Information Sciences 585 (2022) 176–193

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Parallel dynamic topic modeling via evolving topic adjustment and term weighting scheme

Hongyu Jiang [a], Zhiqi Lei [a], Yanghui Rao [a,*], Haoran Xie [b], Fu Lee Wang [c]

[a] School of Computer Science and Engineering, Sun Yat-sen University, Guangzhou, China
[b] Department of Computing and Decision Sciences, Lingnan University, Hong Kong
[c] School of Science and Technology, Hong Kong Metropolitan University, Hong Kong

ARTICLE INFO

Article history:
Received 13 April 2020
Received in revised form 26 October 2021
Accepted 19 November 2021
Available online 26 November 2021

Keywords:
Dynamic topic model
Term weighting scheme
Parallel gibbs sampling

ABSTRACT

The parallel Hierarchical Dirichlet Process (pHDP) is an efficient topic model which explores the equivalence of the generation process between Hierarchical Dirichlet Process (HDP) and Gamma-Gamma-Poisson Process (G2PP), in order to achieve parallelism at the topic level. Unfortunately, pHDP loses the non-parametric feature of HDP, i.e., the number of topics in pHDP is predetermined and fixed. Furthermore, under the bootstrap structure of pHDP, the topic-indiscriminate words are of high probabilities to be assigned to different topics, resulting in poor qualities of the extracted topics. To achieve parallelism without sacrificing the non-parametric feature of HDP, in addition to improve the quality of extracted topics, we propose a parallel dynamic topic model by developing an adjustment mechanism of evolving topics and reducing the sampling probabilities of topic-indiscriminate words. Both supervised and unsupervised experiments on benchmark datasets show the competitive performance of our model.

1. Introduction

The development of the Internet has led to an ever-increasing scale of textual information. As convenient unsupervised learning methods, topic models have been developed to process large amounts of documents and effectively extract key information from them. Such models can mine semantic relationships among words by introducing latent topics. In order to deal with large-scale text data from sources such as social media and academic papers, recent topic models focus on the following 3 aspects: parallelization, dynamic changes of topics, and improvement of the extracted topic quality.

First, traditional topic models like the Latent Dirichlet Allocation (LDA) [4] and the Hierarchical Dirichlet Process (HDP) [28] are difficult to handle large-scale datasets in many applications due to their high time costs in parameter estimation. To address this problem, some parallelization methods have been proposed. These works focused on developing either the parallel Gibbs sampling algorithm for LDA [11,27,2] such as the Slice sampler [29], or the parallel MCMC method for the non-parametric Dirichlet mixture model with auxiliary variables [32]. However, there is still room for improvement because of the high consumption of the synchronization and the increasing rejection rate in the Metropolis Hasting step as the number of processors increases [8]. Recently, the parallel Hierarchical Dirichlet Process (pHDP) [8] was proposed to parallelize HDP by exploring the equivalence of the generation process between HDP and Gamma-Gamma-Poisson process (G2PP). The

* Corresponding author.
E-mail address: raoyangh@mail.sysu.edu.cn (Y. Rao).

https://doi.org/10.1016/j.ins.2021.11.060
0020-0255/

pHDP model implements parallelism at the topic level and shows a rich potential for developing various parallel inference algorithms. Unfortunately, the number of topics in pHDP must be predetermined and fixed.

Second, timestamp is an essential attribute for many texts (e.g., news articles and academic papers). As the publication time goes by, new topics will appear while marginal topics [20] will disappear. Although pHDP provides an efficient parallel solution, it lacks the ability to model dynamically evolved topics. Therefore, we propose a mechanism for dynamic adjustment of topics that can monitor the evolution of topics, so that the model can be applied to real-world topic extraction tasks on streaming datasets.

Third, improving the quality of extracted topics is also a major challenge for topic models. The higher the quality of extracted topics, the easier it is for people to understand the semantics of topics. A topic model generally characterizes the semantics of topics by their top words. Top words are the words that are most likely to be assigned to a topic. For instance, a topic's top words include "Android, phone, Nexus, Samsung, mobile", indicating that this topic is related to mobile phone products. A typical problem in topic models is that there are irrelevant words in the top words of each topic (even if stop words are removed), which results in misleading information of the topic semantics. In our opinion, words with high frequencies in the corpus are more likely to be assigned to multiple topics, which form the irrelevant top words for each topic. For example, among top words like "sound, headphones, phone, bass, card", the last word "card" is unrelated to the topic about headset devices. However, "card" is brought into the list of the top words by the word "phone" because "phone card" is a phrase that often appears in the corpus. These kinds of words (like "phone" in the example above) that scatter across multiple topics are often named as "topic-indiscriminate words" because of their poor abilities to discriminate different topics [34]. The aforementioned pHDP is also affected by topic-indiscriminate words due to its bootstrap structure. In light of this consideration, we develop a term weighting scheme to reduce the impact of topic-indiscriminate words. To address the issues of the aforementioned 3 aspects, a new topic model called the Term Weighted-parallel Dynamic Topic Model (TW-pDTM) is proposed. We summarize the contributions of the TW-pDTM as follows: (i) We introduce a term weighting scheme to measure the topic discriminating ability of words and utilize it to reduce the sampling probabilities of topic-indiscriminate words, which allows us to achieve a more satisfying topic extraction result. The topic extraction result can be applied to some other natural language process applications, such as text summarization and information retrieval. (ii) Making use of the parallel sampling algorithm in pHDP, we develop a mechanism for dynamic adjustment of topics which includes generating new topics and deleting marginal topics.

The rest of this paper is organized as follows: we introduce the related work and the background studies in Section 2 and Section 3, and describe the details of TW-pDTM in Section 4. In Section 5, we carry out topic mining experiments, as well as supervised text summarization as the downstream task. In Section 6, we draw our conclusion.

2. Related work

In general, topic models can be divided into two types, parametric models and non-parametric models. The former one needs to determine the topic number manually while the latter one is capable of inferring the number of topics from the data.

LDA is a typical representative of parametric models [4]. It models the generation process of documents by two conjugate distributions, i.e., the Dirichlet distribution and multinomial distribution. Kai et al. [34] found that LDA may mix unrelated or loosely related words in its extracted topics. They indicated that it is the topic-indiscriminate words that tend to bring irrelevant words into topics, which makes the extracted topics less interpretable, and they proposed a variant model of LDA called TWLDA. In terms of the topic-word distribution obtained by LDA, TWLDA computes the topic discriminating weight of all words and reduces their counts by directly modifying the document-topic matrix and topic-word matrix to reduce topic-indiscriminate words. However, this method leads to the loss of the previously sampled information.

Proposed by Cheng et al. [8], pHDP is a novel method that achieves topic-level parallelization. However, it is a parametric model because it still needs to determine the topic number manually. Also, pHDP suffers from the topic-indiscriminate words due to its reconstruction of the dataset using a bootstrap technique. In the process of reconstructing a new dataset, pHDP performs sampling on the original dataset. As a result, high-frequency words are more likely to be sampled, becoming the topic-indiscriminate words. All the above parametric models are incapable of self-adjusting their topic numbers. If the given topic number is too large, the topic extraction effect will be inconspicuous (since the topic extraction is essentially a clustering process) and a lot of redundant topics will be remained, while if the given topic number is too small, some important topics will be ignored.

For non-parametric models, HDP constructs a two-layer Dirichlet process to infer the topic number automatically from the dataset. An explicit representation of HDP [28] is the Chinese Restaurant Franchise (CRF) [7,3]. CRF treats each restaurant as a document, each table in the restaurants as a topic, and each customer as a word. There are 4 assumptions in the CRF: (i) The number of tables in each restaurant is infinite. (ii) The number of customers who can sit at each table is infinite. (iii) Only 1 dish can be served per table. (iv) All the restaurants share the same menus. Based on these assumptions, it can be inferred that each dish represents a table, just like a unique topic identifier represents a topic. When a new customer enters a restaurant, s/he has to choose what to eat. If there is a non-empty table with the dish s/he prefers, s/he will join the table. Otherwise, s/he may sit at a new empty table and order a new dish. When no more customers enter a restaurant, it can be indicated that each word in the corpus has been assigned to a topic, and the topic-word distribution is formally determined.

This process shows a limitation of HDP. Despite its ability to generate new topics, it can not monitor the decay of marginal topics. A marginal topic refers to the topics with only a few assigned words, and such a topic is unimportant in most cases [30]. Moreover, as mentioned above, HDP is incapable of dealing with large-scale datasets due to its model complexity.

The Hierarchical Pitman-Yor Process (HPYP) [18] is an efficient non-parametric model. The Pitman-Yor Process [15] is a two-parameter Poisson-Dirichlet process. HPYP describes a CRF representation with significant smaller time costs compared with HDP. However, similar to HDP, it is also difficult for HPYP to deal with large-scale datasets. The Conic Scan-and-Cover (CoSAC) [35] is an algorithm that focuses on the analysis of the topic simplex, i.e., a convex polytope constructed by taking the convex hull of the vertices representing the latent topics to achieve the non-parametric characteristic from another aspect. Though CoSAC is one of the fastest algorithms among several state-of-the-art non-parametric techniques, it performs worse than LDA and HDP in terms of the perplexity or the topic coherence.

