Machine Learning (2021) 110:1029–1066
https://doi.org/10.1007/s10994-021-05962-3

Topic extraction from extremely short texts with variational manifold regularization

Ximing Li[1,2] · Yang Wang[3,4] · Jihong Ouyang[1,2] · Meng Wang[3,4]

Received: 11 March 2020 / Revised: 7 January 2021 / Accepted: 17 February 2021 / Published online: 16 April 2021

Abstract

With the emerging of massive short texts, e.g., social media posts and question titles from Q&A systems, discovering valuable information from them is increasingly significant for many real-world applications of content analysis. The family of topic modeling can effectively explore the hidden structures of documents through the assumptions of latent topics. However, due to the sparseness of short texts, the existing topic models, e.g., latent Dirichlet allocation, lose effectiveness on them. To this end, an effective solution, namely Dirichlet multinomial mixture (DMM), supposing that each short text is only associated with a single topic, indirectly enriches document-level word co-occurrences. However, DMM is sensitive to noisy words, where it often learns inaccurate topic representations at the document level. To address this problem, we extend DMM to a novel *Lap*lacian *D*irichlet *M*ultinomial *M*ixture (*LapDMM*) topic model for short texts. The basic idea of LapDMM is to preserve local neighborhood structures of short texts, enabling to spread topical signals among neighboring documents, so as to modify the inaccurate topic representations. This is achieved by incorporating the variational manifold regularization into the variational objective of DMM, constraining the close short texts with similar variational topic representations. To find nearest neighbors of short texts, before model inference, we construct an offline document graph, where the distances of short texts can be computed by the word mover's distance. We further develop an online version of LapDMM, namely *O*nline *LapDMM*, to achieve inference speedup on massive short texts. Carrying this implications, we exploit the spirit of stochastic optimization with mini-batches and an up-to-date document graph that can efficiently find approximate nearest neighbors instead. To evaluate our models, we compare against the state-of-the-art short text topic models on several traditional tasks, i.e., topic quality, document clustering and classification. The empirical results demonstrate that our models achieve very significant performance gains over the baseline models.

Keywords Topic modeling · Short text · Dirichlet multinomial mixture · Variational manifold regularization · Online inference

This article is an extended version of our previous paper of [35] presented at the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI) (Hawaii, USA, January 27–February 1, 2019). The code can be accessed via: https://github.com/li-ximing/LapDMM.

Editor: Byron Wallace.

Extended author information available on the last page of the article

1 Introduction

Nowadays, the emerging social media platforms and Q&A systems produce millions of text documents available everyday on the Internet, such as Twitter posts, question titles and text advertisements *etc*. Due to the big volume of those text data, discovering valuable information and knowledge from them is a hot research topic in the machine learning and data mining communities, and even many real-world applications of content analysis (Liang et al. 2017a, b; Wang et al. 2015b, 2017). However, such texts, formally referred to as **short texts**, are often extremely *short*, *noisy* and *ambiguous*.

Example 1 As shown in Fig. 1, after a removal of the standard stopwords, some question title samples of *Tweets* contain only less than 10 word tokens, and simultaneously mix with many noisy words, e.g., "estimmate" and "rebuiilt"; worst of all, the average text length of *Tweets* is only about 4.9, and many other collections of short texts, e.g., *StackOverFlow* and *BaiduQA*, also averagely contain only less than 5 word tokens (i.e., more details of datasets are presented in Table 3).

Therefore, each short text always involves quite limited valuable information, resulting in the so-called **sparsity problem**. Generally, the topic modeling methods (Blei 2012; Li et al. 2018a; Zhao et al. 2018), such as probabilistic latent semantic indexing (PLSI) (Hofmann

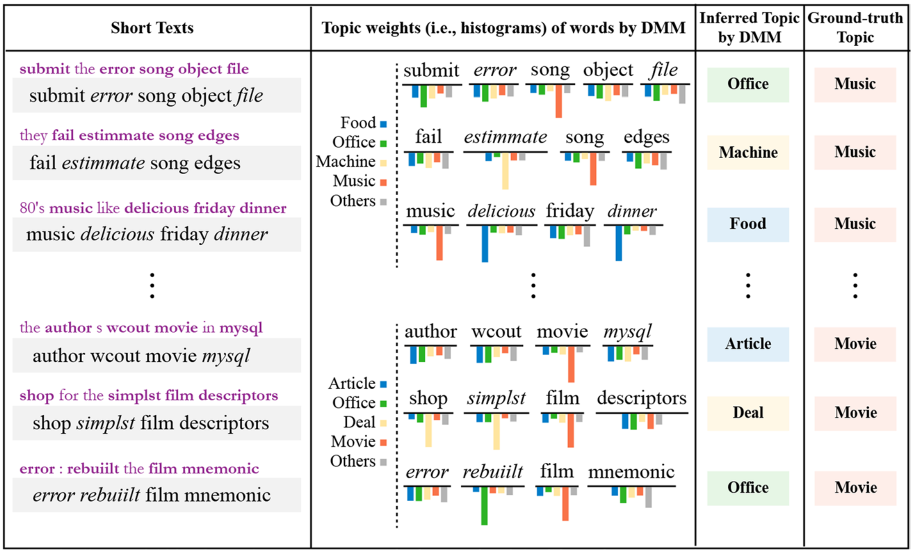
Short Texts
Topic weights (i.e., histograms) of words by DMM
Inferred Topic by DMM
Ground-truth Topic
submit the error song object file
submit error song object file
they fail estimmate song edges
fail estimmate song edges
80's music like delicious friday dinner
music delicious friday dinner
Food
Office
Machine
Music
Others
submit
error
song
object
file
fail
estimmate
song
edges
music
delicious
friday
dinner
Office
Music
Machine
Music
Food
Music
the author s wcout movie in mysql
author wcout movie mysql
shop for the simplst film descriptors
shop simplst film descriptors
error : rebuiilt the film mnemonic
error rebuiilt film mnemonic
Article
Office
Deal
Movie
Others
author
wcout
movie
mysql
shop
simplst
film
descriptors
error
rebuiilt
film
mnemonic
Article
Movie
Deal
Movie
Office
Movie

