# Stochastic Collapsed Variational Bayesian Inference for Biterm Topic Model

Narutaka Awaya, Jun Kitazono, Toshiaki Omori and Seiichi Ozawa
Graduate School of Engineering, Kobe University
1-1, Rokkoudai, Nada, Kobe, Hyogo, Japan
Email: 150t202t@stu.kobe-u.ac.jp, kitazono@eedept.kobe-u.ac.jp, omori@eedept.kobe-u.ac.jp, ozawasei@kobe-u.ac.jp

***Abstract*—It is useful for many applications to find out meaningful topics from short texts, such as tweets and comments on websites. Since directly applying conventional topic models (e.g., LDA) to short texts often produces poor results, as a general approach to short texts, a biterm topic model (BTM) was recently proposed. However, the original BTM implementation uses collapsed Gibbs sampling (CGS) for its inference, which requires many iterations over the entire dataset. On the other hand, for LDA, there have been proposed many fast inference algorithms throughout the decade. Among them, a recently proposed stochastic collapsed variational Bayesian inference (SCVB0) is promising because it is applicable to an online setting and takes advantage of the collapsed representation, which results in an improved variational bound. Applying the idea of SCVB0, we develop a fast one-pass inference algorithm for BTM, which can be used to analyze large-scale general short texts and is extensible to an online setting. To evaluate the performance of the proposed algorithm, we conducted several experiments using short texts on Twitter. Experimental results showed that our algorithm found out meaningful topics significantly faster than the original algorithm.**

## I. Introduction

With the advent of big data era, in addition to normal documents such as academic papers and news articles, a lot of short texts such as tweets and short comments on websites have become available for us. Revealing hidden topics in short texts is important for many tasks, such as user interest profiling [1], event tracking [2] and so on. However, conventional topic models such as latent Dirichlet Allocation (LDA) [3] and probabilistic latent semantic analysis (PLSA) [4] often produce poor results on short texts [5], [6] because they implicitly capture document-level word co-occurrence patterns, which have little information on short texts.

To overcome these limitations of conventional topic models, domain-specific methods such as user-based aggregation have been employed [7]. However, when analyzing a new kind of dataset consisting of short texts, practitioners must come up with effective domain-specific information for each dataset. Moreover, such information is not necessarily available.

As a general approach to topic modeling for short texts with no domain-specific assumptions, a biterm topic model (BTM) [8] and a word network topic model (WNTM) [9] were recently proposed by Yan et al. and Zuo et al., respectively. These two methods make use of corpus-level word co-occurrence patterns for learning topics to overcome the sparsity of document-level word co-occurrence patterns in short texts.

Concretely, BTM first turns a corpus into a *bag-of-biterms*. Here a biterm is defined as an unordered word pair that co-occurs in a short context, which is usually a whole document in a short-text setting. After this procedure, BTM infers topic assignments for each biterm using an assumption that a biterm is generated from a single topic. On the other hand, WNTM takes a different approach. Roughly speaking, WNTM first creates a pseudo-document for each word, where each document consists of words that co-occur with the word in a short context. With these pseudo-documents as an input, WNTM learns topics by the same algorithm as LDA.

Although these two algorithms are reported to perform better on short texts than conventional topic models [8], [9], they cannot be readily applied to large-scale short texts because they use Gibbs sampling for their inference, which is slow to find out meaningful topics and requires many iterations over the entire dataset. For WNTM, we can apply fast inference algorithms such as an online variational Bayes [10] after creating pseudo-documents. However, this procedure is not directly extensible to an online setting because we must recreate pseudo-documents as we get a new document. Actually, for BTM, we can obtain new biterms as a new document arrives and use the biterms to update the model efficiently.

As an online inference algorithm for BTM, Cheng et al. proposed an incremental BTM algorithm (iBTM) and an online BTM algorithm (oBTM) [11]. However, these algorithms still use Gibbs sampling, and they are mainly focused on how to incorporate new information without re-running the algorithm over the whole dataset every time new data has become available. Both iBTM and oBTM do not improve on how fast the model reveals meaningful topics.

In this work, we develop a fast inference algorithm for BTM, which is extensible to an online setting. Our algorithm is based on a stochastic collapsed variational Bayesian inference (SCVB0), which was proposed as a fast inference algorithm for LDA by Foulds et al. [12]. SCVB0 is promising because it is applicable to an online setting and takes advantage of the collapsed representation, which results in an improved variational bound, as opposed to an online variational Bayes [10]. We adapt SCVB0 to BTM and the proposed algorithm is significantly faster in finding out meaningful topics from a

978-1-5090-0620-5/16/$31.00 

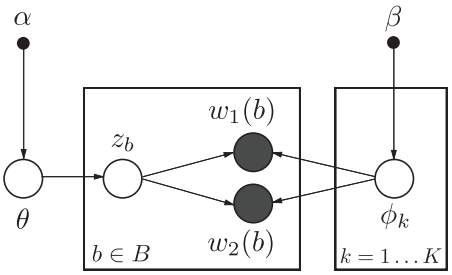


Fig. 1. Graphical model for BTM. The shaded nodes indicate observed variables and the plates indicate replication. $\alpha$ and $\beta$ are hyperparameters of the Dirichlet priors, $\theta$ is a corpus-level topic distribution and $\phi_k$ is a word distribution for topic $k$. $w_1(b)$ and $w_2(b)$ are the pair of words constituting a biterm $b$.

large-scale dataset with short texts than the original algorithm for BTM, which makes use of collapsed Gibbs sampling (CGS) [8].

