Information Sciences 504 (2019) 32–47

Contents lists available at ScienceDirect

Information Sciences

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

A nonparametric model for online topic discovery with word embeddings

Junyang Chen [a], Zhiguo Gong [a,*], Weiwen Liu [b]

[a] State Key Laboratory of Internet of Things for Smart City, Department of Computer Information Science, University of Macau, Macau, China

[b] Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong, China

ARTICLE INFO

Article history:
Received 2 April 2019
Revised 27 June 2019
Accepted 12 July 2019
Available online 13 July 2019

Keywords:
Data mining
Clustering
Topic model
Online topic discovery
Nonparametric model
Word embeddings

ABSTRACT

With the explosive growth of short documents generated from streaming textual sources (e.g., Twitter), latent topic discovery has become a critical task for short text stream clustering. However, most online clustering models determine the probability of producing a new topic by manually setting some hyper-parameter/threshold, which becomes barrier to achieve better topic discovery results. Moreover, topics generated by using existing models often involve a wide coverage of the vocabulary which is not suitable for online social media analysis. Therefore, we propose a **n**on**p**ara**m**etric **m**odel (NPMM) which exploits auxiliary word embeddings to infer the topic number and employs a "spike and slab" function to alleviate the sparsity problem of topic-word distributions in online short text analyses. NPMM can automatically decide whether a given document belongs to existing topics, measured by the squared Mahalanobis distance. Hence, the proposed model is free from tuning the hyper-parameter to obtain the probability of generating new topics. Additionally, we propose a nonparametric sampling strategy to discover representative terms for each topic. To perform inference, we introduce a one-pass Gibbs sampling algorithm based on Cholesky decomposition of covariance matrices, which can further be sped up using a Metropolis-Hastings step. Our experiments demonstrate that NPMM significantly outperforms the state-of-the-art algorithms.

1. Introduction

The streaming data generated from social medias (e.g., Twitter) stimulate more and more research works towards online processing of the short texts, such as online data clustering [34], hot topic detection, and event tracking. Currently, probabilistic graphical models are widely used for the short text stream clustering [1], because the model-based scheme can explore the latent hierarchical information well in the semantic space. By constructing a document-topic distribution as well as a mixture of topic-word distributions, the clustering results generated by topic models are more reasonable and interpretable. Generally speaking, model-based stream clustering approaches can be categorized into two groups: batch-based models and DP-based models.

Batch-based models. Traditional online dynamic topic models proposed by Ahmed and Xing [3–5] are designed to perform analyses on long documents, which are not applicable to the short text clustering. These models cannot deal

* Corresponding author.
E-mail addresses: yb77403@umac.mo (J. Chen), fstzgg@umac.mo (Z. Gong), wwliu@cse.cuhk.edu.hk (W. Liu).

https://doi.org/10.1016/j.ins.2019.07.048
0020-0255/

with the sparsity problem in the topic-word distribution since a short document only contains a narrow range of words due to its length limitation. More recently, [22] proposes a batch process model for short text stream clustering, which is under the assumption that each document is associated with only one topic instead of multiple ones. The model iteratively processes the documents in the current batch and discards them when a new batch arrives. The common limitations of the batch-based models are not only that they lack instant processing capability but also they need to set the size of batches manually. Moreover, all of the above models require a predefined topic number during the topic inferring. This is very difficult for an online processing since the number of topics can dynamically change.

DP-based models. To deal with the problem of the unknown topic number and the requirement of the instant processing in the short text stream clustering, topic models exploiting *Dirichlet process (DP)* [37] are proposed. For example, [44] proposes a one-pass clustering process algorithm based on DP that outperforms the state-of-the-art batch process approaches. In general, the number of topics found by DP-based models such as [14,46] is controlled by a concentration hyper-parameter γ. However, the value of this parameter depends on the loose degree of document collections and needs manually setting.

In essence, existing topic models cannot address the problem of inferring a proper topic number when performing the short text stream clustering. In this paper, we propose a **n**on**p**ara**m**etric **m**odel (NPMM) with word embeddings for the online short document clustering. We notice that lots of well-trained word embedding models learned from comprehensive corpora are available due to the advances of embedding techniques, we can incorporate them into the topic modeling of social media data in order to avoid the subjective setting of the parameters. Our idea is based on the following observation: semantic relations among words (e.g., neural network and AI, living standard and economic development) can be stable for a long period of time. Therefore, given a comprehensive semantic space learned from a large modern external corpus, we can easily infer the topics hidden in the sparse short texts of social media. More specifically, the semantic space of word embeddings can be regarded as the prior knowledge when we derive semantic relations among social media data in the stream.

The key to the success of the word embeddings (widely used for NLP task [11,24,33]) is that a large-scale web-based corpus (about billions of training examples covering a wide range of domain knowledge) is used to model the semantic space. Two implicit assumptions of the precondition using embeddings are summarized by Xu et al. [43]: (1) The training corpus of word embeddings is much larger than the data of the down-stream task. (2) The topics in the training corpus are closely aligned with the topics of the down-stream task. Online datasets used in our experiments, i.e., Google News and Tweets, satisfy both of the assumptions. Firstly, the GloVe word embeddings [31] used in our proposed model is a large trained embeddings with 840 billion examples. This meets the first assumption. As in a specific period of time, the amount of generated documents is much smaller than the training corpus of embeddings. Furthermore, the GloVe covers almost all topics/domains of the web and its embedding representations are similar to our experimental web-based datasets. Therefore, word embeddings can provide significant help for distinguishing a new topic brought by a document and make a more accurate online clustering of the stream of sparse social media data.

To summarize the limitations of existing techniques for the online short text clustering, batch-based models *require manually setting both the batch size and the number of topics*, and *lack the instant processing capability*; DP-based models *need manually setting a proper concentration hyper-parameter* γ to control the probability of a new latent topic generation. In addition, due to the length limitation of short texts, the *sparsity* problem exists in the document-topic distribution as well as in the topic-word distribution. For example, a tweet commonly involves only one topic and a topic only covers very few words. Motivated by the above discussions, in this paper we propose NPMM to address the above stated challenges, which is characterized as follows:

(1) *Online processing.* NPMM has *one-pass clustering* process for each arriving document, which can naturally deal with the streaming data.
(2) *Nonparametric topic discovery.* NPMM incorporates the word embeddings [25,26] in the *online topic discovery*, to release the parameter settings for generating new topics in the processing. Assuming that there are several topics already existing in the clustering, our model can distill representative terms for each topic and further construct a multivariate Gaussian distribution in the global semantic space. Then, we can calculate the *squared Mahalanobis distance* subjected to the *chi-squared distribution* between a newly arriving document and the discovered clusters in the global semantic space to decide whether or not to assign a new topic to this document.
(3) *Sparsity.* NPMM exploits the *"spike and slab"* function following [23,27] to distill the representative terms (a.k.a., the focused words) in each topic to decouple the sparsity and smoothness of the topic-word distribution. Since for the short text clustering, there are small differences between the frequencies of relevant words and the irrelevant ones in a cluster, therefore, we use the "spike and slab" technique to amplify their differences. In addition, we also adopt the assumption as given in [22,44] that each document only contains one topic, in order to alleviate the sparsity in document-topic distribution.

To sum up, we propose a nonparametric model (NPMM) for the steaming clustering of the short texts. The contributions of this paper are summarized as follows:

- To the best of our knowledge, it is the first work to incorporate the word semantic knowledge based on word embeddings into the topic model for the topic discovery. NPMM can automatically discover a new topic by calculating the distance between a newly arriving document and the existing topics, and calculate the probability of the new topic generation with the chi-squared distribution.

- The NPMM can discover the representative words in each topic and employ the "spike and slab" function to alleviate the word sparsity in the topic-word distribution.
- To perform inference, we introduce a one-pass Gibbs sampling algorithm based on the Cholesky decomposition of covariance metrics of the squared Mahalanobis distance. The inference process can be further sped up with a Metropolis-Hastings step.
- We evaluate the proposed model against the state-of-the-art approaches on the real-world short text collections. Experimental results demonstrate the superiority of our model in the online short text clustering tasks.

The rest of this paper is organized as follows. In Section 2, we introduce the related work. Section 3 presents the proposed model. Section 4 performs the parameter inference. We discuss experimental results in Section 5. Section 6 concludes the paper and gives our future work.

