Applied Intelligence
https://doi.org/10.1007/s10489-019-01606-1

A Dirichlet process biterm-based mixture model for short text stream clustering

Junyang Chen[1] · **Zhiguo Gong**[1] · **Weiwen Liu**[2]

Abstract

Short text stream clustering has become an important problem for mining textual data in diverse social media platforms (e.g., Twitter). However, most of the existing clustering methods (e.g., LDA and PLSA) are developed based on the assumption of a static corpus of long texts, while little attention has been given to short text streams. Different from the long texts, the clustering of short texts is more challenging since their word co-occurrence pattern easily suffers from a sparsity problem. In this paper, we propose a Dirichlet process biterm-based mixture model (DP-BMM), which can deal with the topic drift problem and the sparsity problem in short text stream clustering. The major advantages of DP-BMM include (1) DP-BMM explicitly exploits the word-pairs constructed from each document to enhance the word co-occurrence pattern in short texts; (2) DP-BMM can deal with the topic drift problem of short text streams naturally. Moreover, we further propose an improved algorithm of DP-BMM with forgetting property called DP-BMM-FP, which can efficiently delete biterms of outdated documents by deleting clusters of outdated batches. To perform inference, we adopt an online Gibbs sampling method for parameter estimation. Our extensive experimental results on real-world datasets show that DP-BMM and DP-BMM-FP can achieve a better performance than the state-of-the-art methods in terms of NMI metrics.

Keywords Data mining · Stream clustering · Topic modeling

1 Introduction

Short text stream clustering (STSC) has attracted increasing attention due to the explosive volume of short texts on the Internet, e.g., Tweets and Google News. STSC is to cluster the documents as they arrive in a temporal sequence, which has been applied to various applications such as search result diversifications, event detections, and document summarizations [1, 20]. Because of the limited length of short documents, the lack of rich context makes the clustering a challenging problem.

Up to now, many online clustering models have been proposed. DTM [5], DHTM [9] and ST-LDA [3] are proposed for uncovering the hidden topics from text streams and grouping documents with common topics into a cluster. However, their models are designed for long texts such as news articles and academic papers, which face the data sparsity problem in processing short texts. Recently, [27] proposes a model-based clustering algorithm for short text streams, which focuses on the processing of online data but with less consideration on the sparsity problem. In essence, these models all reveal latent topics by implicitly capturing the document-level word co-occurrence pattern [24] in corpora. Nevertheless, this pattern is not applicable for short texts and may suffer from data sparsity problem (the sparse word co-occurrence pattern in short documents [12]). Our experimental results also show that the proposed word-pair pattern is more superior than the mentioned one. In general, the major challenge in short text clustering is the sparse contexts.

✉ Zhiguo Gong
fstzgg@umac.mo

Junyang Chen
yb77403@umac.mo

Weiwen Liu
wwliu@cse.cuhk.edu.hk

[1] State Key Laboratory of Internet of Things for Smart City and Department of Computer and Information Science, University of Macau, Macao, People's Republic of China

[2] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, People's Republic of China

Published online: 01 February 2020

As for the sparsity problem in short texts, one straightforward method to alleviate this problem is to promote the semantically related words under the same topic during the sampling process by using the generalized Pólya urn (GPU) model [18]. For example, GPU-DMM [16] exploits the background knowledge learned from millions of external documents to improve the topic modeling of short texts. However, this model is an off-line model that cannot deal with unseen topics that never occur in the external knowledge before. Moreover, the word semantic relations learned from a massive amount of data can be different from the ones in downstream tasks. In contrast, our proposed DP-BMM is a fully unsupervised model which exploits the word-pairs constructed from the clustering documents to enhance the word co-occurrence pattern in short text streams.

Another way to deal with the sparsity problem is to learn topics by directly modeling the topic generation with the word co-occurrence pattern in the whole corpus. This modeling method ignores the topic distribution at the document-level to avoid the sparsity problem. For example, BTM [26] and iBTM [7] extract the word-pairs co-occurring in a short context and conduct topic modeling using this word-pair pattern. But their models discard the biterm relations during the generative process. By contrast, we utilize this biterm information and assume that biterms in a short document are more likely to be assigned to the same topic in our proposed models.

To tackle the sparsity problem during the short text clustering, we propose a generative Dirichlet process biterm-based mixture model (DP-BMM) which learns the topics over short texts by directly modeling the generation of biterms at the document-level. Here, a *biterm* is an unordered word-pair co-occurring in a short text following the definition in [26]. Compared with the previous bitem-based work [7, 14, 19, 26], the major advantages of our proposed model are that: **(1)** DP-BMM explicitly models the word-pair (i.e., biterm) co-occurrence pattern at document-level to alleviate the sparsity problem. The document generation process under DP-BMM is that a document consists of multiple biterms and these biterms are assigned to the same topic. The assumption of a short context only covering one topic has shown its success in recent work [27–29]. **(2)** DP-BMM can directly obtain the topic probabilities of documents in the modeling. **(3)** DP-BMM can solve the topic drift problem by exploiting Dirichlet process during the online topic learning. Our experiments show that DP-BMM can outperform state-of-the-art models for online document clustering tasks.

Moreover, since we are usually interested in the information within a certain period rather than the whole data stream, we propose an improved algorithm of DP-BMM with forgetting property called DP-BMM-FP. This algorithm, inspired by [27], can efficiently delete biterms of outdated documents by removing clusters of outdated batches, which can prevent the space and time complexity of DP-BMM from growing too large.

The contributions of this paper are summarized as follows:

- We propose a Dirichlet process biterm-based mixture model (DP-BMM) for short text stream clustering, which can alleviate the word sparsity problem in short contexts by explicitly modeling the word-pair (i.e., biterm) co-occurrence pattern at document-level. Moreover, DP-BMM can handle the online topic drift problem by exploiting the Dirichlet process to discover new topics.
- We propose an improved algorithm of DP-BMM with forgetting property called DP-BMM-FP, which can efficiently delete biterms of outdated documents by removing clusters of outdated batches.
- Experimental results on two real-world datasets from Twitter and Google News demonstrate that our proposed model outperforms state-of-the-art approaches in short text stream clustering tasks.