## II. Biterm Topic Model

A biterm topic model (BTM) for short texts [8] is a recently proposed probabilistic topic model. It is the first topic model for short texts with no domain-specific assumptions. On short texts, conventional topic models such as LDA [3] and PLSA [4] do not perform well due to the severe sparsity of word co-occurrence patterns in each short text [5], [6]. To alleviate this problem, BTM simplifies its generation process by employing biterms. A biterm is defined as an unordered word pair that co-occurs in a short context, where a document itself is usually treated as a single short context in a short-text setting. Suppose that $\alpha$ and $\beta$ are hyperparameters of Dirichlet priors, BTM models the generation of biterms as follows:

1) For each topic $k$
   a) Draw a topic-specific word distribution $\phi_k \sim \mathrm{Dir}(\beta)$
2) Draw a topic distribution $\theta \sim \mathrm{Dir}(\alpha)$ for the whole dataset
3) For each biterm $b$ in the set of biterms $B$
   a) Draw a topic assignment $z_b \sim \mathrm{Mult}(\theta)$
   b) Draw words $w_1(b) \sim \mathrm{Mult}(\phi_{z_b})$ and $w_2(b) \sim \mathrm{Mult}(\phi_{z_b})$

A graphical model for BTM is shown in Fig. 1.

For inference, the original implementation makes use of collapsed Gibbs sampling (CGS) [8]. CGS marginalizes out all the parameters in BTM, namely $\theta$ and $\phi$, and iteratively samples $z_b$ from $p(z_b|z^{\setminus b}, B, \alpha, \beta)$. Here $p(z_b|z^{\setminus b}, B, \alpha, \beta)$ is the distribution of the topic assignments of a biterm $b$ given the topic assignments to all the biterms except $b$, a set of biterms $B$, and hyperparameters $\alpha$ and $\beta$. The topic assignments to all the biterms except $b$ is represented as $z^{\setminus b}$. $p(z_b|z^{\setminus b}, B, \alpha, \beta)$ is calculated as

$$p(z_b|z^{\setminus b}, B, \alpha, \beta) \propto \frac{(n_k^{\setminus b} + \alpha)(n_{k,w_1(b)}^{\setminus b} + \beta)(n_{k,w_2(b)}^{\setminus b} + \beta)}{(V\beta + 2n_k^{\setminus b})(V\beta + 2n_k^{\setminus b} + 1)},$$

where $V$ is the vocabulary size, $n_{k,w}$ is the number of times that a word $w$ is assigned a topic $k$, and $n_k$ is the number of biterms that is assigned a topic $k$. Here the superscripts $\setminus b$ indicate removing the contributions of a biterm $b$, and when a biterm $b$ is assigned a topic $k$, $w_1(b)$ and $w_2(b)$ are also considered to be assigned the topic $k$.

The overall algorithm is shown in Algorithm 1. The number of iterations $N_{\mathrm{iter}}$ usually needs to be sufficiently large to obtain meaningful results ($N_{\mathrm{iter}} = 1000$ is used in [8]). After the iterations end, the algorithm compute the parameters $\theta$ and $\phi$ according to

$$\begin{aligned} \theta_k &\propto n_k + \alpha \\ \phi_{k,w} &\propto n_{k,w} + \beta, \end{aligned} \tag{1}$$

where $\theta_k$ is the probability that a topic $k$ is drawn at the corpus-level and $\phi_{k,w}$ is a word distribution for a topic $k$.

**Algorithm 1** CGS for BTM (Yan et al. [8])

1: **Input**: the number of topics $K$, hyperparameters $\alpha$, $\beta$, a set of biterms $B$, the number of iterations $N_{\mathrm{iter}}$
2: **Output**: word distributions for each topic $\phi$, corpus-level topic distribution $\theta$
3: **for** $t = 1$ to $N_{\mathrm{iter}}$ **do**
4:  **for** $b \in B$ **do**
5:  Draw $z_b \sim p(z_b|z^{\setminus b}, B, \alpha, \beta)$
6:  Update $n_k$, $n_{k,w_1(b)}$ and $n_{k,w_2(b)}$
7:  **end for**
8: **end for**
9: Compute the parameters $\theta$ and $\phi$ according to (1).

## III. Proposed Method

In this section, we propose a stochastic collapsed variational Bayesian inference for BTM (SCVB0-BTM), which uses the idea of SCVB0 for LDA [12] and significantly faster than the original CGS inference (CGS-BTM) presented in the previous section. Moreover, the proposed algorithm is extensible to an online setting.

