Sub-Gibbs Sampling: a New Strategy for Inferring LDA

Chuan Hu, Huiping Cao, Qixu Gong

***Abstract*—Latent Dirichlet Allocation (LDA) has been widely used in text mining to discover topics from documents. One major approach to learn LDA is Gibbs sampling. The basic Collapsed Gibbs Sampling (CGS) algorithm requires $O(NZ)$ computations to learn an LDA model with Z topics from a corpus containing N tokens. Existing approaches that improve the complexity of CGS focus on reducing the factor Z.**

In this work, we propose a novel and general *Sub-Gibbs Sampling* (SGS) strategy to improve the Gibbs-Sampling computation by reducing the sample space. This new strategy targets at reducing the factor N by sampling only a subset of the whole corpus. The design of the SGS strategy is based on two properties that we observe: (i) topic distributions of tokens are skewed and (ii) a subset of documents can approximately represent the semantics of the whole corpus. We prove that the SGS strategy can achieve comparable effectiveness (with bounded errors) and significantly reduce the complexity of existing Gibbs sampling algorithms. Extensive experiments on large real-world data sets show that the proposed SGS strategy is much faster than several state-of-the-art fast Gibbs sampling algorithms and the proposed SGS strategy can learn comparable LDA models as other Gibbs sampling algorithms.

***Keywords*-Topic Models, Gibbs Sampling, Sub-sampling**

I. Introduction and Related Works

Latent Dirichlet Allocation (LDA) is a widely used topic model ever since its introduction by Blei et al. [1]. A basic Collapsed Gibbs Sampling (CGS) algorithm to infer the LDA model is introduced by Griffiths et al. [2]. The complexity of this CGS is $O(NZ)$ where N and Z are the total number of observed tokens and the number of latent topics in a text corpus respectively. This $O(NZ)$ complexity makes CGS very expensive to run on large corpora.

To improve the basic CGS algorithm, many fast Gibbs sampling algorithms (e.g., [3], [4], [5], [6], [7]) have been proposed in the literature. Porteous et al. [3] propose the *FastLDA* algorithm to improve the running time (not complexity) of CGS by segmenting and rearranging the conditional probabilities. Yao et al. [4] design the *SparseLDA* algorithm and reduce the complexity to $O(N(Z_w + Z_d))$ where Z_w is the number of distinct topics that are assigned to a token w, Z_d is the number of distinct topics that are assigned to w's corresponding document d, and $Z_w + Z_d$ is generally smaller than Z in practice. *SparseLDA* utilizes an observation that word-topic and document-topic count matrices in Gibbs sampling are sparse. Li et al. [5] design an $O(NZ_d)$ approximate sampling algorithm, *AliasLDA*, by combining the Metropolis-Hasting sampling and the alias table method [8]. Yuan et al. [7] further improve *AliasLDA* by approximating more probability components and propose the *LightLDA* algorithm with an $O(N)$ complexity. Yu et al. [6] design the *F+Nomad LDA* Gibbs sampling algorithm with an $O(N \log Z)$ time complexity by utilizing the *Fenwick tree* [9] data structure. All these works focus on reducing the factor Z through efficient calculation and maintenance of conditional probabilities.

In this paper we propose a totally new and general strategy, *Sub-Gibbs Sampling (SGS)*, to improve Gibbs sampling algorithms for LDA inference by reducing the sample space. I.e, we target at reducing the factor N. SGS is based on two properties which are not utilized in previous works.

- *Skewed topic distributions.* In LDA models, multiple occurrences of one token in a document are assigned with a few distinct topics, and the majority of the occurrences is assigned with the same topic.
- *Approximate semantics.* The semantics of a corpus can be approximately represented by a small subset of the documents. The documents in this representative subset are called *covered-documents* and the remaining documents are called *uncovered-documents*.

These two properties are presented in details in Section III. Utilizing these two properties SGS is designed to run any Gibbs sampling algorithm (i) on a small subset of tokens (instead of *all* the tokens); (ii) on covered-documents for more iterations than on uncovered-documents.

The contributions of this work are as follows.

- We identify two new useful properties in LDA, skewed topic distributions and approximate semantics.
- Utilizing the two properties, we propose a novel Sub-Gibbs Sampling (SGS) strategy to reduce the sampling space of existing Gibbs sampling algorithms.
- We conduct extensive experiments to demonstrate that SGS strategy reduces the running time significantly and achieves similar effectiveness.

II. Background and Notations

Latent Dirichlet Allocation (LDA) is introduced in [1] as a generative probabilistic model to learn topics in text corpora. LDA assumes that the observed tokens in each document are generated from a mixture (document-to-topic distributions θ) of several multinomial distributions (topic-to-word distributions ϕ). The detailed generative process of

C. Hu, H. Cao, Q. Gong are with the Department of Computer Science, New Mexico State University, Las Cruces, NM 88003.
E-mail: chu, hcao, qgong@cs.nmsu.edu
This work is supported by NSF #1633330 and #1345232.