Fig. 1 We illustrate several short text examples of *Tweets*. After removing the standard stopwords, we show the left words in shadow. Such texts are extremely short, and contain many noisy words (denoted by Italics), including misspelled words, e.g., "estimmate" and "rebuiilt", and domain-specific stopwords, e.g., "error" and "file". When using DMM to learn topics from these short texts, it is sensitive to the noisy words, resulting in the so called **sensitivity problem**: The inferred topics (manually denoted by different colors) of documents are mainly depending on the product of topic weights (i.e., histograms) of observable words. Therefore, for short texts with very few words, it is highly affected by every word. However, the misspelled words refer to meaningless topic weights and the domain-specific stopwords often approach uniform weights of topics. With such noisy words, DMM tends to produce less accurate topic estimations of documents. Best viewed in color

1999) and latent Dirichlet allocation (LDA) (Blei et al. 2003), are the mainstream methodologies for discovering and analyzing collections of massive text documents. Basically, they are built on the assumption that there are latent topics beyond the observable word tokens, where each document is a mixture of topics, and each topic is represented by the distributions of words. With topic inference by statistical methods, e.g., variational inference (Blei et al. 2003; Teh et al. 2006) and Gibbs sampling (Griffiths and Steyvers 2004), one can rely on the inferred topics to leverage the latent structure of text documents, and then apply them to the tasks of text discovery and analysis. The existing topic models can effectively learn the topics of normal long text documents (Blei 2012), however, their performance is deteriorated by a large margin when handling short texts due to the aforementioned sparsity problem. The underlying philosophy is that, with statistical methods, the topic inference is mainly depending on the document-level word co-occurrence information (Wang and McCallum 2006) that the short texts lack. This makes the inferred topics much less accurate, hence raising up a significant challenge to topic modeling over short texts.

Many topic modeling efforts have been made to handle the sparsity problem of short texts. Borrowing the taxonomic hierarchy from multi-label learning (Zhang and Zhou 2014), we can organize the existing works on topic modeling of short texts into two categories, i.e., *problem transformation* (**PT**) method and *algorithm adaptation* (**AA**) method.

The **PT** methods (Hong and Davison 2010; Weng et al. 2010; Mehrotra et al. 2013; Quan et al. 2015; Zuo et al. 2016a; Li et al. 2018c) tackle the problem by aggregating short texts into long pseudo-documents and then applying a well-established topic model, e.g., LDA. Specifically, the short texts can be aggregated using side information, e.g., user ID (Mehrotra et al. 2013), or adaptive paradigms (Quan et al. 2015; Zuo et al. 2016a; Li et al. 2018c). However, they suffer from two common drawbacks: (1) Any long pseudo-document may consist of many irrelevant short texts, making the topic inference less effective; (2) their adaptive aggregation steps are computationally expensive, especially for collections of massive short texts.

Unlike the PT methods, the **AA** methods (Nigam et al. 2000; Yan et al. 2013; Cheng et al. 2014; Sridhar 2015; Zuo et al. 2016b; Xin et al. 2011; Wang et al. 2015a, c, 2016a, b, 2018; Yin and Wang 2014; Li et al. 2016a, 2017, 2018e, 2019a; Lu et al. 2017) have been paid more attention due to their superior performance. They directly modify traditional topic models by enriching word co-occurrences, so as to remedy the sparsity problem. First, a straightforward methodology is proposed to model the global word co-occurrences at the corpus level (Yan et al. 2013; Cheng et al. 2014; Sridhar 2015; Zuo et al. 2016b). For example, the biterm topic model (BTM) (Yan et al. 2013; Cheng et al. 2014) learns topics by modeling word co-occurrence pairs over the entire corpus; the word network topic model (WNTM) (Zuo et al. 2016b) refers to each word type as a pseudo-document following a global word co-occurrence network. These models can alleviate the sparsity problem to some extent. However, they may create many meaningless word co-occurrences without any word pair filtering process, and more importantly they lose document-specific topic structures. Second, another methodology is to indirectly enrich document-level word co-occurrences by supposing that each short text covers a small subset of topics. A representative method is the Dirichlet multinomial mixture (DMM) (Nigam et al. 2000; Xin et al. 2011; Yin and Wang 2014) following the assumption that each short text is only associated with a single topic. Recently, two extensions of DMM (Li et al. 2016a, 2017), namely generalized Pólya urn DMM (GPU-DMM) and generalized Pólya urn Poisson-based DMM (GPU-PDMM), incorporate auxiliary word embeddings (Mikolov et al. 2013) to enhance the topic inference of Gibbs sampling, enabling to attract similar words in the same topics. Therefore, they can empirically generate more coherent topic representations.

Orthogonal to BTM and WNTM, the family of DMM can maintain document-specific topic structures, and it has empirically shown very superior performance in many task of short texts (Li et al. 2016a, 2017). However, DMM is sensitive to noisy words, therefore the topic representations of documents can be easily miscalculated. We refer to this as the **sensitivity problem**. That is, if the majority of words in short texts are without any topic-inclination (e.g., domain-specific stopwords) or even errors (e.g., misspelled words), the inferred topics must be dominated by those noisy words, so as to be probably miscalculated. For ease of understanding, we illustrate some examples in Fig. 1. Besides, the recent models of GPU-DMM and GPU-PDMM also suffer from the sensitivity problem, since they focus on leveraging similar words in the same topics, rather than handling noisy ones.

Our contributions: Our goal is to alleviate the sensitivity problem of DMM. To this end, we develop a novel **Lap**lacian **D**irichlt **M**ultinomial **M**ixture (**LapDMM**) topic model for short texts. The basic idea is to extend DMM by preserving local neighborhood structure of short texts using manifold regularization of Laplacian Eigenmap, which has been successfully used for topic models (Mei et al. 2008; Cai et al. 2008, 2009; Huh and Fienberg 2010, 2012; Du et al. 2015; Hu et al. 2017; Li 2018d). The manifold regularization implies that the learned manifolds should be smooth, which here constrains nearby document pairs have similar latent topic representations. This can indirectly spread topical signals among neighboring documents, enabling to modify the miscalculated topic representations, so as to remedy the sensitivity problem of DMM.