### A. CVB0 for BTM

First, following the CVB0 algorithm for LDA [13], we derive the CVB0 algorithm for BTM as a prerequisite for SCVB0-BTM. In the CVB0 inference, we marginalize out the corpus-level distribution over topics $\theta$ and the word distributions $\phi_k$ for each topic $k$, then perform inference only on the topic assignments $z_b$ for each biterm $b$. Since strict evaluation of the expected values appearing in the coordinate ascent, which is often used in variational inference, is intractable, a zero-order approximation is employed as in the original CVB0. The algorithm becomes similar to Algorithm 1 but with deterministic updates. According to

$$\gamma_{bk} \propto \frac{(N_k^{\setminus b} + \alpha)(N_{k,w_1(b)}^{\setminus b} + \beta)(N_{k,w_2(b)}^{\setminus b} + \beta)}{(V\beta + 2N_k^{\setminus b})(V\beta + 2N_k^{\setminus b} + 1)}, \tag{2}$$

the CVB0 algorithm for BTM iteratively updates variational posterior $\gamma_{bk}$, which is a probability of assigning a topic $k$ to

a biterm $b$ given the observation, for each biterm $b$. Here we used the "CVB0 statistics" defined as

$$N_k = \sum_{b \in B} \gamma_{bk} \tag{3}$$

$$N_{k,w} = \sum_{b \in B_w} \gamma_{bk}, \tag{4}$$

where $B_w$ is the set of biterms that contain a word $w$ and the superscripts $\backslash b$ in (2) indicate taking the sum without the term relevant to a biterm $b$.

The derived CVB0 algorithm for BTM uses $\mathcal{O}(|B|K)$ memory, and still requires many iterations over the whole dataset though the number of iterations is usually smaller than that of CGS. For these limitations, CVB0 for BTM usually cannot be applied to large datasets.

### *B. SCVB0 for BTM*

Based on CVB0 for BTM derived in the previous subsection, we derive SCVB0-BTM similar to SCVB0 for LDA [12]. SCVB0 assumes that biterms are drawn from the uniform distribution over the dataset and estimates CVB0 statistics using each biterm. Since these estimations are very crude, to reduce the variance of the estimates, an online average of CVB0 statistics is employed. These stochastic updates enable the model to learn topics faster than the deterministic updates in CVB0.

For BTM, the basic idea is that we do not retain each $\gamma_{bk}$ and stochastically update $N_k$ and $N_{k,w}$ as a new biterm is observed. When a new biterm $b$ arrives, a topic posterior $\gamma_{bk}$ is computed according to (2) using current $N_k$ and $N_{k,w}$ without the superscripts $\backslash b$. This approximation of ignoring the subtraction becomes negligible when the number of biterms is sufficiently large. From this $\gamma_{bk}$, updated $N_k$ and $N_{k,w}$ are estimated according to

$$\hat{N}_k = |B|\gamma_{bk}$$

$$\hat{N}_{k,w} = \begin{cases} |B|\gamma_{bk} & \text{if } w = w_1(b) \text{ or } w = w_2(b) \\ 0 & \text{otherwise} \end{cases},$$

where $\hat{N}_k$ and $\hat{N}_{k,w}$ are very crude estimates of $N_k$ and $N_{k,w}$ after a single update of parallel coordinate ascent, respectively. Note that this interpretation of approximate parallel coordinate ascent comes from [14]. If a biterm $b$ is considered to be drawn from the uniform distribution over $B$, these estimates are unbiased.

Using these estimates, $N_k$ and $N_{k,w}$ are updated according to

$$N_k \leftarrow (1-\rho_t)N_k + \rho_t \hat{N}_k \tag{5}$$

$$N_{k,w} \leftarrow (1-\rho_t)N_{k,w} + \rho_t \hat{N}_{k,w}. \tag{6}$$

For controlling a trade-off between new information and old information as the time step $t$ changes, we use a Robbins-Monro sequence as in [12]. As the time step $t$ increases, it becomes insensitive to new information. For an online setting where convergence to a specific state is not expected, constant learning rates can also be utilized for always adapting to new information. The Robbins-Monro sequence we used here is defined as

$$\rho_t = \frac{1}{(t+\tau)^\kappa}, \tag{7}$$

where $\tau$ and $\kappa$ are hyperparameters of the proposed algorithm. However, for simplicity, we always use $\tau = 1000$.

From the perspective of time complexity, (6) can be $\mathcal{O}(KV)$ in a naive implementation because $N_{k,w}$ contains $KV$ elements, where $K$ is the number of topics and $V$ is the vocabulary size. This $\mathcal{O}(KV)$ update per biterm is very computationally expensive because the overall time complexity of the algorithm becomes $\mathcal{O}(|B|KV)$. To avoid this computationally expensive update, we make use of a lazy update technique taking advantage of the sparsity of $\hat{N}_{k,w}$, which contains only $2K$ non-zero elements.