2. Related work

According to the general survey on the online data clustering in [1,30,34], the stream clustering methods can be categorized into the following two groups: model-based stream clustering and threshold-based stream clustering. Moreover, the model-based methods can be further divided into two branches: DP-based models and batch-based models.

2.1. Model-based stream clustering

Model-based stream clustering methods assume that documents are generated from probabilistic graphical models. They model the latent semantic space by using a pair of distributions (a topic-word distribution and a document-topic distribution), and then use the inference approaches (e.g., Gibbs Sampling [13] and Sequential Monte Carlo [9]) to estimate the parameters in the probabilistic models. In summary, there are two general techniques designed for the topic discovery.

2.1.1. Batch-based models

Variants of the classical LDA [6] have been proposed for the online data clustering, such as Temproral LDA [41], Streaming LDA [4], and Dynamic Topic Model [5]. One major limitation of the above models is that they set a fixed number of the topics. Note that the mentioned batch-based models, if not all, also have this limitation. However, since the optimal number of topics varies with time and dataset, the fixed number setting is not appropriate for the data stream clustering and cannot deal with 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, because in practice, it is more common that a short document only talks about one topic. This assumption induces the sparsity problem in the topic-document distribution. Then, the Dynamic Clustering Topic Model is proposed by Liang et al. [22] to handle the short texts by assigning a single topic to each document and inferring the parameters batch by batch. However, the major problem of an unknown number of topics is still existing. Therefore, to solve this problem, the following DP-based models are proposed.

2.1.2. DP-based models

Dirichlet Process (DP) [37] method, often used in Bayesian nonparametric topic modeling, shows its effectiveness in predicting the number of topics in the existing DP-based models [14,44,46]. A Chinese restaurant process (CRP) [12] is commonly used as an analogy of the DP. Imagine a restaurant with infinite number of tables, each table can seat an infinite number of customers. The first customer sits at the first table. Later, a new arriving customer either chooses a new table with probability $\frac{\gamma}{\gamma+n}$, or chooses an already occupied table k with probability $\frac{n_k}{\gamma+n}$, where n_k is the number of customers of table k, γ is a predefined value and n is the total number of customers. Thus the new customer has higher probability of choosing a new table with a larger γ. And tables with more customers tend to attract more new customers. However, using this method to generate new topics (new tables in CRP) may not be appropriate because the value of γ is a predefined setting and varies according to the specific corpus.

DP-based models are widely used for the evolutionary clustering that can increase the number of topics in data streams. Dirichlet-Hawkes Topic Model (DHTM) [10] and Temporal Dirichlet Process Mixture Model (TDPM) [3] 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 [46] is proposed by [44]. MStream can work well on both the one-pass clustering process and update clustering process. Nevertheless, the limitation of the current DP-based models is that the parameter of the potential topic generation is set manually and it is time-consuming to search for an appropriate value of this parameter.

2.2. Threshold-based stream clustering

Threshold-based stream clustering methods employ the distance metrics like the cosine distance to evaluate the similarity between documents and clusters. For the online text clustering, the following methods have been proposed. A condensation-based approach is proposed by Aggarwal and Philip [2] for the text and categorical data stream clustering, which summarizes the stream into a number of fine grained cluster droplets. Zhong [48] combines an efficient online spherical k-means algorithm with an existing scalable clustering strategy to achieve fast and adaptive clustering of text streams.

Yoo et al. [47] propose a one-pass clustering process which maintains an approximation of the normalized Laplacian of the data stream over time and efficiently updates the changing eigenvectors of this Laplacian in a streaming fashion. Kalogeratos et al. [19] exploit temporal information (terms that appear in many documents during a short time period) to improve the performance of the text clustering algorithm. Specially, for the tweet stream clustering, Shou et al. [32] propose an online tweet stream clustering algorithm to cluster tweets and maintain distilled statistics called Tweet Cluster Vectors in an online fashion.

The main limitation of the aforementioned threshold-based stream clustering methods is that the threshold is set manually to determine the topic assignments of online documents. For example, a newly arriving document choose the nearest cluster or a new one by comparing the distance with the predefined threshold. However, it is a time-consuming operation to search for a proper threshold in different datasets.

3. The NPMM model

In this section, we first formulate the problem of the nonparametric topic modeling, then present the graphic model, and finally give the generative process and the algorithms of NPMM.

3.1. Problem formulation

As mentioned in Section 1, the proposed NPMM constructs a multivariate Gaussian distribution of the discovered topics in the global semantic space using word embeddings. When a new document comes, we can obtain the probability of a new topic generation by calculating the distance between this document and the Gaussian distribution.

Nevertheless, simply using all the words in the existing topics to construct a global Gaussian distribution based on word embeddings may bring much noise, because there are some words weakly connected to their topics. To solve the problem, we need to define the following auxiliary components in the clustering.

3.1.1. Representative terms

In each topic, there are some words having strong ties with the topic, which are called representative terms or focused words. This situation can be understood intuitively. Although all words in a short text likely belong to the same topic, as mentioned in [21,46], some words may be weakly correlated to this topic. Taking a short text *"microsoft sony battle gaming supremacy holiday season"* as an example, this document is associated with Sony-product topic. However, words like *"holiday, season"* could be less relevant to this topic. Therefore, we select "*microsoft sony battle gaming supremacy*" as the representative terms of this sentence. In [21], the authors propose a nonparametric probabilistic sampling strategy to determine whether a word in a document has strong ties with the sampled topics. Intuitively, all documents in the same topic can be regarded as a large pseudo document. We can obtain the representative terms in a topic as follows:

$$\begin{aligned} &\beta_{z,w} \sim Bernoulli(\lambda_{z,w}) \\ &\lambda_{z,w} = \frac{p(w|z)}{\max_{w'} p(w'|z)}, \quad \forall w' \in z, \end{aligned} \tag{1}$$

where $\beta_{z,w}$ indicates whether word w is a representative term in topic z, $\lambda_{z,w}$ is related to the ratio of the conditional word probability of given word w to the maximal word probability of topic z in terms of $p(w'|z)$.

3.1.2. Global semantic space

The global semantic space G is represented as a multivariate normal distribution in the word embedding space, which is constructed by the representative terms from each topic during the clustering. In the scenario of social media streams, the newly coming topic is often outside the convex of those discovered topics, to be free of parameters and make the computation simple, we use just only one multivariate Gaussian to represent those discovered topics. Specifically we use $\boldsymbol{x}_w$ to represent the corresponding vector of w in word embeddings. When a new document d comes, the squared Mahalanobis distance [42] between each word w in d and G is calculated as follows:

$$d_M^2(w, G) = (\boldsymbol{x}_w - \boldsymbol{\mu_0})^T \boldsymbol{\Sigma_0}^{-1} (\boldsymbol{x}_w - \boldsymbol{\mu_0}), \tag{2}$$

where $\boldsymbol{\mu_0}$ and $\boldsymbol{\Sigma_0}$ are the mean and the covariance of the global space G, respectively. As d_M^2 has a chi-square χ^2_{dim} distribution with dim degrees of freedom (where dim is the vector dimension of word embeddings), the probability of document d generated by space G is defined as follows:

$$p(\eta_d = 1|G) = \frac{\prod_{w \in d} \chi^2_{dim}(w|G)}{\prod_{w \in d} \chi^2_{dim}(w|G) + \prod_{w \in d}(1 - \chi^2_{dim}(w|G))} \tag{3}$$

Similarly, the probability of document d not generated by space G is defined as:

$$p(\eta_d = 0|G) = \frac{\prod_{w \in d}(1 - \chi^2_{dim}(w|G))}{\prod_{w \in d} \chi^2_{dim}(w|G) + \prod_{w \in d}(1 - \chi^2_{dim}(w|G))} \tag{4}$$

where G is represented by $\{\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0}\}$, and $\chi^2_{dim}(w|G)$ denotes the probability of word w generated by G. In Eqs. (3) and (4), we make the Naive Bayes assumption that the words in a document are generated independently when the global semantic space G is known. And we further assume that the probability of a word is independent of its position within the document.

