Knowl Inf Syst
DOI 10.1007/s10115-017-1099-0

REGULAR PAPER

Short text topic modeling by exploring original documents

Ximing Li[1,2] · Changchun Li[1,2] · Jinjin Chi[1,2] ·
Jihong Ouyang[1,2]

Received: 7 May 2016 / Revised: 26 December 2016 / Accepted: 1 September 2017

Abstract Topic modeling for short texts faces a tough challenge, owing to the sparsity problem. An effective solution is to aggregate short texts into long pseudo-documents before training a standard topic model. The main concern of this solution is the way of aggregating short texts. A recent developed self-aggregation-based topic model (SATM) can adaptively aggregate short texts without using heuristic information. However, the model definition of SATM is a bit rigid, and more importantly, it tends to overfitting and time-consuming for large-scale corpora. To improve SATM, we propose a generalized topic model for short texts, namely latent topic model (LTM). In LTM, we assume that the observable short texts are snippets of normal long texts (namely original documents) generated by a given standard topic model, but their original document memberships are unknown. With Gibbs sampling, LTM drives an adaptive aggregation process of short texts, and simultaneously estimates other latent variables of interest. Additionally, we propose a mini-batch scheme for fast inference. Experimental results indicate that LTM is competitive with the state-of-the-art baseline models on short text topic modeling.

Keywords Short text · Topic modeling · Original document · Fast inference

1 Introduction

Short texts, such as microblogs, are nowadays increasingly available on the Internet. Inferring latent topics and discovering knowledge from these short texts are significant for many real-world applications [7,12,19]. However, unlike normal long texts, short texts contain a few word tokens, e.g., the lengths of microblogs are commonly less than 10, resulting in the so called sparsity problem. This introduces a tough challenge for short text applications.

✉ Jihong Ouyang
ouyj@jlu.edu.cn

1 College of Computer Science and Technology, Jilin University, Changchun, China

2 Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Changchun, China

In this paper, we investigate topic modeling for short texts. Standard topic models such as latent Dirichlet allocation (LDA) [3] consider that each document is a mixture of topics, where each topic is a multinomial distribution over a fixed vocabulary. The topic-related variables learnt by statistical inference algorithms are used to uncover the hidden structure and other thematic information of documents. The standard topic modeling family is acknowledged as a very powerful analysis tool for normal long texts, however, it loses effectiveness on short texts [9,31]. This is mainly caused by the sparsity problem mentioned above, where only a few word token samples are available in short texts.

A straightforward methodology to alleviate the sparsity problem is to aggregate short texts into long pseudo-documents before training a standard topic model. The intuition is that a group of short texts may share similar topic mixture structures, and aggregating these texts leads to a single text sample that contains rich word token samples. We fully agree with this methodology. For example, a piece of dialogues (i.e., short texts) from microblogs can be seen as an interview (i.e., a normal long text) from newsgroups; a quantity of online product reviews (i.e., short texts) can be seen as a user feedback report (i.e., a normal long text).

The main concern is how to aggregate short texts. Some investigations use observable heuristic information to guide the aggregation. For example, the popular microblog Twitter includes abundant side information such as authorship and hashtag. Researches on Twitter use these heuristic information to aggregate tweet texts, and then propose user-based aggregation [9], word-based aggregation [24] and combination aggregations [14]. Many short text collections such as search queries lack useful heuristic information. A recent work, namely self-aggregation based topic model (SATM) [20], models these general collections by integrating clustering and topic modeling. It introduces an additional generative process for short snippets that helps adaptively aggregate short texts. SATM is more robust than aggregation methods that are dependent on heuristic information. However, it tends to overfitting and time-consuming, especially for large-scale corpora.

To improve SATM, in this paper we propose a generalized topic model for short texts, namely latent topic model (LTM). In contrast with SATM, the model definition of LTM is straightforward to be understood. The key idea behind LTM can be described as follows: Given a standard topic model, LTM follows its rules to generate a number of normal long texts, where each of them consists of a set of short snippets (i.e., short texts). The short texts are observable, but their normal long text memberships are unknown. We refer to the normal long text as original document. In contrast with SATM, LTM involves only one new latent variable, i.e., the original document assignment for each short text. With approximating inference algorithms (Gibbs sampling is used in this work), LTM iteratively updates the original document assignments along with other latent variables of interest. This is in fact an adaptive original document exploration procedure. For fast inference, we further develop a mini-batch scheme by breaking short texts into different original document blocks. Experimental results indicate that LTM is competitive with the state-of-the-art baseline models.

The rest of this paper is organized as follows: In Sect. 2, we briefly review LDA and SATM. The proposed LTM algorithm is introduced in Sect. 3. Section 4 describes related works, and Sect. 5 presents experimental results. Conclusions are stated in Sect. 6.

2 Background

In this section, we briefly review the standard latent Dirichlet allocation (LDA) model [3] and self-aggregation-based topic model (SATM) for short texts [20]

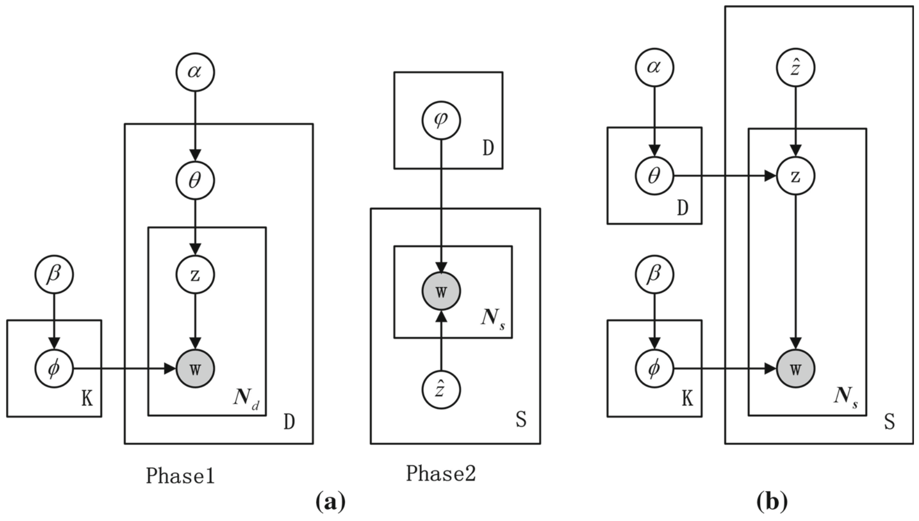

Fig. 1 Graphical model representations of **a** SATM and **b** LTM

2.1 LDA

LDA [3] is a representative topic model used to uncover collections of documents. The model consists of K topics, where each topic k is a multinomial distribution ϕ_k over a fixed V-dimensional vocabulary, drawn from a Dirichlet prior β. Given the topics, each document d is generated as follows: (1) draw a multinomial distribution θ_d over topics from a Dirichlet prior α. (2) To generate the n-th word token, it first draws a topic indicator $z_{dn} \in \{1, \ldots, K\}$ from θ_d, and then draws a word w_{dn} from the selected topic $\phi_{z_{dn}}$.

2.2 SATM

SATM [20] models short texts by integrating clustering and topic modeling. Its generative process for short texts can be represented in two phases. The first phase follows LDA to generate D normal long texts. Suppose that there are D long text-specific multinomial distributions φ over words. In the second phase, each normal long text d is divided into a set of short texts by independently sampling word tokens from φ_d. The graphical model representation of SATM is shown in Fig. 1a.

SATM builds a bridge between observed short texts and latent normal long texts in the document generation level. With approximating inference algorithms, it aggregates short texts without using heuristic information. This can enrich text samples, and can alleviate the sparsity problem in short text topic modeling.