2374-8486/17 $31.00 © 2017 IEEE
DOI 10.1109/ICDM.2017.113

LDA is in Fig. 1. The Collapsed Gibbs Sampling (CGS) algorithm [2] is a widely accepted approach to learn the parameter values in LDA. CGS estimates latent topics for tokens by iteratively drawing samples for each token from conditional probabilities. The CGS algorithm is shown in Fig. 2. The notations of LDA and CGS are listed in TABLE I.

Symbol	Meaning
α, β	Hyper-parameters of Dirichlet distribution
$\vec{\theta}_i$	Multinomial topic distribution of document d_i
$\vec{\phi}_i$	Multinomial word distribution of topic z_i
D	# of documents
W	# of distinct tokens (vocabulary size)
N	Total # of tokens in the corpus
Z	# of latent topics
d_i	The ith document in the corpus
l_i	The length of the document d_i
w_{ij}	The jth token in the document d_i
z_{ij}	The latent topic of the token w_{ij}
n_{dz}	# of times that topic z is assigned to document d
n_{dw}	# of times that token w occurs document d
n_d	Total # of tokens in document d
n_{wz}	# of times that topic z is assigned to token w
n_z	# of times that topic z is assigned for the corpus

Table I
NOTATIONS FOR LDA AND CGS

1) For each latent topic z_i, $i \in \{1, 2, \ldots, Z\}$, draw $\vec{\phi}_i \sim Dirichlet(\beta)$
2) For each document d_i, $i \in \{1, 2, \ldots, D\}$
   a) Draw $\vec{\theta}_i \sim Dirichlet(\alpha)$
   b) For $j \in \{1, 2, \ldots l_i\}$, where l_i is the length of document d_i
      i) Draw the latent topic $z_{ij} \sim multinomial(\vec{\theta}_i)$
      ii) Draw the observed token $w_{ij} \sim multinomial(\vec{\phi}_{z_{ij}})$

Figure 1. Generative Process of LDA [1]

1) For each iteration $i = 1, 2, \cdots$
   a) For each token w in each document d
      i) Let z be the topic assignment to w in the previous iteration
      ii) Decrement n_{dz}, n_d, n_{wz}, n_z by one
      iii) Sample a new topic z for w from a multinomial distribution $p(z = k|\neg z),\ k \in \{1, 2, \ldots, Z\}$

$$p(z = k|\neg z) \propto \frac{n_{wk} + \beta}{n_k + W\beta} \cdot \frac{n_{dk} + \alpha}{n_d + Z\alpha} \quad (1)$$

      iv) Increment n_{dz}, n_d, n_{wz}, n_z by one

Figure 2. Collapsed Gibbs Sampling (CGS) algorithm to learn LDA [10]

III. SUB-GIBBS SAMPLING STRATEGY

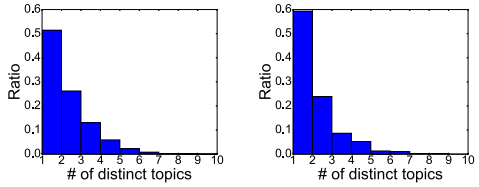
This section presents the strategy of ***Sub-Gibbs Sampling (SGS)***. SGS utilizes two properties in LDA models to reduce the sample space: (i) tokens have very skewed topic distribution patterns, and (ii) the semantics of a corpus can be approximated using a subset of documents. We denote these two properties as ***Skewed Topic Distributions*** and ***Approximate Semantics***. The idea of SGS is using a subset of tokens in a document and a subset of documents in a corpus to learn LDA models.

A. SGS Utilizing Skewed Topic Distributions

We conduct a pre-study to examine the topic distributions of tokens. In this pre-study, we run the CGS algorithm on the *NYTimes* data set. After CGS terminates, we group tokens by their *term frequency* (tf) and examine the topic distribution of tokens. Let the group of tokens with $tf = m$ be denoted as G_m. I.e., $G_m = \{w_{ij} | i \in 1 \ldots D,\ j \in 1 \ldots l_i,\ n_{d_i w_{ij}} = m\}$. The topic distribution of tokens in the token group G_m is the proportion of tokens that are assigned with $Z'(\leq Z)$ distinct topics, which is calculated as

$$r(Z', m) = \frac{\left|\left\{w_{ij} \middle| w_{ij} \in G_m,\ |\{z_{ij'} | j' \in 1 \ldots l_i,\ w_{ij'} = w_{ij}\}| = Z'\right\}\right|}{|G_m|}.$$

We plot the topic distributions of tokens with different term frequencies for several m values in Fig. 3.