We kindly remind that the manifold regularization cannot be directly applied to DMM, since it supposes that each document is only associated with a single topic. To address this, we train LapDMM following the spirit of collapsed variational inference (Teh et al. 2006), a more accurate inference method for topic models (Chi et al. 2018) than Gibbs sampling used in GPU-DMM and GPU-PDMM (Li et al. 2016a, 2017). We then incorporate a manifold regularizer with respect to variational distributions, referring to as **variational manifold regularization**, into the original variational objective of DMM, such that the close short texts tend to have similar variational topic representations. LapDMM is optimized by maximizing the regularized variational objective. Besides, the variational manifold regularization is built on the offline document graph, indicating the nearest neighbors of all short texts, before training LapDMM. To better capture distances between short texts, we employ the word mover's distance (WMD) (Kusner et al. 2015) with word embeddings (Mikolov et al. 2013), which describes document distances at the semantic level. We employ a regularized version of WMD with an entropic regularizer (Cuturi 2013) for efficient computations.

Furthermore, we propose two ideas to achieve inference speedup when facing massive short texts. First, inspired by Hoffman et al. (2010, 2013), Foulds et al. (2013), we exploit the spirit of stochastic optimization with mini-batches. That is, at each iteration, we only exploit a small mini-batch of short texts to update variational parameters of interest, instead of the whole corpus, so as to accelerate the inference procedure. Second, since the time cost of the document graph construction is quadratic with the number of short texts, the step becomes much expensive given massive instances. Motivated by this, we aim to efficiently find approximate nearest neighbors, and therefore construct an up-to-date document graph with mini-batches of short texts instead. Upon these two ideas, we develop an online version of LapDMM, namely **O**nline **LapDMM** (**OLapDMM**).

Empirically, we evaluate LapDMM and OLapDMM on various datasets of short texts across various tasks, i.e., topic quality, document clustering and classification. Experimental results indicate that our models significantly outperform the state-of-the-art baseline topic models of short texts.

In a nutshell, the major contributions of this paper are listed below:

- We develop a novel topic model for short texts, namely **LapDMM**, which handles the sensitivity problem of DMM by preserving local neighborhood structure of texts using manifold regularization of Laplacian Eigenmap. We train LapDMM following the spirit of collapsed variational inference, and then leverage a variational manifold regularization that refers to an offline document graph, indicating the nearest neighbors of all short texts. Therefore, we achieve a regularized variational objective of LapDMM. and optimize it using generalized expectation maximization.
- We develop an online version of LapDMM, namely **OLapDMM**, for inference speedup with even massive short texts. OLapDMM is built on the spirit of stochastic optimization with mini-batches. Besides, we develop an up-to-date document graph with mini-batches of short texts, enabling to efficiently find approximate nearest neighbors.
- We conduct a number of experiments on several benchmark datasets of short texts. Empirical results demonstrate that our models significantly performs better than the state-of-the-art baselines on topic quality, clustering and classification tasks. Specifically, the performance gain achieves even above 160% in many cases.

We kindly remind that this article is an extension of our previous conference paper of Li et al. (2019b). The extended works include: (1) We propose an online version of LapDMM, namely OLapDMM; (2) we develop an up-to-date document graph for efficiently finding approximate nearest neighbors; (3) we discuss the time complexities of LapDMM and OLapDMM; (4) more experimental results are presented to validate the effectiveness of LapDMM and OLapDMM.

The rest of this article is organized as follows: Some recent related works are introduced in Sect. 2. In Sect. 3, we describe LapDMM and OLapDMM in detail. In Sect. 4, we present and discuss the empirical results. The conclusions are given in Sect. 5.

2 Related work

We review some recent related works on topic models for short texts and topic modeling with manifold regularization.

2.1 Topic models for short texts

Conventional topic models, such as PLSI and LDA, suffer from the sparsity problem of short texts, because they are lacking of word co-occurrences at the document level. To effectively extract topics from short texts, many topic modeling attempts have been recently proposed, and they can be roughly divided into two categories, i.e., *problem transformation* (**PT**) method (Hong and Davison 2010; Weng et al. 2010; Mehrotra et al. 2013; Quan et al. 2015; Zuo et al. 2016a; Li et al. 2018c) and *algorithm adaptation* (**AA**) method (Nigam et al. 2000; Cheng et al. 2014; Sridhar 2015; Zuo et al. 2016b; Xin et al. 2011; Yin and Wang 2014; Li et al. 2016a, 2017, 2018e, 2019a; Lu et al. 2017).

The idea of **PT** methods is to aggregate the short texts into long pseudo-documents and then applying a well-established topic model, e.g., LDA. For example, some models attempt to aggregate Twitter posts using the user information (Hong and Davison 2010), shared words (Weng et al. 2010) and combinations of various side messages (Mehrotra et al. 2013). However, they are highly data-dependent, and cannot be applied to short texts without any side information. For more practical models, some recent works (Quan et al. 2015; Zuo et al. 2016a; Li et al. 2018c) propose to adaptively aggregate short texts. For example, the self-aggregation based topic model (SATM) (Quan et al. 2015) integrates topic modeling with clustering; the latent topic model (LTM) (Li et al. 2018c) supposes that the long pseudo-documents are composed of short texts, and then alternatively draws the topic assignments for short texts and word tokens using Gibbs sampling. Roughly this kind of adaptively aggregated models is equivalent to an EM-like iteration procedure, i.e., clustering short texts (E-step) and LDA optimization (M-step). In some sense, our LapDMM is also aggregating short texts by linking neighboring ones. In contrast to them, LapDMM is much safer since it learns topics with the help of the neighboring document graph, rather than short text clusters, i.e., long pseudo-documents, that may consist of many irrelevant short texts. Besides, these adaptively aggregated models are sensitive to the number of long pseudo-documents, and become computationally expensive for collections of massive short texts, since more long pseudo-documents are often required given more short texts.