3 Latent topic model

In SATM, short texts are generated from an additional generative process (i.e., the second phase). Each normal long text generated by LDA (i.e., the first phase) is randomly divided into a set of short texts. In the context of SATM, it involves two new latent variables, i.e., the long text-specific distribution φ and the long text assignment for each short text $\hat{z}$. This leads to much more time and memory requirements for model inference, especially for large-scale collections.

To improve SATM, we propose a new generalized topic model for short texts, namely latent topic model (LTM). We omit the additional short text generation process in SATM. Instead, we suppose that short texts are snippets of normal long texts generated by a standard topic model, but their long text memberships are unobservable. Our assumption doesn't change the structure of standard topic models. Additionally, we develop a mini-batch scheme for fast model inference. The details of LTM are presented in the following subsections.

3.1 Model

LTM provides a new perspective to describe the relationship between observed short texts and latent normal long texts. Given a standard topic model, LTM follows its rules to generate D normal long texts, where each one consists of a set of short snippets (i.e., short texts). All S unordered short texts are observable, but their normal long text assignments are unknown. We refer to the given standard topic model as candidate topic model, and refer to the normal long text as original document.

More formally, each short text is a pair $\{s, \hat{z}\}$, where the short text s itself is observable, but its original document assignment $\hat{z}$ is latent. We use LDA as the candidate topic model. LTM generates a collection of short text pairs as follows: Let K, D and S be the number of topics, original documents and short text pairs, respectively. (1) For each topic k, draw a multinomial distribution ϕ_k over a fixed vocabulary, from a Dirichlet prior β. (2) For each original document d, draw a multinomial distribution θ_d over topics, from a Dirichlet prior α. (3) For each short text pair $\{s, \hat{z}\}$, generate word tokens as in LDA given $\theta_{\hat{z}_s}$. As shown in Fig. 1b, the generative process of LTM is outlined by:

1. For each topic k, draw a distribution over words: $\phi_k \sim \text{Dirichlet}(\beta)$
2. For each original document d, draw a distribution over topics: $\theta_d \sim \text{Dirichlet}(\alpha)$
3. For each short text pair $\{s, \hat{z}\}$
   a. For each of the N_s words w_{sn}
      i. Draw a topic: $z_{sn} \sim \text{Multinomial}\left(\theta_{\hat{z}_s}\right)$
      ii. Draw a word: $w_{sn} \sim \text{Multinomial}\left(\phi_{z_{sn}}\right)$

SATM versus LTM Both of them adaptively aggregates short texts into normal long texts, and alleviate the sparsity problem in short text topic modeling. However, there are some differences: (1) SATM is a two-stage topic model, where it first uses LDA to generate normal long texts, and then generates short texts using long text-specific distributions. This model structure is complex and rigid. In contrast, LTM doesn't change the structure of the candidate topic model, and the relationship between short texts and original documents is straightforward. (2) In SATM the words in a short text may be drawn from different normal long texts in the second phase, resulting in a loss of short text-level word co-occurrence information in some degree. (3) SATM involves two new latent variables, and LTM only involves one new latent variable (i.e., the original document assignment $\hat{z}$). LTM thus uses less memory. More discussions on space complexity are presented in Sect. 3.4.

3.2 Inference

Given a short text collection W, LTM needs to estimate the posterior distribution over the original document assignment for each short text $\hat{z}$ and three latent variables from LDA, including the topic-word distribution ϕ, the original document-topic distribution θ and the topic assignment for each word token z.

The exact posterior distribution is intractable to compute, we thus use Gibbs sampling to estimate an approximation. Following [8], we marginalize out the multinomial distributions ϕ and θ, and only sample the original document assignment $\hat{z}$ and the topic assignment z. Using the blocked Gibbs sampling framework, the two assignments can be alternately sampled by holding all other variables fixed. We directly present the final Gibbs sampling equations. Derivations are presented in "Appendix".

First, the conditional posterior probability for $\hat{z}$ over D original documents is given by:

$$p\left(\hat{z}_s = d|\hat{z}^{-s}, z, W, \alpha\right) \propto \frac{\prod_{k=1}^{K}\prod_{n=1}^{N_{sk}}\left(N_{dk}^{-s} + n - 1 + \alpha\right)}{\prod_{n=1}^{N_s}\left(N_d^{-s} + n - 1 + K\alpha\right)} \tag{1}$$

where N_{sk} and N_{dk} are the number of word tokens assigned to the topic k in the short text s and the original document d, respectively; N_s and N_d are the total number of word tokens in the short text s and the original document d, respectively; the superscript "$-s$" means a quantity that excludes the short text s.

Second, the conditional posterior probability for z over K topics is given by:

$$p\left(z_{dn} = k|\hat{z}, z^{-dn}, W, \beta, \alpha\right) \propto \frac{N_{kw_{dn}}^{-dn} + \beta}{N_k^{-dn} + V\beta}\left(N_{dk}^{-dn} + \alpha\right) \tag{2}$$

where w_{dn} is the *n-th* word in the original document d; $N_{kw_{dn}}$ is the number of word type w_{dn} assigned to the topic k and N_k is the total number of word tokens assigned to the topic k; the superscript "$-dn$" denotes a quantity that excludes z_{dn} from the position (d, n).

We alternatively sample $\hat{z}$ and z using Eqs. (1) and (2) until convergence. An algorithmic presentation of LTM is outlined in Algorithm 1.

Algorithm 1 Gibbs sampling for LTM

```
1: Initialization
2: For $t = 1, 2, \ldots, Max\_iter$ do
3:     For short text $s = 1$ to $S$ do
4:         Draw an original document $d$ using Eq. (1)
5:         For word token $n = 1$ to $N_s$ do
6:             Draw a topic $k$ using Eq. (2)
7:         End for
8:     End for
9: End for
```

Given final samples, the two distributions with respect to topics, i.e., θ and ϕ, can be estimated by:

$$\theta_{dk} = \frac{N_{dk} + \alpha}{N_d + K\alpha} \tag{3}$$

$$\phi_{kv} = \frac{N_{kv} + \beta}{N_k + V\beta} \tag{4}$$

Discussion We compare LDA with LTM from the model inference perspective. The samples they face are different, i.e., short text pairs for LTM and original documents for LDA. Abstractedly, the inference for LTM is similar to an expectation maximization inference for the pair $\{\hat{z}, LDA\}$. In the E-step, approximate the expectation of $\hat{z}$ by holding the current

LDA variables fixed, i.e., Eq. (1); and in the M-step, optimize LDA variables given $\hat{z}$ obtained in the previous E-step, i.e., Eq. (2). We can see that the essence of the E-step is to assign each short text to an original document. This is in fact an adaptive aggregation process for short texts. If all original document assignments $\hat{z}$ can be exactly estimated, LTM is equivalent to LDA.

3.3 Mini-batch scheme

To draw a sample $\hat{z}$ over D original documents, one must compute D corresponding probability values. That is, the time complexity of sampling $\hat{z}$ is proportional to the number of original documents D. If D is set to a very large value, this sampling is significantly time-consuming. Unfortunately, common sense tells us that there must be more original documents (i.e., larger D values) when modeling a bigger collection of short texts (i.e., larger S values). This limits the use of LTM for many real-world applications.

To accelerate the sampling of $\hat{z}$, we propose a mini-batch scheme inspired by [26]. The main idea is to randomly break the entire short text collection into a number of disjoint short text blocks, and short texts in different blocks are aggregated independently. For simplicity, only short text blocks of the same size are taken into consideration. We set that each block contains R short texts, and these R texts are aggregated into L original documents. As a result, in the sampling of $\hat{z}$, each short text only needs to sample over L original documents from its block, instead of all D original documents. Given a short text s belonging to the block b, its sampling equation about $\hat{z}_s$, i.e., Eq. (1), is replaced by:

$$p\left(\hat{z}_s = d_b | \hat{z}^{-s}, z, W, b, \alpha\right) \propto \frac{\prod_{k=1}^{K} \prod_{n=1}^{N_{sk}} \left(N_{d_b k}^{-s} + n - 1 + \alpha\right)}{\prod_{n=1}^{N_s} \left(N_{d_b}^{-s} + n - 1 + K\alpha\right)} \tag{5}$$

where d_b is the original document indicator belonging to the block b. An algorithmic presentation of LTM using the mini-batch scheme is outlined in Algorithm 2.