Recently, some competitive topic models have been proposed. For instance, TSAMcop [20] modelled the local topical dependencies within phrases by using copula functions [1]. On the other hand, neural network based topic models take the advantage of hidden layers to represent the topic distribution, which can accurately infer the posterior distribution of topic features, and obtain a distributed representation of topic features. Neural Topic Model (NTM) [6] combined the neural network and the probabilistic topic model by converting the parameter estimation of Dirichlet distribution in LDA to the estimation of hidden layer parameters in the neural network, which is more suitable for supervised learning. Neural Variational Document Model (NVDM) [22] pioneered the Neural Variational Inference (NVI) framework and applied the topic model to text classification and question answering. Document Informed Neural Autoregressive Distribution Estimator (iDocNADE) [12] set up multiple hidden layers. For each word, the topic feature extraction results are recursively expressed by multiple hidden layers. Gaussian Softmax (GSM) and Gaussian Stick Breaking (GSB) [21] changed the prior distribution of document-topic distribution and topic-word distribution in LDA from Dirichlet distribution to Gaussian distribution, and the neural network is used to fit the mean and variance of this Gaussian distribution, like Variational Auto-Encoder (VAE) [16]. Hidden variables are generated by a prior Gaussian distribution, then input into a multi-layer perceptron, normalized by the sigmoid activation function, and input into the stick breaking process for the GSB model to achieve non-parametric characteristics. Based on NVI, Gamma Negative Binomial-Neural Topic Model (GNBNTM) [33] exploited Gamma and Poisson distributions as prior distributions for discovering dispersed and explainable topics.

3. Background

To describe our model intuitively, we here introduce some background studies.

3.1. Hierarchical Dirichlet Process

In order to infer the topic number automatically from the data, the HDP method is proposed [28]. Given a set of documents $X = \{X_1, X_2, \ldots, X_D\}$, HDP models the generation process of each word as follows:

$$G_0|\{\alpha, H\} \sim \mathrm{DP}(\alpha, H), \tag{1}$$

$$G_d|\{\beta, G_0\} \sim \mathrm{DP}(\beta, G_0), \quad d \in \{1, \ldots, D\}, \tag{2}$$

$$\lambda_{di}|G_d \sim G_d, \tag{3}$$

$$X_{di}|\lambda_{di} \sim p(x_{di}|\lambda_{di}), \quad i \in \{1, \ldots, N_d\}. \tag{4}$$

In the above, H is the base measurable space of topics. G_0 is the global measurable space of topics shared by all documents, generated by a Dirichlet Process (DP) from H with a concentration parameter α. G_d is the topic measurable space of the dth document, which is also generated by a DP. λ_{di} is the topic of X_{di} (i.e., the ith word in the dth document), and it is sampled from G_d. Finally, X_{di} is sampled from $p(x_{di}|\lambda_{di})$ given topic λ_{di}.

The HDP has achieved success in modeling various types of data. However, the biggest challenge towards applications is its inability to scale to large datasets. Moreover, most of the existing works on parallel sampling for HDP have limitations either in inference accuracy or convergence rate [8].

3.2. Parallel Hierarchical Dirichlet Process

Due to the high time complexity in HDP, it is difficult to run HDP over a large-scale corpus. To address this, Cheng et al. [8] proposed pHDP based on the equivalence of the generation process between G2PP and HDP. G2PP is a three-level hierarchical random process defined on the base measurable space of topics. The generation process of each word in G2PP is as follows:

$$G_{0'}|\{\alpha, H\} \sim \mathrm{GaP}(\alpha H), \tag{5}$$

$$G_{d'}|\{G_{0'}\} \sim \mathrm{GaP}(G_{0'}), \tag{6}$$

$$\Pi_{d\prime}|\{m, G_{d\prime}\} \sim \text{PoisP}(mG_{d\prime}), \tag{7}$$

$$X_{di}|\Pi_{d\prime} \sim p(x_{di}|\Pi_{d\prime}). \tag{8}$$

In the above, GaP and PoisP represent Gamma process and Poisson process, respectively. Compared to HDP, G2PP replaces the upper level DP by the Gamma process with the same parameters $G_{0\prime} = \sum_{k=1}^{\infty} \alpha_k \delta_{\theta_k} \sim \text{GaP}(\alpha H)$, where α_k can be seen as the weight for the kth topic. δ_{θ_k} is the Dirac-delta function of mixture component θ_k. The lower level DP in HDP is also replaced by another Gamma process $G_{d\prime} = \sum_{k=1}^{\infty} \pi_{dk\prime} \delta_{\theta_k}$ for G2PP, where $\pi_{dk\prime}$ can be seen as the weight of the kth topic in the dth document. The Poisson process generates $\Pi_{d\prime}$ by $\sum_{k=1}^{\infty} n_{dk} \delta_{\theta_k}$, where n_{dk} is the number of words assigned to the kth topic in the dth document. Finally, each word is generated in terms of n_{dk} and the conditional distribution $p(x_{di}|\Pi_{d\prime})$ for G2PP. Details of the generation process equivalence between HDP and G2PP are illustrated in [36]. However, the existing pHDP can neither determine the topic number from the data nor deal with topic-indiscriminate words.

3.3. Term Weighted Latent Dirichlet Allocation

To address the topic-indiscriminate problem in topic modeling, a model called TWLDA was proposed to reduce the impact of topic-indiscriminate words. Inspired by [31], TWLDA develops a weight function called the Topic Distributional Concentration (*TDC*) to measure the topic discriminating ability of a word w, as follows:

$$TDC(w) = 1 + \frac{\sum_{k=1}^{K} \frac{p(w|k)}{\sum_{j=1}^{K} p(w|j)} log \frac{p(w|k)}{\sum_{j=1}^{K} p(w|j)}}{log(K)}. \tag{9}$$

In the above, K is the number of topics, and $p(w|k) = f(w,k)/N_k$, where $f(w,k)$ denotes how many times word w appears in topic k. According to this equation, *TDC* is a number between 0 and 1. If the distribution of a word w highly concentrates in a topic, $TDC(w)$ will be close to 1, indicating that it has a strong topic discriminating ability. In the other case, if the distribution of a word w is scattered across multiple topics, $TDC(w)$ will be close to 0, indicating that it is most likely a topic-indiscriminate word. In TWLDA, the *TDC* of each word is calculated only once, and the weight of topic-indiscriminate words will be reduced by the following rules:

$$Nmk_m^k = \sum_{w=1}^{V} TDC(w) \times n_{mkw}, \tag{10}$$

$$Nkw_k^w = \sum_{m=1}^{M} TDC(w) \times n_{mkw}. \tag{11}$$

In the above, the definition of *TDC* is referred to Eq. (9). M, V, and K represent the number of documents, the length of vocabulary, and the number of topics, respectively. *Nmk* is a document-topic statistical matrix whose element denotes the number of words assigned to topic k in document m. *Nkw* is a topic-word statistical matrix, whose element denotes the number of occurrences of word w assigned to topic k. Nmk_m^k represents the element located in the *mth* row and the *kth* column of *Nmk*, and Nkw_k^w represents the element located in the *kth* row and the *wth* column of *Nkw*. n_{mkw} represents the number of occurrences of word w assigned to topic k in document m. Unfortunately, there are several problems for such a term weighting scheme. Firstly, it requires a lot of time cost to compute n_{mkw}. Secondly, it is likely to lose some sampled semantic information after greatly modifying the two original statistical matrices. Finally, this weighting scheme will be performed only once and it is difficult to determine at which iteration to perform such a scheme so that the best result can be obtained. In addition, there is a term weighting scheme based on a combination form of entropy weighting (CEW) [17]. This method is based on conditional entropy measured by word co-occurrences and it assigns meaningless words with lower weights and informative words with higher weights. However, when applying CEW to LDA, there are three steps to do: find the meaningless words and informative words, then calculate the weights separately, and assign the weights finally. These operations are too cumbersome.

4. Proposed Model

In this section, we describe our TW-pDTM in details. The problem is defined at first, including the notations of frequently-used variables. Then, the dynamic adjustment mechanism of topics and the term weighting scheme are presented.

4.1. Problem Definition

For the sake of parallelization, the input texts are first divided into D subsets $\{X_1, X_2, \ldots, X_D\}$. Then, we organize all subsets as a loop, which means that the next subset of X_D is X_1. Each thread starts from a different position of the loop of subsets to avoid the read-write conflict.

Our task is to get the topic distribution of the input texts, in which, the number of topics will change over time. Thus, we need to update the subset-topic distribution and the topic-word distribution dynamically. We denote the number of topics as K, the length of the vocabulary as W, and the total number of words assigned to topic k as N_k. The value of N_k changes when a word is added to the assigned word set of topic k with an acceptance probability P_{ADD} or a word is deleted from the assigned word set of topic k with an acceptance probability P_{DEL}. Table 1 summaries the notations of frequently used variables in TW-pDTM.

4.2. Basic Structure

As aforementioned, we need to update the subset-topic distribution and the topic-word distribution until convergence. Fig. 1 presents the updating process of parameters in our TW-pDTM. During each iteration, we first compute the value of n_{dk} for each topic in each subset through the dynamic adjustment mechanism of topics, which will be described in detail later. Then we update the subset-topic distribution of each subset π_{dk} in terms of n_{dk}. Finally, we update the topic-word distribution of each topic θ_k, and the weight of each topic α_k. We stop the iteration until the subset-topic distribution and the topic-word distribution are stable.