To update $N_{k,w}$ using (6), the following steps are needed:

1) Multiply all the elements in $N_{k,w}$ by $1-\rho_t$.
2) For $k$ and $w$ where $\hat{N}_{k,w}$ is non-zero, perform updates $N_{k,w} \leftarrow N_{k,w} + \rho_t \hat{N}_{k,w}$.

Although the first step appears to be $\mathcal{O}(KV)$ computation, by using a proper data structure, this computation becomes $\mathcal{O}(1)$. To represent $N_{k,w}$, which consists of $KV$ elements, in the algorithm, we maintain an array $A_{k,w}$ with $KV$ elements and a scalar $a$ instead of directly maintaining $N_{k,w}$. Throughout the algorithm, it is always assumed that an equation $N_{k,w} = aA_{k,w}$ holds for any $k$ and $w$.

Concretely, when we want to multiply all the elements by a scalar $1-\rho_t$ as in the first step, we just perform $a \leftarrow (1-\rho_t)a$, which is $\mathcal{O}(1)$. For the second step, we need to know the current value of $N_{k,w}$ and set the value of $N_{k,w}$ to a specific value. According to the equation $N_{k,w} = aA_{k,w}$, to get the value of $N_{k,w}$ for specific $k$ and $w$, $aA_{k,w}$ is just used. To set $N_{k,w}$ for specific $k$ and $w$ to the value of a scalar $c$, $A_{k,w} \leftarrow c/a$ is performed.

By using this internal representation of $N_{k,w}$, namely $A_{k,w}$ and $a$, the time complexity for an update using (6) now becomes $\mathcal{O}(K)$. Note that since $1-\rho_t$ is repeatedly multiplied to $a$ during the algorithm, it may result in overflow or underflow in floating-point arithmetic. For avoiding this situation, we must regularly perform $A_{k,w} \leftarrow aA_{k,w}$ and $a \leftarrow 1$ for resetting $a$, which is $\mathcal{O}(KV)$. Although this $\mathcal{O}(KV)$ operation appears to be problematic, in the algorithm, the situation rarely occurs because $1-\rho_t$ is approaching 1 as the value of $t$ increases. Actually, in our experiments, the situation did not occur during each iteration over the whole dataset. Moreover, even if the $\mathcal{O}(KV)$ operation is needed, it would be negligible compared with the number of other operations.

For $N_{k,w}$, we always utilize this internal representation for computational efficiency. Making use of this lazy update technique, the overall time complexity per biterm now becomes $\mathcal{O}(K)$, which is much more efficient than $\mathcal{O}(KV)$ because $V$ is relatively large in real-world settings. Using the lazy update, the overall proposed algorithm is shown in Algorithm 2.

**Algorithm 2** SCVB0 for BTM (Our Method)

**Input**: the number of topics $K$, hyperparameters $\alpha$, $\beta$, a set of biterms $B$
**Output**: corpus-level topic distribution $\theta$, word distributions for each topic $\phi$
Randomly initialize $N_k$ and $N_{k,w}$. ▷ See the text for the internal representation of $N_{k,w}$.
Let the time step $t = 1$.
**for** $b \in B$ **do** ▷ the biterms in $B$ are randomly ordered
    Compute $\gamma_{bk}$ according to (2) ignoring the superscripts $\backslash b$.
    Update $N_k$ and $N_{k,w}$ according to (5) and (6).
    Update the time step $t \leftarrow t + 1$.
**end for**
Compute the corpus-level topic distribution $\theta_k \propto N_k + \alpha$.
Compute word distributions for each topic $\phi_{k,w} \propto N_{k,w} + \beta$.

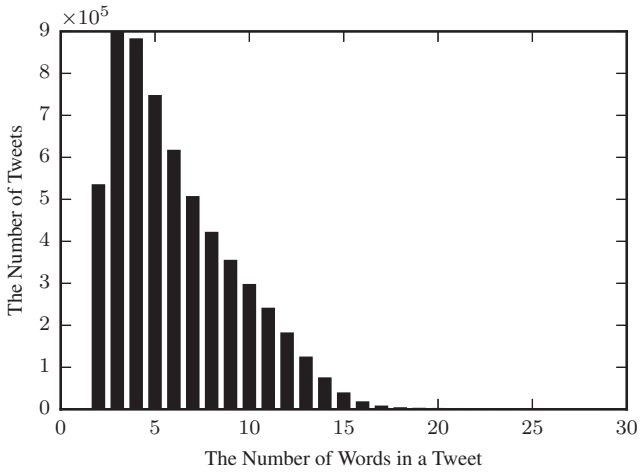


Fig. 2. Histogram of tweet lengths (i.e., the number of words in a tweet) in the dataset after the preprocessing. The average number of words in a tweet is 6.1.