3.1.3. The spike and slab priors

The spike and slab priors introduced to topic modeling [23,39,40] are able to address the sparsity of topic-word distribution by using auxiliary Bernoulli variables to present the "on" and "off" of the priors, which indicate whether or not a term is selected by a topic as the representative term (or called the focused word). The selection process is given in Eq. (1). As in the real-world scenarios of short texts, a topic covers a narrow range of terms instead of a wide coverage of the vocabulary. It is hard for the sampling algorithm to distinguish relevant words from the irrelevant ones due to the small difference in word frequencies of short texts.

3.1.4. The cluster feature (CF) vector

The cluster feature vector is used to represent a cluster[1], along with an *addible property* and a *deletable property*. We follow the notations as in [44] that the CF vector for a cluster z is defined as a tuple $(\{n_z^w\}, m_z, n_z)$, where n_z^w is the number of occurrences of word w in cluster z, m_z is the number of documents in cluster z, n_z is the number of words in cluster z. During the clustering, we denote D as the set of recorded documents and V as the set of vocabulary of the recorded documents.

(a). Addible property. When a new document d comes, it will be firstly assigned to a topic z in the one-pass clustering process, the detail is given in the later section. Then, the CF vector and the recorded sets (i.e., D and V) are updated in the following way:

$$\begin{aligned} n_z^w &= n_z^w + N_d^w, \quad \forall w \in d \\ m_z &= m_z + 1 \\ n_z &= n_z + N_d \\ D &= D \cup \{d\} \\ V &= V \cup \{w|w \in d\}, \end{aligned} \tag{5}$$

where N_d is the number of words in document d, N_d^w is the number of occurrences of word w in document d.

(b). Deletable property. Accordingly, when a document d is deleted from cluster z in the update clustering process, the CF vector and the recorded sets are updated as follows:

$$\begin{aligned} n_z^w &= n_z^w - N_d^w, \quad \forall w \in d \\ m_z &= m_z - 1 \\ n_z &= n_z - N_d \\ D &= D \setminus \{d\} \\ V &= V \setminus \left\{ w|w \in d, \ \sum_z n_z^w = 0 \right\} \end{aligned} \tag{6}$$

After defining the above auxiliary components, the major tasks of **NPMM** given a data stream in a certain time range can be defined as:

1. Sample a set of representative terms from the existing clusters (or called topics);
2. Construct a semantic space G with the representative terms using a multivariate normal distribution;
3. Calculate the probability of a newly arriving document d belonging to the semantic space G;
4. Learn the topic-word distribution $\boldsymbol{\varphi}$ and document-topic distribution $\boldsymbol{\theta}$, respectively.

All the notations used in this paper are summarized in Table 1.

3.2. Generative process

As mentioned in Section 1, [23,28,38,44] assume that each document only focuses on one topic in short texts. In our model, we follow this assumption as it holds up well in practice. The graphical model of the proposed NPMM is given in Fig. 1. In the generative process, we use the inverse Wishart distribution as the conjugate prior for the covariance matrix of a multivariate normal distribution. In Bayesian statistics, it is widely used for estimating a multivariate normal distribution with unknown mean and covariance matrix. Since a document is just related to one topic, there are two possible statuses for each newly arriving document in the clustering, either relevant or irrelevant to the existing global semantic space G. This status variable is denoted by r. In addition, $\lambda_{z,w}$ is as defined in Eq. (1) and other notations are summarized in Table 1. The generative process is presented as follows:

[1] Notice that we use topic and cluster in this paper interchangeably, as the documents in a cluster represent the same topic.

Table 1
Notations.

Symbol	Description
V	Set of the vocabulary of recorded documents
D	Set of recorded documents
$\boldsymbol{z}$	Topic assignments of documents
$\beta_{z,\,w}$	Representative term indicator of term w in topic z
m_z	The number of documents in topic z
n_z	The number of words in topic z
n_z^w	The number of occurrences of word w in topic z
d	Document in the clustering
I	The number of iterations
N_d	The number of words in document d
N_d^w	The number of occurrences of word w in document d
r	Relevance indicator
G	global semantic space represented as a multivariate normal distribution
S	Set of representative terms in G
π	Representative/focused term indicator in G
$\boldsymbol{\mu_0}$, $\boldsymbol{\Sigma_0}$	Mean and covariance metric of G
$\boldsymbol{\mu}$, $\boldsymbol{\Psi}$, ν, τ	Normal-Wishart prior set for $\boldsymbol{\mu_0}$, $\boldsymbol{\Sigma_0}$
δ, ϵ	Spike and slab priors of Dirichlet distribution
η	Bernoulli distribution over the relevance indicator
d_M^2	Squared Mahalanobis distance
α	Symmetric prior of Dirichlet distribution
$\boldsymbol{\varphi}$	Multinomial distribution over topical words
$\boldsymbol{\theta}$	Multinomial distribution over topics
χ^2_{dim}	Chi-squared distribution with *dim* degrees of freedom

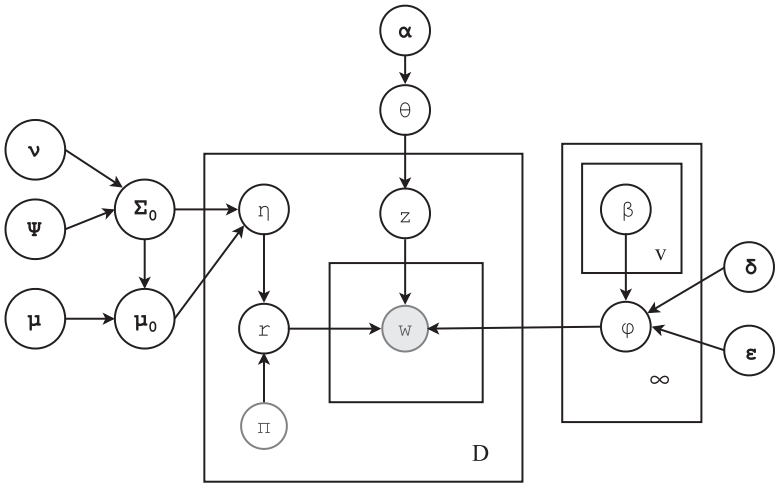


Fig. 1. The graphical model of NPMM.

1. Draw $\boldsymbol{\Sigma_0} \sim \mathcal{W}^{-1}(\boldsymbol{\Psi}, \nu)$.
2. Draw $\boldsymbol{\mu_0} \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{\tau}\boldsymbol{\Sigma_0})$.
3. Draw a topic distribution $\boldsymbol{\theta} \sim Dirichlet(\alpha)$.
4. For each topic $z \in \{1, 2, \ldots, |\boldsymbol{z}|\}$:
   (a) For each term $w \in \{1, 2, \ldots, |V|\}$:
      i. Draw a focused term $\beta_{z,\,w} \sim Bernoulli(\lambda_{z,\,w})$.
   (b) Draw a word distribution $\boldsymbol{\varphi_z} \sim Dirichlet(\delta\boldsymbol{\beta}_z + \epsilon\mathbf{1})$, $\boldsymbol{\beta}_z = \{\beta_{z,w}\}_{w=1}^{|V|}$.
5. For each document $d \in \{1, 2, \ldots\}$:
   (a) Draw $\eta_d \sim \chi^2_{dim}(\boldsymbol{\mu_0}, \boldsymbol{\Sigma_0})$.
   (b) Draw relevance r based on the representative term indicator π and $p(\eta_d)$ (refer to Eqs. (3) and (4)).
   (c) If the document is relevant to the global semantic space G, i.e., $r = 1$:
      i. Draw a topic $z \sim Multinomial(\boldsymbol{\theta})$.
      ii. Emit a word $w \sim Multinomial(\boldsymbol{\varphi_z})$.
   (d) If the document is irrelevant to the global semantic space G, i.e., $r = 0$:
      i. Create a new topic z.

For a better understanding of the process, we give an example to illustrate the clustering when documents stream in. The following documents are extracted from Google News dataset after preprocessing (refer to Section 5.1).