Unlike the PT methods, the **AA** methods directly modify traditional topic models by enriching word co-occurrences. The models of BTM (Yan et al. 2013; Cheng et al. 2014) and Gaussian mixture topic model (Sridhar 2015) consider a corpus as a single big document, and they then model all word co-occurrence patterns extracted from documents and word embeddings of observable word tokens, respectively. Another representative WNTM (Zuo et al. 2016b) is built on the word type pseudo-documents, which are constructed by word co-occurrences over the whole corpus. Besides these models, which may mix with many noisy word co-occurrence patterns, DMM directly handles the sparsity problem by assuming that each short text is drawn from a single topic. Given the sparse content of short texts, this assumption is more reasonable, making DMM more effective than traditional topic models (Xin et al. 2011). Recently, two extensions of DMM, i.e., GPU-DMM and GPU-PDMM (Li et al. 2016a, 2017), incorporate a generalized Pólya urn process into the topic inference process, so that similar words measured by word embeddings should be clustered in the same topics. In contrast to GPU-DMM and GPU-PDMM, our LapDMM not only captures the semantic information of word embeddings, but also further preserves the neighborhood structure of short texts by manifold constraints.

Besides, there are some other short text topic models (Yan et al. 2013; Shi et al. 2018; Li et al. 2020), built on non-negative matrix factorization (NMF) (Lee and Seung 1999). For example, the recent semantics-assisted NMF model (Shi et al. 2018) alleviates the sparsity problem by referring to the word contexts of short texts as auxiliary pseudo-texts.

2.2 Topic modeling with manifold regularization

The manifold regularization has been successfully used for topic modeling (Mei et al. 2008; Cai et al. 2008, 2009; Huh and Fienberg 2010, 2012; Du et al. 2015; Hu et al. 2017; Li 2018d). The prior works mainly investigate various ways to implement the manifold constraint between documents. For example, Cai et al. (2008) incorporate manifold structure information, i.e., a manifold regularizer with the Euclidean distance, into the log-likelihood objective of PLSI (Hofmann 1999). The locally-consistent topic model (Cai et al. 2009) relies on a manifold regularizer with Kullback-Leibler divergence by replacing the

Table 1 A summary of important notations

Notation	Description
D	Number of short texts
V	Number of words
K	Number of topics
$\boldsymbol{\phi}$	Topic distributions over words
β	Dirichlet prior of $\boldsymbol{\phi}$
$\boldsymbol{\theta}$	Corpus-level distribution over topics
α	Dirichlet prior of $\boldsymbol{\theta}$
z	Topic indicator of documents
$\boldsymbol{\gamma}$	Variational parameter
W	Edge weight of the document graph
R	Number of nearest neighbors in the document graph
M	Mini-batch size in OLapDMM

Euclidean distance. Besides, the discriminative topic model (Huh and Fienberg 2010) develops a manifold regularizer, which not only pulls neighboring document pairs closer together, but also separates non-neighboring document pairs from each other. Additionally, several recent works develop manifold-based topic models in specific learning scenarios, e.g., semi-supervised learning (Hu et al. 2017) and weakly supervised learning with seed words (Li 2018d). However, those models are mainly designed for modeling normal long texts, therefore they are not applicable to short texts, due to the sparsity problem.

Discussion We discuss several differences between prior topic models with manifold regularization and our LapDMM. First, the prior models mainly extend the traditional topic models, e.g., PLSI and LDA, directly incorporating the manifold constraint on the K-dimensional topic representations of documents. However, the manifold regularization cannot be directly applied to DMM when modeling short texts, because DMM supposes that each document is only associated with a single topic. Instead, in LapDMM we leverage a manifold regularizer with respect to variational distributions under the framework of collapsed variational inference. Besides, LapDMM further utilizes the recent WMD to measure document distances at the semantic level. Second, to our knowledge, these prior manifold-based topic models are all built on pre-computed offline document graph, which is computationally expensive given massive short texts. In contrast, we develop an online version of LapDMM, i.e., OLapDMM, with an up-to-date document graph for finding approximate neighbors, instead of exact ones, enabling to be applicable for larger collections of short texts.

3 Model

Before shedding light on our method, we first give a brief introduction to Dirichlet multinomial mixture (DMM) (Nigam et al. 2000; Yin and Wang 2014), which paves the way to our **Lap**lician **D**irichlet **M**ultinomial **M**ixture (**LapDMM**) topic model for short texts. For clarity, we outline some important notations in Table 1.

3.1 Dirichlet multinomial mixture

DMM is a generative topic model with the assumption that each document covers only a single topic. Actually, this assumption can indirectly enrich word co-occurrences at the document level, making the model more effective for short texts than LDA and its variants.

Formally, DMM consists of **(1)** K topic distributions $\boldsymbol{\phi}$ over the vocabulary of V words, drawn from a Dirichlet prior β and **(2)** a corpus-level distribution $\boldsymbol{\theta}$ over topics, drawn from a Dirichlet prior α. For each document d, DMM first draws a topic indicator z_d from $\boldsymbol{\theta}$, and subsequently draws each word token w_{dn} from the selected topic $\boldsymbol{\phi}_{z_d}$. Its generative process of D short texts can be described as follows:

1. Draw a distribution over topics: $\boldsymbol{\theta} \sim \mathbf{Dir}(\alpha)$
2. For each topic k

    a. Draw a topic distribution over words $\boldsymbol{\phi}_k \sim \mathbf{Dir}(\beta)$

3. For each document d

    a. Draw a topic : $z_d \sim \mathbf{Multinomial}(\boldsymbol{\theta})$
    b. For each of the N_d words w_{dn}

i. Draw a word token: $w_{dn} \sim \mathbf{Multinomial}\left(\boldsymbol{\phi}_{z_d}\right)$

3.2 LapDMM with variational manifold regularization

The basic idea of LapDMM is to extend DMM by preserving local neighborhood structure of short texts, enabling to spread topical signals among neighboring documents, so as to remedy the sensitivity problem. This is achieved by using the manifold regularization methodology. We now describe the manifold regularization in topic modeling, and then the objective of LapDMM.

Manifold regularization In the context of topic modeling, the manifold regularization constrains that the latent topic representations of document pairs should be similar to each other if they are nearest neighbors in the document manifold.