The source code and datasets are available at https://github.com/junyachen/BMM. The rest of this paper is organized as follows. In Section 2, we report the related work. Section 3 introduces the core idea of our proposed models and presents the parameter inference. We discuss experimental results in Section 4. Section 5 concludes our work and gives future work.

2 Related work

According to the general survey on the online data clustering in [1, 20], model-based stream clustering methods can be categorized into the following two categories: fixed-topic-number-based models and DP-based models. These approaches model the latent semantic space by using a pair of distributions (a topic-word distribution and a document-topic distribution) and then use inference approaches (e.g., Gibbs Sampling [10] and Sequential Monte Carlo [8]) to estimate the parameters.

2.1 Fixed-topic-number-based models

Variants of the classical LDA [6] have been proposed for the online data clustering, such as Streaming LDA [3], Dynamic Topic Model [5] and Topic Tracking Model [15]. One major limitation of the above models is that they set a fixed number of topics. The same limitation also occurs in the previous short text clustering models, such as [22, 31]. Since the optimal number of topics varies with time

and dataset, the fixed number way is not appropriate for the data stream clustering and cannot deal with the topic evolution problem. Moreover, these models assume that each document contains multiple topics, which are designed for long texts but do not hold well in short text streams. Then, Dynamic Clustering Topic Model [17] handles the short texts by assigning a single topic to each document and inferring the parameters batch by batch. However, the major challenge of the topic drift problem still exists. Therefore, the following DP-based models are proposed to solve this problem.

2.2 DP-based models

Dirichlet Process (DP) method [23] is often used in Bayesian nonparametric topic modeling, and shows its effectiveness in predicting the number of topics in the existing DP-based models [27, 29]. The models with DP are widely used for the evolutionary clustering that can dynamically increase the number of topics in data streams. Dirichlet-Hawkes Topic Model (DHTM) [9] and Temporal Dirichlet Process Mixture Model (TDPM) [2] are proposed for the automatic topic discovery. However, DHTM is not designed for the short text clustering and TDPM is an offline model. Then, a model-based clustering for short text stream algorithm (MStream) based on the Dirichlet Process Multinomial Mixture Model [29] is proposed by [27]. To sum up, all of the above models still suffer from the word sparsity problem in short texts because they all follow the sparse word co-occurrence pattern.

3 Proposed approaches

Conventional topic models may suffer from the word sparsity problem since they all make use of the sparse word co-occurrence pattern of short texts during the topic learning. To tackle this problem, we propose a Dirichlet process biterm-based mixture model which can model the document generation process by explicitly exploiting the word-pair (i.e., biterm) co-occurrence pattern at document-level.

3.1 Biterm construction

Following the similar way as in [26], a *biterm* is defined as an unordered word-pair extracted from words in a short document. For example, a tweet "*ai accelerate robot*", after stopword removing and word stemming, can be extracted to be the following biterms: "ai accelerate", "ai robot" and "accelerate robot". One of the major differences compared with the previous biterm-based work [7, 14, 26] is that all the biterms constructed from a short context are assigned to the same topic in our work. Since online texts from social media are usually short and specific (e.g., the average length of tweets after preprocessing is around 8), it is reasonable to assume that each short document only covers one topic [27]. After constructing biterms, the number of words in each short text can be promoted to a reasonable amount (e.g., a document containing n words is promoted to a document having $\frac{n*(n-1)}{2}$ biterms), which greatly alleviate the word sparsity problem in short documents. The construction of biterms is formulated as follows:

$$\begin{aligned} B_d &= \{(w_i, w_j) | w_i, w_j \in d, i \neq j\} \\ B &= \bigcup_{d \in D} B_d \end{aligned} \tag{1}$$

where B_d is the set of biterms extracted from document d, and each biterm $b \in B_d$ contains two unordered words (w_i, w_j), D and B are the sets of current recorded documents and biterms, respectively.

3.2 Data structure of DP-BMM

The fine design of data structure can help models to add or delete information effectively during the online clustering. Yin et al. [27] represent a document with its word frequencies and represent a cluster with a cluster feature (CF) vector which can efficiently delete words of outdated documents from clusters. In contrast, for DP-BMM, a document consists of multiple biterms. The data structure of the CF vector in DP-BMM can be described as follows:

CF vector The cluster feature (CF) vector for a cluster z is defined as a tuple $\{\mathbf{n}_z, m_z, n_z\}$, where

- $\mathbf{n}_z$ represents a list of word occurrences in biterms assigned to cluster z
- m_z is the number of documents in cluster z
- n_z is the number of words in cluster z

The CF vector can be used to conduct operations of effectively adding/deleting information of document d into/from cluster z, which can be formulated as follows:

$$\begin{aligned} n_z^{w_i} &= n_z^{w_i} \pm b^{w_i}, n_z^{w_j} = n_z^{w_j} \pm b^{w_j} \quad \forall b \in d \\ m_z &= m_z \pm 1 \\ n_z &= n_z \pm N_d \end{aligned} \tag{2}$$

where b^{w_i} and b^{w_j} are the numbers of occurrences of word w_i and word w_j in biterm b (the values of them are one in the biterm model), $n_z^{w_i}$ and $n_z^{w_j}$ are the numbers of occurrences of word w_i and word w_j in cluster z, N_d is the total number of words in document d, that $N_d = \sum_{b \in d}(b^{w_i} + b^{w_j})$. These effective operations of CF vectors are useful during the online document clustering.