Algorithm 2 Gibbs sampling for LTM using mini-batch scheme

```
1: **Initialize** $K$, $R$ and $L$
2: Generate short text blocks $B$ randomly
3: **For** $t = 1, 2, \ldots, Max\_iter$ **do**
4:     **For** short text $s = 1$ to $S$ **do**
5:         **Draw** an original document $d_b$ using Eq. (5)
6:         **For** word token $n = 1$ to $N_s$ **do**
7:             **Draw** a topic $k$ using Eq. (2)
8:         **End for**
9:     **End for**
10: **End for**
```

Discussion on parallelization Following the mini-batch scheme, we straightforwardly implement a parallel version of LTM using GPUs (based on CUDA). At one iteration, Gibbs sampling updates for different short text blocks are parallel, and we synchronize global assignment counts until all sampling updates are finished. In our early experiments, we observed that the GPU version almost performs the same with the CPU version. and it yields significant speedups. For fair run-time comparison, we only report the run-time of the CPU LTM version in the experiment section.

Table 1 Time and space complexities of LDA, SATM and LTM with and without using the mini-batch scheme (abbr. LTM$_b$ and LTM$_{nb}$, respectively)

Model	Time complexity	Space complexity
LDA	$O(KN_V)$	$O(KD + KV + N_V)$
SATM	$O(DN_V + KDN_V)$	$O(KD + KV + SD + 2N_V)$
LTM$_b$	$O(KN_V + LN_V)$	$O(KD + KV + S + N_V)$
LTM$_{nb}$	$O(KN_V + DN_V)$	$O(KD + KV + S + N_V)$

3.4 Complexity analysis

This subsection presents the time and space complexities of Gibbs sampling for LDA, SATM and LTM with and without using the mini-batch scheme. These are listed in Table 1, where N_V is the total number of word tokens in a collection of short texts, i.e., $N_V = \sum_{s=1}^{S} N_s$.

Time complexity What we concern is the per-iteration time cost. For LDA, at one iteration each word token needs to sample a topic assignment from K topics, requiring $O(K)$ time. The overall time complexity of LDA is thus $O(KN_V)$. SATM first estimates a relation matrix between short texts and normal long texts, requiring $O(D \sum_{s=1}^{S} N_s)$ time, and then for each word token samples a pair of the normal long text assignment and topic assignment, requiring $O(KDN_V)$ time. The overall time complexity of SATM is thus $O(DN_V + KDN_V)$. Besides a same topic sampling as in LDA, LTM has to additionally draw original document assignments for short texts. Reviewing Eq. (1), for each short text s, the original document sampling uses $O(DN_s)$ time, leading to $O(D \sum_{s=1}^{S} N_s)$ time cost for all short texts. This gives an overall time complexity of $O(KN_V + DN_V)$ for LTM without using the mini-batch scheme. Thanks to the mini-batch scheme, the original document sampling reduces to $O(L \sum_{s=1}^{S} N_s)$ time, and its overall time complexity is thus $O(KN_V + LN_V)$. We see that the mini-batch scheme can effectively reduce the time complexity of LTM, owing to the fact $L << D$. Fortunately, empirical results indicated that small L values can achieve competitive performance, e.g., setting $L = 2^3$ in our experiments. In this sense, the run-time of LTM with the mini-batch scheme is even comparable with LDA.

Space complexity In terms of models we concern, the variables necessary to be cached include the count numbers, topic assignments, normal long text assignments and relation matrices. For LDA, we need to maintain the counts of word tokens assigned to topics at document level (i.e., $K \times D$), the counts of word types assigned to topics at corpus level (i.e., $K \times V$), and the topic assignment for each word token (i.e.,N_V) in memory. The space complexity of LDA is thus $O(KD + KV + N_V)$. Besides these variables, in SATM we additionally maintain a relation matrix between short texts and normal long texts (i.e., $S \times D$), and the normal long text assignment for each word token (i.e., N_V) in memory, giving a space complexity $O(KD + KV + SD + 2N_V)$. Turn to LTM, we only additionally maintain the original document assignment for each short text (i.e., S) in memory beyond LDA. This gives that the space complexity of LTM is $O(KD+KV+S+N_V)$. In summary, we can see that the memory cost of LTM is comparable with that of LDA, and LTM is much more memory-efficient than SATM.

Table 2 Statistics of the short text datasets, where *TotalV* and *AvgD* represent the total number of word tokens and average document length, respectively

Dataset	*D*	*V*	*TotalV*	*AvgD*
Snippets	12,340	30,452	215,367	17.5
Tweets	1,500,000	97,823	9,450,000	6.3
Newsgroup	323,149	88,857	2,108,284	6.5
NIPS	571,164	29,371	5,810,779	10.2

4 Related work

Besides the methods that aggregate short texts [9,14,20,24], other existing topic models for short texts are mixture of unigrams [10], biterm topic model (BTM) [6,28], and word network topic model (WNTM) [32]. Both BTM and WNTM consider the corpus level word co-occurrence information. BTM breaks short texts into biterms, and assumes that biterms are generated at corpus level. WNTM regroups a collection of short texts by constructing a global word co-occurrence network. Each word type is treated as a pseudo-document with content consisting of its adjacent word types in this network. They can alleviate the sparsity problem, however, BTM is sensitive with the average document length of a corpus, and WNTM is much less time-efficient for very large word co-occurrence networks. Some other recent topic models for short texts include Twitter-LDA [31], Twitter-TTM [22] and Twitter-BTM [5]. These models further improve short text topic modeling by introducing background topics to distill discriminative words. However, they are limited in real-world applications, because they are highly dependent on heuristics provided by tweets, e.g., user information.

5 Experiment

In this section, we empirically compare the propose LTM algorithm with the existing topic models for short texts.

Dataset Four datasets listed in Table 2 are used in our experiments. The *Snippets*[1] dataset is selected from the results of web search transaction using predefined phrases of 8 different categories [18]. The *Tweets* dataset is a collection of Twitter microblogs downloaded from the web. Additionally, we generate two short text datasets by breaking the normal long text datasets into sentences. The first normal dataset is *Newsgroup*,[2] which contains 18,821 normal long texts. The second normal dataset is *NIPS*, which contains 3066 NIPS (from 2000 to 2012) conference papers[3] without reference and bibliography sections. In the preprocessing, we remove the stop words.

Baseline model Four baseline topic models are used in our experiments, including LDA, Mixture of unigrams (Mix), BTM and SATM. For fair comparison, all models are learnt by Gibbs sampling. We implement in-house codes for LDA, Mix and SATM, and use the source code of BTM[4] provided by its authors. We set the iteration number of Gibbs sampling to 1000,

[1] http://jwebpro.sourceforge.net/data-web-snippets.tar.gz.

[2] http://web.ist.utl.pt/~acardoso/datasets/.

[3] http://papers.nips.cc/.

[4] http://code.google.com/p/btm/.

and the Dirichlet hyper-parameters α and β to $50/K$ and 0.01, respectively. In particular, for SATM, we set the number of long pseudo-documents to 1000 as suggested in [20]. For LTM, we use the mini-batch scheme with the setting of $(R, L) = (2^9, 2^3)$.

5.1 Qualitative Evaluation

This subsection presents qualitative evaluation results.

