichlet processes. *Journal of the American Statistical Association*, 101(476), 2006.

[26] L. Yao, D. Mimno, and A. McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 937–946. ACM, 2009.