($d1$) *nawaz sharif lt gen raheel sharif Pakistan army chief.*
($d2$) *Pakistan building mw nuclear power project.*
($d3$) *china public back air defence zone.*
($d4$) *pm meet lt general rashid mahmood raheel sharif.*

We assume that the above documents come in order and there is only one cluster/topic containing $d1$ and $d2$ as the initialization. Since a short document only talks about one topic, a new document (e.g., $d3$) can be identified as relevant or irrelevant to the existing topics when it comes. A multivariate normal distribution is used for fitting the semantic space G of the recorded clusters. We use r to represent the relevance status, that is $r \in \{0, 1\}$, where $r = 1$ indicates that $d3$ is relevant to G, and $r = 0$ is irrelevant. The probability of $d3$ having this multivariate normal distribution is based on the Mahalanobis distance as given in Eq. (2).

In addition, we can get the representative terms (marked in bold in the example) following Eq. (1). Another related variable is $\pi \in \{0, 1\}$, which represents whether a document d contains at least one of the representative terms S, where $\pi = 1$ means that the document d contains the representative term, whereas $\pi = 0$ means not. In the above example, S is {"sharif", "Pakistan"}. For $d4$, its term indicator is $\pi = 1$ because it contains the term "sharif". In this case, we believe that $d4$ is relevant to G because it is unlikely that a short sentence contains the representative term "sharif" but not talking about it.

3.3. The spike and slab priors

When document d is sampled as relevant to the global semantic space G, we draw a topic assignment for it and apply the spike and slab priors on its word generation. The key idea of employing the priors is to restrict the size of the word simplex over Dirichlet distribution to reduce the sparsity in the topic-word distribution by incorporating a Bernoulli variable. The spike prior δ is much larger than the slab prior ϵ, that is $\delta \gg \epsilon$. The representative term indicator $\beta_{z,w}$ serves as a switch of "on" and "off" to determine which term is a representative term. In the previous case, when $r = 1$, a topic z is sampled from $\boldsymbol{\theta}$, and a word is emitted from $\boldsymbol{\varphi_z} \sim Dirichlet(\delta\boldsymbol{\beta}_z + \epsilon\mathbf{1})$. This does not violate the definition of a multinomial distribution as $\sum_{w=1}^{|V|} \varphi_{z,w} = 1$, where $|V|$ is the size of the vocabulary of current recorded documents. The spike-and-slab prior setting enables the model to generate more coherent topics. Besides, when $r = 0$ that d is sampled as irrelevant to G, we create a new topic for d and also update the global space G as well.

3.4. The NPMM algorithms

The details of the clustering process for the proposed NPMM model are shown in Algorithms 1 and 2 . The meanings of the notations are as given in Table 1. As mentioned in the introduction section, NPMM has one-pass clustering process and update clustering process. The one-pass scheme (lines 5–8 in Algorithm 1) grants NPMM the instant processing capability to handle the massive amount of short text streams. And the update clustering scheme (or called batch scheme) (lines 9–25 in Algorithm 1) enables NPMM to achieve a better performance through multiple iterations. In addition, Algorithm 2 is a common operation for sampling the latent variables (details in the following section) in NPMM.

4. Inference

We use Gibbs sampling [13] for the parameter inference. There are five latent variables in the model, including topic assignments $\boldsymbol{z}$, relevance indicator r, the posterior normal distribution parameters $\boldsymbol{\mu_0}$ and $\boldsymbol{\Sigma_0}$, and the focused terms $\boldsymbol{\beta}_z$ in each topic. The conditional probabilities of document d being relevant or not are computed as in the followings and the meanings of notations are given in Table 1.

Sampling the relevance status r for document d: Combining Eqs. (3) and (4), we can sample the relevance status r for document d as in the following way:

$$p(r_d = e|\pi_d = f, \eta_d) \propto \begin{cases} p(\eta_d = e|G) & f = 0, e = 0 \text{ or } 1 \\ C * p(\eta_d = e|G) & f = 1, e = 1 \\ (1 - C) * p(\eta_d = e|G) & f = 1, e = 0 \end{cases}, \tag{7}$$

where π_d is the representative/focused term indicator, $\pi_d = 1$ indicates document d containing a focused term and $\pi_d = 0$ indicates not, $r_d = 1$ denotes that document d is relevant to the global space G and $r_d = 0$ denotes not, $p(\eta_d)$ is the probability of document d relating to the global space G. In addition, C is the soft constraint of the relevant confidence level when document d contains a focused term, the value of C is between 1 and 0.

Algorithm 1: NPMM.

Input: Document stream $d \in \{1, 2, \ldots\}$, iteration number I
Result: Number of topics K, topic assignments $\boldsymbol{z}$

```
begin
    //Initialization;
    K = 1;
    Zero the CF vector (see Section 3.1.4);
    //One-pass clustering process;
    for document d = 1,2,... do
        Clustering process(d) (see Algorithm 2) ;
    end
    //Update clustering process;
    for iter = 2 to I do
        for document d = 1,2,... do
            Record the current topic z of d;
            //Updating CF with deletable property (see Eq. (6));
            m_z = m_z − 1,  n_z = n_z − N_d, D = D \ {d};
            for each word w in d do
                n_z^w = n_z^w − N_d^w;
                V = V \ {w| Σ_z n_z^w = 0};
            end
            for each topic z in z do
                Sampling the representative terms (see Eq. (9));
            end
            Updating the mean μ_0 and the covariance Σ_0 of the global space G (see Eq. (10));
            Clustering process(d);
        end
    end
end
```

Sampling the topic indicator z for document d: If $r_d = 0$, a new topic is created and assigned to document d. If $r_d = 1$, then we sample a topic z for d from the current recorded topics as follows:

$$p(z_d = z|\boldsymbol{z}_{\neg d}, D, \alpha, \delta, \epsilon) \propto \frac{m_{z,\neg d}}{|D| - 1 + \alpha} \frac{\prod_{w \subset d} \prod_{j=1}^{N_d^w} (n_{z,\neg d}^w + \beta_{z,w}\delta + \epsilon + j - 1)}{\prod_{i=1}^{N_d} (n_{z,\neg d} + |\boldsymbol{\beta}_z|\delta + |V|\epsilon + i - 1)} \tag{8}$$

where $|\boldsymbol{\beta}_z|$ is the size of all representative terms in topic z. All the CF (described in Section 3.1.4) counters in Eq. (8) are calculated with the current document d excluded.

Sampling the representative terms in topics: We follow Section 3.1.1 to sample the representative/focused terms in each topic. Combined with Eq. (1), the probability of word w being a representative/focused term is formulated as:

$$p(\beta_{z,w} = s|z) = \begin{cases} \frac{n_z^w / n_z}{max_{w'}(n_z^{w'} / n_z)} & \forall w' \in z, \quad s = 1 \\ 1 - \frac{n_z^w / n_z}{max_{w'}(n_z^{w'} / n_z)} & \forall w' \in z, \quad s = 0 \end{cases} \tag{9}$$

Updating the semantic space G: Every time we re-sample the representative terms, we need to update the normal distribution of G because a representative term w is either leaving or joining G. Following [29], the parameters ($\boldsymbol{\mu_0}$ and $\boldsymbol{\Sigma_0}$) of the posterior predictive distribution are given as follows:

$$\begin{aligned} \tau_G &= \tau + |S|, \qquad \boldsymbol{\mu_0} = \frac{\tau \boldsymbol{\mu} + |S| \bar{\mathbf{v}}_{\boldsymbol{G}}}{\tau_G} \\ \nu_G &= \nu + |S|, \qquad \boldsymbol{\Sigma_0} = \frac{\boldsymbol{\Psi_G}}{\nu_G - dim + 1} \\ \boldsymbol{\Psi_G} &= \Psi + \boldsymbol{C_G} + \frac{\tau |S|}{\tau_G} (\bar{\mathbf{v}}_{\boldsymbol{G}} - \boldsymbol{\mu})(\bar{\mathbf{v}}_{\boldsymbol{G}} - \boldsymbol{\mu})^T \\ \bar{\mathbf{v}}_{\boldsymbol{G}} &= \frac{\sum_{i:S} (\mathbf{v_i})}{|S|}, \qquad \boldsymbol{C_G} = \sum_{i:S} (\mathbf{v_i} - \bar{\mathbf{v}}_{\boldsymbol{G}})(\mathbf{v_i} - \bar{\mathbf{v}}_{\boldsymbol{G}})^T \end{aligned} \tag{10}$$

where $\boldsymbol{v_i}$ denotes ith word vector of the representative terms S, $\bar{\mathbf{v}}_{\boldsymbol{G}}$ is the sample mean and $\boldsymbol{C_G}$ is the scaled form of the sample covariance of the representative term vectors in the global space G, $|S|$ is the total number of representative terms,

Algorithm 2: Clustering process(d).