Formally, consider a directed document graph with D vertices, where each vertex corresponds to a document in the corpus. Each component of the edge weight matrix W is defined by:

$$W_{ij} = \begin{cases} 1 & \text{if } d_i \in \Omega(d_j) \text{ or } d_j \in \Omega(d_i) \\ 0 & \text{otherwise} \end{cases}, \tag{1}$$

where $\Omega(d)$ is a document set containing R nearest neighbors of document d. Specifically, let $\boldsymbol{\omega}_d$ denote a latent K-dimensional topic representation of document d. We can define a least square manifold regularization term as follows:

$$\mathcal{R}(\omega) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j=1}^{D} \left(\boldsymbol{\omega}_{ik} - \boldsymbol{\omega}_{jk}\right)^2 W_{ij} \tag{2}$$

Regularized variational objective of LapDMM Note that we can not directly incorporate the manifold regularization term of Eq. 2 into DMM inference. Because in DMM each document is only associated with a single topic, there are no explicit K-dimensional topic representations $\boldsymbol{\omega}$ for documents.

To break this limitation, we resort to the collapsed variational inference optimization (Teh et al. 2006), and propose a manifold regularizer with respect to the variational distribution instead.

Thanks to the conjugate Dirichlet-multinomial design in DMM, the two distributions $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be directly marginalized out. We then define a mean-field variational distribution with respect to the topic assignment $\boldsymbol{z}$ of documents,

$$q(z|\boldsymbol{\gamma}) = \prod_{d=1}^{D} q\left(z_d|\boldsymbol{\gamma}_d\right), \tag{3}$$

where each $q(z_d|\boldsymbol{\gamma}_d)$ is a multinomial distribution with a K-dimensional variational parameter vector $\boldsymbol{\gamma}_d$, i.e., $\sum_{k=1}^{K} \boldsymbol{\gamma}_{dk} = 1$. Given a short text collection S, we train DMM by maximizing the following variational objective with respect to γ:

$$\mathcal{L}(\boldsymbol{\gamma}) = \mathbb{E}_{q(z|\boldsymbol{\gamma})}\left[\log p(S, z|\alpha, \beta) - \log q(z|\boldsymbol{\gamma})\right] \tag{4}$$

Since each document-specific variational distribution $q(z_d|\boldsymbol{\gamma}_d)$ is used as an approximation to the latent topic representation of the current document, we can define a manifold regularizer with respect to $q(z)$, i.e., referring to as **variational manifold regularization**, to achieve manifold constraints. That is, we re-write the manifold regularization of Eq. 2 by replacing $\boldsymbol{\omega}$ with the K-dimensional variational parameter $\boldsymbol{\gamma}$ as follows:

$$\mathcal{R}(\boldsymbol{\gamma}) = \frac{1}{2} \sum_{k=1}^{K} \sum_{i,j=1}^{D} \left(\boldsymbol{\gamma}_{ik} - \boldsymbol{\gamma}_{jk}\right)^2 W_{ij} \tag{5}$$

By combining Eqs. 4 and 5, we reach the final regularized variational objective of LapDMM with respect to $\boldsymbol{\gamma}$:

$$\widehat{\mathcal{L}}(\boldsymbol{\gamma}) = \frac{1}{D}\mathcal{L}(\boldsymbol{\gamma}) - \lambda \mathcal{R}(\boldsymbol{\gamma}), \tag{6}$$

where $\lambda \in [0, 1]$ is a regularization parameter.

Discussion We now discuss the manifold formulation of LapDMM, described by Euclidean distance between variational parameters, i.e., Eq. 5. Prior works suggest some other manifold formulations, e.g., the manifold term with Kullback-Leibler divergence (Cai et al. 2009) and the one considering neighboring and non-neighboring document pairs simultaneously (Huh and Fienberg 2010). Actually, we have examined those popular manifold formulations in our early experiments. We found that all of them performed very similar performance in the scenario of modeling short texts, but the manifold term with Euclidean distance is more tractable to compute than other ones. Therefore, this formulation of Eq. 5 is leveraged in LapDMM.

3.3 Optimization

We use a double-loop optimization procedure (Cai et al. 2008) to maximize the objective of LapDMM $\widehat{\mathcal{L}}(\boldsymbol{\gamma})$ (i.e., Eq. 6). In the outer iteration, we optimize $\boldsymbol{\gamma}$ by maximizing the first term of Eq. 6, i.e., the original variational objective of DMM; in the inner iteration, we use the Newton-Raphson method to update $\boldsymbol{\gamma}$ by minimizing the second term of Eq. 6, i.e., the variational manifold regularization, until the value of $\widehat{L}(\boldsymbol{\gamma})$ decreases. We now describe the optimization details.

Outer iteration Actually, the outer update is a standard step of collapsed variational inference for training DMM. Following Bishop (2006), for each document d, the optimum of $\boldsymbol{\gamma}_d$, holding all other variational distributions fixed, can be computed by (derivation details are shown in the "Appendix"):

$$\begin{aligned}\boldsymbol{\gamma}_{dk} &\propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\log p(S, z^{\neg d}, z_d = k|\alpha, \beta)\right]\right)\\ &\propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\log\left(\widehat{N}_k^{\neg d} + \alpha\right) + \sum_{v\in d}\sum_{n=1}^{N_{dv}}\log\left(N_{kv}^{\neg d} + \beta + n - 1\right)\right.\right.\\ &\quad\left.\left. - \sum_{n=1}^{N_d}\log\left(N_k^{\neg d} + V\beta + n - 1\right)\right]\right),\end{aligned} \tag{7}$$

where N_{dv} is the number of times word v occurring in document d; $\widehat{N}_k$ is the number of documents assigned to topic k; N_{kv} and N_k are the number of word v assigned to topic k and total number of words assigned to topic k, respectively; the superscript "$\neg d$" means the corresponding variables and counts with document d excluded.