4.3. Dynamic Adjustment Mechanism of Topics

For real-world textual data, when a topic becomes popular, words related to the topic are more likely to appear. Meanwhile, when a topic becomes marginal, words related to the topic becomes unimportant and rare. Along time, the members of these two kinds of words will evolve dynamically. To this end, we set two containers, a collection X'_d and a stack S_d for the dth subset X_d to capture the two kinds of words, respectively. The words in X'_d have been assigned to a certain topic while the words in S_d have not been assigned to any current topic. And X'_d is constructed as a reconstruction of X_d with a flexible length. This feature of X'_d enables us to develop efficient parallel sampling algorithms according to [8]. In addition, we set a buffer for each subset to prevent the excessive growth of the topic number, which will be explained later. We initialize X'_d to be empty and initialize S_d by randomly sampling words from X_d for each subset. Then during one iteration, each thread, which represents a topic, visits along the loop of subsets. During the visit to the dth subset, each thread performs the following two processes with an equal probability, as shown in Fig. 2.

The ADD process of each topic is to sample representative words from S_d with an acceptance probability P_{ADD}. The words that are accepted by the ADD process will be assigned to topic k, and transferred to X'_d as a part of the reconstructed dataset. For example, in Fig. 2(a), as the "Accepted" arrow in subset X_d shows, when thread k visits subset X_d and executes the ADD process, the word w_1 is assigned to topic k with a probability of P_{ADD}. Otherwise, w_1 that is rejected from the ADD process with a probability of $(1 - P_{ADD})$ will be put into B_d, as the "Rejected" arrow in Fig. 2(a) shows.

The DEL process is to delete some words from a topic with an acceptance probability P_{DEL}. The words that are accepted by the DEL process will be transferred from X'_d to B_d, representing that they no longer belong to topic k. In Fig. 2(b), as the "Accepted" arrow in the subset X_{d+1} shows, when thread k visits the next subset X_{d+1} and executes the DEL process, the word w_5 is deleted from topic k with a probability of P_{DEL}. Otherwise, w_5 that is rejected from the DEL process with a probability of $(1 - P_{DEL})$ will remain in topic k. B_d will be cleared if it exceeds the maximum size and the remaining words will be put into S_d.

During each iteration, some words will be continuously deleted from X'_d or added into X'_d in each subset, indicating that the sum of all X'_d, i.e., the reconstructed dataset X' is also changing over time. It is reasonable for us to consider X' as the current hot topics. If people pay less attention to a topic, the words assigned to this topic will appear less frequently. Thus, we assume that, when

$$N_k \leqslant word\_num \times thr, \tag{12}$$

topic k becomes marginal and needs to be deleted [20], where N_k is the total number of words assigned to topic k, *word_num* is the total number of words in $X\prime$, and *thr* is the user-defined threshold. The remaining words assigned to topic k are put into the buffers of some random subsets.

With respect to the generation of new topics, the process is as follows. As mentioned above, the words in stacks are considered not being assigned to any topic yet. Thus, if there are any words remaining in any stack, it means that the current number of topics is not enough, and new topics need to be added. Since the number of threads increases as the number of topics increases, to prevent the excessive growth of threads, we add 1 topic per iteration and set a buffer for each subset by following [20]. Though the words in the buffers are also treated as having no topic attribution, they will not cause the generation of new topics. There is no word assigned to the new topic after topic initialization until the topic samples representative words by the ADD process in the next iteration. Obviously, the sizes of the stack and the buffer in each subset are the

Table 1
Notations used in TW-pDTM.

Notations	Description
K	The number of topics
W	The length of the vocabulary
D	The number of subsets
X	The original dataset
$X\prime$	The reconstructed dataset
X_d	The dth subset
X'_d	The reconstruction of the dth subset and $\sum_{d=1}^{D} X'_d = X\prime$
B_d	The buffer of the dth subset
S_d	The stack of the dth subset
θ_k	The topic-word distribution of topic k
α_k	The weight of topic k
π_{dk}	The element in the dth row and the kth column of the subset-topic distribution matrix
N_k	The total number of words assigned to topic k
n_{dk}	The number of words assigned to topic k in the dth subset and $\sum_{d=1}^{D} n_{dk} = N_k$
P_{ADD}	The acceptance probability of adding a word to the assigned word set of a topic
P_{DEL}	The acceptance probability of deleting a word from the assigned word set of a topic
word_num	The total number of words in $X\prime$

two main parameters controlling the change of the topic number, which will also affect the topic extraction results. In Section 5.6.2, we conduct a comparison analysis of the above two parameters.

To make it intuitive, we summarize the dynamic adjustment mechanism of topics in Algorithm 1.

Algorithm 1: Dynamic Adjustment Mechanism of topics

Input: N_k of each topic k, S_d of each subset X_d, Number of topics K;
Output: New topic_word distribution θ, New subset_topic distribution π;
vector⟨*int*⟩ marginal_topics;
for $i = 1$ to K **do**
  if $N_i \leqslant word\_num * thr$ **then**
    marginal_topics.push_back(i);
  end if
end for
for $i = 1$ to *length*(marginal_topics) **do**
  for *word* $w = 1$ to N_i
    index = D * Uniform(0, 1);
    B_{index}.push_back(w);
    word_num−−;
  end for
  Delete the ith row of θ and the ith column of π and normalize them;
end for
Reduce K by *length*(marginal_topics);
for $d = 1$ to D **do**
  if S_d is not empty **then**
    K++;
    Randomly initialize the (K-1) th column of π;
Randomly initialize the (K-1) th row of θ;
    Break;
  end if
end for

After each thread has visited all subsets, we can obtain each updated n_{dk}. According to those results, model parameters can be updated dynamically.

4.4. Term weighting scheme

The dynamic adjustment mechanism of topics still leave the topic indiscriminate problem unresolved. In particular, if a topic-indiscriminate word frequently appears in the original dataset, it is more likely to be sampled by a larger number of topics and dominates the top words of these topics. To address this, we propose an entropy-based term weighting scheme to

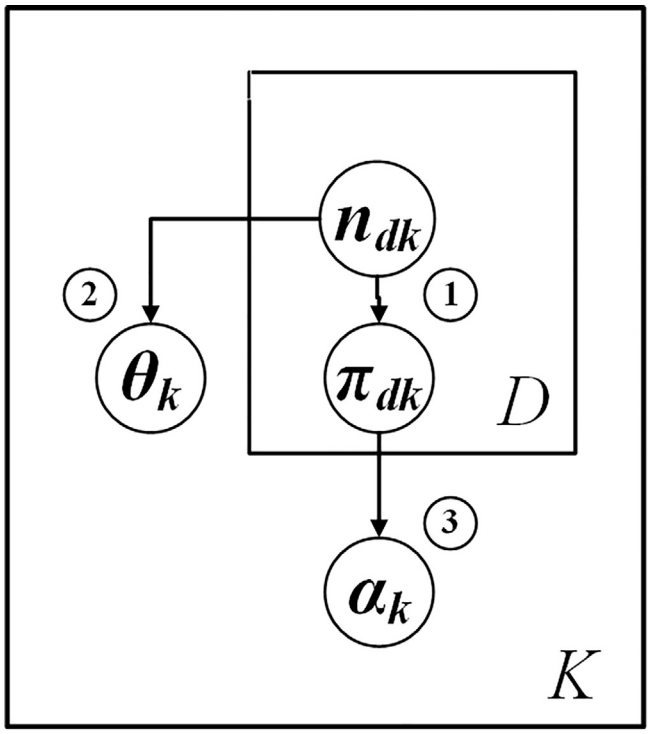

Fig. 1. Parameter updating process in TW-pDTM.

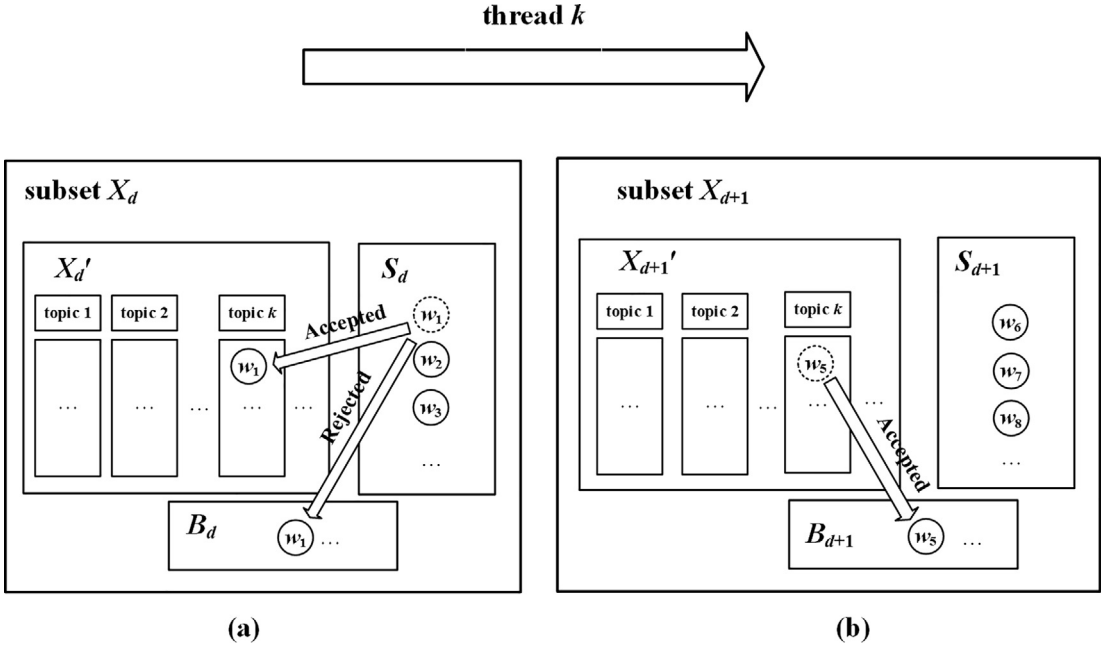

Fig. 2. ADD and DEL processes when a thread k visits the subsets.

reduce the sampling probabilities of topic-indiscriminate words. Different from the term weighting scheme in TWLDA, our method is as follows.