Input: Document d
Result: Updating the latent variables in NPMM
begin
  Check whether d contains a word in the set of representative terms S. $\pi = 1$ means that d contains at least one representative term, otherwise $\pi = 0$;
  if $\pi == 1$ **then**
    A new topic z is created and assigned to d;
    $K = K + 1$;
  end
  else if $\pi == 0$ **then**
    Sampling the relevance status r for d (see Eq. (7)), where $r = 1$ denotes d has a multivariate normal distribution, otherwise $r = 0$;
    if $r == 1$ **then**
      Compute the probability of d choosing each of the topics in $\boldsymbol{z}$ (see Eq. (8));
      Sample a topic z for d;
    end
    else
      A new topic z is created and assigned to d;
      $K = K + 1$;
    end
  end
  //Updating CF with addible property (see Eq. (5));
  $m_z = m_z + 1,\ n_z = n_z + N_d, D = D \cup \{d\}$;
  for *each word w in d* **do**
    $n_z^w = n_z^w + N_d^w$;
    $V = V \cup \{w\}$;
  end
  for *each topic z in* $\boldsymbol{z}$ **do**
    Sampling the representative terms (see Eq. (9));
  end
  Updating the mean $\boldsymbol{\mu_0}$ and the covariance $\boldsymbol{\Sigma_0}$ of the global space G (see Eq. (10)) ;
end

$\boldsymbol{\mu_0}$ and $\boldsymbol{\Sigma_0}$ represent the posterior mean and covariance of the normal distribution of G, τ_G and ν_G are the strength of the priors of mean and covariance, respectively, and $\boldsymbol{\mu}$, $\boldsymbol{\Psi}$, ν, τ are the priors of Normal-Wishart.

4.1. Reduction of the sampling complexity with cholesky decomposition

As shown in Algorithm 2, we need to sample a relevance status for each document d not containing the representative term. The evaluation is based on the squared Mahalanobis distance as given in Eq. (2). The computation time of the inverse of the posterior covariance matrix requires $O(n^3)$ operations. Moreover, when a document is assigned to a topic, the representative terms in this topic change following Eq. (1). These terms form the semantic space G (as described in Section 3.1.2) and we need to recompute the covariance following Eq. (10) as well as the inverse of the covariance matrix used in Eq. (2). In the worst case, the time complexity of the recomputation is $O(IDn^3)$, where I is the iteration number and D is the size of the recoded documents. Since D will become much larger along the processing of the online clustering, it is important to speed up the process. Following the formula in [8], the posterior equation of $\boldsymbol{\Psi_G}$ as shown in Eq. (10) can be updated as follows:

$$\boldsymbol{\Psi_G} \leftarrow \boldsymbol{\Psi_G} + \frac{\tau_G}{\tau_G - 1}(\boldsymbol{\mu_0} - \mathbf{v_i})(\boldsymbol{\mu_0} - \mathbf{v_i})^T \tag{11}$$

Eq. (11) meets the form of a rank 1 update and thus Cholesky decomposition can be used intuitively. Since $\boldsymbol{\Sigma_0}$ is equal to $\boldsymbol{\Psi_G}$ times a scalar factor as shown in Eq. (10), the decomposition can be applied to $\boldsymbol{\Sigma_0}$ as well. According to the proof in [8,35] (covariance $\boldsymbol{\Sigma_0}$ is said to be a positive definite matrix if $\forall \boldsymbol{m} \in \mathbb{R}^{dim}$ has $\boldsymbol{m}^T\boldsymbol{\Sigma_0}\boldsymbol{m} > 0$, where *dim* is the dimensionality of $\boldsymbol{\Sigma_0}$), we can get the updated value of $\boldsymbol{\Sigma_0}$ as:

$$\boldsymbol{\Sigma_0} \leftarrow \boldsymbol{\Sigma_0} + \boldsymbol{m}\boldsymbol{m}^T \tag{12}$$

In Eq. (12), the update operation takes quadratic time. When computing the squared Mahalanobis distance in the form of $(\boldsymbol{x} - \boldsymbol{\mu_0})^T\boldsymbol{\Sigma_0}^{-1}(\boldsymbol{x} - \boldsymbol{\mu_0})$, we can use this operation with the Cholesky decomposition [35] to obtain the factorization of

Table 2
Statistics of the experimental datasets.

Dataset	Documents	Topics	Vocabulary	Avg. len
Google News & Google News-T	11,108	152	7830	6.13
TweetSet & TweetSet-T	30,289	269	10,448	7.72

the covariance $\boldsymbol{\Sigma_0} = \boldsymbol{LL}^T$ in a more efficient manner which takes quadratic time instead of cubic time. Then, Eq. (2) can be further derived with Cholesky decomposition as follows:

$$\begin{aligned}(\boldsymbol{x}-\boldsymbol{\mu_0})^T\boldsymbol{\Sigma_0}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0}) &= (\boldsymbol{x}-\boldsymbol{\mu_0})^T(\boldsymbol{LL}^T)^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0}) \\ &= (\boldsymbol{L}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0}))^T(\boldsymbol{L}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0})),\end{aligned} \tag{13}$$

where $\boldsymbol{L}$ is a lower triangular matrix. The operation of $(\boldsymbol{L}^{-1}(\boldsymbol{x}-\boldsymbol{\mu_0}))$ can be calculated using forward substitution [35] which takes quadratic time. Therefore, the time complexity of calculating the squared Mahalanobis distance is reduced to calculating an inner product which also takes quadratic time (i.e., the time complexity is reduced from $O(IDn^3)$ to $O(IDn^2)$).

4.2. Improved reduction of the sampling complexity with metropolis hastings step

The Metropolis Hastings (MH) algorithm [7] is a Markov chain Monte Carlo method for obtaining a certain number of random samples from a stale probability distribution. The precondition of using MH is that the objective distribution changes relatively slowly and its changing distribution is similar to the stale one. From the observation of the parameters ($\boldsymbol{\mu_0}$ and $\boldsymbol{\Sigma_0}$) in the global space G, they do not change drastically during the clustering. One possible explanation is that in the real world, the number and the content of topics are relatively stable in a certain time window (e.g., people always like to talk about the latest news on social media platforms). We can exploit this observation and employ MH step in our sampling process (i.e., obtaining a few samples with the stale distribution of the global space G).

Therefore, combining Cholesky decomposition with the MH step, the sampling complexity of calculation can be brought down from $O(IDn^2)$ to $O(I\frac{D}{H}n^2)$, where H represents the number of MH steps used in the clustering and $\frac{D}{H} \ll D$. The experiment results (i.e., sampling naively, with Cholesky decomposition (CH), with CH+MH step) of the running time will be demonstrated in the following experiment section.

5. Experiment

In this section, we will evaluate the performance of the proposed NPMM by comparing with the state-of-the-art models. Our experiments show that NPMM can discover a more appropriate number of topics and outperform those competitors in the clustering results.

5.1. Datasets

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

Google News[2]. This dataset is one of the labeled collections which has been used in [45,46] to evaluate the clustering performance. It contains 11,108 news articles (including titles and snippets) grouping into 152 topics. In our experiments, we follow the usage of [44] and take the titles of the news as the short text documents of which the average length is 6.13.

TweetSet. This dataset contains tweets which are labeled in the 2011–2015 microblog tracks at Text Retrieval Conference.[3] The NIST assessors have evaluated all the submitted tweets and retained the quality ones, which involve 269 topics and 30,289 tweets. The average length of the documents in TweetSet is 7.72.

Synthetic datasets. Google News-T and TweetSet-T are two variants of the above datasets to simulate a situation where topics only appear in a certain time period and then disappear. We follow the process in [44] (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 lowercase; (2) Remove non-latin characters and stop words; (3) Remove words not in the space of word embeddings; (4) Conduct word stemming. After the preprocessing, the statistics of these datasets are given in Table 2.