We can efficiently compute an approximation of Eq. 7 by using the first-order Taylor expansion (Asuncion et al. 2009; Sato and Nakagawa 2012) at the expectation values of number counts in Eq. 7:

$$\boldsymbol{\gamma}_{dk} \propto \left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\widehat{N}_k^{\neg d}\right] + \alpha\right) \times \frac{\prod_{v\in d}\prod_{n=1}^{N_{dv}}\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[N_{kv}^{\neg d}\right] + \beta + n - 1\right)}{\prod_{n=1}^{N_d}\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[N_k^{\neg d}\right] + V\beta + n - 1\right)}, \tag{8}$$

where for example the expectation of $\widehat{N}_k^{\neg d}$ is $\sum_{i\neq d}^{D}\boldsymbol{\gamma}_{ik}$, and the other two expected number counts are similar.

Inner iteration In the inner iteration, we focus on minimizing the variational manifold regularization $\mathcal{R}(\boldsymbol{\gamma})$. We continue updating $\boldsymbol{\gamma}$ using Newton-Raphson iterations until the value of the overall objective $\widehat{\mathcal{L}}(\boldsymbol{\gamma})$ decreases (Cai et al. 2008). The update equation is as follows:

$$\begin{aligned}\boldsymbol{\gamma}_{dk} &\leftarrow \boldsymbol{\gamma}_{dk} - \rho\frac{\mathcal{R}'(\boldsymbol{\gamma}_{dk})}{\mathcal{R}''(\boldsymbol{\gamma}_{dk})}\\ &\leftarrow (1-\rho)\boldsymbol{\gamma}_{dk} + \rho\frac{\sum_{i=1}^{D}\boldsymbol{\gamma}_{ik}W_{di}}{\sum_{i=1}^{D}W_{di}},\end{aligned} \tag{9}$$

where $\rho \in [0, 1]$ is the learning rate. Note that this update equation guarantees $\sum_{k=1}^{K}\boldsymbol{\gamma}_{dk} = 1$ for any document d.

Remark The learning rate ρ can be roughly considered as a tuning parameter used to balance the two terms in Eq. 9. When $\rho = 0$, LapDMM is downgraded to the standard DMM without manifold constraints.

Estimations of topic distributions Given the optimum of $\boldsymbol{\gamma}$, the point estimates of $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ can be computed by:

$$\boldsymbol{\phi}_{kv} = \frac{\mathbb{E}_{q(z|\boldsymbol{\gamma})}[N_{kv}] + \beta}{\mathbb{E}_{q(z|\boldsymbol{\gamma}))}[N_k] + V\beta}, \tag{10}$$

$$\boldsymbol{\theta}_k = \frac{\mathbb{E}_{q(z|\boldsymbol{\gamma})}[\widehat{N}_k] + \alpha}{D + K\alpha}, \tag{11}$$

where for example the expectation of $\widehat{N}_k$ is $\sum_{d=1}^{D} \boldsymbol{\gamma}_{ik}$, and the other two expected number counts in Eqs. 10 and 11 are similar.

3.3.1 Document graph construction

To achieve manifold constraints, we need to construct an offline document graph before LapDMM inference, i.e., finding R nearest neighbours for all short texts (ref. Eq. 1). In this article, we exploit two ways to measure distances between document pairs detailed below.

Measuring document distances in the original term space: Straightforwardly, we employ the popular cosine distance of documents' term frequency vectors.

Measuring document distances in a latent semantic space with word embeddings: In the sparse short text context, semantically related documents may not contain any same word, so that they are far away in the term space. To alleviate this, we employ the Word Mover's Distance (WMD) (Kusner et al. 2015) to measure document distances at the semantic level. The formulation of WMD of a document pair (d_i, d_j) with an entropic regularization term (Cuturi 2013) is given by:

$$W_c\left(d_i, d_j\right) = \inf_{P \in \Pi(d_i, d_j)} \langle P, C \rangle - \frac{1}{\lambda'} H(P) \tag{12}$$

where d_i denotes the normalized term frequency vector of document i that can be considered as a multinomial distribution; $\Pi(d_i, d_j)$ is the set of the joint distributions of d_i and d_j; $H(\cdot)$ denotes the entropy; λ' is a regularization parameter[1]; and C is the cost matrix measured by the cosine distances of word embedding pairs (i.e., semantic distances between words measured by the corresponding word embeddings). In summary, the WMD actually measures the optimal (i.e., cheapest) transport from one document to any other in a semantic space with word embeddings. We can use the method proposed in Cuturi (2013) to efficiently optimize Eq. 12 and then obtain the WMD values.

3.3.2 Summary of LapDMM optimization

For clarity, the full optimization process of LapDMM can be summarized as follows: **(1)** After initializing the parameters of LapDMM, i.e., R, λ and ρ, we construct the document graph by computing the distances of all short text pairs; **(2)** we iteratively optimize the variational parameter $\boldsymbol{\gamma}$ by performing the outer and inner iterations.

In summary, we outline the optimization of LapDMM in **Algorithm 1**.

[1] Following previous studies, in this work we set λ' to 10.

Algorithm 1 Optimization for LapDMM

```
Set model and training parameters, including the number of nearest neighbors $R$, the regularization parameter $\lambda$ and Newton-Raphson learning rate $\rho$
Construct the offline document graph
Initialize $\boldsymbol{\gamma}$ randomly and then expected number counts
For $t = 1, 2, \ldots,$ MaxIter
    Update $\boldsymbol{\gamma}$ using Eq.8
    $\widehat{\gamma} \leftarrow \gamma$
    While $\left(\widehat{\mathcal{L}}(\gamma) \leq \widehat{\mathcal{L}}(\widehat{\gamma})\right)$ Do
        $\gamma \leftarrow \widehat{\gamma}$
        Update $\widehat{\gamma}$ using Eq.9
    End While
    Update expected number counts with the current $\boldsymbol{\gamma}$
End for
Compute $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ using Eqs.10 and 11
```

3.3.3 Time complexity of LapDMM

In this subsection, we mainly discuss the time complexities of collapsed variational inference for DMM and LapDMM.

For clarity, we again declare that D and K denote the numbers of short texts and topics, respectively. Besides, let $\overline{N}$ denote the average document length of a corpus, T and $\widehat{T}$ denote the numbers of outer iteration and inner iteration in LapDMM, respectively.