## IV. Experiments

We carried out experiments to evaluate the performance of SCVB0-BTM with a real-world dataset. Here, we collected English tweets on Twitter as short text data. We compared the time to learn topics between CGS-BTM and the proposed SCVB0-BTM. Note that unless otherwise specified, the number of topics $K = 50$, Dirichlet hyperparameters $\alpha = 50/K$ and $\beta = 0.01$ were used as in [8].

### A. Dataset

We used 50 000 000 English tweets sampled from October 9 to November 18, 2015, as the dataset. In the dataset, there were a lot of noisy data (e.g., repeated advertisements, comments by nonsense bots and so on) that prevent from estimating accurate topics of common users' interests. Since advertisements often contain a URL, we removed tweets that contain a URL. Moreover, to reduce comments by nonsense bots, we used only a single tweet per user. To remove excessively duplicate contents, retweets were removed and tweets that share the same text were treated as a single tweet.

After that, to normalize contents, stop words were removed and all words are lowercased. To simplify the dataset, we treated duplicate words in a tweet as a single word. Since infrequent words have little information, we removed words that occur in only less than 10 tweets. Moreover, we removed tweets with only a single word because they also have little information.

After the preprocessing, 5 919 659 tweets were left and the time period still spanned from October 9 to November 18, 2015. The vocabulary size was $V = 117\,449$, the number of biterms was $|B| = 121\,825\,490$, and the average number of words in a tweet in the dataset was 6.1. Fig. 2 shows the histogram of tweet lengths (i.e., the number of words in a tweet) in the used dataset. As seen from Fig. 2, we used short texts that typically include 3 to 10 words.

In the following experiments, the set of biterms extracted from these preprocessed tweets is used as an input for algorithms.

### B. Topic Quality Evaluation

To investigate how the quality of topics changes during the learning process of SCVB0-BTM and CGS-BTM, they were iterated multiple times over the whole dataset. For SCVB0-BTM, Robbins-Monro parameters $\tau = 1000$ and $\kappa = 0.8$ are used. Several selected topics for different iteration steps are shown in Table I, II and III, which show topics after 1, 10 and 1000 iterations, respectively. For both of the algorithms, it took about 2 minutes per iteration in our environment.

For each table, a double line separates topics. Each topic contains the "top words" (shown upper) and the "non-top words" (shown lower), which are defined later. A coherence score for each topic is also shown in the upper-left in the cell which contains the top words for a topic. The coherence score for a topic is defined in [15] as

$$\sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D(v_m, v_l) + 1}{D(v_l)},$$

where $(v_1, \ldots, v_M)$ is a list of the most probable $M$ words for the topic (in our experiments, $M = 20$ is used). Here $D(v_i)$ is the number of documents which contains the word $v_i$ and $D(v_i, v_j)$ is the number of documents which contains both $v_i$ and $v_j$. A higher coherence score indicates a more "coherent" topic.

Since in the middle of the learning process, topics with higher coherence scores almost always consist of a lot of common words, we want to see topics with lower coherence scores in each setting. Thus, we selected the most "incoherent" four topics for each setting for comparison.

In the tables, the top words for each topic are the most probable 20 words in the topic, and the non-top words are

TABLE I
TOPICAL WORDS BY SCVB0-BTM AND CGS-BTM AFTER A SINGLE ITERATION

| SCVB0 (after 1 iteration) | CGS (after 1 iteration) |
|---|---|
| **(Score: -1019)** ur @gurmeetram-rahim sir love best good india @iamsrk like hai day one test ji happy #indvsa ho movie @be-ingsalmankhan time | **(Score: -684)** im like dont get love one good know time go cant day people see got today want back need would |
| akram saying ghar #fan kajol loves bye gy broad month jaye apna leave mil chase tera share thinking @faf1307 @zoomtv | clothes mark mess itll sold cream holiday clear screen snap personal usually sharing died caught de-cided sex shopping waste company |
| **(Score: -985)** de la que en el love con por mi se le es te un los ya eu dia com ni | **(Score: -684)** im like dont get love one good time know go day cant people see got want back today need would |
| steal yah streaming asi liga schrank kl fome wah chorando video david #1dmx memang causa quando living ano true walk | stream area stage blame lit company grade community greatest worked queen none energy nap action drinking clothes @real_liam_payne holiday five |
| **(Score: -972)** #amas artist favorite year female @onedirection #emabiggestfansjustinbieber soulr&b vote @arianagrande #emabiggestfans1d poprock @rihanna group @beyonce @taylorswift13 voting one taylor raphip-hop | **(Score: -684)** im like dont get love one know good time go cant day people see got want today back need would |
| siete roman shippo feat gap character @projetooohelp x8 @shaeera_k esse cen backkkk @applemusic porra merda achieved acabou away ote ew | sell apple twice dying practice itll whenever sexy helping singing plays @5sos wars btw kiss five falling chat sharing main |
| **(Score: -937)** #love #art follow #teamfollowback love sexy #halloween #music #travel #follow #fashion new day beautiful want happy followers #photography morning good | **(Score: -684)** im like dont get love one good know time go cant day people see got back want today need would |
| #meow wa #trees #beer #nudes #newday #toes #cold #sunshine beach two #original #golf #and #late #ladyboy #wcw cant #whores #follow2befollowed | price wednesday student stage crap unless happiness cream five retweet enjoyed channel interested hurts link ways learning space breaking version |