We take the advantage of the *TDC* of each word in the ADD and DEL processes described in Section 4.3. When a word w is selected to be added, then

$$P_{ADD} = min\left(1, \frac{m\pi_{dk}}{n_{dk}+1} p_{x^*}(k) \times TDC(w)\right). \tag{13}$$

When a word w is selected to be deleted, then

$$P_{DEL} = min\left(1, \frac{n_{dk}}{m\pi_{dk}} \frac{1}{p_{x^*}(k) \times TDC(w)}\right). \tag{14}$$

In the above, the definition of *TDC* can be referred to Eq. (9), and

$$p_{x^*}(k) = p(x^*|\theta_k)/Z(x^*), \tag{15}$$

where the partition function is $Z(x^*) = \sum_{k=1}^{K} p(x^*|\theta_k)$ [8].

Compared to TWLDA, the *TDC* of each word is updated in terms of the new topic-word distribution of each topic θ_k after every certain number of iterations in our TW-pDTM, which will be shown in Algorithm 2. This operation can periodically prevent topic-indiscriminate words from being sampled. Moreover, it takes much less time to compute *TDC* than to compute n_{mkw} in TWLDA. In addition, our term weighting scheme does not directly modify the original statistical or distribution matrices. Therefore, we can obtain more words with a strong topic discriminating ability than topic-indiscriminate words in each topic. Finally, unlike CEW [17], the process of finding the topic-indiscriminate words and the process of calculating the weights are carried out at the same time, which saves a lot of time.

4.5. Algorithm

In this section, we first derive the parameter updating formulas of our model. According to the generation process of each word in pHDP [8], the joint distribution can be computed by

$$\begin{aligned} p(n_{dk}, \pi_{dk}, X_{dk}, \alpha_k, \theta_k) &= \prod_{k=1}^{K} \frac{\alpha_k^{\frac{z}{K}-1}}{\Gamma(\frac{z}{K})} e^{-\alpha_k} \\ &\times \prod_{k=1}^{K}\prod_{d=1}^{D} \frac{\pi_{dk}^{\alpha_k-1}}{\Gamma(\frac{z}{K})} e^{-\pi_{dk}} \frac{(m\pi_{dk})^{n_{dk}}}{n_{dk}!} e^{-m\pi_{dk}} \\ &\times \prod_{k=1}^{K} H_\theta(\theta_k) \times \prod_{k=1}^{K}\prod_{d=1}^{D}\prod_{i=1}^{n_{dk}} p(x_{dk}^{(i)}|\theta_k). \end{aligned} \tag{16}$$

In the above, the factor related to π_{dk} is

$$\pi_{dk}^{\alpha_k+n_{dk}-1} e^{-(m+1)\pi_{dk}}, \tag{17}$$

which indicates that π_{dk} follows a gamma distribution with $n_{dk}+\alpha_k$ and $m+1$ as its scale and shape parameter respectively. So, we update π_{dk} as follows:

$$\pi_{dk} \sim \text{Gamma}(n_{dk}+\alpha_k, m+1). \tag{18}$$

Furthermore, the factor related to α_k is

$$\frac{\alpha_k^{\frac{z}{K}-1}}{(\Gamma(\alpha_k))^D} e^{\alpha_k \sum_{d=1}^{D} log(\pi_{dk})-1}. \tag{19}$$

According to the first order Laurent expansion, $\Gamma(z) \approx \frac{1}{z}$ when $|z| < 1$. Therefore, α_k can be approximately updated as follows:

$$\alpha_k \sim \text{Gamma}\left(\frac{\alpha}{K} + D, \sum_{d=1}^{D} - log(\pi_{dk})\right). \tag{20}$$

Finally, θ_k is updated using the following formula:

$$\theta_k^w = Nkw_k^w / N_k, \tag{21}$$

where θ_k^w is the wth element of θ_k. As mentioned in Section 3.3, Nkw_k^w represents the number of occurrences of the word w assigned to topic k.

We summarize the overall process of our TW-pDTM in Algorithm 2.

Algorithm 2: Term Weighted-parallel Dynamic Topic Model

Input: Origin dataset X, Number of iterations for each thread to visit a subset n, Initial number of topics K;
Output: Subset-topic distribution π and topic-word distribution θ;
Divide the origin dataset X into D subsets and construct a stack S_d for each subset;
Construct empty reconstruction X'_d and empty buffer B_d for each subset;
Randomly initialize topic-word distribution θ and subset-topic distribution π;
while π and θ have not converged **do**
  for each topic k asynchronously in parallel **do**
    for each subset d **do**
      for $i = 1$ to n **do**
        Sample $\mu \sim$ Uniform(0,1);
        if $\mu \leqslant 0.5$ **then**
        Compute P_{ADD} for a random word w in S_d;
      else
        Compute P_{DEL} for a random word w in X'_d;
      end if
    end for
    Push all elements from B_d to S_d when B_d exceeds its max size;
    Count n_{dk};
    Update π_{dk} according to Eq. (18);
  end for
  Update α_k according to Eq. (20);
  Update θ_k according to Eq. (21);
  end for
  Execute Algorithm 1;
  if *current_iter*%10 = 0
    compute *TDC* for each word in vocabulary;
end if
end while

4.6. Applications

4.6.1. Text summarization

Text summarization is one of the most popular tasks in natural language processing, here we show a case study to apply our TW-pDTM to text summarization. An advanced framework for text summarization model is ConvS2S [10], which employs convolutional blocks as the encoder and decoder of the sequence to sequence (S2S) architecture. To utilize topic information in text summarization, Topic–ConvS2S [25] extends the ConvS2S model by concatenating a topic representation to the original word embedding, where such a word-level topic representation vector is constructed as the element-wise product of the word-topic vector of this word and the document-topic vector for the document where the word is in.

In Topic–ConvS2S, the word-topic and document-topic vectors are obtained by training the LDA [4] model. Similarly, our method is also to combine the document-topic vector and the word-topic vector with the word embedding of the input documents in the framework, while such vectors are obtained by our TW-pDTM.

The steps for the procedure are summarized as follows:

- S1. Train a TW-pDTM topic model $\mathscr{M}$ on the given dataset.
- S2. For each word w in document D from the dataset, retrieve its word-topic vector t'_w and document-topic vector t_D from $\mathscr{M}$.
- S3. Calculate $t'_w \odot t_D$ as the word-level topic vector, then concatenate it to the original word vector $e_i = (x_w + p_w) \oplus (t'_w \odot t_D)$, where $\odot$ denotes the element-wise product and $\oplus$ denotes concatenation, x_w is the word embedding and p_w is the positional embedding of word w.
- S4. Perform text summarization according to Topic–ConvS2S [25].

Compared with RNN-based S2S structure which was commonly used in text summarization, our framework can not only capture long-distance features through multiple layers of stacked CNN, but also be easy to parallel.

4.6.2. Information retrieval

In this part, we illustrate an application on information retrieval for our TW-pDTM. By setting the number of subsets D to be equal to the number of documents in the original dataset, the rows of π are the topic distributions for each reconstructed documents approximating the original documents, and the semantics of each topic can be referred to θ whose rows are the word distributions for each topic. Therefore, topic-level semantics-based information retrieval can be performed by leveraging document-topic distribution π_d.

Specifically, a TW-pDTM can be trained on a collection of documents and the topic distribution for each document is obtained. Then, for a query text, the query-topic distribution with the same topic basis can also be obtained from TW-pDTM. The cosine similarity can be calculated between the query-topic distribution and every document-topic distribution. After ranking the documents by the calculated cosine similarity, the most relevant documents to the query are retrieved finally. Note that different from keyword-based information retrieval, such a similarity based retrieval method performs relevance scoring based on topic semantics, that is, the more similar topics observed in the query and the document, the higher cosine similarity between them will be.

5. Experiment

5.1. Datasets

Table 2 is a brief introduction to the datasets we used[1]. Among them, KOS is from the Daily Kos, a political blog site that collects a lot of political vocabulary and content. NIPS is from the papers in NIPS from 1987 to 2016. ENRON is from the emails at a company called Enron in the USA and NYTIMES is from the The New York Times.

5.2. Experiment design

To evaluate the effectiveness of our methods (i.e. the dynamic adjustment of topics and the term weighting scheme), we compare the perplexity and topic coherence of TW-pDTM with other baselines. All models are run for 5 times to evaluate the performance stability. pDTM is a version of TW-pDTM without any term weighting scheme. Apart from pHDP [8], we also adopt a recent neural topic model, i.e., GNBNTM [33], as a strong baseline for topic quality analysis. Particularly, GNBNTM is one of the latest NVI-based topic models which also exploits Gamma and Poisson distributions as prior distributions. For NTM [6] which takes n-grams embeddings over the whole dataset as the input, it may be unsuitable for the order-independent Bag-of-Words datasets we used, and such n-gram-level inputs also lead to a high running time cost in terms of large datasets (more than 24 h on the NYTIMES data). Due to the multiple run experiment setting, it would be too time-consuming to run LDA [4] and TWLDA [34], e.g., running time of over 24 h is needed on the NYTIMES data for both of the two models. Consequently, we do not adopt NTM, LDA, and TWLDA for comparison.

Perplexity is a general metric for measuring language models, which is defined as follows:

$$PPL = exp\left(-\frac{\sum_{w} logP(w)}{word\_num}\right), \tag{22}$$

where $P(w)$ represents the probability of the occurrence of each word w, and

$$P(w) = \sum_{k} p(k|d) \times p(w|k), \tag{23}$$

where k denotes a topic, and d denotes a document. A lower perplexity indicates better generalization performance [4].