Topic visualization First, we select two *NIPS* topics (i.e., about computer vision and network, respectively) existing in all models, and directly present the 10 most probable words in Table 3. Overall, we can observe that the topics of LTM are more coherent than the topics learnt by baseline models. The topics of LDA contain some noisy words (e.g., "hopfield" and "result") and less relevant words (e.g., order). Mix, BTM and SATM perform better than LDA, however, they still output a few unrelated words (e.g., "delay", "weight" and "time"). In contrast, the topics of LTM are more coherent.

Word intrusion task Additionally, we qualitatively evaluate topics by the word intrusion task [4], In this task, each topic is represented by its top six words. Randomly remove one of six top words and then randomly select an intruder word to replace the removed top word. Users are asked to identify the intruder word. More coherent topics imply higher intrusion accuracy (AC) given by:

$$\mathrm{AC} = \frac{\sum_{k=1}^{K} \mathbb{1}(i_k = w_k)}{K} \tag{6}$$

where i_k is the intruder word of the topic k selected by human beings, and w_k is the true intruder word of the topic k. For convenience, we use an automatic intruder detector [11], which can emulate the performance of human judgements.

The results of AC are shown in Fig. 2. In terms of the two sentence datasets, LTM outperforms baseline models in all the settings. For example, LTM has AC values on *NIPS* about 0.02 better than those of BTM and AC values on *Newsgroup* about 0.03 better than those of SATM. Roughly, Mix, BTM and SATM are competitive with each other, and they perform about $0.01 \sim 0.04$ better than LDA. In terms of the two real-world datasets, we see very similar AC results. LTM is competitive with BTM, and outperforms other three baseline models. Compared with BTM, the AC results of LTM are about 0.015 higher on *Tweets*, and are slightly 0.01 lower on *Snippets*. That may be because *Snippets* contains fewer texts, resulting in difficulty in aggregating short texts. Overall, our LTM gains better word intrusion performance than existing models. Further paired t tests (p value $<$0.01) indicate that the improvement over baseline models is statistically significant.

5.2 Evaluation on TC and PMI

We evaluate whether the topics learnt by LTM are quantitatively coherent. To this end, we employ two metrics, i.e., topic coherence (TC) [15] and pointwise mutual information (PMI) [16,21]. Both metrics follow the assumption that a topic is more coherent if its most probable words are more frequently co-occurring. A basic difference is that the word co-occurrences of TC are computed on the current dataset, but those of PMI are computed on an extra (large-scale) reference corpus. Given a topic k and its M most probable words, the TC value of topic k is given by:

$$\mathrm{TC}(k) = \sum_{m=2}^{M} \sum_{l=1}^{m-1} \log \frac{D\left(v_m^k, v_l^k\right) + 1}{D\left(v_l^k\right)} \tag{7}$$

Table 3 The top 10 word lists on *NIPS*

LDA	Mix	BTM	SATM	LTM
Image, images, vision, feature, error, visual, figure, **result**, **hopfield**, face	Image, vision, images, feature, **delay**, visual, clustering, face, training, classifier	Image, images, vision, clustering, feature, visual, **weight**, learning, face , motion	Image, figure, images, feature, **time**, space, visual, set, pixel, vision	Image, video, vision, feature, classification, visual, recognition, training, face , motion
Network, neuron, **order**, output, figure, **result**, **experiment**, training, hidden, node	Network, learning, input, output, activation, **result**, error, learning, hidden, node	Network, neurons, learning, synaptic, output, **result**, figure, activation	Network, neuron, neurons, **state**, input, networks, output, dynamics, figure, function	Network, neuron, layer, neural, networks, output, training, input, node, hidden

The top section is a topic about computer vision, and the bottom section is a topic about network. Words in boldface are less relevant or noisy words

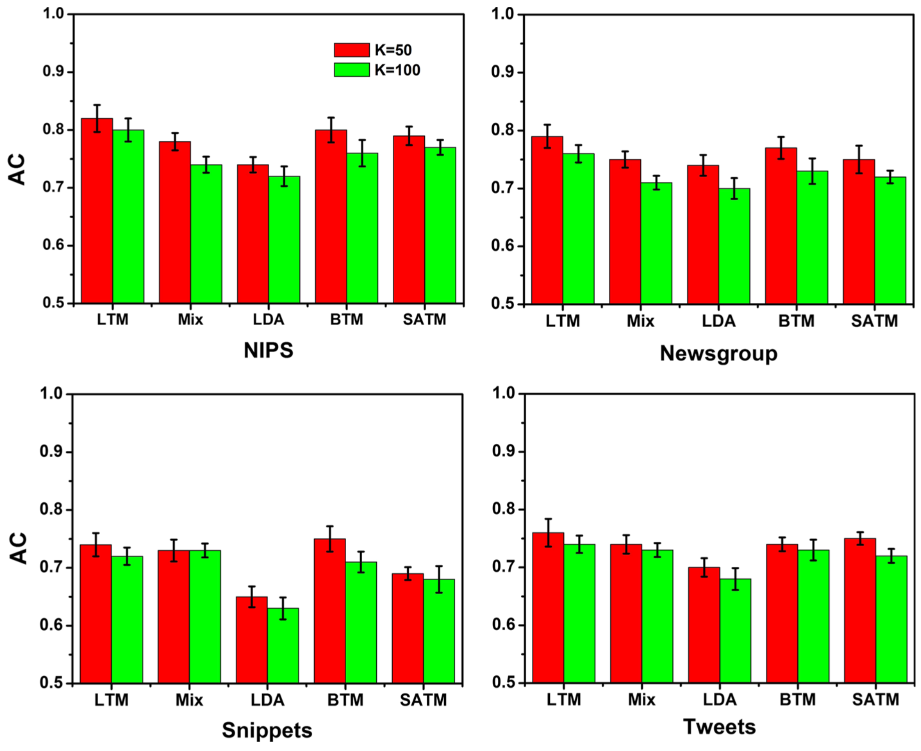

Fig. 2 The AC performance of the word intrusion task. A higher value implies better performance

where $D(v)$ is the number of documents containing word type v; $D(v_1, v_2)$ is the number of documents containing both word type v_1 and v_2. Meanwhile, the PMI value of topic k is given by:

$$\mathrm{PMI}(k) = \frac{1}{M(M-1)} \sum_{1 \le i \le j \le M} \log \frac{p\left(w_i, w_j\right)}{p(w_i)\, p\left(w_j\right)} \tag{8}$$

where $p(w_i)$ and $p\left(w_i, w_j\right)$ are the probabilities of occurring word w_i and co-occurring word pair $\left(w_i, w_j\right)$ computed on the reference corpus, respectively. For both metrics, we set M to 10, and report the average values over all K topics finally.

The results of TC are shown in Fig. 3. Roughly, we see that the performance order is given by LTM > SATM ≈ BTM > Mix > LDA. For example, the TC value of LTM is about 8 better than that of SATM on *NIPS* when $K = 50$, and about 5 better than that of BTM on *Tweets* when $K = 100$. The TC values of Mix are always higher than those of LDA, agreeing with previous empirical results in [10]. It is surprisingly that BTM performs even obviously worse than LDA on *NIPS*. The possible reason is that the average document length of *NIPS* attains 10.2, resulting in about 45 ($\frac{10\times 9}{2} = 45$) biterms for each short text averagely. This introduces many low-quality biterms. The results of PMI are shown in Fig. 4. Similar with TC results, the performance order of PMI is also roughly given by LTM > SATM ≈ BTM > Mix > LDA. SATM and BTM are competitive with each other, and they gain better PMI scores than Mix.