TABLE II
TOPICAL WORDS BY SCVB0-BTM AND CGS-BTM AFTER 10 ITERATIONS

| SCVB0 (after 10 iterations) | CGS (after 10 iterations) |
|---|---|
| **(Score: -1152)** ur @gurmeetram-rahim sir india best good @iamsrk love hai like #indvsa movie ji test ho @biggboss cricket match @be-ingsalmankhan #indvssa | **(Score: -885)** today im rain wind like weather get day pressure falling dont high good time love one go current humidity low |
| beat sobti fir tere taylor official strike something surely mohit gt #srk taking kab dec interesting ma-sood uu dats @kbfcofficial | experience los clean turned #pushawardslizquens space lights putting average practice river drizzle shut van valley lead airport changed thu mi |
| **(Score: -1140)** #love day new #art follow love today morning #hal-loween beautiful #fashion sexy #music #travel #teamfollowback #follow want #fitness #photogra-phy happy | **(Score: -855)** one #mtvstars im like love lady dont get justin direc-tion know good go time summer seconds see cant got bieber |
| #learning #throwback #joy sea-son #dallas @dirtyoldman_68 be-gins quick brand #meditation #to month mmm #omg street #ifollow-back store #authors removed light | photos possible given hopefully spend experience sent worse imma performing meeting walked dress pm record spot fix ll gods wanting |
| **(Score: -1025)** #amas artist favorite year #pushaward-skathniels one female vote @onedirection hundred #emabiggestfansjustinbieber soulr&b #emabiggestfans1d @arianagrande @rihanna poprock two @beyonce group three | **(Score: -804)** im like get dont love one free go know good time cant see trade got tweet day follows want need |
| ashley hate jadine drag theres dulce steal rr #1dca drive slayed brasil tattooed work kb @hill-songunited belong even tv dezoito | explain deep thursday fuckin greatest eye chat per human thru ryan spot wins doubt @justinbieber member roll mention given finished |
| **(Score: -975)** love im please dont like @sbs_mtv miss thanks go time want hahaha pls haha thank hello ur hi ni ya | **(Score: -794)** im like #aldubeb-tamangpanahon dont love good get one time @mainedcm go day know happy cant people see make @aldenrichards02 back |
| wedding guna straight loves np un-pretty model shut deh abeg holiday @cassandrasleee hihi laugh pt3 tq etc jakarta jan diamond | minute gods club mouth slow nov grow difference alive low hearts practice study french france kayo type brain system @foxnews |

the 20 words ranked from 1000 to 1020 in the order. As top words and even non-top words are located in a rather higher position in the order in the whole vocabulary, non-top words should be consistent with the top words in the same topic.

After only a single iteration over the whole dataset, the proposed SCVB0-BTM has found out meaningful topics as in the left column of Table I. For example, the topic in the top-left cell of Table I is related to India. On the other hand, all the topics found by CGS-BTM after a single iteration, which is shown in the right column of Table I, are almost the same. CGS-BTM was not able to find any meaningful topics after a single iteration.

After 10 iterations over the whole dataset, apparently CGS-BTM appears to have found meaningful topics, but they still contain a lot of common words. For example, although the topic at the top of the right column of Table II appears to be a "weather" topic, this topic contains relatively irrelevant common words in its top words (e.g., dont, love, go and get). Ideally, such common words should only be in the top words of "common-word topics", which mainly contain common words as its top words. Actually, SCVB0-BTM has found a better "weather" topic after only a single iteration, which is shown in Table IV.

The topics extracted by CGS-BTM after 1000 iterations are shown in Table III. This setting is similar to the experiment in the original BTM paper [8]. CGS-BTM have found meaningful topics in this setting. However, it is uncertain whether CGS-BTM (after 1000 iterations) is better than SCVB0-BTM (after 1 or 10 iterations) or not.

CGS-BTM requires many iterations because it uses col-

TABLE III
TOPICAL WORDS BY CGS-BTM AFTER 1000 ITERATIONS

| CGS (after 1000 iterations) |
|---|
| **(Score:-1061)** #mtvstars one direction justin lady gaga #emabiggestfansjustinbieber bieber ariana summer seconds fifth #madeintheam harmony #1dharry #1dlouis grande #1dliam #1d nicki |
| indonesia @jbkidrauhlhelp spamming @1027kiisfm scared #1dpt @official1dmex fancy soundwave cita fireproof x19 jane @britawards 22x dms current normani dnce zedd |
| **(Score:-1032)** ur love @gurmeetramrahim good sir @iamsrk hai best like nd plz movie ho ji hi happy @biggboss show @beingsalmankhan na |
| married believe #sanayairani awaited no1 shame gave waste net madam @zoomtv @flipkart festival bcos sen wali cricket releasing @shaheer_saka |
| **(Score:-1024)** day new today love #love morning #halloween follow great #art beautiful happy us good get ready time #travel #fashion #music |
| #greece #clouds train #africa #loveit competition #ny hope oct #tired simply #sydney ahead #thailand road naughty building #happymonday happen success |
| **(Score:-1007)** #amas artist de favorite year female @onedirection que en soulr&b love la el con por #emabiggestfans1d @rihanna te es se |
| monkeys dx nominated min cute soir puede idk gg amigo bebe aq confident cumple plz outfit forget nights #otrasheffield3 meme |