$tf = 20$ $tf = 40$

Figure 3. Topic distribution patterns learned by CGS for several token groups on NYTimes data set with $Z = 100$; $tf = X$ represents G_X

We can observe (Fig. 3) that multiple occurrences ($tf = m$) of the same token in a document are only assigned with a few distinct (much fewer than $\min(m, Z)$) topics.We denote this observation as the property of ***skewed topic distributions***. Previous works [5], [4], [7] utilize the sparsity in parameters n_{wz} and n_{dz} to design fast Gibbs sampling algorithms. This parameter-sparsity property denotes that as the sampling process proceeds most elements in n_{wz}, n_{dz} become zero. Our skewed topic distributions property is very different. We study the effect of the sparsity on the distributions of the number of distinct topics.

Input: base Gibbs sampling algorithm for LDA (*GS*), group partition algorithm (*GP*)
1) For each iteration $i = 1, 2, \cdots$
   a) For each *distinct* token w in each document d
      i) Partition m occurrences of w into groups $g_1, g_2, \ldots, g_s$ of size $m_1, m_2, \ldots, m_s$ ($\sum_{i=1}^{s} m_i = m$) utilizing *GP*
      ii) For each group g_i for token w
         A) Let z be the topic assignment to g_i in the previous iteration
         B) Decrement n_{dz}, n_d, n_{wz}, and n_z by m_i
         C) Sample z for g_i using *GS*
         D) Increment n_{dz}, n_d, n_{wz}, n_z by m_i

Figure 4. SGS utilizing skewed topic distributions

The property of skewed topic distributions suggests that, when running CGS to learn LDA, it is unnecessary to sample each individual occurrence of a token. Instead, we can draw one topic and assign this topic to several occurrences of a token. We propose the SGS strategy by utilizing the skewed topic distribution property in Fig. 4. The SGS strategy takes as input (a) a base Gibbs sampling algorithm for LDA (GS), which can be any LDA Gibbs Sampling algorithm, and (b) a group partition algorithm (GP), which is explained in Section III-A1. It works in two steps. The first step partitions the multiple occurrences of token w, whose term frequency is m, into s ($s \leq m$) groups $g_1, g_2, \ldots, g_s$ of

size $m_1, m_2, \ldots, m_s$ ($\sum_{i=1}^{s} m_i = m$) by utilizing GP. The second step samples **one** representative topic for each group g_i by calling GS. These two steps are repeated on all the distinct tokens in each document. This strategy can reduce the time complexity of sampling m occurrences of a token from $O(Zm)$ to $O(Zs)$ ($s < m$) if CGS is taken as the base GS. We denote a base GS algorithm implementing the SGS strategy as an **SGS algorithm** in later discussions.

1) Group Partition Algorithm: We design a uniform size group partition (USGP) algorithm. In USGP, a subsampling ration $q \in [0, 1.0]$ is defined to control the group size. Given q, the m occurrences of a token is partitioned into s groups. The group sizes are $\lceil mq \rceil, \lceil mq \rceil, \ldots, m - (s-1)\lceil mq \rceil$. If $\lceil mq \rceil < 1$, we put all m occurrences into one group. Even USGP generates uniform size groups, the LDA model learned by SGS algorithms that utilize USGP still shows skewed topic distribution patterns. This has been verified with detailed experimental results in our technical report [11].

2) Error Bound Analysis: The sampling process of an SGS algorithm and its corresponding base GS algorithm (as shown in Fig. 2 and Fig. 4) are different. It has shown [12] that it is *intractable* to derive their accumulated difference through the *whole* sampling process. Nevertheless, to get a basic idea of the error that the subsampling brings, we analyze the difference between an SGS algorithm and its corresponding base GS algorithm in the sampling process for one token. We define $\delta_{dw} = ||p - p'||_1$, in which p' and p are conditional probabilities (Eq. (1)) of an SGS algorithm and its corresponding GS respectively. We prove that, given the same count matrices (n_{dz}, n_{wz}, and n_z), δ_{dw} has a limited upper bound $\delta_{dw} < (1 + a)\epsilon$ which is close to 0. The detailed proof and analysis are shown in [11].

B. SGS Utilizing Approximate Semantics

To further reduce N we propose the SGS utilizing approximate semantics which is described in the remaining of this section. The document-word representation of a corpus can be viewed as a bipartite graph. Fig. 5 shows an example of a small corpus. Let us use **semantics** to denote the set of distinct tokens (vocabulary) in the corpus. The idea of approximate semantics is inspired by the observation that a few documents can cover the overall semantics of a corpus.