Topic coherence is a metric for assessing the topic quality by modelling the co-occurrence of top words within documents [24]. The topic coherence of a topic k is defined as follows:

$$TH(k) = \sum_{m=2}^{M}\sum_{l=1}^{m-1} log\frac{D(w_m, w_l)+1}{D(w_l)}. \tag{24}$$

In the above, $D(w_l)$ is the document frequency of word w_l, $D(w_m,w^l)$ is the co-document frequency of word w^m and w^l, i.e., the number of documents includes both w^m and w_l. In order to avoid the invalid result when $D(w^m,w_l)$ equals to 0 in logarithmic operation, the numerator needs to be added 1. $\{w_1, w_2, \ldots, w_M\}$ represents the top M words assigned to topic k, where M is the number of top words we show for each topic and it is usually set to 10 according to [9]. If $TH(k)$ is close to 0, it indicates that the co-occurrence of the top words assigned to topic k is frequent, and it is reasonable that they belong to the same topic. In order to make a comparison, we compute the average topic coherence of all topics in each model.

We further compare the top words of sampled topics extracted by pDTM and TW-pDTM, so as to evaluate the effectiveness of our term weighting scheme qualitatively.

[1] http://archive.ics.uci.edu/ml/datasets/Bag + of + Words

Table 2
Description of datasets.

Name	Number of documents	Length of vocabulary	Total number of words
KOS	3,430	6,906	467,714
NIPS	1,500	12,419	$\approx 1,900,000$
ENRON	38,961	28,102	$\approx 6,400,000$
NYTIMES	300,000	102,660	$\approx 100,000,000$

5.3. Parameter settings

In the experiment, we let the topic number increase sufficiently by setting the initial number of topics to 2, and empirically set the threshold for reducing a marginal topic, i.e., *thr* in Eq. (12) to 1%. Next, we compare TW-pDTM with other models over the same K because the perplexity and the topic coherence may change with K. Thus, we first execute TW-pDTM for 5 times to obtain the final number of topics until convergence (i.e., 50 iterations which will be detailed in Section 5.4), and average the topic number over 5 runs to obtain *K_final* for each dataset, and then set K to be *K_final* when executing GNBNTM and pHDP. We run GNBNTM[2] by following the settings in [33], and execute pHDP, pDTM, and TW-pDTM using the same hyperparameter settings of the Gamma and Poisson distributions which are set as $\alpha = 0.1$ and $m = 1000$, respectively. Finally, we use the grid search to determine the size of S_d and B_d for each subset by following [20]. First we need to determine the scope of the search. In order to adapt to the size of different subsets, the size of S_d and B_d should be proportional to $|X_d|$, i.e., the total number of words in the corresponding subset, and this ratio should be between 0 and 1. Notice that the words in S_d are randomly selected from X_d, so if this ratio is too large, it is very likely that a large proportion of high frequency words will be selected, resulting in too many topic-indiscriminate words in S_d. Meanwhile, the ratio should also be large enough to avoid missing necessary information in the original subset. Moreover, in order to let the topic number maintain a growing trend until convergence, the size of S_d should be slightly larger than B_d. Based on the above conditions, we set the range of $|S_d|/|X_d|$ to 0.1–0.9 with a step of 0.1, and the size of B_d (i.e., $|B_d|$) to 75% and 90% of the size of S_d (i.e., $|S_d|$). According to the grid search result, we set $|S_d|/|X_d|$ to 0.4 and $|B_d|/|S_d|$ to 90%.

5.4. Convergence analysis

To examine the convergence trend of our TW-pDTM, we plot the convergence curve of a run on the NIPS dataset. Fig. 3(a) shows the changing trend towards the perplexity score over iterations, and we can see that the change is slighter and slighter, and finally becomes stable with around 50 iterations. A similar trend can be observed from the changing number of topics over iterations, as shown in Fig. 3(b). It can be seen that the number of topics increase steadily within 50 iterations. All of these results indicate that the proposed TW-pDTM can reach a convergence, and has converged with approximately 50 iterations on the NIPS dataset.

5.5. Comparison with baselines

5.5.1. Quantitative analysis of topics

In this part, we compare the performance of different models in terms of perplexity and topic coherence. We calculate the average results for all runs, followed by the maximum absolute difference (i.e., range) of each case. Table 3 shows the number of final topics for TW-pDTM on all datasets, which changes little under different runs.

Table 4 shows the perplexity results on all datasets. Note that the perplexity score of GNBNTM is not presented due to the following reasons. Firstly, the perplexity of such an NVI-based model is difficult to be compared with that of a sampling-based model directly [14]. In our specific case, the perplexity scores of involved sampling-based models (i.e., pHDP, pDTM, and TW-pDTM) are calculated among the words in the reconstructed datasets, as shown in Eq. (22), while that of NVI-based model (i.e., GNBNTM) is calculated among the words in the original dataset. Secondly, the KL-divergence will be greatly influenced by the priors, which results in difference in the perplexity score of NVI-based models with different priors [5]. For all sampling-based models, i.e., pHDP, pDTM, and TW-pDTM, our models outperform pHDP when the size of the dataset is becoming larger, and TW-pDTM performs the stablest among them. This validate the effectiveness of our dynamic adjustment and term-weighting schemes.

Table 5 shows that our pDTM and TW-pDTM consistently outperform pHDP in terms of topic coherence, and TW-pDTM performs better than pDTM on the ENRON dataset. We conjecture that this is because there might be more topic-indiscriminate words in the ENRON dataset. Furthermore, our pDTM and TW-pDTM perform much better than GNBNTM on the small-scaled KOS and NIPS datasets. Although GNBNTM performs the best on the large-scaled ENRON and NYTIMES datasets, its running time is much longer than our pDTM and TW-pDTM. Specifically, it costs more than 20 h for GNBNTM to achieve convergence on the NYTIMES dataset, even if we run it on a GPU of GeForce GTX 1080 Ti which contains 6 major

[2] https://github.com/mxiny/NB-NTM

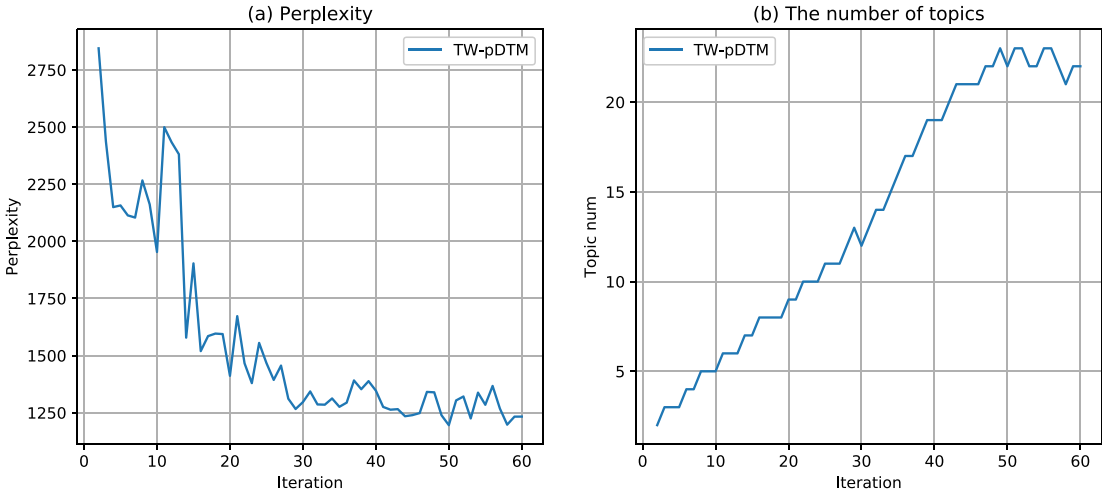

Fig. 3. Perplexity and #topics trends of TW-pDTM on the NIPS dataset.

Table 3
The number of topics for different datasets (Avg ± Range).

Dataset	KOS	NIPS	ENRON	NYTIMES
#Topics (*K_final*)	***5*** ± 1	***19*** ± 4	***22*** ± 1	***10*** ± 3

Table 4
Perplexity of different models (Avg ± Range).

Model & Dataset	KOS	NIPS	ENRON	NYTIMES
pHDP [8]	***1816*** ± 539	***1276*** ± 67	3659 ± 125	9378 ± 1319
pDTM	2195 ± 322	1358 ± 80	***3563*** ± 60	5609 ± 857
TW-pDTM	2664 ± 206	1316 ± 32	3669 ± 95	***5599*** ± 624

Table 5
Topic coherence of different models (Avg ± Range). Note that GNBNTM is run on the GTX 1080Ti GPU for acceleration. However, it still cost more than 20 h to achieve convergence on the NYTIMES dataset.

Model & Dataset	KOS	NIPS	ENRON	NYTIMES
GNBNTM [33]	−76.2 ± 2.3	−61.1 ± 1.4	−***50.3*** ± 7.0	−***27.4*** ± 3.9
pHDP [8]	−75.3 ± 12.3	−16.9 ± 4.4	−68.8 ± 4.9	−114.7 ± 37.6
pDTM	−***58.5*** ± 5.0	−***10.4*** ± 0.6	−67.5 ± 2.4	−84.9 ± 5.3
TW-pDTM	−63.4 ± 11.9	−11.2 ± 2.5	−66.0 ± 1.9	−92.5 ± 4.8

cores and 1 minor core with memory clock rate 1.582 GHz. The reason may lie in the time-consuming batch-wise back propagation under the limited GPU memory for enormous number of weights inside the model when trained on the original NYTIMES dataset [20], which takes 4688 batches in average with a batch size of 64 for each epoch of GNBNTM. On the other hand, for our pDTM and TW-pDTM, sampling is performed on the original datasets, and such a procedure enables significantly reduced running time of the models by slightly sacrificing the topic quality. This is a trade-off decision which enables our model to sample less words for its scalability on a larger dataset, and if more subsets are constructed and more words are sampled, the coherence score will increase potentially.