5.2. Evaluation metrics

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

[2] https://news.google.com/news/.
[3] https://trec.nist.gov/data/microblog.html.

Table 3
NMI results of the experimental datasets.

	Google News-T	Google News	TweetSet-T	TweetSet
NPMM	**0.879 ± 0.002**	**0.885 ± 0.002**	**0.854 ± 0.003**	**0.862 ± 0.003**
MStream	0.861 ± 0.003	0.855 ± 0.004	0.851 ± 0.004	0.849 ± 0.004
DTM	0.808 ± 0.003	0.796 ± 0.003	0.803 ± 0.002	0.801 ± 0.002
Sumblr	0.723 ± 0.002	0.575 ± 0.006	0.699 ± 0.002	0.691 ± 0.002

Normalized Mutual Information (NMI). It has been used in [36,44,46] to evaluate the clustering results with ground truth, which is formally defined as follows:

$$NMI = \frac{\sum_{h,l} n_{h,l} \log\left(\frac{n * n_{h,l}}{n_h * n_l}\right)}{\sqrt{\left(\sum_h n_h \log\frac{n_h}{n}\right)\left(\sum_l n_l \log\frac{n_l}{n}\right)}}, \tag{14}$$

where n_h is the number of documents in topic h assigned by models, n_l is the number of documents in topic l assigned by the ground truth, $n_{h,l}$ is the number of documents in topic h as well as in topic l, n is the total number of documents. When the clustering results perfectly match the ground truth, the NMI value will be one. In the worst case, the NMI value will be zero.

5.3. Methods for comparison

We compare NPMM with the following state-of-the-art models in the document clustering field.

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

MStream. Model-based short text stream clustering algorithm [44] is based on the Dirichlet process multinomial mixture (DPMM) model [46]. MStream can work well on both of one-pass clustering process and update clustering process.

MStreamOne. It is the version of MStream with only the one-pass clustering process. The speed of MStreamOne is faster than MStream for the online data processing. According to the claims of the authors [44], MStreamOne can achieve the state-of-the-art performance with only one-pass of the stream.

Sumblr. It is proposed in [32], which can handle the online tweet stream clustering with one-pass of the stream.

NPMM. The nonparametric model is proposed in this work for the online clustering. With the auxiliary word embeddings, NPMM can automatically discover a more appropriate number of topics and outperform the state-of-the-art models in clustering results.

NPMMOne. It is a variant of NPMM with only the one-pass clustering process. We use it to test whether the proposed model can work well on both the one-pass clustering process and the update clustering process.

5.4. Parameter setting

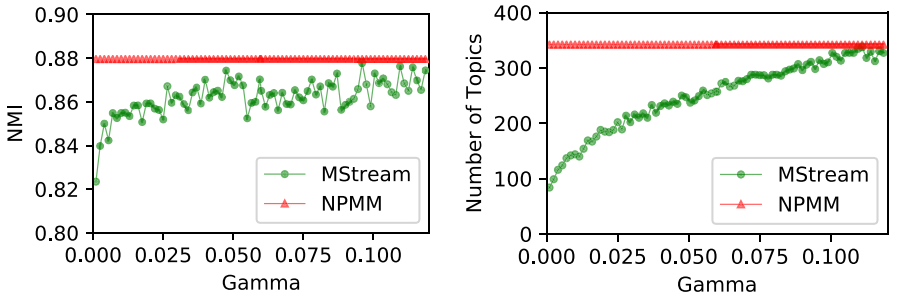
For MStream and MStreamOne, we follow the setting in [44] and set $\gamma = 0.03$ (the hyper-parameter in DP), $\beta = 0.03$. For DTM and Sumblr, we set $\alpha = 0.1$ and $\beta = 0.02$, respectively. For the predefined number of topics in DTM and Sumblr, we set it as 170 and 300 for Google News and TweetSet accordingly. For the spike and slab priors in the proposed NPMM and NPMMOne, we set $\delta = 30, \epsilon = 0.03$ and $\delta = 0.03, \epsilon = 3.0 \times 10^{-7}$, respectively. C in Eq. (7) is set to 1 for NPMM and NPMMOne. In general, for the parameter settings of all the above models except MStream and MStreamOne (where we use their preferred settings), we employ grid search method to find out the parameters with the average best performance for these models. Furthermore, we uniformly set the iteration number $I = 10$ for DTM, MStream, and NPMM, and run 10 independent trials for all of them.

5.5. Comparison with existing models

As introduced before, NMI metric can be used to evaluate the clustering results of the algorithms with ground truth. In this part, we try to compare the performance of NPMM with those of MStream, DTM, and Sumblr in terms of NMI on the four labeled datasets. The mean and the standard deviation of NMI of these models are shown in Table 3. The parameter settings of the experiments are referred in Section 5.4. From Table 3, we can see that the proposed NPMM can achieve the highest performance on the four datasets.

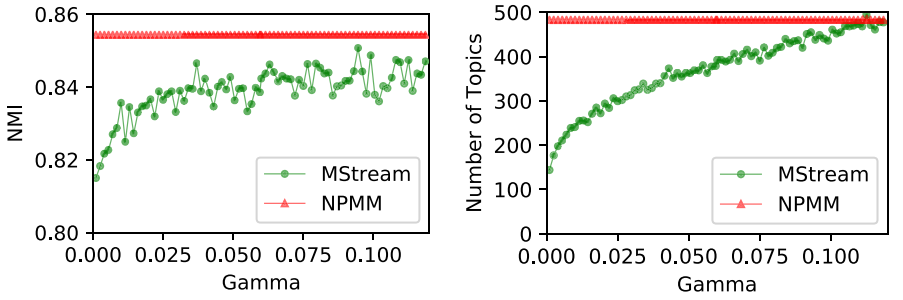
5.5.1. Comparison with the DP-based model

Note that MStream can achieve a competitive performance with an appropriate parameter setting. In order to get better understanding of the contribution of NPMM in terms of the nonparametric topic discovery, we compare NPMM with MStream and DTM to investigate the influence of the topic number found to the clustering performance. Figs. 2 and 3 show



(a) NMI performance (b) Number of topics

Fig. 2. Influence of the hyper-parameter γ to NMI performance and the number of topics found by NPMM and MStream on Google News-T.



(a) NMI performance (b) Number of topics

Fig. 3. Influence of the hyper-parameter γ to NMI performance and the number of topics found by NPMM and MStream on TweetSet-T.

the influence of the hyper-parameter γ in DP to the NMI performance and the number of topics found by NPMM and MStream on Google News-T and TweetSet-T, respectively.

As mentioned in Section 2, the performances of the DP-based models (e.g., MStream) are easily affected by the hyper-parameter of γ which represents the probability of a new topic generation. From Figs. 2 and 3, we can see that the NMI performance and the number of topics found by MStream increase with γ becoming large. The reason for the increment of the topic number is that the probability of the document choosing a potential new topic grows with γ. Contrastively, the proposed NPMM can automatically infer the number of topics without γ setting and achieve higher NMI performance compared with MStream. Note that we get similar results in Google News and TweetSet.

5.5.2. One-pass clustering process capability

In this part, we try to investigate the capability of the one-pass clustering process for the proposed model.

Fig 4 shows the NMI performance of different models with the one-pass scheme in each batch. We discard DTM here as it only has the update clustering process. NPMMOne and MStreamOne are the versions of NPMM and MStream with only the one-pass clustering process. We only report the case that each batch contains 500 documents, and we have similar results on the cases that each batch contains 700 to 1000 documents. From Fig 4, we can see that NPMMOne can outperform MStream and Sumblr in terms of NMI performance. Note that we have tuned the DP parameter γ for MStream and show the best cases. In contrast, the proposed NPMMOne is a nonparametric model that can get rid of the burden of the parameter tunning and achieve a better NMI performance.

5.6. Running time of NPMM

In this part, we try to investigate the speed of NPMM, NPMM with Cholesky decomposition (CH) and NPMM with CH+MH step, which have been introduced in Section 4.1 and Section 4.2, respectively. NPMM is implemented in Python and run on a Windows server with Intel Core i7-6700 3.40 GHz CPU and 16 GB memory. We set the number of iterations to 10 for all sampling methods.