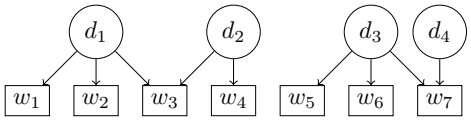


Figure 5. Bipartite representation of a small corpus

A natural question is which subset we should choose to approximately represent the semantics of the corpus. We formulate the problem of choosing the representative document subset from a corpus as a set cover problem on a bipartite graph. The document-word representation of a corpus is formulated as a bipartite graph $G = (V, E)$. The graph has two types of nodes $V = \mathcal{D} \cup \mathcal{W}$, in which $\mathcal{D} = \{d_1, d_2, \cdots, d_D\}$ is the set of documents and $\mathcal{W} = \{w_1, w_2, \cdots, w_W\}$ is the set of distinct tokens. A graph edge $(d_i, w_j) \in E$ indicates that document d_i contains token w_j. To find a document subset to represent the semantics of the corpus, we adopt the greedy set cover algorithm [13] to find a subset of documents that contains all the distinct tokens in the corpus. Recall that the documents in the representative set are called covered-documents, and the remaining documents are called uncovered-documents.

Input: bipartite graph representation of a corpus $G = (V, E)$ in which $V = \mathcal{D} \cup \mathcal{W}$, base Gibbs sampling algorithm for LDA GS, subsampling ratio r

1) Use greedy set cover algorithm [13] to find a subset of documents S that covers the semantics of G
2) For each iteration $i = 1, 2, \cdots$
   a) Run GS on S
   b) If $i \mod (10 * r) = 0$, run GS on $\mathcal{D} - S$

Figure 6. SGS utilizing approximate semantics

The SGS strategy utilizing the approximate semantics property is shown in Fig. 6. For this strategy, we define a subsampling ratio $r \in [0, 1.0]$ to control the frequency that a GS algorithm runs on uncovered-documents. This strategy works as follows. It first uses a greedy set cover algorithm to find the representative subset S. It then runs a base Gibbs sampling algorithm GS on S in each iteration. If the iteration number i satisfies that $i \mod (10 * r) = 0$, GS also runs on uncovered-documents $\mathcal{D} - S$. We limit the subsampling ratio r in range $[0, 1.0]$ and set the consistent factor as 10 because out experiments show that when $10 * r > 10$ the effectiveness of SGS algorithms is unacceptable. Assuming that each document has L words on average and the base GS algorithm has complexity $O(Z_f)$ for sampling each token, then the complexity of the base GS algorithm is $O(DLZ_f)$ and the corresponding SGS algorithm has complexity $O((|S| + (D - |S|) * \frac{1}{10*r})LZ_f)$. Because S is a small subset of $\mathcal{D}$, $|S| + (D - |S|) * \frac{1}{10*r} \ll D$, which means that the complexity of the SGS algorithm is much lower than the base GS algorithm. The cost of greedy set cover algorithm is demonstrated in [11].

Data set	W	D	N
NIPS	12,419	1,500	746,316
ENRON	28,102	37,861	3,710,420
NYTimes	102,660	300,000	69,679,427
PubMed	141,043	8,200,000	483,450,157

Table II
STATISTICS OF DATA SETS

C. Discussions

We use ***sgs_GS***(q, r) to denote the SGS algorithm utilizing the skewed topic distribution property with subsampling ratio q and the approximate semantics property with subsampling ratio r. ***GS*** denotes the input base Gibbs Sampling algorithm. Parameters $q = 0$ or $r = 0$ means

that the corresponding property is not utilized. To achieve the maximum speedup, we set $q = 1.0$ and $r = 1.0$.

IV. Experiments

We implement the proposed SGS strategy on four Gibbs sampling algorithms, *CGS* [2], *SparseLDA* [4], *AliasLDA* [5], and *F+Nomad LDA* [6]. They are denoted as ***cgs***, ***splda***, ***alias***, and ***flda***. The Gibbs sampling algorithms implementing the SGS strategy are denoted as ***sgs_cgs***, ***sgs_splda***, ***sgs_alias***, and ***sgs_flda***, which are called ***sgs_GS*** or SGS algorithms in later discussions. All the implementations are based on the open-source code in [6]. Due to limit of space, we only present representative experimental results and their analysis. More detailed experimental analysis is shown in our technical report [11].

A. Data Sets

We conduct extensive experiments on four data sets, *NIPS, ENRON, NYTimes*, and *PubMed*, which are downloaded from UCI Bag of Words data collection [14]. The statistics of these data sets are shown in TABLE II.