In summary, LTM outperforms baseline models in the most settings. In particular, the empirical improvement over SATM indicates that LTM drives a more reasonable adaptive

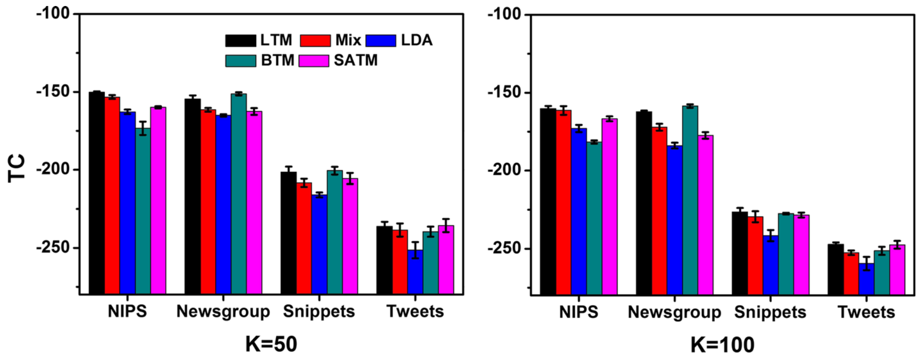

Fig. 3 TC performance of LDA, Mix, BTM, SATM and LTM. A higher value implies better performance

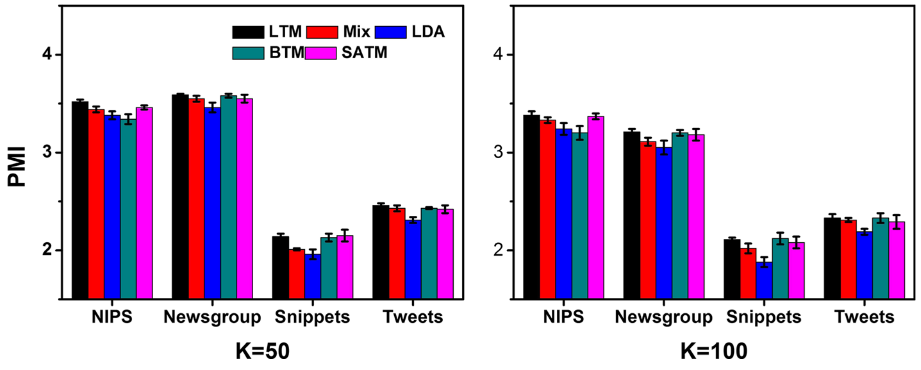

Fig. 4 PMI performance of LDA, Mix, BTM, SATM and LTM. A higher value implies better performance

aggregation process. The performance of BTM heavily drops down on *NIPS*, whose average document length is relatively longer (i.e., 10.2). In contrast, LTM is stronger in robustness. Additionally, the results of paired t tests (p value <0.01) indicate that the improvement over baseline models is statistically significant.

5.3 Evaluation on purity

We evaluate whether the topics learnt from the short text dataset are in maximum alignment with the topics learnt from the corresponding normal dataset. A purity metric [20], coming from the clustering problem, is used. Taking LTM as an example, we train a LTM model on the short text dataset, obtaining the topics $\mathcal{T}^s$, and simultaneously train a LDA model with the same number of topics on the corresponding normal dataset, obtaining the topics $\mathcal{T}^l$. The purity score of LTM can be computed by:

$$Purity = \frac{1}{KM} \sum_{i=1}^{K} \max_j \left| \mathcal{T}_i^s \cap \mathcal{T}_j^l \right| \tag{9}$$

Note that in this definition, only the most M probable words are compared between topics in $\mathcal{T}^s$ and $\mathcal{T}^l$. We set M to 10.

We train all models on *NIPS* and *Newsgroup*, and compare the purity scores of models in Fig. 5. In terms of *NIPS*, we can see that the purity scores of LTM are higher than those of

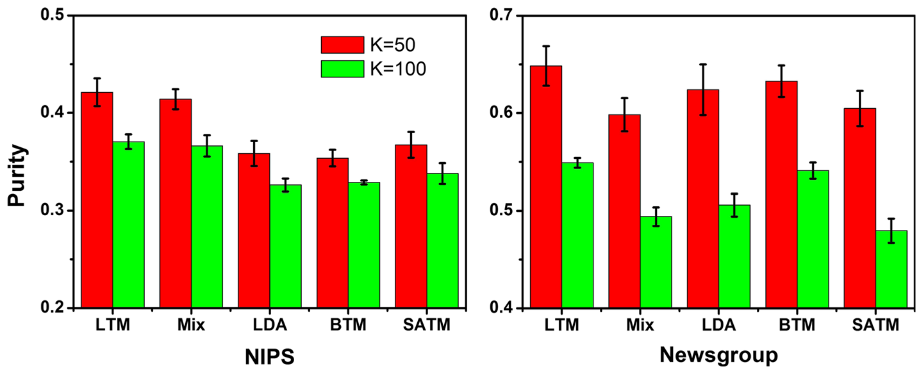

Fig. 5 Purity performance of LDA, Mix, BTM, SATM and LTM. A higher value implies better performance

LDA, BTM and SATM, and a slightly higher than those of Mix, e.g., about 0.01 improvement over Mix while $K = 50$. BTM still performs worse than LDA, since the average document length of *NIPS* is relatively longer. In terms of *Newsgroup*, we still see that LTM outperforms all baseline models, e.g., about 0.04 improvement over Mix and about 0.02 improvement over LDA. Different from early results, the purity scores of LDA are higher than those of Mix. The possible reason is that many sentences in the original *Newsgroup* is incomplete, resulting in many meaningless short texts. This seriously influences the modeling performance of Mix.

In summary, LTM outperforms baseline models in all the settings. The improvement over SATM further testifies that LTM drives a more reasonable adaptive aggregation process. The results of paired t tests (p value <0.01) guarantee that the improvements over baseline models is statistically significant.

5.4 Evaluation on clustering and classification

We evaluate LTM by clustering and classification tasks on *Snippets*, which is labeled.

The metric used for clustering is rand index (RI) given by:

$$\mathrm{IR} = \frac{2 \times (TP + TN)}{S \times (S - 1)} \tag{10}$$

where TP denotes the number of text pairs that are in the same cluster both in predicted results and the ground truth; TN denotes the number of text pairs that don't occur in the same cluster in predicted results and the ground truth. The cluster (topic) number is set to 8, which is the true category number of *Snippets*. The IR results are shown in Fig. 6a. It can be seen that the IR value of LTM is higher than those of LDA, Mix and SATM, and slightly lower than that of BTM.

The metric used for classification is classification accuracy (CAC). For all models, the topical representation $p(k|d)$ [6] is used as the feature vector, and SVMs implemented by LibSVM[5] is employed as the downstream classifier. The parameters of SVMs are tuned by a grid search on the set $\{10^i | i = -5, -4, \ldots, 4, 5\}$. The average CAC values of fivefold cross-validation are reported in Fig. 6b. We see that LTM outperforms baseline models, e.g., about 0.015 better than BTM and about 0.01 better than SATM. Additionally, the performance of LTM is more stable as the number of topics K increases. For both IR and CAC experiments,

5 http://www.csie.ntu.edu.tw/~cjlin/libsvm/.