Fig. 5 shows the NMI performance with the running time after applying different sampling methods on Google News-T. From Fig. 5, we can see that the time per iteration of CH+MH100 is 4.52 times less than CH and 4.78 times less than the naive method on average. Moreover, the CH+MH100 method can achieve the similar NMI performance in less time compared with the naive and CH methods. Other observations are that the running time of CH+MH500 is about 1.47 times more than the CH+MH100 and the NMI performance of CH+MH500 is about 1.03 times less than CH+MH100. This is because more

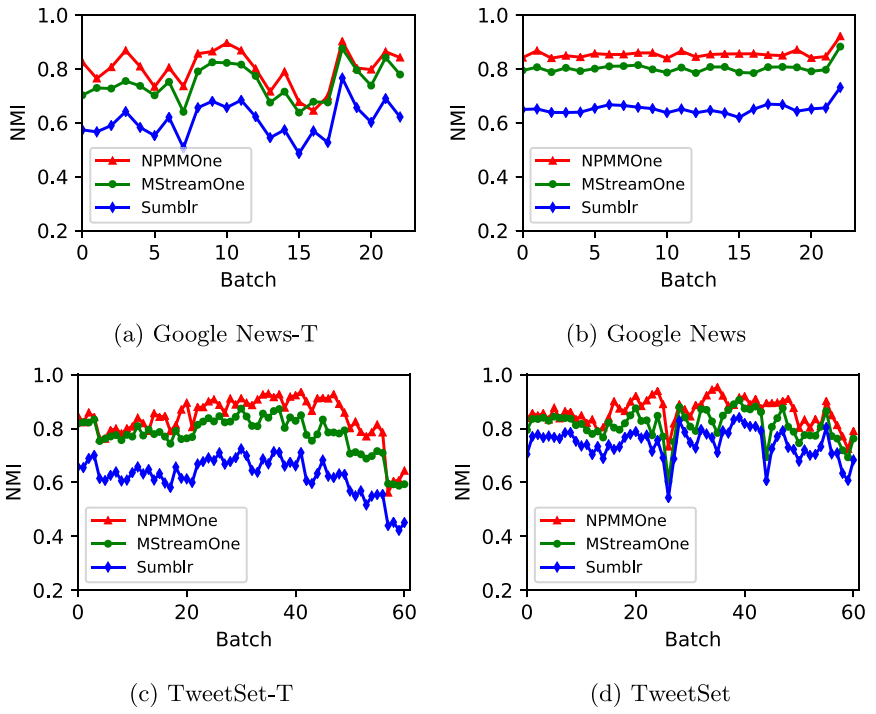


Fig. 4. NMI results of models applying the one-pass clustering process on each batch over the four datasets. We assign 500 documents to each batch.

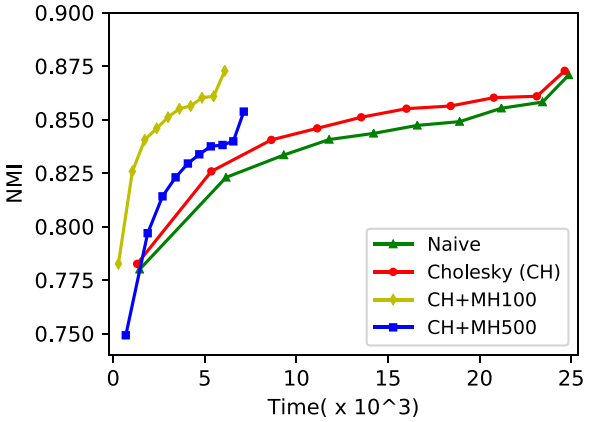


Fig. 5. Plot comparing the NMI performance with time (in second) achieved after applying each sampling method on Google News-T. The shapes on each curve denote end of each iteration and the number of iterations is set to 10.

samples obtained from the stale distribution (details in Section 4.2) may induce coarse generation of topics, which will incur more time to reach the same NMI performance. We found that NPMM with CH+MH100 step is more appropriate than the other sampling methods. In addition, since we have the similar results on Google News, TweetSet-T and TweetSet, we omit their performances for easy presentation.

5.7. Statistical tests

To test whether the difference of clustering performance between NPMM and the state-of-the-art algorithms is statistically significant, we adopt *t*-testing [18] and the *p*-value can be found from Student's t-distribution. If the *p*-value is below the threshold chosen for statistical significance (usually 0.10, 0.05, or 0.01 level), then the difference can be regarded as statistical significance. Table 4 shows the t-testing results of algorithms on different datasets in terms of NMI metrics, where we can find that the *p*-value is generally very small even close to zero, indicating that the superiority of NPMM over other algorithms is statically significant.

Table 4
Significance tests of algorithms on clustering performance.

	p-value of NMI			
	Google News-T	Google News	TweetSet-T	TweetSet
MStream	5.461e-12	3.464e-14	0.074	1.656e-7
DTM	<1e-19	<1e-19	<1e-19	<1e-19
Sumblr	<1e-19	<1e-19	<1e-19	<1e-19

Table 5
Examples of discovered topics with top-10 representative words on Google News. Words italicized and marked in red are not so relevant to the topics.

lakers			nokia			hiv		
NPMM	MStream	DTM	NPMM	MStream	DTM	NPMM	MStream	DTM
kobe	kobe	lakers	nokia	nokia	nokia	hiv	hiv	hiv
bryant	bryant	wizard	lumia	lumia	*corporation*	aid	greek	greek
lakers	lakers	*xbox*	phone	window	*adr*	greek	greece	benefit
extension	extension	wall	window	phone	*nyse*	greece	*flu*	claim
contract	contract	*user*	tablet	tablet	nok	claim	death	infecting
year	*nokia*	*microsoft*	uk	uk	*facebook*	testing	claim	report
deal	*year*	*live*	launch	launch	*nasdaq*	*day*	report	*euro*
sign	*lumia*	john	ipad	camera	*fb*	benefit	*bmw*	*error*
gasol	*ipad*	*video*	price	*december*	*notable*	report	benefit	*ecb*
nba	deal	washington	*december*	*inch*	lumia	adolescent	toll	injecting
typhoon			mcdonald			fda		
NPMM	MStream	DTM	**NPMM**	MStream	DTM	**NPMM**	MStream	DTM
typhoon	typhoon	typhoon	burger	burger	burger	fda	*pope*	*moto*
philippine	philippine	philippine	king	king	king	test	*francis*	*motorola*
relief	relief	relief	france	france	france	dna	fda	fda
aid	haiyan	haiyan	mcdonald	mcdonald	mcdonald	genetic	test	test
haiyan	aid	aid	worldwide	*peter*	*african*	testing	dna	dna
effort	effort	*flu*	expand	*nyse*	*central*	selling	genetic	selling
victim	victim	*death*	*take*	expand	*republic*	kit	testing	sale
help	help	toll	joint	*take*	expand	halt	*call*	genetic
military	*chemical*	*time*	venture	*nokia*	venture	*google*	*google*	*early*
farmer	*key*	*swine*	*form*	worldwide	*troop*	*tell*	kit	*order*

5.8. Qualitative assessment of topic discovery

To evaluate the performance of different models for topic discovery, Table 5 shows six topics (including “lakers”, “nokia”, “hiv”, “typhoon”, “mcdonald” and “fda”) extracted from Google News dataset. Since Sumblr performs considerably inferior to all other methods, we omit its results for easy presentation. From Table 5, we can see that the top-10 representative words discovered by the proposed NPMM can perfectly represent these topics. Irrelevant terms are italicized in the table. Besides, the words extracted by MStearm are less consistent, and DTM achieves the worst performance. In general, the topic discovery performance of these models coincides with the NMI performance as shown in Table 3, which also demonstrates the effectiveness of our NPMM for online topic discovery.

6. Conclusion and future work

In this paper, we have proposed a nonparametric topic model (NPMM) with auxiliary word embeddings for online topic discovery. NPMM can discover a new topic by computing the probabilities of a document belonging to the existing topics and a new one. NPMM can achieve the state-of-the-art performance of online clustering with one-pass process, and can have even better performance with multiple iterations. Moreover, after obtaining the representative terms from each topic, NPMM can exploit the spike and slab priors function to amplify the contrast of word generation probabilities between the relevant words and the irrelevant ones, which alleviates the sparsity problem of the topic-word distribution in the short text clustering. In addition, in order to speed up the sampling process, we have proposed two improved sampling methods (i.e., NPMM with CH and NPMM with CH+MH step) to compare with the naive sampling one. Our extensive experimental study has shown that NPMM with CH+MH100 step can achieve similar performance in less time compared with the naive sampling method on real-life datasets.