3.3 The generative process of DP-BMM

The proposed DP-BMM learns topics based on the word-pair (i.e., biterm) pattern to alleviate the sparsity problem in short texts. Moreover, we exploit Dirichlet process (DP) [23] for the assignments of topic proportions. DP is a widely used method in non-parametric Bayesian topic modeling and show its effectiveness in predicting the number of topics in documents [11, 27, 29]. In the generative process of DP-BMM, we utilize the assumption that biterms in a short text are more likely to be assigned to the same topic, which help to ease the topic sparsity problem in a short context.

The generative process of the proposed DP-BMM can be described as follows:

1. Sample a global base distribution over topic proportions, $G|\gamma \sim GEM(\gamma)$
2. Sample a topic distribution $\theta \sim DP(\alpha, G)$
3. For each topic $z \in \{1, 2, ...\}$:
   (a) Draw a topic-specific word distribution $\phi_z \sim Dirichlet(\beta)$
4. For each document $d \in \{1, 2, ...\}$:
   (a) Draw a topic assignment $z \sim Multinomial(\theta)$
   (b) For each biterm b in the biterm set B_d:
      (i) Emit a biterm $b \sim Multinomial(\phi_z)$

where (α, γ) are the hyper-parameter in DP, GEM stands for *Griffiths-Engen-McCloskey* distribution [21], and β is the parameter of the Dirichlet distribution [4]. Note that, GEM and Dirichlet are the conjugate distributions of Multinomial distribution, which are usually regarded as the prior distributions in Bayesian inference [4]. Following the above generative process, the probability of a biterm b drawing from cluster z can be formulated as:

$$p(b|\phi_z) = p(w_i, w_j|\phi_z) \propto p(w_i|\phi_z)p(w_j|\phi_z) \tag{3}$$

Then, the probability of document d generated by cluster z is defined as follows:

$$p(d|\phi_z) = \prod_{b \in d} p(b|\phi_z) \tag{4}$$

where we make the Naive Bayes assumption that each biterm in a document is generated independently when the cluster assignment of this document is given. And the probability of a word in a biterm is also independent of its position.

From the generation process, we explicitly model the word-pair pattern at the document-level and directly obtain the topic probability of a document. This pattern alleviates the word sparsity problem which is better than the single word pattern used in the traditional topic models. In addition, the modeling process of a document generation can be directly applied to the document clustering tasks. Moreover, the topic generation is constructed by $\theta \sim DP(\alpha, G)$ [23, 27], which can deal with the topic drift problem during the stream clustering.

3.4 Topic inference

For DP-BMM, we adopt the online Gibbs sampler [27] to perform parameter inference. Gibbs sampling is a widely applicable Markov chain Monte Carlo algorithm. In DP-BMM, there are three latent variables (i.e., z, θ, ϕ) required to be inferred. With the technique of Gibbs sampling, the variables of θ, ϕ can be integrated out for the conjugate prior distributions with the parameters α and β. Therefore, the inference can be simplified to sample the topic assignments of each document from its posterior probability. The derivation process will be introduced in the followings.

Given the recorded documents $\mathbf{d}$ except the current document d, the probability of document d choosing cluster z is formulated as:

$$\begin{aligned} &p(z_d = z|\mathbf{z}_{\neg d}, \mathbf{d}, \alpha, \beta) \\ &\propto p(z_d = z|\mathbf{z}_{\neg d}, \alpha)p(d|z_d = z, \mathbf{d}_{z,\neg d}, \beta) \end{aligned} \tag{5}$$

where the first term in (5) indicates the topic probability of document d after being given the topic assignments of other documents. The derivation of the first term is not different from the inference of traditional DP-based topic models and the details can be found in [29]. The final formula of the first term in (5) is shown as follows:

$$p(z_d = z|\mathbf{z}_{\neg d}, \alpha) \propto \begin{cases} \frac{m_{z,\neg d}}{D-1+\alpha D}, & \text{if } z \text{ exists} \\ \frac{\alpha D}{D-1+\alpha D}, & \text{if } z \text{ is new} \end{cases} \tag{6}$$

where $m_{z,\neg d}$ indicates the number of documents in cluster z except document d, and αD is the pseudo number of documents [27] in the new cluster.

The second term in (5) considers the similarity between the words in clusters z and document d, which can be further rewritten as follows:

$$\begin{aligned} p(d|z_d &= z, \mathbf{d}_{z,\neg d}, \beta) \\ &= \int p(\phi_z|\mathbf{d}_{z,\neg d}, \beta)p(d|\phi_z, z_d = z)d\phi_z \end{aligned} \tag{7}$$

where ϕ_z is the posterior distribution of Dirichlet prior, and $p(d|\phi_z)$ is the product of of multinomial distributions described in (4). In the derivation, we utilize Bayesian inference and exploit the property that Dirichlet distribution is a conjugate prior of multinomial distribution.

Without loss of generality, we follow the traditional models presenting topics as groups of correlated words in DP-BMM. We use Dirichlet distribution as the prior distribution of the topic-specific word distribution as shown in the first term of (7). For the second term of (7), we exploit

the word-pair (i.e., biterm) pattern in the generation. Then, combined with (3) and (4), (7) can be derived as follows:

$$
\begin{aligned}
&p(d|z_d = z, \mathbf{d}_{z,\neg d}, \beta) \\
&= \int \frac{\Gamma(\sum_{v=1}^{V}(n_{z,\neg d}^{v}+\beta))}{\prod_{v=1}^{V}\Gamma(n_{z,\neg d}^{v}+\beta)} \prod_{v=1}^{V} \phi_{z,v}^{n_{z,\neg d}^{v}+\beta-1} \prod_{b\in d} \phi_{z,w_i}^{n_{z,\neg d}^{w_i,b}} \phi_{z,w_j}^{n_{z,\neg d}^{w_j,b}} \\
&= \frac{\Gamma(\sum_{v=1}^{V}(n_{z,\neg d}^{v}+\beta))}{\prod_{v=1}^{V}\Gamma(n_{z,\neg d}^{v}+\beta)} \frac{\prod_{v=1}^{V}\Gamma(n_{z}^{v}+\beta)}{\Gamma(\sum_{v=1}^{V}(n_{z}^{v}+\beta))} \\
&= \frac{\prod_{b\in d}\prod_{q=1}^{N_d^b}(n_{z,\neg d}^{w_i,b}+n_{z,\neg d}^{w_j,b}+\beta+q-1)}{\prod_{p=1}^{N_d^b}(n_{z,\neg d}+V\beta+p-1)}
\end{aligned} \tag{8}
$$