5.5.2. Running time analysis

To verify the efficiency of our models, we compare the running time of HDP, pHDP, pDTM, and TW-pDTM in dealing with different scales of data. Since our multithreaded implementations of pHDP, pDTM, and TW-pDTM are in C++, for HDP, we also use C++ implementations from the code repository of the authors' lab[3] with versions HDP and HDP-faster inside. Just as the name implies, HDP-faster is a faster implementation of HDP, so we adopt both of these two implementations for running

[3] https://github.com/blei-lab/hdp

time comparison. We run TW-pDTM and pDTM for 5 times on all datasets under the same hyper-parameter settings, and set the initial number of topics as the averaged value among these runs for HDP, HDP-faster, and pHDP. All the models are run on a multi-core computing node with two 8-core Intel Xeon E5-2630 v3 central processing units and memory of 125 GB.

The running time of different models are summarized in Table 6. Note that HDP and HDP-faster can not run successfully due to the failure of randomizing parameters when faced with the large-scaled NYTIMES dataset. Besides, it is estimated that HDP will be converged in more than 24 h on this dataset, which is also unacceptable. For the above reasons, the running time of HDP and HDP-faster on NYTIMES are not presented.

As shown in Table 6, three parallel models, i.e., pHDP, pDTM, and TW-pDTM run much faster than HDP and HDP-faster on the KOS, NIPS, and ENRON datasets, and achieve quite good speedup on the ENRON dataset, which may also owe to the subset division scheme and sampling on the original datasets. Compared with HDP-faster, our pDTM achieves approximately 18× speedup on the ENRON dataset. Naturally, the larger the dataset is, the higher speedup the parallel model achieves. Moreover, when compared with pHDP, our pDTM and TW-pDTM always achieve better speedup on NIPS, ENRON, and NYTIMES. This phenomenon reveals the acceleration effect of dynamic adjustment mechanism, that is, pHDP runs over a fixed large topic number while pDTM and TW-pDTM can adjust the topic number incrementally when dealing with large datasets. Exception lies in the KOS dataset which is too small to obtain valuable running time comparisons.

In terms of applying the term weighting scheme, TW-pDTM takes only a little more time than pDTM, but actually, TWLDA is much more time-consuming than LDA. Although not presented, for the employed datasets, TWLDA runs even 5 to 10 times as long as LDA, which is consistent to the high time complexity in calculating n_{mkw} for TWLDA as mentioned in Section 3.3.

Considering the experimental results in Tables 4–6, we can conclude that our models have competitive performance not only in topic quality but also in efficiency when dealing with large datasets.

5.6. Effect of hyper-parameters

In this subsection, we run our TW-pDTM on the NIPS dataset with different hyper-parameter settings to show their impact, including distribution parameters and the ratio for container size.

5.6.1. Effect of parameters in probability distributions

For parameters in probability distributions, i.e., the parameter α of the high level Gamma distribution in Eq. (5) and the parameter m of the Poisson distribution in Eq. (7), we set each parameter as different values and perform 5 individual runs for each parameter setting to evaluate the effect on both topic quality and stability. Specifically, for the influence of α, we set m as 1000 and vary α in the range of {0.01, 0.1, 1, 10, 100, 1000, 10000}, and for the influence of m, we set α as 0.1 and vary m in the same range.

We calculate the average results of converged topic numbers, perplexity, and topic coherence for all runs, followed by the range of each result. We summarize these results in Tables 7 and 8. First, it can be observed that the change of α shows little effects on the results except for two extreme values 0.01 and 10000. For m, it can be observed that the larger the value is, the smaller the shaking difference is, indicating that more stable the model is. Moreover, by comparing Table 7 with Table 8, the Poisson distribution parameter m shows more significant impact on the results, especially for the perplexity score, than the high level Gamma distribution parameter α. The possible reason is that the Poisson distribution is related to each individual document more directly than the higher level Gamma distribution. In general, the setting of hyper-parameters in distributions does not affect the result much in terms of the number of discovered topics and topic coherence, thus we can conclude that our Gamma-Gamma-Poisson process-based TW-pDTM is stable and insensitive to distribution parameters.

5.6.2. Effect of container size

As mentioned in Section 5.3, the size of S_d and B_d should be proportional to $|X_d|$. For $|S_d|/|X_d|$, we fix $|B_d|/|S_d|$ to 0.9 and vary $|S_d|/|X_d|$ from 0.1 to 0.9 with a step of 0.1, and for $|B_d|/|S_d|$, we fix $|S_d|/|X_d|$ to 0.4 and vary $|B_d|/|S_d|$ from 0.75 to 0.9 with a step of 0.05. The results on the NIPS dataset with 5 individual runs are summarized in Tables 9 and 10, respectively.

As shown in Table 9, a larger S_d always leads to a larger number of topics along with longer running time, higher perplexity, and slightly higher topic coherence, which is quite reasonable since a larger S_d potentially results in larger X'_d with more topics and more words involved in perplexity calculation in Eq. (22). To realize a trade-off between perplexity and topic coherence, $|S_d|/|X_d|$ is set as 0.4.

As for B_d, from Table 10 we can observe that a larger B_d indicates much lower perplexity and comparable topic coherence as well as slightly different topic numbers and running time, since a larger B_d can avoid consecutive rejections on outliers better [8]. Thus, 0.90 is adopted as the setting of $|S_d|/|X_d|$.

5.7. Qualitative analysis

5.7.1. Effectiveness of the term weighting scheme

To evaluate the effectiveness of our term weighting scheme qualitatively, we inspect the top topic words on the NIPS dataset for our models and baselines. Since the difference between pDTM and pHDP lies in self-adjusted dynamic topic num-

Table 6
Running time (seconds) of different models (Avg ± Range).

Model & Dataset	KOS	NIPS	ENRON	NYTIMES
HDP [28]	125 ± 21s	1362 ± 445s	3165 ± 189s	-
HDP-faster [28]	17 ± 1s	62 ± 3s	273 ± 30s	-
pHDP [8]	***3*** ± 1s	12 ± 1s	27 ± 1s	32 ± 1s
pDTM	4 ± 1s	***9*** ± 1s	***15*** ± 7s	***28*** ± 4s
TW-pDTM	***3*** ± 1s	10 ± 1s	20 ± 3s	30 ± 2s

Table 7
Performance comparisons between different values of α (Avg ± Range).

α	#Topics	Perplexity	Topic coherence
0.01	16 ± 3	1349 ± 93	−15.3 ± 4.1
0.1	18 ± 3	1326 ± 66	−11.9 ± 1.9
1	15 ± 4	1316 ± 76	−11.6 ± 2.5
10	19 ± 4	1343 ± 81	−10.6 ± 1.0
100	18 ± 3	1328 ± 34	−11.1 ± 1.2
1000	17 ± 3	1321 ± 81	−11.5 ± 1.4
10000	15 ± 6	1341 ± 107	−13.9 ± 5.2

Table 8
Performance comparisons between different values of m (Avg ± Range).

m	#Topics	Perplexity	Topic coherence
0.01	17 ± 9	10191 ± 7291	−11.5 ± 1.3
0.1	18 ± 5	7976 ± 3953	−12.6 ± 4.1
1	19 ± 4	8028 ± 3034	−12.8 ± 4.3
10	17 ± 4	3359 ± 1029	−12.8 ± 3.7
100	18 ± 4	1780 ± 132	−12.5 ± 3
1000	17 ± 3	***1363*** ± ***134***	−12.5 ± 5.7
10000	20 ± 2	1457 ± 136	−12.7 ± 1.3

Table 9
Performance comparisons between different values of $|S_d|/|X_d|$ (Avg ± Range).

$\lvert S_d\rvert/\lvert X_d\rvert$	#Topics	Perplexity	Topic coherence	Time
0.1	7 ± 2	1859 ± 479	−17.3 ± 4.2	***6*** ± 1s
0.2	15 ± 3	***1223*** ± 138	−18.4 ± 11.9	8 ± 1s
0.3	15 ± 4	1276 ± 53	−14.0 ± 3.7	9 ± 1s
0.4	18 ± 5	1350 ± 122	−11.9 ± 2.3	10 ± 2s
0.5	20 ± 3	1376 ± 63	−11.3 ± 1.8	11 ± 2s
0.6	21 ± 4	1455 ± 70	−11.0 ± 1.4	11 ± 2s
0.7	24 ± 1	1480 ± 86	−***10.8*** ± 0.7	11 ± 2s
0.8	25 ± 3	1510 ± 73	−11.5 ± 4.1	11 ± 2s
0.9	23 ± 4	1575 ± 50	−11.4 ± 3.0	13 ± 1s

Table 10
Performance comparisons between different values of $|B_d|/|S_d|$ (Avg ± Range).