First, for the outer iteration of LapDMM, i.e., the DMM optimization, the main time cost is the update of the variational parameter $\boldsymbol{\gamma}$. Referring to Eq. 8, for each short text this step requires $O(K\overline{N})$ time. Therefore the time complexity of the outer iteration of LapDMM is $O(TDK\overline{N})$ in total and so does DMM. **Second**, the inner iteration of LapDMM refers to the Newton-Raphson update of $\boldsymbol{\gamma}$. For each short text d, its corresponding $\boldsymbol{\gamma}_d$ is iteratively updated using the moving average of its R nearest neighbors' $\boldsymbol{\gamma}_i$ such that $W_{di} = 1$. We thus present the time complexity of the inner iteration of LapDMM by $O(T\widehat{T}DKR)$. **Finally**, before LapDMM inference we need to construct an offline document graph, which stores the distances of all short text pairs. Naturally, this step requires $O(D^2)$ time by considering the distance operation is a constant.

In summary, we conclude that the overall time complexity of LapDMM is $O(TDK\overline{N} + T\widehat{T}DKR + D^2)$ by summing the time costs of the three steps discussed above. We outline the time complexities of DMM and LapDMM in Table 2.

Additionally, we briefly compare the time complexity of LapDMM with two traditional topic models, i.e., GPU-DMM and BTM, optimized by Gibbs sampling. First, the per-iteration time cost of GPU-DMM contains the topic assignment sampling for documents with $O(DK\overline{N})$ time and sample enrichment by the GPU model requiring $O(D\overline{N}l)$ time, where l denotes the average number of similar words stored in the GPU model. Therefore, the overall time complexity of GPU-DMM is about given by $O(TDK\overline{N} + TD\overline{N}l)$ (Li et al. 2016a). Second, in the context of BTM, it generates word co-occurrence patterns from each document, and draws topic assignments from K topics for them. Roughly, each document can totally generate $\mathrm{C}_{\overline{N}}^{2}$ word co-occurrence patterns, thus the overall time complexity of BTM can be directly measured by $O(TDK\mathrm{C}_{\overline{N}}^{2})$ (Cheng et al. 2014). Obviously, the time cost of BTM is

Table 2 Summary of time complexities

Model	Time complexity
DMM	$O(TDK\overline{N})$
LapDMM	$O(TDK\overline{N} + T\hat{T}DKR + D^2)$
OLapDMM	$O(T\widehat{M}K\overline{N} + T\hat{T}\widehat{M}KR + T(MBER + MR^2))$

T: outer iteration number; $\widehat{T}$: inner iteration number of LapDMM and OLapDMM; $\overline{N}$: the average document length: M: the mini-batch size; $\widehat{M}$ is equivalent to $M(R+1)$

affected by the average length of documents, and it becomes less efficient with relatively longer documents. Comparing with GPU-DMM and BTM, our LapDMM is less efficient mainly due to its inner iteration. Fortunately, empirical results indicate that LapDMM can achieve competitive performance with very few inner iterations, e.g., $\widehat{T} = 5$, in our experiments. In this sense, LapDMM is practical in real applications. More details of empirical efficiency evaluation are presented in Sect. 4.3.3.

3.4 Online Optimization of LapDMM

Following the time complexity analysis above, LapDMM becomes computationally expensive as the number of short texts (i.e., D) increases, and worse of all, it involves a more expensive step of document graph construction, requiring even $O(D^2)$ time. For inference speedup, we propose an online version of LapDMM, namely **O**nline **LapDMM** (**OLapDMM**).

Our OLapDMM simultaneously exploits the spirit of stochastic optimization with mini-batches (Hoffman et al. 2010, 2013; Foulds et al. 2013; Li et al. 2016b) and an up-to-date document graph that can efficiently find approximate nearest neighbors instead. We now describe them one by one.

3.4.1 Stochastic optimization

Reviewing the LapDMM optimization (i.e., ***Algorithm 1***), at each outer iteration we need to update the expected number counts with the current $\boldsymbol{\gamma}$ of all short texts. Specifically, we define some new notations to denote those expected number counts of interest for convenience:

$$\mathbb{N}_k^{\boldsymbol{\theta}} \overset{\Delta}{=} \mathbb{E}_{q(z|\boldsymbol{\gamma})}\left[\widehat{N}_k\right] = \sum\nolimits_{d=1}^{D} \boldsymbol{\gamma}_{dk} \tag{13}$$

$$\mathbb{N}_{kv}^{\boldsymbol{\phi}} \overset{\Delta}{=} \mathbb{E}_{q(z|\boldsymbol{\gamma})}\left[N_{kv}\right] = \sum\nolimits_{d=1}^{D} \sum\nolimits_{v \in d} N_{dv} \boldsymbol{\gamma}_{dk} \tag{14}$$

$$\mathbb{N}_k^{\boldsymbol{\phi}} \overset{\Delta}{=} \mathbb{E}_{q(z|\boldsymbol{\gamma})}\left[N_k\right] = \sum\nolimits_{d=1}^{D} N_d \boldsymbol{\gamma}_{dk} \tag{15}$$

This batch optimization is inefficient for big corpora of massive short texts. Inspired by Hoffman et al. (2010, 2013), Foulds et al. (2013), Li et al. (2016b), we refer to $\{\mathbb{N}_k^{\boldsymbol{\theta}}, \mathbb{N}_{kv}^{\boldsymbol{\phi}}, \mathbb{N}_k^{\boldsymbol{\phi}}\}$

as the global variables of LapDMM, and then update them using an online update manner with mini-batches of short texts.