where $n_{z,\neg d}^{v}$ denotes the number of word v in cluster z except the biterms in document d, $n_{z,\neg d}^{w_i,b}$ and $n_{z,\neg d}^{w_j,b}$ represent the number of word w_i and w_j of biterm b in cluster z except the biterms in document d, respectively, N_d^b represents the number of biterms in document d, V denotes the size of the vocabulary of recorded documents. In addition, for the above derivation, we adopt the following property that $\frac{\Gamma(x+m)}{\Gamma(x)} = \prod_{i=1}^{m}(x+i-1)$.

To sum up, we can obtain the topic probability of document d by adopting Gibbs sampling technique. Combining (5) with (6) and (8), the cluster assignment of document d can be formulated as follows:

$$
p(z_d = z|\mathbf{z}_{\neg d}, \mathbf{d}, \alpha, \beta) \propto
\begin{cases}
\frac{m_{z,\neg d}}{D-1+\alpha D} \frac{\prod_{b\in d}\prod_{q=1}^{N_d^b}(n_{z,\neg d}^{w_i,b}+n_{z,\neg d}^{w_j,b}+\beta+q-1)}{\prod_{p=1}^{N_d^b}(n_{z,\neg d}+V\beta+p-1)}, & \text{if } z \text{ exists} \\
\frac{\alpha D}{D-1+\alpha D} \frac{\prod_{b\in d}\prod_{q=1}^{N_d^b}(n_{z,\neg d}^{w_i,b}+n_{z,\neg d}^{w_j,b}+\beta+q-1)}{\prod_{p=1}^{N_d^b}(n_{z,\neg d}+V\beta+p-1)}, & \text{if } z \text{ is new}
\end{cases} \tag{9}
$$

3.5 The DP-BMM-FP algorithm

The DP-BMM-FP is an improved algorithm of DP-BMM with a forgetting property. Since users are usually more interested in the topics within a certain time period, the proposed DP-BMM-FP can efficiently delete the information of outdated batches from the clusters. Specifically, we first separate the documents into batches based on the stream sequences. Then we set the number of stored batches to n. For the streaming clustering, we only keep the most recent batches and regard the previous ones are outdated. Similar to the subtraction operation proposed by [27], we go a step further to subtract the biterms of documents from outdated batches. The detail of the forgetting property is: For all documents in outdated batches $\forall d \in \mathbf{d}_p$, we perform the subtraction formulated in (2). By this way, we can perform focused analysis on the newly arrived batch data.

4 Experiment

In this section, we evaluate the performance of our proposed models by comparing with state-of-the-art models.

4.1 DataSets

Two real-life datasets from Google News and Twitter and two variants of them are used in the experimental study.

Google news [1] This dataset is one of the labeled collections used to evaluate the clustering performance with the ground truth. It contains 11,108 news articles (including titles and snippets) grouping into 152 topics. In our experiments, we follow the usage of [27] to take the titles of the news as the short text documents. The original average number of words in the documents is 6.23 and the average number of biterms is 18.04.

TweetSet [2] This dataset contains tweets which are labeled in the 2011-2015 microblog tracks at Text REtrieval Conference. The NIST assessors have evaluated all the submitted tweets and retained the quality ones, which involve 269 topics and 30322 tweets. The original average number of words in the documents is 7.97 and the average number of biterms is 32.37.

Synthetic Datasets Google News-T and TweetSet-T are two variants of the above datasets to simulate the situation where topics only appear in a certain time period and then disappear. We follow the process in [27] (sorting Tweets and News by topics, dividing them into 16 equal parts and shuffling them, respectively).

For the above datasets, we only apply a simple preprocessing on them: (1) Convert all letters into the lowercase ones; (2) Remove non-latin cahracters and stop words; (3) Conduct word stemming. After the preprocessing, the statistics about these datasets are given in Table 1.

4.2 Evaluation metrics

In the experiment, we employ a widely used metrics to evaluate the document clustering results.

[1] https://news.google.com/news/

[2] https://trec.nist.gov/data/microblog.html

Table 1 Statistics of the experimental datasets

Dataset	Documents	Topics	Vocabulary	Avg. words	Avg. biterms
Google News & Google News-T	11108	152	8110	6.23	18.04
TweetSet & TweetSet-T	30322	269	12301	7.97	32.37

Normalized Mutual Information (NMI) It has been used in [27, 29] to evaluate clustering results with ground truth, which is formally defined as follows:

$$NMI = \frac{\sum_{h,l} n_{h,l} \log(\frac{n * n_{h,l}}{n_h * n_l})}{\sqrt{(\sum_h n_h \log \frac{n_h}{n})(\sum_l n_l \log \frac{n_l}{n})}} \tag{10}$$

where n_h is the number of documents in cluster h, n_l is the number of documents in topic l, and $n_{h,l}$ is the number of documents in cluster h as well as in topic l, n is the total number of documents. When the clustering results perfectly match the ground truth clusters, the NMI value is one, whereas the NMI value is zero when the clustering results are randomly generated.

4.3 Methods for comparison

We compare DP-BMM and DP-BMM-FP with the following state-of-the-art models in the document clustering field.

DTM Dynamic topic models [5] is an extension of classical topic models (e.g., LDA [6]), which can be used to analyze evolving topics in a sequential collection of documents.