B. Evaluation Measures and Parameter Settings

1. Log-likelihood (*llh*). We use the likelihood defined in [15] which is defined as follows.

$$p(w, z|\alpha, \beta) = \Big(\prod_{i=1}^{D} \frac{\prod_{j=1}^{Z} \Gamma(\alpha + n_{d_i z_j})}{\Gamma(Z\alpha + n_{d_i})}\Big) \cdot \Big(\prod_{i=1}^{Z} \frac{\prod_{j=1}^{W} \Gamma(\beta + n_{w_j z_i})}{\Gamma(W\beta + n_{z_i})}\Big)$$

We define log-likelihood ratio (*llh-ratio*) to measure the effectiveness difference between ***GS*** and ***sgs_GS***. The *llh-ratio* of ***sgs_GS*** over ***GS*** is defined as

$$\textit{llh-ratio}(sgs\_GS, GS) = abs\Big(\frac{llh(sgs\_GS) - llh(GS)}{llh(GS)}\Big)$$

Smaller *llh-ratio* indicates ***sgs_GS*** and ***GS*** have similar *llh*.

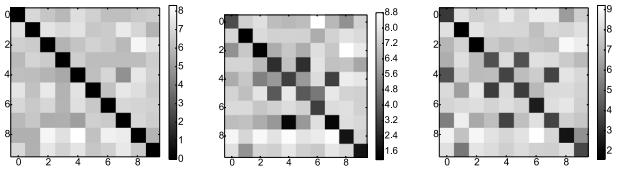
2. KL-divergence. To further examine whether ***sgs_GS*** can learn similar topics as what ***GS*** learns, we utilize KL-divergence. Given that $\vec{\phi}_i = (\phi_{i1}, \phi_{i2}, \ldots, \phi_{iW})$ (a topic learned by ***GS***) and $\vec{\phi}'_j = (\phi'_{j1}, \phi'_{j2}, \ldots, \phi'_{jW})$ (a topic learned by ***sgs_GS***), the KL-divergence from $\vec{\phi}'_j$ to $\vec{\phi}_i$ is defined as $K_{i,j} = \sum_{w=1}^{W} \phi_{iw} \ln \frac{\phi_{iw}}{\phi'_{jw}}$.

3. Running time and speedup ratio. To measure how SGS strategy improves ***GS*** in efficiency, we record the average running time per iteration, and also calculate speedup ratios of ***sgs_GS*** and ***GS*** algorithms over ***cgs*** algorithm.

4. Parameter Settings. We set $\alpha = 50/Z$ and $\beta = 0.01$, which are also used in [6].

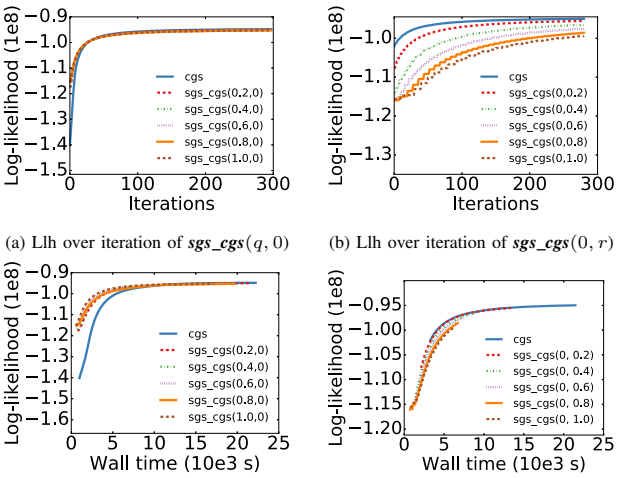
C. Effectiveness Analysis

First, we examine the quality of the topics learned by SGS algorithms. Let the topics learned by ***cgs*** and ***sgs_cgs***$(q, 0)$ with $q = 0.5$, and 1.0 be denoted as ϕ, $\phi'_{0.5}$, and $\phi'_{1.0}$ respectively. Fig. 7 shows the *KL-divergence matrix* of the NIPS data set. In TABLE III we show the representative words of the topics learned from the NIPS data set using



(a) From ϕ to ϕ (b) From $\phi'_{0.5}$ to ϕ (b) From $\phi'_{1.0}$ to ϕ

Figure 7. KL-divergence matrix from ϕ' (learned by *sgs_cgs*) to ϕ (learned by *cgs*) on NIPS ($Z = 10$)



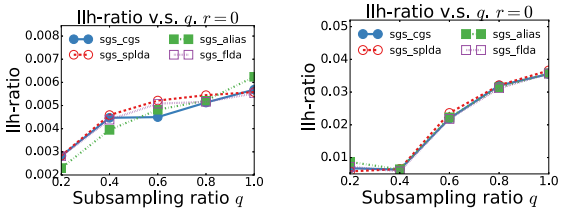
(a) Llh over iteration of ***sgs_cgs***$(q, 0)$ (b) Llh over iteration of ***sgs_cgs***$(0, r)$

(c) Llh over time of ***sgs_cgs***$(q, 0)$ (d) Llh over time of ***sgs_cgs***$(0, r)$

Figure 8. Log-likelihood trend over iterations and wall time of ***sgs_cgs***(q, r) and ***cgs*** on NYTimes ($Z = 1000$)

cgs and ***sgs_cgs***$(q, 0)$. These observations demonstrate that SGS algorithms learn similar topics as CGS.