TABLE IV
A "WEATHER" TOPIC FOUND BY SCVB0-BTM AFTER A SINGLE ITERATION

| **(Score:-734)** wind humidity rain pressure temperature cloudy today weather falling low high temp current rising partly mm visibility mostly mph kmh |
|---|
| #landoph waukesha los cst #ldn aedt @fox5atlanta jersey rockford 70mph venice operations monday wagoner #morning lashing okt diminishing hubbard #2016election |

lapsed Gibbs sampling, which is governed by the randomly initialized state for a long time. From the perspective of wall-clock time in our experimental environment, it took about 2 minutes for a single iteration over the whole dataset for both SCVB0-BTM and CGS-BTM. In this experiment, the proposed SCVB0-BTM was able to find meaningful topics much faster than CGS-BTM.

### C. Test-set Perplexity Evaluation

To see how the generalization performance of the topical representation of each document changes during the learning process for the algorithms, we used test-set perplexity, which measures how well the model predicts missing words by using the topic posterior for documents and the learned topics. Note that lower test-set perplexity is better. For calculating test-set perplexity, we first removed a randomly selected word from each tweet that contains more than 2 words. After the removal, $|B| = 92\,193\,504$ biterms were left. In this experiment, we learned the model using these incomplete documents $D_m$.

Now test-set perplexity $PPL$ is defined as

$$PPL = \frac{1}{\left(\prod_{d \in D_m} p(m_d|d)\right)^{\frac{1}{|D_m|}}}.$$

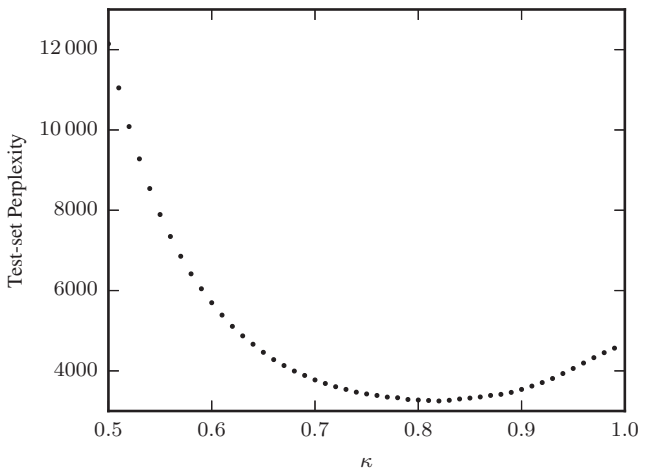


Fig. 3. Test-set perplexity for different Robbins-Monro parameters $\kappa$. SCVB0-BTM was iterated over the dataset $D_m$ only once.

Here $p(m_d|d)$ is the probability of the occurrence of the missing word $m_d$ given a document $d$ and defined as

$$p(m_d|d) = \sum_{k=1}^{K} p(m_d|z=k)p(z=k|d),$$

where $p(m_d|z=k)$ is the probability of the occurrence of a word $m_d$ given a topic $k$ and $p(z=k|d)$ is the probability that a document $d$ is assigned a topic $k$. $p(m_d|z=k)$ is directly obtained from the learned parameter $\phi$ and $p(z=k|d)$ is calculated using $B_d$, which is the set of biterms contained in a document $d$, as

$$p(z=k|d) \propto \sum_{b \in B_d} p(z=k|b).$$

This equation for the topic posterior for a document is rather ad hoc because BTM does not directly model the generation process of documents. However, despite this limitation, this topical representation of a document was shown to good in the original BTM paper [8].

As a first experiment, we examined how the test-set perplexity changes in the proposed SCVB0-BTM when we use different Robbins-Monro parameters $\kappa$ in (7). We ran the algorithm for a single iteration over the whole dataset for each $\kappa$. Among parameters for the Robbins-Monro sequence, $\kappa$ determines a trade-off between newer information and older information as the time step $t$ increases. The result is shown in Fig. 3. As lower test-set perplexity indicates the progress of learning, we see that using better $\kappa$ is very important for faster learning.