MStream Model-based short text stream clustering algorithm [27] is based on the Dirichlet process multinomial mixture (DPMM) model [29]. MStream can work well on both of the one-pass clustering process and the update clustering process.

MStreamF It is an improved algorithm of MStream [27], which can delete outdated documents efficiently and perform analysis on the current batch.

4.4 Parameter setting

In Google News and Google News-T, we uniformly set $\alpha = 0.6$, $\beta = 0.02$ for the proposed DP-BMM and DP-BMM-FP. In TweetSet and TweetSet-T, $\alpha = 0.3$, $\beta = 0.02$ for the proposed models. In general, for the hyper-parameter settings, we employ a grid-search method to find out the ones with the average best performance. More details of parameter analysis for our models are reported in Section 4.6 and Section 4.7. Besides, for MStream and MStreamF, we follow the setting in [27] and set $\alpha = 0.03$, $\beta = 0.03$. To make a comparison between DP-BMM-FP and MStreamF, we both set the number of stored batches to one. The predefined number of topics in DTM is set at 170 and 300 for Google News and TweetSet, respectively. Furthermore, we uniformly set the iteration number $I = 10$ and run 10 independent trials for all models.

4.5 Comparision with existing models

In this part, we compare the performance of the proposed models with DTM, MStream and MStreamF in terms of NMI metrics. We show the mean and the standard deviation of NMI performance after running all models in Table 2. From Table 2, we can see that the proposed DP-BMM always achieves the highest NMI performance compared with the other clustering models on the four datasets.

In addition, to make a further comparison between DP-BMM and DP-BMM-FP, we report the running time and the

Table 2 NMI results of the experimental datasets

	Google News-T	Google News	TweetSet-T	TweetSet
DP-BMM	0.882 ± 0.004	0.881 ± 0.006	0.865 ± 0.006	0.864 ± 0.004
DP-BMM-FP	0.837 ± 0.005	0.865 ± 0.005	0.850 ± 0.004	0.845 ± 0.006
MStream	0.866 ± 0.002	0.864 ± 0.007	0.851 ± 0.006	0.847 ± 0.006
MStreamF	0.810 ± 0.003	0.853 ± 0.004	0.845 ± 0.003	0.861 ± 0.005
DTM	0.808 ± 0.003	0.796 ± 0.003	0.803 ± 0.002	0.801 ± 0.002

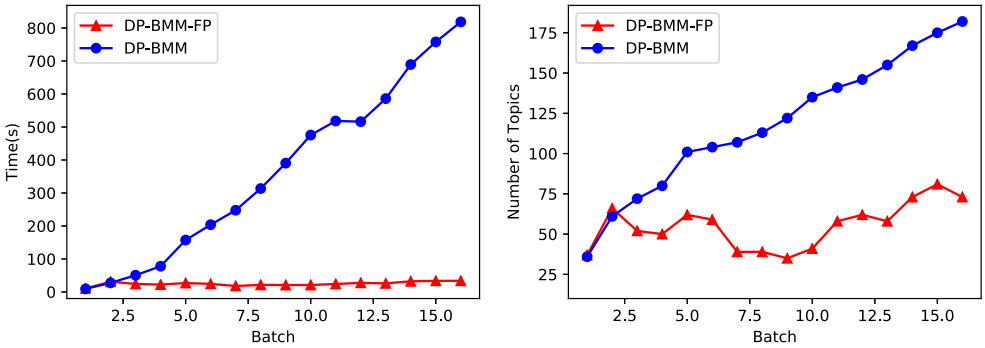


Fig. 1 The running time and the number of topics found on each batch of Google News-T

(a) Running time (b) Number of topics

number of topics found on arriving batches in Figs. 1 and 2. Note that we only show the results on Google News -T and TweetSet-T for easy presentation since we have similar results on Google News and TweetSet. DP-BMM and DP-BMM-FP are both implemented in Python and run on a Windows server with Intel Core i7-6700 3.40GHz CPU and 16GB memory. We set the number of iterations to 10 in the sampling and set the total number of batches to 16. From Figs. 1 and 2, we can see that the running time and the number of topics found of DP-BMM are approximately linear to the size of arriving batches. Compared with DP-BMM, DP-BMM-FP gets similar running time and has approximate number of topics found on the first and second batches. But when dealing with the subsequent batches, DP-BMM-FP can discard outdated batches with a forgetting property (described in Section 3.5) and take shorter running time. In contrast, DP-BMM obtains larger numbers of topics and takes more running time when batches arrive in the subsequent time.

Moreover, we also provide the overall running time compared with the baselines. The comparison results are reported in Fig. 3 where we can find that the sort of time spent follows the sequence: DTM > **DP-BMM** > MStream > **DP-BMM-FP** > MStreamF. Note that MStream is faster than the proposed DP-BMM and MStream-F is faster than DP-BMM-FP. This is because we have data promotion after Biterm Construction (Section 3.1). A document containing n-word is promoted to a document having $\frac{n*(n-1)}{2}$ biterms. The statistics of the datasets are shown in Table 1. In this way, the proposed models can alleviate the data sparsity in short texts and achieve better clustering performance than MStream and MStream-F. Therefore, our proposed models are competitive with the state-of-art ones.