In the future work, we intend to exploit NPMM to improve the performance of other text mining applications, such as event detection [14,15], search result diversification [22], and text classification [16,17,20].

Conflict of interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgments

This work was supported in part by the Fund of Science and Technology Development of Macau Government under FDCT/007/2016/AFJ and FDCT/0045/2019/A1, in part by the University of Macau under MYRG2017-00212-FST and MYRG2018-00129-FST, and in part by Guangzhou Science and Technology Innovation and Development under EF005/FST-GZG/2019/GSTIC.

References

[1] C.C. Aggarwal, A survey of stream clustering algorithms, in: Data Clustering, Chapman and Hall/CRC, 2018, pp. 231–258.
[2] C.C. Aggarwal, S.Y. Philip, On clustering massive text and categorical data streams, Knowl. Inf. Syst. 24 (2) (2010) 171–196.
[3] A. Ahmed, E. Xing, Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering, in: Proceedings of the SIAM International Conference on Data Mining, SIAM, 2008, pp. 219–230.
[4] H. Amoualian, M. Clausel, E. Gaussier, M.-R. 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, ACM, 2016, pp. 695–704.
[5] D.M. Blei, J.D. Lafferty, Dynamic topic models, in: Proceedings of the 23rd International Conference on Machine Learning, ACM, 2006, pp. 113–120.
[6] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (Jan) (2003) 993–1022.
[7] S. Chib, E. Greenberg, Understanding the metropolis-hastings algorithm, Am. Stat. 49 (4) (1995) 327–335.
[8] R. Das, M. Zaheer, C. Dyer, Gaussian LDA for topic models with word embeddings, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 1, 2015, pp. 795–804.
[9] A. Doucet, N. De Freitas, N. Gordon, An introduction to sequential monte carlo methods, in: Sequential Monte Carlo Methods in Practice, Springer, 2001, pp. 3–14.
[10] N. Du, M. Farajtabar, A. Ahmed, A.J. Smola, L. Song, 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, 2015, pp. 219–228.
[11] G. Durrett, D. Klein, Neural CRF parsing, ACL 1 (2015) 302–312.
[12] T.S. Ferguson, A Bayesian analysis of some nonparametric problems, Ann. Stat. (1973) 209–230.
[13] T.L. Griffiths, M. Steyvers, Finding scientific topics, Pro. Natl. Acad. Sci. 101 (suppl 1) (2004) 5228–5235.
[14] J. Guo, Z. Gong, 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, 2016, pp. 499–508.
[15] J. Guo, Z. Gong, A density-based nonparametric model for online event discovery from the social media data, in: Proceedings of the IJCAI, 2017, pp. 1732–1738.
[16] L. Jiang, C. Li, S. Wang, L. Zhang, Deep feature weighting for naive Bayes and its application to text classification, Eng. Appl. Artif. Intell. 52 (2016) 26–39.
[17] L. Jiang, S. Wang, C. Li, L. Zhang, Structure extended multinomial naive Bayes, Inf. Sci. 329 (2016) 346–356.
[18] L. Jiang, H. Zhang, Z. Cai, A novel Bayes model: hidden naive Bayes, IEEE Trans. Knowl. Data Eng. 21 (10) (2008) 1361–1371.
[19] A. Kalogeratos, P. Zagorisios, A. Likas, Improving text stream clustering using term burstiness and co-burstiness, in: Proceedings of the 9th Hellenic Conference on Artificial Intelligence, ACM, 2016, p. 16.
[20] S.-B. Kim, H.-C. Rim, H.-S. Lim, A new method of parameter estimation for multinomial naive Bayes text classifiers, in: Proceedings of the 25th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2002, pp. 391–392.
[21] C. Li, H. Wang, Z. Zhang, A. Sun, Z. Ma, 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, 2016, pp. 165–174.
[22] S. Liang, E. Yilmaz, E. Kanoulas, Dynamic clustering of streaming short documents, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 995–1004.
[23] T. Lin, W. Tian, Q. Mei, H. Cheng, The dual-sparse topic model: mining focused topics and focused terms in short text, in: Proceedings of the 23rd International Conference on World Wide Web, ACM, 2014, pp. 539–550.
[24] A.L. Maas, R.E. Daly, P.T. Pham, D. Huang, A.Y. Ng, C. Potts, Learning word vectors for sentiment analysis, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 142–150.
[25] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, ICLR (Workshop), 2013.
[26] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Proceedings of the Advances in Neural Information Processing Systems, 2013, pp. 3111–3119.
[27] T.J. Mitchell, J.J. Beauchamp, Bayesian variable selection in linear regression, J. Am. Stat. Assoc. 83 (404) (1988) 1023–1032.
[28] A. Mukherjee, B. Liu, Aspect extraction through semi-supervised modeling, in: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, 2012, pp. 339–348.
[29] K.P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.
[30] H.-L. Nguyen, Y.-K. Woon, W.-K. Ng, A survey on data stream clustering and classification, Knowl. Inf. Syst. 45 (3) (2015) 535–569.
[31] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1532–1543.
[32] L. Shou, Z. Wang, K. Chen, G. Chen, Sumblr: continuous summarization of evolving tweet streams, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2013, pp. 533–542.
[33] S.K. Sienčnik, Adapting word2vec to named entity recognition, in: Proceedings of the 20th Nordic Conference of Computational Linguistics, Linköping University Electronic Press, 2015, pp. 239–243. may 11–13, 2015
[34] J.A. Silva, E.R. Faria, R.C. Barros, E.R. Hruschka, A.C. De Carvalho, J. Gama, Data stream clustering: a survey, ACM Comput. Surv. (CSUR) 46 (1) (2013) 13.
[35] G.W. Stewart, Matrix Algorithms: Basic Decompositions, SIAM, Society for Industrial and Applied Mathematics, 1998.
[36] A. Strehl, J. Ghosh, Cluster ensembles—a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (Dec) (2002) 583–617.
[37] Y.W. Teh, Dirichlet process, in: Encyclopedia of Machine Learning, Springer, 2011, pp. 280–287.
[38] I. Titov, R. McDonald, Modeling online reviews with multi-grain topic models, in: Proceedings of the 17th International Conference on World Wide Web, ACM, 2008, pp. 111–120.
[39] C. Wang, D.M. Blei, Decoupling sparsity and smoothness in the discrete hierarchical dirichlet process, in: Proceedings of the Advances in Neural Information Processing Systems, 2009, pp. 1982–1989.

[40] S. Wang, Z. Chen, G. Fei, B. Liu, S. Emery, Targeted topic modeling for focused analysis, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1235–1244.
[41] Y. Wang, E. Agichtein, M. Benzi, Tm-lda: efficient online modeling of latent topic transitions in social media, in: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2012, pp. 123–131.
[42] R. Warren, R.F. Smith, A.K. Cybenko, Use of Mahalanobis distance for detecting outliers and outlier clusters in markedly non-normal data: a vehicular traffic example, Technical Report, SRA International INC Dayton OH, 2011.
[43] H. Xu, B. Liu, L. Shu, P.S. Yu, Lifelong domain word embedding via meta-learning, IJCAI, 2018, pp. 4510–4516.
[44] J. Yin, D. Chao, Z. Liu, W. Zhang, X. Yu, J. Wang, Model-based clustering of short text streams, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, ACM, 2018, pp. 2634–2642.
[45] J. Yin, J. Wang, 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, 2014, pp. 233–242.
[46] J. Yin, J. Wang, A model-based approach for text clustering with outlier detection, in: Proceedings of the IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 625–636.
[47] S. Yoo, H. Huang, S.P. Kasiviswanathan, Streaming spectral clustering, in: Proceedings of the IEEE 32nd International Conference on Data Engineering (ICDE), IEEE, 2016, pp. 637–648.
[48] S. Zhong, Efficient streaming text clustering, Neural Netw. 18 (5–6) (2005) 790–798.