Second, we examine how the subsampling ratio q affects the effectiveness of SGS algorithms. Fig. 8(a) and Fig. 8(c) show the *llh* trends of ***sgs_cgs***$(q, 0)$ and ***cgs*** on the NYTimes data set. Fig. 8(a) shows that the models learned by ***sgs_cgs***$(q, 0)$ and ***cgs*** have similar *llh* values at the end. Fig. 8(c) shows that ***sgs_cgs***$(q, 0)$ can achieve higher *llh* than ***cgs*** within the same amount of time and ***sgs_cgs***$(q, 0)$ converges faster than ***cgs***. The ***sgs_GS*** algorithms with other base ***GS*** show similar patterns. This demonstrates that SGS algorithms can learn models that are similar to the ***GS*** and converge as the base ***GS***. We further examine *llh-*

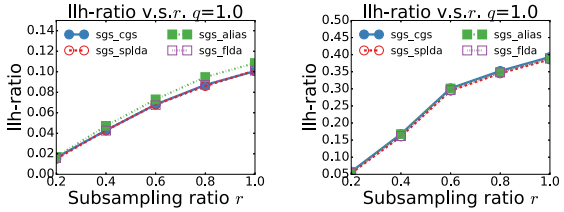


(a) NYTimes (b) PubMed

Figure 9. llh-ratio v.s. subsampling ratio q ($Z = 1000$)

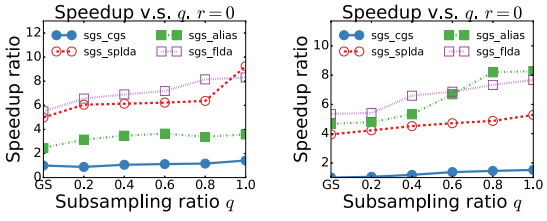
Algorithm	Top-2 words from 10 topics	Overlapping ratio
cgs	function, neuron, network, algorithm, image, object, system, cell, circuit, speech, learning, error, model, data, signal, unit	1.0
sgs_cgs$(0.2, 0)$	**function**, training, **neuron**, word, **network**, **algorithm**, **image**, **object**, **system**, **cell**, **circuit**, set, **learning**, input, **model**, **data**	0.82
sgs_cgs$(0.6, 0)$	**function**, control, training, **object**, word, **network**, neural, **image**, **neuron**, **system**, **cell**, set, **learning**, input, **model**, **data**, recognition	0.80
sgs_cgs$(1.0, 0)$	**function**, control, set, **image**, **object**, **learning**, **data**, recognition, **network**, neural, pattern, presented, **system**, **cell**, memory, **model**, **neuron**	0.78

Table III
REPRESENTATIVE WORDS OF CGS AND SGS_CGS ON NIPS ($Z = 10$); OVERLAPPED WORDS ARE HIGHLIGHTED



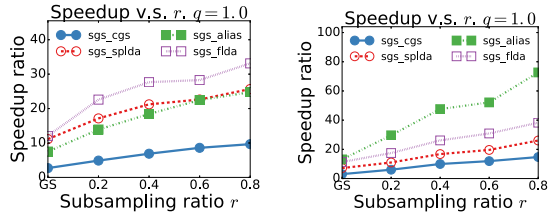
(a) NYTimes (b) PubMed

Figure 10. llh-ratio v.s. subsampling ratio r ($Z = 1000$)



(a) NYTimes (b) PubMed

Figure 11. Comparison of speedup of ***sgs_GS*** methods v.s. subsampling ratio q ($Z = 1000$)



(a) NYTimes (b) PubMed

Figure 12. Comparison of speedup of ***sgs_GS*** methods v.s. subsampling ratio r ($Z = 1000$)

ratio(***sgs_GS***, ***GS***) which is shown in Fig. 9. We can observe that *llh-ratio*(***sgs_GS***, ***GS***) is higher when q is greater. This indicates that ***sgs_GS*** with higher subsampling ratio q learns a LDA model whose *llh* is more biased from that of ***GS***. The small absolute *llh-ratio* shows that ***sgs_GS***$(q, 0)$ can achieve very similar *llh* as ***GS***.