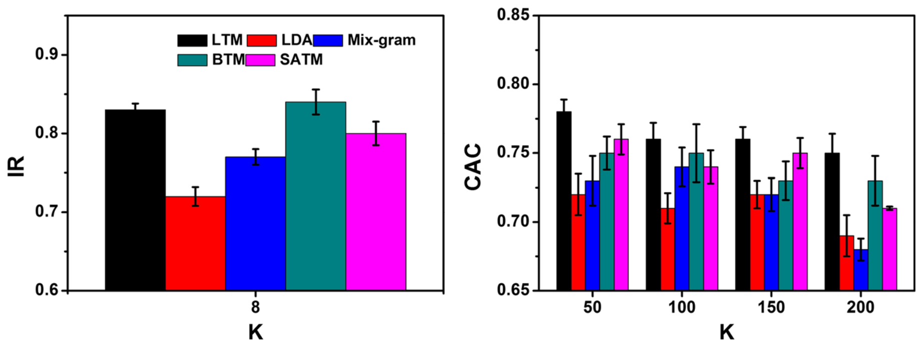

Fig. 6 IR and CAC performance of LDA, Mix, BTM, SATM and LTM. Higher values imply better performance

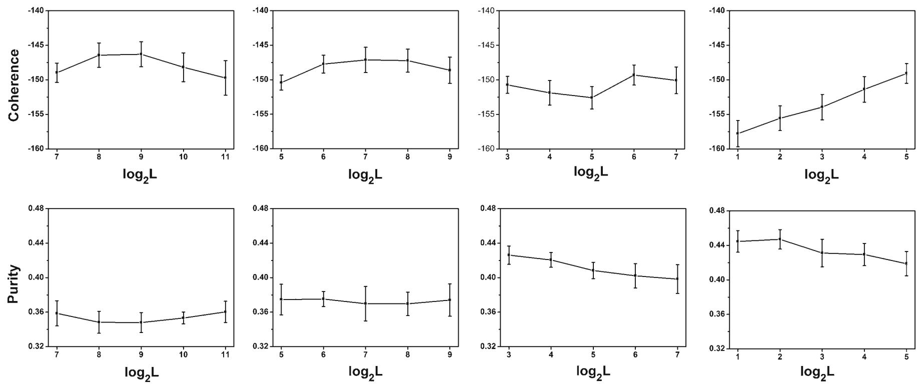

Fig. 7 Performance on different values of parameters R and L across *NIPS*. The first/second/third/fourth column shows the results while $R = 2^{13}/2^{11}/2^9/2^7$

the results of paired t tests (p value <0.01) indicate that the improvement over baseline models is statistically significant.

5.5 Evaluation on LTM parameters

In this subsection, we empirically investigate how to set the two LTM parameters: R and L. For this objective, we train 50-topic LTM models, and compare both TC and purity values on *NIPS* with different R and L combinations, where $R \in \left\{2^i | i = 7, 9, 11, 13\right\}$ and $L \in \left\{2^{i-j} | j = 2, 3, 4, 5, 6\right\}$ given $R = 2^i$. The experimental results are shown in Fig. 7.

First, we analyze the setting of R. Overall, the performance trends on the two evaluation metrics are somewhat different. For TC, the results of larger short text blocks (i.e., larger R) are better than these of smaller blocks (i.e., smaller R). We can see that the TC values are almost above -150 when $R = 2^{13}, 2^{11}$, but below -150 when $R = 2^9, 2^7$. This indicates that exploring original documents over a wide range is beneficial for generating high quality topics. For purity, smaller short text blocks perform better. The purity performance boundary between $R = 2^9, 2^7$ and $R = 2^{13}, 2^{11}$ is 0.4. The nature of the purity metric is to measure whether the topics learnt from short texts are closer to the topics learnt from the corresponding normal long texts (i.e., gold-standard topics). So the purity results in Fig. 7 (bottom) are in

Table 4 Per-iteration time cost (s) of Mix, LDA, BTM and LTM

Dataset	LDA	Mix	BTM	LTM
NIPS	6.45	5.9	37.2	14.1
Newsgroup	2.74	2.46	8.3	4.37

conveying that setting relatively smaller R values are enough to close the gold-standard topics in some degree.

Second, we analyze the setting of L. In our experiments, we find that compared to L, the $\frac{R}{L}$ value (i.e, the average original document length) is more significant. For TC, smaller $\frac{R}{L}$ values perform better in the most settings, e.g., -147.2 with $(R, L) = (2^{11}, 2^8)$ versus -150.4 with $(R, L) = (2^{11}, 2^5)$, and -150.5 with $(R, L) = (2^9, 2^7)$ versus -154.6 with $(R, L) = (2^9, 2^5)$. For purity, smaller $\frac{R}{L}$ values perform worse, but the performance gap is not obvious, e.g., the gap between $(R, L) = (2^9, 2^5)$ and $(R, L) = (2^9, 2^7)$ is only about 0.05. These interesting results indicate that aggregating fewer short texts is positive for topic modeling.

Summary Because the two evaluation metrics, i.e., TC and purity, follow different assumptions, we see some conflict results between the two in some settings. For the parameter R, the TC metric prefers larger R values, but the purity metric prefers smaller R values. Fortunately, we see that the results of both metrics are acceptable and stable when $R = 2^9$. For the parameter L, fixed R we see that the results of both metrics are almost at high level, e.g., -150.8 TC and 0.426 purity with $(R, L) = (2^9, 2^3)$. Further considering the run-time, we prefer smaller L values. We thus use the default setting of $(R, L) = (2^9, 2^3)$ in the above experiments.

5.6 Run-time

This subsection presents run-time comparisons among four relatively faster models: Mix, LDA, BTM and LTM. We run each model with 50 topics two hours regardless of the results. In our early experiments, we observed that the convergence curves of models have no obvious difference. We thus only report the average per-iteration run-time.

As shown in Table 4, we can see that LDA and Mix are faster than BTM and LTM. Although our LTM drives an additional original document exploration procedure, the run-time of LTM is only about 2.2 and 1.8 times of LDA on *NIPS* and *Newsgroup* in practice, respectively. This profits from the mini-batch scheme that significantly reduces the time cost. LTM runs faster than BTM. For example, BTM needs 8.3 s to finish one iteration on *Newsgroup*, and in contrast LTM only needs 4.37 s. The run-time of BTM on *NIPS* attains 37.2 s. It is much slower than other models (e.g., 6.45 s for LDA and 14.1 seconds for LTM). That is because the average document length of *NIPS* is relatively longer, resulting in too many biterms in BTM.

6 Conclusion and discussion

We investigate topic modeling for short texts. Following the idea of adaptively aggregate short texts into long pseudo-documents, we propose a generalized short text topic model, namely LTM. The proposed LTM is based on the fragmentary original document assumption. If a candidate topic model is chosen, the inference of LTM is equivalent to iteratively exploring the original documents and inferring the parameters of the candidate topic model. Experimental results indicate that our LTM is on a par with the state-of-the-art topic models for short texts.

In the future, we plan to further investigate the following issues.

1. In this work, we only refer to the candidate topic model as the standard LDA model. Actually, other representative topic models, such as correlated topic model [2] and hierarchical Dirichlet process [25], can be directly used as the candidate topic model in LTM. It is interesting to evaluate whether based on these models LTM can achieve better topic modeling performance for short texts.
2. Another problem we concern is that the mini-batch scheme of LTM introduces two new parameters (i.e., R and L), however it lacks setting guidelines for them in theory. This is also the very problem in SATM, which lacks guidelines to set the number of pseudo-documents.
3. Finally, a new raised issue is bursty topic discovery. For short texts such as microblog stream, a topic becomes bursty in a certain time slice if it is heavily discussed in that time slice, but not in most of other time slices [29]. In the future, we plan to investigate how to capture such bursty topics.

Acknowledgements This work was supported by the National Natural Science Foundation of China (NSFC) Grant Numbers 61602204 and 61472157).