4.6 Influence of alpha to the proposed models

In this part, we try to investigate the influence of hyper-parameter α to the NMI performance and the number of topic found of the proposed models. The value of α ranges from 0.1 to 1.0. Figure 4a shows the NMI performance of DP-BMM with different values of α. From Fig. 4a, we can see that the NMI performance of DP-BMM keep stable with

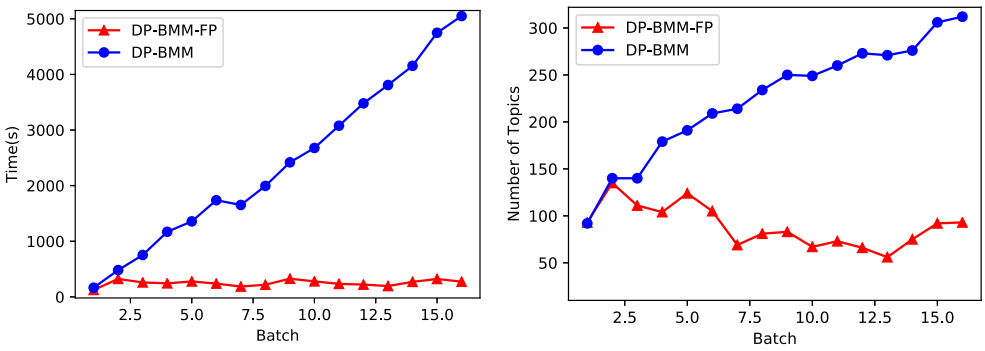


Fig. 2 The running time and the number of topics found on each batch of TweetSet-T

(a) Running time (b) Number of topics

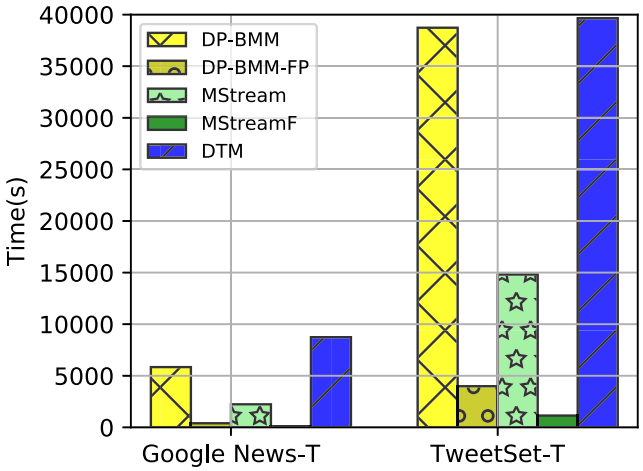


Fig. 3 The overall running time on Google News-T and TweetSet-T

the growing of α on the four datasets. Figure 4b shows the number of topics found by DP-BMM with different values of α. We can observe that the number of topics found increases slowly with α. This is because the probability of a new topic generation (formulated in (6)) grows with α. Figure 5 shows the average NMI performance and the average number of topics found by DP-BMM-FP with different values of α on batches in four datasets. From Fig. 5, we can observe that DP-BMM-FP can achieve stable performance with different α. Another observation is that the average number of topics found by DP-BMM-FP also grows with α.

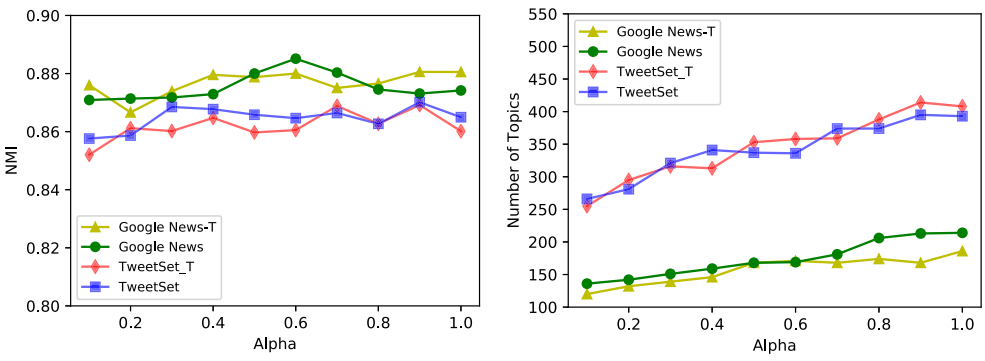
4.7 Influence of beta to the proposed models

In this part, we try to investigate the influence of parameter β to the NMI performance and the number of topics found by the proposed models. We vary the value of β from 0.01 to 0.1. Figure 6a shows the NMI performance of DP-BMM with different values of β. From Fig. 6a, we can see that the changes of the NMI performance of DP-BMM occur slowly with β on the four datasets. Figure 6b shows the number of topics found by DP-BMM with different values of β. We can see that the number of topics found decreases when β becomes larger. The reason of the contrary trend is that the probability of a document choosing a topic becomes less sensitive to the coherence between the document and the topics when the parameter β gets larger. As a result, "richer-get-richer" phenomenon in probabilistic sampling [23] makes larger topics tend to attract more documents, which resulting in generating a smaller number of total topics. In addition, Fig. 7 shows the average NMI performance and the average number of topic found by DP-BMM-FP with different values of β on batches in four datasets. From Fig. 7, we can see that DP-BMM-FP can achieve relatively stable performance with different β. And the average number of topics found by DP-BMM-FP also drops with β.

4.8 Statistical tests

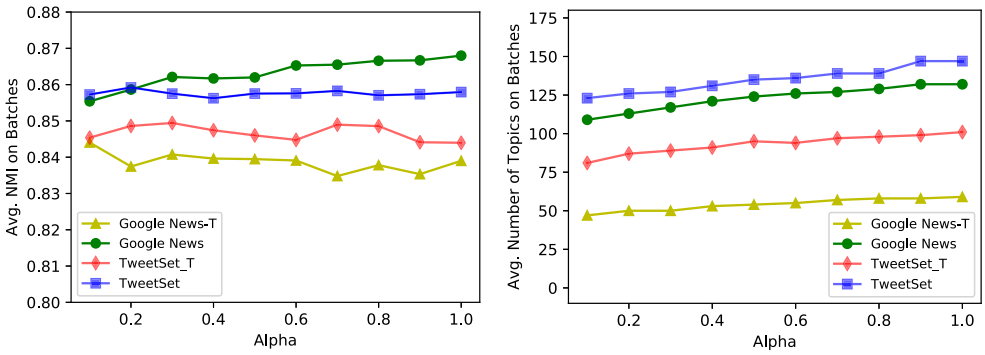
To test whether the clustering performance of our proposed DP-BMM achieves statistically significant results compared with the baselines, we adopt a t-test for the significance test. Then, the p-value can be found from the Student's t-distribution. If it is below the threshold (usually 0.05) chosen for statistical significance, then the difference can be regarded as statistical significance. Table 3 shows the t-test results on different datasets in terms of NMI metrics, where we can find that the p-value is generally very small even close to zero. Therefore, the performance of our proposed model is statically significant.