Third, we examine how the subsampling ratio r affects the effectiveness of SGS algorithms. Fig. 8(b) and Fig. 8(d) show the *llh* trends of ***sgs_cgs***$(0, r)$ and ***cgs***. From Fig. 8(b) we can observe that ***sgs_cgs***$(0, r)$ with the lower subsampling ratio r learns models with the more similar *llh* as the models learned by ***cgs***. ***sgs_cgs***$(0, r)$ shows a zig-zag *llh* trend because ***cgs*** runs on uncovered-documents every $10{*}r$ iterations in ***sgs_cgs***. Fig. 8(d) shows that ***sgs_cgs***$(0, r)$ achieve lower *llh* than ***cgs*** within the same amount of time and ***sgs_cgs***$(0, r)$ still converges as the ***cgs*** finally. We also observe that the final *llh* of ***sgs_cgs***$(0, r)$ with higher subsampling ratio r is biased more from ***cgs***. To study the effect of r on effectiveness of SGS algorithms, we present the *llh-ratio*(***sgs_GS***, ***GS***) in Fig. 10. We can observe that higher r results in larger ratios.

D. Efficiency Analysis

We investigate how the subsampling ratio q affects the efficiency of SGS algorithms. The speedup ratios of ***sgs_GS***$(q, 0)$ and ***GS*** over ***cgs*** are shown in Fig. 11. First, ***sgs_GS***$(q, 0)$ consistently has higher speedup ratio than the corresponding ***GS***. It means ***sgs_GS***$(q, 0)$ is faster than ***GS***. Second, when q grows, the speedup ratio also grows because higher q means that less groups (more tokens within one group) are sampled in every iteration.

We also examine how the subsampling ratio r affects the efficiency of SGS algorithms. The speedup ratios of ***GS*** and ***sgs_GS***$(1.0, r)$ over ***cgs*** are shown in Fig. 12. The figure shows that ***sgs_GS***$(1.0, r)$ consistently has higher speedup ratio than the corresponding ***GS***. In addition, when r grows, the speedup ratio of ***sgs_GS***$(1.0, r)$ also grows.

Finally, we check the efficiency of ***sgs_GS*** under different model complexities Z. The speedup ratios of ***GS*** and ***sgs_GS*** over ***cgs*** are shown in TABLE IV. Overall the ***sgs_GS*** algorithms significantly improves the corresponding ***GS*** algorithms. An impressive example is that, ***cgs*** takes 5 hours to run one iteration on PubMed with $Z = 5000$; with the same setting, ***sgs_cgs***$(1.0, 0.3)$ takes only 0.5 hours.

Summary: In real applications, it is a tradeoff to choose proper subsampling ratios q and r to have expected speedup and acceptable effectiveness. Since q does not affect effectiveness too much as shown in Fig. 9, we suggest to choose $q = 1.0$ to have the maximum speedup. Since large r results in unacceptable effectiveness, we recommend to choose a relatively small r value depending on the data set size.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we propose a novel and general strategy Sub-Gibbs Sampling (SGS), to improve the efficiency of any Gibbs sampling algorithms for LDA. The SGS strategy

Algorithms	$Z = 2000$	$Z = 5000$	$Z = 8000$	$Z = 10000$	$Z = 2000$	$Z = 5000$	$Z = 8000$	$Z = 10000$
	NIPS				Enron			
sgs_cgs$(1.0, 0)$	2.61	3.00	2.67	2.54	1.71	1.62	1.72	1.74
sgs_cgs$(1.0, 0.3)$	4.89	4.68	4.74	4.10	3.63	3.92	3.89	3.93
splda	23.56	33.90	40.66	48.57	14.75	14.76	17.03	19.46
sgs_splda$(1.0, 0)$	50.87	67.66	74.34	84.45	20.66	20.68	23.13	25.68
sgs_splda$(1.0, 0.3)$	84.50	143.44	124.05	134.91	42.13	47.84	53.72	60.24
alias	5.55	12.37	17.93	29.82	6.24	11.53	20.75	24.63
sgs_alias$(1.0, 0)$	19.03	41.00	60.69	74.82	12.73	26.73	41.91	48.44
sgs_alias$(1.0, 0.3)$	30.30	64.19	98.53	117.34	27.38	54.86	79.90	106.21
flda	33.36	52.77	67.02	82.82	18.17	30.66	33.39	34.00
sgs_flda$(1.0, 0)$	76.58	149.99	204.64	286.78	33.19	35.65	46.62	56.84
sgs_flda$(1.0, 0.3)$	141.40	244.71	325.33	423.65	71.44	98.90	104.25	134.32
Algorithms	$Z = 2000$	$Z = 5000$	$Z = 8000$	$Z = 10000$	$Z = 1000$	$Z = 2000$	$Z = 5000$	
	NYTimes				PubMed			
sgs_cgs$(1.0, 0)$	1.43	1.43	1.42	1.44	1.52	1.54	1.57	
sgs_cgs$(1.0, 0.3)$	4.23	4.22	4.23	4.27	14.74	15.10	13.74	
splda	5.02	7.35	8.94	8.47	3.95	3.08	3.49	
sgs_splda$(1.0, 0)$	8.02	9.47	10.15	12.53	5.28	4.23	5.50	
sgs_splda$(1.0, 0.3)$	16.41	19.53	28.68	29.27	11.00	12.48	13.90	
alias	4.34	10.21	13.62	18.86	4.68	7.96	13.41	
sgs_alias$(1.0, 0)$	7.06	14.24	20.88	32.13	8.27	11.45	23.80	
sgs_alias$(1.0, 0.3)$	17.15	50.77	75.22	113.48	29.63	46.91	93.02	
flda	7.37	8.31	8.98	10.29	5.41	3.73	4.63	
sgs_flda$(1.0, 0)$	8.51	11.51	13.31	14.95	7.67	5.92	7.79	
sgs_flda$(1.0, 0.3)$	18.07	29.83	36.48	29.91	17.48	15.67	19.43	