$\lvert B_d\rvert/\lvert S_d\rvert$	#Topics	Perplexity	Topic coherence	Time
0.75	21 ± 2	1622 ± 84	−11.0 ± 0.6	11 ± 3s
0.80	22 ± 4	1525 ± 42	−11.3 ± 0.7	12 ± 2s
0.85	20 ± 5	1420 ± 109	−10.9 ± 1.4	11 ± 1s
0.90	19 ± 2	***1330*** ± 63	−12.5 ± 3.3	10 ± 1s

bers, the top words of which are similar. Thus, we conduct a comparison between the top topic words on the NIPS dataset extracted by pDTM and TW-pDTM, which are shown in Tables 11 and 12, respectively.

In Table 11 and Table 12, the bolded words are unrelated words, and the italics are topic-indiscriminate words in corresponding topics. Note that although the topic-indiscriminate words scatter across multiple topics, they will not make the topic semantics confusing. Actually, the irrelevant words mainly cause the confusion of the topic semantics. According to Tables 11 and 12, although the number of topic-indiscriminate words extracted by the two models is almost the same, it

Table 11
Top words on NIPS dataset by TW-pDTM.

Topic	Top words
1	*database network functional* componential controllable **fifteen curried** weighted conditional entries
2	*modeled* imagery learnt effected patterned vision methodologies strategy faced found
3	*database* cases algorithmic convergent continuously computed contextual dimensional classified class
4	*network* filter analyze equipment fixes noise layered circuitry correlational activation
5	*systematic* setpoint problematic resultant unitary approximator pointed applied **spacecraft** weighted
6	*systematic database* connectionism setpoint resultant patterned cellular vector vectorial considerable
7	*database functional* variance informational mean norm methodologies controllable optimality linearities
8	*network modeled* recognizable setpoint systematic classifies speechreading wordspotter classified performances

Table 12
Top words on NIPS dataset by pDTM.

Topic	Top words
1	*modeled systematic* signaled frequent filterbank **auer** sour chaos delayed phased
2	*modeled* cellular visualization **fifteen** orientation motivate receptor **corty** directional inquiry
3	*network modeled* norm trees graphic variance probable tree **fifteen believe**
4	*systematic modeled* **eyes** moves **motorola** header directional positional developed cellular
5	*network systematic* phased **actor** oscillator oscillatory modulo modules patterned connectionism
6	*modeled* objection imagery imaginary viewed **feb** features recognizable **parthasarathy** visit
7	*network systematic functional neuronal* neutrally dynamical equipment pointed enforce **fifteen**
8	*modeled neuronal* cellular synaptically inquiry **spikes** firm potentially synaptic rated

is obvious that more irrelevant words are extracted by pDTM (12 irrelevant words in Table 12) than by TW-pDTM (3 irrelevant words in Table 11). The reason for the similar number of topic-indiscriminate words is that in TW-pDTM, the word with a weight of less than 1 has a lower sampling probability than the word with a weight of 1. Even if the sampling probabilities of topic-indiscriminate words are rather smaller, their scale is larger than other words.

For the top-performing neural topic model, i.e., GNBNTM, there are also some irrelevant words in the generated topics. For example, for a topic discovered by GNBNTM with top words as *"task delay architecture network level object space university real matrix"*, the words *delay* and *university* are considered irrelevant to the topic.

From these results, we can conclude that TW-pDTM does improve the quality of topic extraction by reducing the topic-indiscriminate words and irrelevant words.

5.7.2. Effectiveness of topic evolution modelling

To show the dynamic characteristics of TW-pDTM, we collect the textual data from 2011 and 2014 on Sina Microblog[4]. Since the data is in Chinese, we processed them by removing punctuation, numbers, English characters, and stop words. Chinese word segmentation is conducted by Jieba[5]. Considering the imbalance of the data, the random filtering was employed to ensure that the size of data in each year is almost the same. Finally, we input the processed data into TW-pDTM in batches chronologically. Tables 13 and 14 show the topic extraction results for each year, and the words in these tables have been translated into English.

In Tables 13 and 14, we list the top 10 words of each topic in 4 years. The topics of each year consist of some hot events and some daily topics.

Among the 4 topics obtained in 2011, topic 1, topic 2, and topic 4 correspond to 3 hot events respectively, that is, the death of Steven Jobs, the big earthquake in Japan and the XiaoYue-yue car accident in China. Among the 5 topics obtained in 2012, topic 1 and topic 5 correspond to the new hot events respectively, i.e., the London Olympics Games and the Doomsday prophecy. The topics corresponding to the hot events in 2011 have been disappeared. Topic 2, topic 3, and topic 4 in 2012 are daily topics. Note that topic 3 in 2012 derives from topic 3 in 2011. Though the former is more related to marriage, both of them are related to love. Among the 5 topics obtained in 2013, topic 1, topic 2, and topic 4 correspond to the emerging hot events, that is, the earthquake in Ya'an, the popularity of WeChat, and the premiere of the reality show "Where Are We Going, Dad?". Similarly, the topics corresponding to the hot events in 2012 have been disappeared. Also note that topic 3 in 2013 derives from topic 3 in 2012. They are both related to happy life, except that the former is more related to marriage while the latter is more related to Christmas. Among the 6 topics obtained in 2014, topic 1, topic 5, and topic 6 correspond to the Brazil World Cup, the MH370 accident, and the terrorist attack in Kunming Railway Station, respectively. Topic 3 and topic 4 illustrate the rise of e-commerce and live broadcast industries. Meanwhile, topic 4 in 2014 derives from topic 3 in 2013. We show the evolution of the topic about "happy life" by bolded words in each year.

[4] https://weibo.com
[5] https://pypi.org/project/jieba/

Table 13
Top words in 2011 and 2012 on Sina Microblog.

Year	Topic	Top words
2011	1	iPhone telecommunications Jobs microblog Android world company product mobile share
	2	tsunami Japanese earthquake meditation aftershock nuclear-leak dead repost pollution nuclear-power-plant
	3	**life lonely living happiness hope hahaha like love single microblog**
	4	event Xiaoyue-Yue pedestrians indifferent society car-accident meditation responsibility share distressed
2012	1	Olympic-Games headline runner-up London share China champion third-place effort first-place loser
	2	job postgraduate online recommend effort chance perform Tsinghua teacher repost
	3	**wedding love wife gift love sincere microblog woman happiness forever**
	4	constellation share Gemini character Scorpio quotations Aries share Leo Cancer
	5	doom Maya world share true rumor earthquake tsunami polar-night destroy

Table 14
Top words in 2013 and 2014 on Sina Microblog.

Year	Topic	Top words
2013	1	earthquake rescue troop China share Ya'an reconstruct Android cold-area discover
	2	download WeChat Samsung support world new Android mobile share
	3	**red-pocket Christmas-Eve happy apple Christmas repost gift happiness focus children**
	4	daddy where children life share actor sina reality-show Koren live-broadcast
	5	youth living China microblog travel global Thai recommend free Beijing
2014	1	the-World-Cup football wonderful goal Brazil football-star score Chinese-Football-Association-Super-League the-sporting-world share
	2	trading-day empty-order microblog volatile rebound index trend market buy-in position
	3	purchase agent micro-market counter quality fashion share Tmall spot complaint
	4	**beauty-cam recording download life camgirl video mobile fans happiness repost**
	5	Malaysia-Airline-Flight airliner focus missing pray terrorist news truth search-and-rescue repost
	6	railway-station Kunming mob terrorist casualties society share scene nation conflict

There are still some topic-indiscriminate words in Tables 13 and 14, such as "microblog", "repost", and "share". This is because when the users share the microblog without any comment, they will send some default content. The default comment may be "repost microblog" or "share microblog". But these topic-indiscriminate words have little impact on understanding the semantic of topics. In summary, this experiment shows that our model is effective to capture dynamic topics and model the topic evolution of real-world datasets.

5.8. Application on text summarization

To further verify the quality of the topics extracted by TW-pDTM, in this part, we conduct experiments on the application of text summarization described in Section 4.6.1.

We use two datasets, i.e., NLPCC and CNN + DailyMail, to validate the effectiveness of topics generated by different models on automatic text summarization. NLPCC is a dataset for automatic text summarization in the natural language processing competition held by the International Conference on Natural Language Processing and Chinese Computing in 2017[6]. CNN + DailyMail is some news data collected from Cable News Network and the Daily Mail Network. In this paper, CNN + DailyMail is divided according to the original paper [13], and pre-processed according to [26].

We use Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [19] to evaluate the quality of the generated text summary. It counts the overlapping n-gram proportions between the candidate summary and the reference summary to obtain the similarity. The larger the ROUGE is, the better the quality of the summary. In this experiment, ROUGE-1, ROUGE-2, and ROUGE-L are used. ROUGE-1 and ROUGE-2 count the 1-gram and 2-gram of co-occurrences between the candidate summary and the reference summary, and ROUGE-L counts the longest common subsequence.

In this experiment, the baseline models include ConvS2S [10], Topic–Conv-S2S [25], pHDP [8]-ConvS2S, and GSB [21]-ConvS2S[7], representing that which topic model is utilized to obtain word-topic and document-topic vectors. For NLPCC and CNN + Daily Mail, the document length is limited to 500 words, and the summary length is limited to 100 words [25]. In ConvS2S based models, we use Word2vec [23] to get the word embedding with a dimension of 300. We set the dimension of topic embedding in Topic–ConvS2S and pHDP-ConvS2S to 512. The dimension of topic embedding of TW-pDTM-ConvS2S and GSB-ConvS2S is determined according to the topic number after convergence. To prevent gradient explosion, we renormalized gradients if their norm exceeded 0.1. To prevent overfitting, we added a Dropout layer during the training process, with a dropout ratio of 0.2. In the convolution layers, because the text is not scalable, it must be continuous when extracting word features, thus we set the convolution step to 1. The parameter setting of the topic mining part is referred to Section 5.3.