Appendix

A. Derivation of Gibbs sampling LTM

We derive the Gibbs sampling equations for LTM. In terms of LTM, four latent variables of interest exist, including the topic-word distribution ϕ, the original document-topic distribution θ, the original document assignment for short texts $\hat{z}$ and the topic assignment for word tokens z. Thanks to the conjugate Dirichlet-Multinomial design for ϕ and θ, we can effectively marginalize out these two variables, and turn to the joint distribution of the observation W and the two assignment variables $\hat{z}$ and z as follows:

$$\begin{aligned}
p\left(W, \hat{z}, z|\beta, \alpha\right) &= \int\int p\left(W, \phi, \theta, \hat{z}, z|\beta, \alpha\right) \mathrm{d}\phi\mathrm{d}\theta \\
&= \int\int \prod_{k=1}^{K} Dir(\phi|\beta) \prod_{d=1}^{D} Dir(\theta|\alpha) \prod_{v=1}^{V}\prod_{k=1}^{K}\prod_{d=1}^{D} \phi_{kv}^{N_{kv}} \theta_{dk}^{N_{dk}} \mathrm{d}\phi\mathrm{d}\theta \\
&= \left(\prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma\left(N_{kv}+\beta\right)}{\Gamma\left(N_k+V\beta\right)} \frac{\Gamma\left(V\beta\right)}{\prod_{v=1}^{V}\Gamma\left(\beta\right)}\right) \left(\prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma\left(N_{dk}+\alpha\right)}{\Gamma\left(N_d+K\alpha\right)} \frac{\Gamma\left(K\alpha\right)}{\prod_{k=1}^{K}\Gamma\left(\alpha\right)}\right) \\
&\propto \left(\prod_{k=1}^{K} \frac{\prod_{v=1}^{V} \Gamma\left(N_{kv}+\beta\right)}{\Gamma\left(N_k+V\beta\right)}\right) \left(\prod_{d=1}^{D} \frac{\prod_{k=1}^{K} \Gamma\left(N_{dk}+\alpha\right)}{\Gamma\left(N_d+K\alpha\right)}\right) \\
&\stackrel{\Delta}{=} B\left(N_{kv}, N_{k,}\beta\right) B\left(N_{dk}, N_{d,}\alpha\right)
\end{aligned} \tag{11}$$

where the fourth line in Eq. (11) follows that β and α are constant; notations $B\left(N_{kv}, N_{k,}\beta\right)$ and $B\left(N_{dk}, N_{d,}\alpha\right)$ are used to denote $\prod_{k=1}^{K} \frac{\prod_{v=1}^{V}\Gamma(N_{kv}+\beta)}{\Gamma(N_k+V\beta)}$ and $\prod_{d=1}^{D} \frac{\prod_{k=1}^{K}\Gamma(N_{dk}+\alpha)}{\Gamma(N_d+K\alpha)}$ for convenience.

We employ the blocked Gibbs sampling framework, where in each iteration $\hat{z}$ and z are alternately sampled given the other one. We derive the Gibbs sampling equations for $\hat{z}$ and z one by one.

Sampling equation of $\hat{z}$ given the current z To draw a sample of $\hat{z}$, following Gibbs sampling idea we consecutively draw a single original document assignment $\hat{z}_s$ from a posterior conditioned on all other original document assignments $\hat{z}^{-s}$:

$$\begin{aligned} p\left(\hat{z}_s|\hat{z}^{-s}, z, W, \beta, \alpha\right) &= \frac{p\left(W, \hat{z}|z, \beta, \alpha\right)}{p\left(W, \hat{z}^{-s}|z, \beta, \alpha\right)} \propto \frac{p\left(W, \hat{z}|z, \beta, \alpha\right)}{p\left(W^{-s}, \hat{z}^{-s}|z, \beta, \alpha\right)} \\ &= \frac{p\left(W, \hat{z}, z|\beta, \alpha\right)}{p\left(W^{-s}, \hat{z}^{-s}, z|\beta, \alpha\right)} \propto \frac{p\left(W, \hat{z}, z|\beta, \alpha\right)}{p\left(W^{-s}, \hat{z}^{-s}, z^{-s}|\beta, \alpha\right)} \end{aligned} \tag{12}$$

By combing Eqs. (11) and (12), we have:

$$\begin{aligned} p\left(\hat{z}_s = d|\hat{z}^{-s}, z, W, \beta, \alpha\right) &\propto \frac{B\left(N_{kv}, N_k, \beta\right) B\left(N_{dk}, N_d, \alpha\right)}{B\left(N_{kv}^{-s}, N_k^{-s}, \beta\right) B\left(N_{dk}^{-s}, N_d^{-s}, \alpha\right)} \\ &\propto \frac{B\left(N_{dk}, N_d, \alpha\right)}{B\left(N_{dk}^{-s}, N_d^{-s}, \alpha\right)} \end{aligned} \tag{13}$$

The second line in Eq. (13) follows that the terms with respect to topic-word counts are independent of the current assignment $\hat{z}_s$. We then expand Eq. (13) and obtain the final Gibbs sampling equation for $\hat{z}$ (i.e., Eq. 1):

$$\begin{aligned} p\left(\hat{z}_s = d|\hat{z}^{-s}, z, W, \alpha\right) &\propto \frac{\prod_{j=1}^{D} \frac{\prod_{k=1}^{K} \Gamma(N_{jk}+\alpha)}{\Gamma(N_j+K\alpha)}}{\frac{\prod_{k=1}^{K} \Gamma(N_{dk}^{-s}+\alpha)}{\Gamma(N_d^{-s}+K\alpha)} \prod_{j\neq d} \frac{\prod_{k=1}^{K} \Gamma(N_{jk}+\alpha)}{\Gamma(N_j+K\alpha)}} \\ &= \frac{\prod_{k=1}^{K} \Gamma\left(N_{dk}+\alpha\right)}{\prod_{k=1}^{K} \Gamma\left(N_{dk}^{-s}+\alpha\right)} \frac{\Gamma\left(N_d^{-s}+K\alpha\right)}{\Gamma\left(N_d+K\alpha\right)} \\ &= \frac{\prod_{k=1}^{K} \prod_{n=1}^{N_{sk}} \left(N_{dk}^{-s}+n-1+\alpha\right)}{\prod_{n=1}^{N_s} \left(N_d^{-s}+n-1+K\alpha\right)} \end{aligned} \tag{14}$$

The third line in Eq. (14) follows the fact ($m > n$):

$$\frac{\Gamma(n)}{\Gamma(m)} = \frac{\Gamma(n)}{\Gamma(n+1)} \frac{\Gamma(n+1)}{\Gamma(n+2)} \frac{\Gamma(n+2)}{\Gamma(n+3)} \cdots \frac{\Gamma(m-1)}{\Gamma(m)} = \frac{1}{\prod_{i=1}^{m-n} (n+i-1)}$$

Sampling equation of z given the current $\hat{z}$ We now turn to the sampling of z. Note that given $\hat{z}$, the case is completely equivalent to the standard LDA Gibbs sampling. Similar with the derivations of Eqs. (12) and (13), the posterior of a single topic assignment z_{dn} conditioned on all other topic assignments z^{-dn} is given by:

$$\begin{aligned} p\left(z_{dn}|\hat{z}, z, W, \beta, \alpha\right) &\propto \frac{p\left(W, \hat{z}, z|\beta, \alpha\right)}{p\left(W^{-s}, \hat{z}^{-s}, z^{-s}|\beta, \alpha\right)} \\ &= \frac{B\left(N_{kv}, N_k, \beta\right) B\left(N_{dk}, N_d, \alpha\right)}{B\left(N_{kv}^{-s}, N_k^{-s}, \beta\right) B\left(N_{dk}^{-s}, N_d^{-s}, \alpha\right)} \end{aligned} \tag{15}$$

By expanding Eq. (15), we derive the final Gibbs sampling equation for z (i.e., Eq. 2):

$$p\left(z_{dn}=k|\hat{z},z^{-dn},W,\beta,\alpha\right)\propto\frac{\frac{\Gamma(N_{kw_{dn}}+\beta)}{\Gamma(N_k+V\beta)}\frac{\Gamma(N_{kw_{dn}}+\beta)}{\Gamma(N_k+V\beta)}}{\frac{\Gamma\left(N_{kw_{dn}}^{-dn}+\beta\right)}{\Gamma\left(N_k^{-dn}+V\beta\right)}\frac{\Gamma\left(N_{dk}^{-dn}+\alpha\right)}{\Gamma\left(N_d^{-dn}+K\alpha\right)}}$$
$$=\frac{N_{kw_{dn}}^{-dn}+\beta}{N_k^{-dn}+V\beta}\frac{N_{dk}^{-dn}+\alpha}{N_d^{-dn}+K\alpha}$$
$$\propto\frac{N_{kw_{dn}}^{-dn}+\beta}{N_k^{-dn}+V\beta}\left(N_{dk}^{-dn}+\alpha\right) \quad (16)$$

The third line in Eq. (16) follows that the term $(N_d^{-dn}+K\alpha)$ is independent of the current topic assignment z_{dn}.