Fig. 4 Influence of Alpha to the NMI performance and the number of topics found by DP-BMM



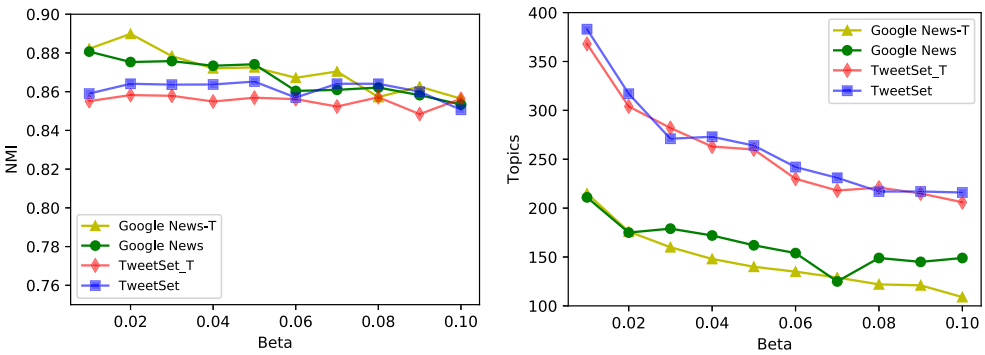
(a) NMI performance

(b) Number of topics



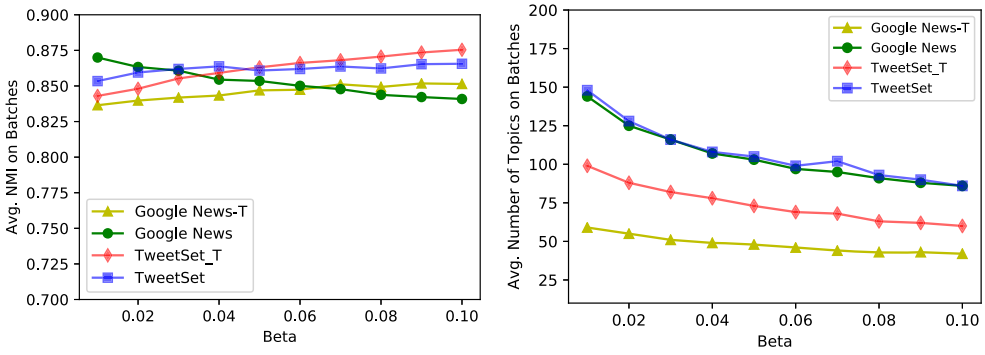
(a) NMI performance (b) Number of topics

Fig. 5 Influence of Alpha to the average NMI performance and the average number of topics found by DP-BMM-FP on batches



(a) NMI performance (b) Number of topics

Fig. 6 Influence of Beta to NMI performance and the number of topics found by DP-BMM



(a) NMI performance (b) Number of topics

Fig. 7 Influence of Beta to the average NMI performance and the average number of topics found by DP-BMM-FP on batches

Table 3 Significance Tests of DP-BMM compared with baselines on NMI

	P-value of NMI			
	Google News-T	Google News	TweetSet-T	TweetSet
MStream	1.295e-9	1.596e-5	5.812e-5	6.595e-7
MStreamF	<1e-19	1.073e-10	8.931e-9	4.804e-2
DTM	<1e-19	<1e-19	<1e-19	<1e-19

5 Conclusion and future work

In this paper, we first propose a Dirichlet process biterm-based mixture model (DP-BMM) to deal with the short text stream clustering. DP-BMM can alleviate the word sparsity problem by constructing biterms and explicitly exploiting the biterm co-occurrence pattern in the modeling. Besides, by employing the Dirichlet process, DP-BMM can handle the topic drift problem during the online document clustering. Our experiments show that DP-BMM can achieve better performance than the state-of-the-art models on real-life datasets. Moreover, we proposed an improved algorithm of DP-BMM with a forgetting property, called DP-BMM-FP, which can efficiently delete biterms of outdated documents by deleting clusters of outdated batches. Comparing with DP-BMM, DP-BMM-FP takes less running time to deal with sequential batches, which is a trade-off between the efficiency of online data processing and the performance of clustering results.

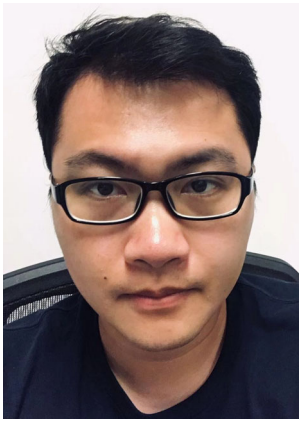
In future work, we intend to exploit our models to improve the performance of other text mining applications, such as word sense disambiguation [25], review mining [30], and time-series tasks [13].

Acknowledgments This work was supported by the Science and Technology Development Fund, Macau SAR (SKL-IOTSC-2018-2020, FDCT/0045/2019/A1, FDCT/007/2016/AFJ), Guangzhou Science and Technology Innovation and Development Commission (EF005/FST-GZG/2019/GSTIC), Research Committee of University of Macau (MYRG2017-00212-FST , MYRG2018-00129-FST).