Table IV
SPEEDUP OF SGS_GS OVER CGS VARYING Z

utilizes two properties that we observed in text corpora, the tokens in a document have skewed topic distributions and the semantics of a corpus can be approximately covered by a subset of documents in this corpus. We theoretically prove that the error of SGS algorithm is bounded by a small upper bound. We implemented the SGS strategy on the traditional Collapsed Gibbs Sampling (*CGS*) algorithm and three state-of-the-art Gibbs sampling algorithms (*FastLDA*, *AliasLDA*, and *F+Nomad LDA*). The experimental results on four real data sets showed that the SGS algorithms can learn similar models as those of other Gibbs sampling algorithms with much better efficiency. In particular the proposed strategy is $2 \sim 100$ times faster than CGS and $2 \sim 5$ times faster than *FastLDA*, *AliasLDA*, and *F+Nomad LDA* algorithms. In the future we will explore better group partition algorithms and find a close form error bound for the SGS strategy.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *JMLR*, vol. 3, pp. 993–1022, 2003.

[2] T. L. Griffiths and M. Steyvers, "Finding scientific topics," *Proceedings of the National Academy of Sciences*, vol. 101, no. suppl 1, pp. 5228–5235, 2004.

[3] I. Porteous, D. Newman, A. T. Ihler, A. U. Asuncion, P. Smyth, and M. Welling, "Fast collapsed gibbs sampling for latent dirichlet allocation," in *SIGKDD*, 2008, pp. 569–577.

[4] L. Yao, D. M. Mimno, and A. McCallum, "Efficient methods for topic model inference on streaming document collections," in *SIGKDD*, 2009, pp. 937–946.

[5] A. Q. Li, A. Ahmed, S. Ravi, and A. J. Smola, "Reducing the sampling complexity of topic models," in *SIGKDD*, 2014, pp. 891–900.

[6] H. Yu, C. Hsieh, H. Yun, S. V. N. Vishwanathan, and I. S. Dhillon, "A scalable asynchronous distributed algorithm for topic modeling," in *WWW*, 2015, pp. 1340–1350.

[7] J. Yuan, F. Gao, Q. Ho, W. Dai, J. Wei, X. Zheng, E. P. Xing, T. Liu, and W. Ma, "Lightlda: Big topic models on modest computer clusters," in *WWW*, 2015, pp. 1351–1361.

[8] A. J. Walker, "An efficient method for generating discrete random variables with general distributions," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 253–256, 1977.

[9] P. M. Fenwick, "A new data structure for cumulative frequency tables," *Softw., Pract. Exper.*, vol. 24, no. 3, pp. 327–336, 1994.

[10] G. Heinrich, "Parameter estimation for text analysis," Technical Note, University of Leipzig, Germany., 2008.

[11] C. Hu, H. Cao, , and Q. Gong, "Sub-gibbs sampling: a new strategy for inferring lda (theoretical proof and experiment details)," Technical Report TR-CS-NMSU-2017-08-21, 2017. [Online]. Available: https://www.cs.nmsu.edu/wp/wp-content/uploads/2017/06/icdm2017-tr.pdf

[12] A. T. Ihler and D. Newman, "Understanding errors in approximate distributed latent dirichlet allocation," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 952–960, 2012.

[13] C. Lund and M. Yannakakis, "On the hardness of approximating minimization problems," *J. ACM*, vol. 41, no. 5, pp. 960–981, Sep. 1994.

[14] "UCI bag of words data sets," https://archive.ics.uci.edu/ml/datasets/Bag+of+Words.

[15] A. J. Smola and S. M. Narayanamurthy, "An architecture for parallel topic models," *PVLDB*, vol. 3, no. 1, pp. 703–710, 2010.