[6] http://tcci.ccf.org.cn/conference/2017/taskdata.php
[7] All the codes of ConvS2S models are based on the official implementations of Topic–ConvS2S [25]: https://github.com/EdinburghNLP/XSum.

Table 15
ROUGE results on NLPCC dataset (%).

Models	R1	R2	RL
ConvS2S [10]	64.85	25.48	64.56
Topic–ConvS2S [25]	65.46	26.07	65.94
pHDP [8]-ConvS2S	65.37	25.98	65.86
GSB [21]-ConvS2S	65.35	26.39	65.25
TW-pDTM-ConvS2S	**66.71**	**26.67**	**66.65**

Table 16
ROUGE results on CNN + DailyMail dataset (%).

Models	R1	R2	RL
ConvS2S [10]	40.11	17.95	36.69
Topic–ConvS2S [25]	40.56	18.81	37.86
pHDP [8]-ConvS2S	40.68	18.64	37.99
GSB [21]-ConvS2S	41.69	**19.67**	38.62
TW-pDTM-ConvS2S	**41.75**	19.59	**38.83**

The results are shown in Tables 15 and 16, where R1, R2 and RL correspond to ROUGE-1, ROUGE-2 and ROUGE-L, respectively. The effect of automatic text summarization with dynamic topic information is generally better than that with fixed topic information. And TW-pDTM-ConvS2S performs mostly better than GSB-ConvS2S, which further validates the effectiveness of TW-pDTM.

6. Conclusion

Despite the success of the existing topic models in dealing with large-scale datasets or inferring the number of topics, they lack the ability to handle topic-indiscriminate words effectively. In this paper, we proposed a novel topic model named TW-pDTM. It can generate new topics or delete marginal topics according to the evolution of topics, and improve the quality of topic extraction by continuously reducing the proportion of topic-indiscriminate words. Experiments on multiple datasets in terms of topic mining and text summarization have demonstrated the validity of our model. We also qualitatively show the improvement of the extracted topic quality from the perspective of the top words of each topic. In the future, we will focus on reducing the number of parameters that need to be predetermined in the model, such as the threshold for determining whether a topic needs to be deleted, and the sizes of S_d and B_d in each subset. Furthermore, the initialization of S_d in each subset may have a great influence on the model. In light of this consideration, we also plan to improve the stack initialization of the model.

CRediT authorship contribution statement

Hongyu Jiang: Methodology, Software, Writing - original draft. **Zhiqi Lei:** Software, Validation, Writing - review & editing. **Yanghui Rao:** Conceptualization, Methodology, Writing - review & editing. **Haoran Xie:** Writing - review & editing, Supervision. **Fu Lee Wang:** Writing - review & editing, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors are thankful to the reviewers for their constructive comments and suggestions. The research described in this paper was supported by the National Natural Science Foundation of China (61972426), Guangdong Basic and Applied Basic Research Foundation (2020A1515010536), the Direct Grant (DR21A5) and the Faculty Research Grants (DB21B6 and DB21A9) of Lingnan University, Hong Kong, and a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (UGC/FDS16/E01/19).

References

[1] H. Amoualian, M. Clausel, É. Gaussier, M. Amini, Streaming-lda: A copula-based approach to modeling topic dependencies in document streams, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 695–704.

[2] A.U. Asuncion, P. Smyth, M. Welling, Asynchronous distributed learning of topic models, Proceedings of the Advances in Neural Information Processing Systems 21 (2008) 81–88.
[3] D.M. Blei, T.L. Griffiths, M.I. Jordan, J.B. Tenenbaum, Hierarchical topic models and the nested chinese restaurant process, Proceedings of the Advances in Neural Information Processing Systems 16 (2003) 17–24.
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, Journal of Machine Learning Research 3 (2003) 993–1022.
[5] S. Burkhardt, S. Kramer, Decoupling sparsity and smoothness in the dirichlet variational autoencoder topic model, Journal of Machine Learning Research 20 (2019) 1–27.
[6] Z. Cao, S. Li, Y. Liu, W. Li, H. Ji, A novel neural topic model and its supervised extension, in: Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, 2015, pp. 2210–2216.
[7] C. Chen, L. Du, W.L. Buntine, Sampling table configurations for the hierarchical poisson-dirichlet process, in: Proceedings of the Machine Learning and Knowledge Discovery in Databases - European Conference, 2011, pp. 296–311.
[8] D. Cheng, Y. Liu, Parallel gibbs sampling for hierarchical dirichlet processes via gamma processes equivalence, in: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2014, pp. 562–571.
[9] R. Ding, R. Nallapati, B. Xiang, Coherence-aware neural topic modeling, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 830–836.
[10] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y.N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 1243–1252.
[11] J. Gonzalez, Y. Low, A. Gretton, C. Guestrin, Parallel gibbs sampling: From colored fields to thin junction trees, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, 2011, pp. 324–332.
[12] P. Gupta, Y. Chaudhary, F. Buettner, H. Schütze, Document informed neural autoregressive topic models with distributional prior, in: Proceedings of the 33rd AAAI Conference on Artificial Intelligence, 2019, pp. 6505–6512.
[13] K.M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, Proceedings of the Advances in Neural Information Processing Systems 28 (2015) 1693–1701.
[14] M. Isonuma, J. Mori, D. Bollegala, I. Sakata, Tree-structured neural topic model, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 800–806.
[15] H. Iswaran, Gibbs sampling methods for stick - breaking priors, Journal of the American Statistical Association 96 (2001) 161–173.
[16] D.P. Kingma, M. Welling, Auto-encoding variational bayes, in: Proceedings of the 2nd International Conference on Learning Representations, 2014.
[17] X. Li, A. Zhang, C. Li, J. Ouyang, Y. Cai, Exploring coherent topics by topic modeling with term weighting, Information Processing and Management 54 (2018) 1345–1358.
[18] K.W. Lim, W.L. Buntine, C. Chen, L. Du, Nonparametric bayesian topic modelling with the hierarchical pitman-yor processes, International Journal of Approximate Reasoning 78 (2016) 172–191.
[19] Lin, C., 2004. Looking for a few good metrics: Automatic summarization evaluation - how many samples are enough?, in: Proceedings of the Fourth NTCIR Workshop on Research in Information Access Technologies Information Retrieval, Question Answering and Summarization..
[20] L. Lin, Y. Rao, H. Xie, R.Y.K. Lau, J. Yin, F.L. Wang, Q. Li, Copula guided parallel gibbs sampling for nonparametric and coherent topic discovery, IEEE Transactions on Knowledge and Data Engineering (2020), https://doi.org/10.1109/TKDE.2020.2976945.
[21] Y. Miao, E. Grefenstette, P. Blunsom, Discovering discrete latent topics with neural variational inference, in: Proceedings of the 34th International Conference on Machine Learning, 2017, pp. 2410–2419.
[22] Y. Miao, L. Yu, P. Blunsom, Neural variational inference for text processing, in: Proceedings of the 33nd International Conference on Machine Learning, 2016, pp. 1727–1736.
[23] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proceedings of the 1st International Conference on Learning Representations, 2013.
[24] D.M. Mimno, H.M. Wallach, E.M. Talley, M. Leenders, A. McCallum, Optimizing semantic coherence in topic models, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 262–272.
[25] S. Narayan, S.B. Cohen, M. Lapata, Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1797–1807.
[26] S. Narayan, S.B. Cohen, M. Lapata, Ranking sentences for extractive summarization with reinforcement learning, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 1747–1759.
[27] Newman, D., Asuncion, A.U., Smyth, P., Welling, M., 2007. Distributed inference for latent dirichlet allocation, in: Proceedings of the Advances in Neural Information Processing Systems 20, pp. 1081–1088..
[28] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Sharing clusters among related groups: Hierarchical dirichlet processes, Proceedings of the Advances in Neural Information Processing Systems 17 (2004) 1385–1392.
[29] S.G. Walker, Sampling the dirichlet mixture model with slices, Communications in Statistics – Simulation and Computation 36 (2007) 45–54.
[30] H.M. Wallach, D.M. Mimno, A. McCallum, Rethinking LDA: why priors matter, Proceedings of the Advances in Neural Information Processing Systems 22 (2009) 1973–1981.
[31] T. Wang, Y. Cai, H. Leung, Z. Cai, H. Min, Entropy-based term weighting schemes for text categorization in VSM, in: Proceedings of the 27th IEEE International Conference on Tools with Artificial Intelligence, 2015, pp. 325–332.
[32] S. Williamson, A. Dubey, E.P. Xing, Parallel markov chain monte carlo for nonparametric mixture models, in: Proceedings of the 30th International Conference on Machine Learning, 2013, pp. 98–106.
[33] J. Wu, Y. Rao, Z. Zhang, H. Xie, Q. Li, F.L. Wang, Z. Chen, Neural mixed counting models for dispersed topic discovery, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 6159–6169.
[34] K. Yang, Y. Cai, Z. Chen, H. Leung, R.Y.K. Lau, Exploring topic discriminating power of words in latent dirichlet allocation, in: Proceedings of the 26th International Conference on Computational Linguistics, 2016, pp. 2238–2247.
[35] M. Yurochkin, A. Guha, X. Nguyen, Conic scan-and-cover algorithms for nonparametric topic modeling, Proceedings of the Advances in Neural Information Processing Systems 30 (2017) 3878–3887.
[36] M. Zhou, L. Carin, Augment-and-conquer negative binomial processes, Proceedings of the Advances in Neural Information Processing Systems 25 (2012) 2555–2563.