References

1. Aggarwal CC (2013) A survey of stream clustering algorithms. In: Data clustering. Chapman and Hall/CRC, pp 231–258
2. Ahmed A, Xing E (2008) Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In: Proceedings of the 2008 SIAM international conference on data mining. SIAM, pp 219–230
3. Amoualian H, Clausel M, Gaussier E, Amini MR (2016) 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. ACM, pp 695–704
4. Bernardo JM, Smith AF (2009) Bayesian theory, vol 405. Wiley, New York
5. Blei DM, Lafferty JD (2006) Dynamic topic models. In: Proceedings of the 23rd international conference on machine learning. ACM, pp 113–120
6. Blei DM, Ng AY, Jordan MI (2003) Latent dirichlet allocation. J Mach Learn Res 3(Jan):993–1022
7. Cheng X, Yan X, Lan Y, Guo J (2014) Btm: Topic modeling over short texts. IEEE Trans Knowl Data Eng 26(12):2928–2941
8. Doucet A, De Freitas N, Gordon N (2001) An introduction to sequential monte carlo methods. In: Sequential Monte Carlo methods in practice. Springer, pp 3–14
9. Du N, Farajtabar M, Ahmed A, Smola AJ, Song L (2015) Dirichlet-Hawkes processes with applications to clustering continuous-time document streams. In: Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 219–228
10. Griffiths TL, Steyvers M (2004) Finding scientific topics. Proceedings of the National Academy of Sciences 101(suppl 1):5228–5235
11. Guo J, Gong Z (2016) A nonparametric model for event discovery in the geospatial-temporal space. In: Proceedings of the 25th ACM international on conference on information and knowledge management. ACM, pp 499–508
12. Hong L, Davison BD (2010) Empirical study of topic modeling in twitter. In: Proceedings of the first workshop on social media analytics. ACM, pp 80–88
13. Hu J, Zheng W (2019) Transformation-gated lstm: efficient capture of short-term mutation dependencies for multivariate time series prediction tasks. In: 2019 International joint conference on neural networks (IJCNN). IEEE, pp 1–8
14. Hu X, Wang H, Li P (2018) Online biterm topic model based short text stream classification using short text expansion and concept drifting detection. Pattern Recogn Lett 116:187–194
15. Iwata T, Watanabe S, Yamada T, Ueda N (2009) Topic tracking model for analyzing consumer purchase behavior. In: IJCAI, vol 9, pp 1427–1432
16. Li C, Wang H, Zhang Z, Sun A, Ma Z (2016) Topic modeling for short texts with auxiliary word embeddings. In: Proceedings of the 39th international ACM SIGIR conference on research and development in information retrieval. ACM, pp 165–174
17. Liang S, Yilmaz E, Kanoulas E (2016) Dynamic clustering of streaming short documents. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 995–1004
18. Mahmoud H (2008) Pólya urn models. Chapman and Hall/CRC, London
19. Mai K, Mai S, Nguyen A, Van Linh N, Than K (2016) Enabling hierarchical Dirichlet processes to work better for short texts at large scale. In: Pacific-asia conference on knowledge discovery and data mining. Springer, pp 431–442
20. Nguyen HL, Woon YK, Ng WK (2015) A survey on data stream clustering and classification. Knowledge and Information Systems 45(3):535–569
21. Pitman J et al (2002) Combinatorial stochastic processes. Tech. rep. Technical Report 621, Dept. Statistics, UC Berkeley. Lecture notes for St. Flour Summer School
22. Quan X, Kit C, Ge Y, Pan SJ (2015) Short and sparse text topic modeling via self-aggregation. In: Twenty-fourth international joint conference on artificial intelligence
23. Teh YW (2011) Dirichlet process. In: Encyclopedia of machine learning. Springer, pp 280–287
24. Wang X, McCallum A (2006) Topics over time: a non-markov continuous-time model of topical trends. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, pp 424–433
25. Wang Y, Wang M, Fujita H (2019) Word sense disambiguation: a comprehensive knowledge exploitation framework. Knowledge-based Systems, p 105030
26. Yan X, Guo J, Lan Y, Cheng X (2013) A biterm topic model for short texts. In: Proceedings of the 22nd international conference on World Wide Web. ACM, pp 1445–1456

27. Yin J, Chao D, Liu Z, Zhang W, Yu X, Wang J (2018) Model-based clustering of short text streams. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. ACM, pp 2634–2642
28. Yin J, Wang J (2014) A dirichlet multinomial mixture model-based approach for short text clustering. In: Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 233–242
29. Yin J, Wang J (2016) A model-based approach for text clustering with outlier detection. In: 2016 IEEE 32nd international conference on data engineering (ICDE). IEEE, pp 625–636
30. Yuan C, Zhou W, Ma Q, Lv S, Han J, Hu S (2019) Learning review representations from user and product level information for spam detection. arXiv:1909.04455
31. Zuo Y, Wu J, Zhang H, Lin H, Wang F, Xu K, Xiong H (2016) Topic modeling of short texts: a pseudo-document view. In: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. ACM, pp 2105–2114



Junyang Chen is currently a Ph.D. student in the Faculty of Science and Technology, University of Macau, Macau, China. He received Msc degree in Electronic Commerce from City University of Hong Kong, China in 2014. He received B.S. degree in software engineering from Guangdong University of Technology, China in 2013. His research interests include machine learning algorithms, information retrieval, and data mining.

Zhiguo Gong received the Ph.D. degree in computer science from the Institute of Mathematics, Chinese Academy of Science, Beijing, China. He is currently a Professor with the Faculty of Science and Technology, University of Macau, Macau, China. His current research interests include machine learning, data mining, database, and information retrieval.

Weiwen Liu is currently a Ph.D. student in Computer Science and Engineering at the Chinese University of Hong Kong. She received a B.S. degree in computer science and technology from the South China University of Technology in 2016. Her research interests include machine learning algorithms, information retrieval, and recommender systems.