References

1. Bouma G (2009) Normalized (pointwise) mutual information in collocation extraction. In: Proceedings of the Biennial GSCL conference, pp 31–40
2. Blei DM, Lafferty JD (2007) A correlated topic model fo science. Ann Appl Stat 1(2):17–35
3. Blei DM, Ng AY, Jordan MI (2003) Latent Dirichlet allocation. J Mach Learn Res 3:993–1022
4. Chang J, Boyd-Graber J, Gerrish S, Wang C, Blei DM (2009) Reading tea leaves: How humans interpret topic models. In: Neural information processing systems, pp 288–296
5. Chen W, Wang J, Zhang Y, Yan H, Li X (2015) User based aggregation for biterm topic model. In: Annual meeting of the association for computational linguistics and international joint conference on natural language processing of the Asian Federation of Natural Language Processing, pp 489–494
6. Cheng X, Yan X, Lan Y, Guo J (2014) BTM: topic modeling over short texts. IEEE Trans Knowl Data Eng 26(12):2928–2941
7. Gangemi A, Presutti V, Recupero DR (2014) Frame-based detection of opinion holders and topics: a model and a tool. IEEE Comput Intell Mag 9(1):20–30
8. Griffiths TL, Steyvers M (2004) Finding scientific topics. Natl Acad Sci USA 101(Suppl. 1):5228–5235
9. Hong L, Davison BD (2010) Empirical study of topic modeling in Twitter. In: Proceedings of the first workshop on social media analytics. ACM, New York, pp 80–88
10. Lakkaraju H, Bhattacharya I, Bhattacharyys C (2012) Dynamic multi-relational Chinese restaurant process for analyzing influences on users in social media. In: International conference on data mining. IEEE, pp 389–398
11. Lau JH, Newman D, Baldwin T (2014) Machine reading tea leaves: automatically evaluating topic coherence and topic model quality. In: Conference of the European chapter of the Association for Computational Linguistics, pp 530–539
12. Lau RYK, Xia Y, Ye Y (2014) A probabilistic generative model for mining cybercriminal networks from online social media. IEEE Comput Intell Mag 9(1):31–43
13. Li X, Ouyang J, You L, Zhou X, Tian T (2015) Group topic model: organizing topics into groups. Inf Retr J 18(1):1–25
14. Mehrotra R, Sanner S, Buntine W, Xie L (2013) Improving LDA topic models for microblogs via tweet pooling and automatic labeling. In: International ACM SIGIR conference on research and development in information retrieval. ACM, New York, pp 889–892
15. Mimno D, Wallach HM, Talley E, Leenders M, McCallum A (2011) Optimizing semantic coherence in topic models. In: Conference on empirical methods in natural language processing, pp 262–272
16. Newman D, Lau HJ, Grieser K, Baldwin T (2010) Automatic evaluation of topic coherence. In: Conference of the North American chapter of the Association for Computational Linguistics: Human Language Technologies, pp 100–108
17. Nigam K, Mccallum AK, Thrun S, Mitchell T (2000) Text classification from labeled and unlabeled documents using EM. Mach Learn 39(2):103–134
18. Phan X, Nguyen, L (2008) Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In: International conference on world wide web. ACM, New York, pp 91–100

19. Poria S, Chaturvedi I, Cambria E, Bisio F (2016) Sentic LDA: improving on LDA with semantic similarity for aspect-based sentiment analysis. In: International joint conference on neural networks, pp 4465–4473
20. Quan X, Kit C, Ge Y, Pan SJ (2015) Short and sparse text topic modeling via self-aggregation. In: International joint conference on artificial intelligence. AAAI Press, pp 2270–2276
21. Roder M, Both A, Hinneburg A (2015) Exploring the space of topic coherence measures. In: ACM international conference on web search and data mining, pp 399–408
22. Sasaki K, Yoshikawa T, Furuhashi T (2014) Twitter-TTM: an efficient online topic modeling for Twitter considering dynamics of user interests and topic trends. In: 15th international symposium on soft computing and intelligent systems (SCIS), joint 7th international conference on and advanced intelligent systems (ISIS). IEEE, pp 440–445
23. Sridhar VKR (2015) Unsupervised topic modeling for short texts using distributed representations of words. In: Proceedings of NAACL-HLT, pp 192–200
24. Weng J, Lim E, Jiang J, He Q (2010) TwitterRank: finding topic-sensitive influential twitterers. In: Proceedings of the third ACM international conference on web search and data mining. ACM, New York, pp 261–270
25. Teh YW, Jordan MI, Beal MG, Blei DM (2007) Hierarchical Dirichlet Processes. J Am Stat Assoc 101(476):1566–1581
26. Tsoumakas G, Katakis I, Vlahavas I (2011) Random k-labelsets for multilabel classification. IEEE Trans Knowl Data Eng 23(7):1079–1089
27. Xia Y, Tang N, Hussain A, Cambria E (2015) Discriminative bi-term topic model for headline-based social news clustering. In: International Florida artificial intelligence research society conference. AAAI Press, pp 311–316
28. Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: international conference on world wide web. ACM, New York, pp 1445–1456
29. Yan X, Guo J, Lan Y, Xu J, Cheng X (2015) A probabilistic model for bursty topic discovery in microblogs. In: Association for the advancement of artificial intelligence, pp 353–359
30. Yin J, Wang J (2014) A Dirichlet multinomial mixture model-based approach for short text clustering. In: ACM SIGKDD international conference on knowledge discovery and data mining. ACM, New York, pp 233–242
31. Zhao WX, Jiang J, Weng J, He J, Lim E, Yan H, Li X (2011) Comparing Twitter and traditional media using topic models. In: Proceedings of the 33rd European conference on advances in information retrieval. Springer, Heidelberg, pp 338–349
32. Zuo Y, Zhao J, Xu K (2016) Word network topic model: a simple but general solution for short and imbalanced texts. Knowl Inf Syst 48(2):379–398

Ximing Li is currently Assistant Professor at the College of Computer Science and Technology, Jilin University of China. He received his Ph.D. degree in Jilin university in 2011. His main research interests include machine learning, more specifically in topic modeling and approximating inference.

Changchun Li received the B.S. degree in Computer Science from Jilin University, China, in 2015. Currently he is an M.S. candidate in the College of Computer Science and Technology at Jilin University. His main research interests include topic modeling and classification.

Jinjin Chi received the M.S. degree in Computer Science from Jilin University, China, in 2015. Currently she is a Ph.D. candidate in the College of Computer Science and Technology at Jilin University. Her main research interests include data mining and topic modeling.

Jihong Ouyang is a professor at the College of Computer Science and Technology, Jilin University of China. She received her Ph.D. degree in Jilin university in 2005. Her main research interests include artificial intelligence and machine learning, more specifically in spatial reasoning, multi-label learning, topic modeling, and online learning.