As a second experiment, we examined how the test-set perplexity changes when we run CGS-BTM and SCVB0-BTM for multiple iterations. The result is shown in Fig. 4. Among different runs for each algorithm, the variance of the test-set perplexity was so small that it is negligible at this scale. For

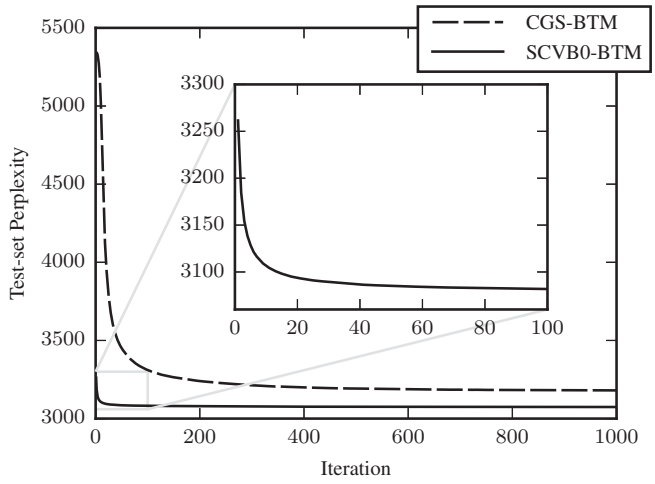


Fig. 4. Test-set perplexity $PPL$ per iteration for the two algorithms. After a single iteration, which took about two minutes for both of the algorithms in our environment, SCVB0 reached $PPL = 3262$ while CGS was still $PPL = 5344$.

simplicity, only a single instance is plotted in Fig. 4 for each setting.

As the wall-clock time per iteration over the whole dataset in our environment was about 2 minutes for both of the algorithms, SCVB0-BTM has reached much lower test-set perplexity significantly faster than CGS-BTM. In addition to faster learning, SCVB0-BTM appears to get lower test-set perplexity than CGS-BTM even after a sufficient number of iterations ($\sim 1000$).

## V. Conclusions

In this work, we developed a faster inference algorithm for BTM than the original CGS inference [8]. By using the proposed SCVB0-BTM, we can find out meaningful topics significantly fast from a large static dataset that mainly consists of short texts. In addition, it can be adapted to an online setting in practice if we roughly estimate the number of biterms $|B|$ in advance. As the learning rates for SCVB0-BTM, constant rates can be used in an online setting because it does not need to converge to a specific state.

As far as we know, this is the first work to improve the inference speed of BTM using stochastic optimization. Not just it is fast, in our experiments, it also performed better than CGS-BTM even with a sufficient number of iterations in terms of test-set perplexity. To see how the proposed algorithm performs better, further analysis and experiments are needed. Although the algorithm needs to be set proper learning rates $\rho_t$, recent works on determining learning rates automatically (e.g., [14]) may be applied as a future work.

## References

[1] J. Weng, E.-P. Lim, J. Jiang, and Q. He, "TwitterRank: Finding Topic-sensitive Influential Twitterers," in *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, 2010, pp. 261–270.

[2] C. X. Lin, B. Zhao, Q. Mei, and J. Han, "PET: A Statistical Model for Popular Events Tracking in Social Communities," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2010, pp. 929–938.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[4] T. Hofmann, "Probabilistic Latent Semantic Analysis," in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI'99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

[5] X.-H. Phan, L.-M. Nguyen, and S. Horiguchi, "Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections," in *Proceedings of the 17th International Conference on World Wide Web*, ser. WWW '08. New York, NY, USA: ACM, 2008, pp. 91–100.

[6] K. Puniyani, J. Eisenstein, S. Cohen, and E. P. Xing, "Social links from latent topics in Microblogs," in *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, 2010, pp. 19–20.

[7] L. Hong and B. D. Davison, "Empirical Study of Topic Modeling in Twitter," in *Proceedings of the First Workshop on Social Media Analytics*, 2010, pp. 80–88.

[8] X. Yan, J. Guo, Y. Lan, and X. Cheng, "A Biterm Topic Model for Short Texts," in *Proceedings of the 22nd International Conference on World Wide Web*, 2013, pp. 1445–1456.

[9] Y. Zuo, J. Zhao, and K. Xu, "Word network topic model: a simple but general solution for short and imbalanced texts," *Knowledge and Information Systems*, pp. 1–20, 2015.

[10] M. Hoffman, F. R. Bach, and D. M. Blei, "Online Learning for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 856–864.

[11] X. Cheng, X. Yan, Y. Lan, and J. Guo, "BTM: Topic Modeling over Short Texts," in *IEEE Transactions on Knowledge and Data Engineering*, 2014.

[12] J. Foulds, L. Boyles, C. DuBois, P. Smyth, and M. Welling, "Stochastic Collapsed Variational Bayesian Inference for Latent Dirichlet Allocation," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2013, pp. 446–454.

[13] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On Smoothing and Inference for Topic Models," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, 2009, pp. 27–34.

[14] N. Houlsby and D. M. Blei, "A Filtering Approach to Stochastic Variational Inference," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2114–2122.

[15] D. Mimno, H. M. Wallach, E. Talley, M. Leenders, and A. McCallum, "Optimizing Semantic Coherence in Topic Models," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2011, pp. 262–272.