At each outer iteration t, in OLapDMM we randomly draw a mini-batch (i.e., denoted by $\mathit{\Delta}_t$) of M short texts, and only update the variational parameters of $\mathit{\Delta}_t$ and its nearest neighboring short texts. Then, following Foulds et al. (2013), update $\{\mathbb{N}_k^{\boldsymbol{\theta}}, \mathbb{N}_{kv}^{\boldsymbol{\phi}}, \mathbb{N}_k^{\boldsymbol{\phi}}\}$ by using an online average of their current values and expected values as below:

$$\mathbb{N}_k^{\boldsymbol{\theta}} \leftarrow (1-\hat{\rho}_t)\mathbb{N}_k^{\boldsymbol{\theta}} + \hat{\rho}_t \frac{D}{|\hat{\mathit{\Delta}}_t|} \sum\nolimits_{d \in \hat{\mathit{\Delta}}_t} \boldsymbol{\gamma}_{dk}, \tag{16}$$

$$\mathbb{N}_{kv}^{\boldsymbol{\phi}} \leftarrow (1-\hat{\rho}_t)\mathbb{N}_{kv}^{\boldsymbol{\phi}} + \hat{\rho}_t \frac{D}{|\hat{\mathit{\Delta}}_t|} \sum\nolimits_{d \in \hat{\mathit{\Delta}}_t} \sum\nolimits_{v \in d} N_{dv}\boldsymbol{\gamma}_{dk}, \tag{17}$$

$$\mathbb{N}_k^{\boldsymbol{\phi}} \leftarrow (1-\hat{\rho}_t)\mathbb{N}_k^{\boldsymbol{\phi}} + \hat{\rho}_t \frac{D}{|\hat{\mathit{\Delta}}_t|} \sum\nolimits_{d \in \hat{\mathit{\Delta}}_t} N_d\boldsymbol{\gamma}_{dk}, \tag{18}$$

where $\hat{\rho}_t$ is the learning rate; and $\hat{\mathit{\Delta}}_t$ denotes the set of $\mathit{\Delta}_t$ and its nearest neighboring short texts. Since the size of $\hat{\mathit{\Delta}}_t$ is always much less than that of the whole corpus, the $\{\mathbb{N}_k^{\boldsymbol{\theta}}, \mathbb{N}_{kv}^{\boldsymbol{\phi}}, \mathbb{N}_k^{\boldsymbol{\phi}}\}$ updates of OLapDMM are much faster than those of batch LapDMM, especially for big corpora of massive short texts.

3.4.2 Up-to-date document graph construction

In OLapDMM, we replace the expensive offline document graph with a much efficient up-to-date alternative.

The pre-computed offline document graph stores the exact R nearest neighbors of all short texts, so as to achieve the variational manifold constraints. In some sense, the goal of variational manifold regularization in LapDMM is to spread topical signals among similar short texts, but not limited to the most similar pairs. Therefore, we can utilize the variational manifold regularization with approximate nearest neighbors to replace the one with the exact nearest neighbors. Following this idea, we develop an up-to-date document graph that can efficiently find approximate R nearest neighbors of short texts. Formally, it involves the following key steps:

- *Initialization* At the first iteration of OLapDMM, we initialize a document graph $\mathbb{U}_1$, which stores the exact R nearest neighbors of the first selected mini-batch $\mathit{\Delta}_1$.
- *Search* At each iteration $t \geq 2$, we search the approximate R nearest neighbors of the current mini-batch $\mathit{\Delta}_t$ using the graph nearest neighbor search (GNNS) algorithm (Hajebi et al. 2011) based on the document graph $\mathbb{U}_{t-1}$
- *Update* Incorporate[2] the mini-batch $\mathit{\Delta}_t$ with its approximate R nearest neighbors into the document graph $\mathbb{U}_{t-1}$. Besides, continue to update $\mathbb{U}_{t-1}$ by referring to the recent obtained neighbor pairs. We finally achieve an updated document graph $\mathbb{U}_t$

[2] Note that if OLapDMM has swept the whole corpus, the document graph size will no longer increase.

The implementation details of the **Search** and **Update steps** are presented below.

Search details: For a short text being absent in the document graph $\mathbb{U}_{t-1}$, we can employ the GNNS algorithm (Hajebi et al. 2011) to efficiently find its approximate R nearest neighbors. The GNNS algorithm is built on the spirit of hill-climbing, starting from a random sampled node of short text stored in $\mathbb{U}_{t-1}$. Formally, given a query short text s, the GNNS algorithm repeats the following steps B times for collecting its candidate neighbor set $\mathbb{C}$:

- **Step 1**: Randomly select a node of short text $\widehat{s}_0$ stored in $\mathbb{U}_{t-1}$.
- **Step 2**: At each iteration $\widehat{t} \geq 1$, scan all R neighbors of $\widehat{s}_{\widehat{t}-1}$ stored in $\mathbb{U}_{t-1}$ (i.e., denoted by $\mathbb{U}_{t-1}(\widehat{s}_{\widehat{t}-1})$). Then, return the closest short text of the query s (i.e., denoted by $\widehat{s}_{\widehat{t}} = \mathbb{U}_{t-1}(s, \widehat{s}_{\widehat{t}-1})$) as the next node, and merge $\mathbb{U}_{t-1}(\widehat{s}_{\widehat{t}-1})$ into the candidate neighbor set $\mathbb{C}$. i.e., $\mathbb{C} = \mathbb{C} \bigcup \mathbb{U}_{t-1}(\widehat{s}_{\widehat{t}-1})$.
- **Step 3**: Repeat the **Step 2** E times.

After obtaining $\mathbb{C}$, we can find the approximate R nearest neighbors (i.e., denoted by $\mathbb{C}(s)$) of the query short text s from it. The parameters of GNNS are empirically set as follows: $B = 5$ and $E = 5$.

Besides, in OLapDMM we note that after sweeping the whole corpus, the document graph $\mathbb{U}_{t-1}$ involves neighbors of all short texts. We use the notation $\exists_s \mathbb{U}_{t-1}$ to denote that $\mathbb{U}_{t-1}$ stores R neighbors of short text s. In this situation, we not only use the candidate neighbor set $\mathbb{C}$ obtained by the above loops, but also refer to its approximate R nearest neighbors stored in $\mathbb{U}_{t-1}$, i.e., $\mathbb{U}_{t-1}(s)$, as candidates, i.e., $\mathbb{C} = \mathbb{C} \bigcup \mathbb{U}_{t-1}(s)$. Then, we search its approximate R nearest neighbors, i.e., $\mathbb{C}(s)$, from the aggregated $\mathbb{C}$.

For clarity, the full **Search step** of OLapDMM is outlined in **Algorithm 2**.

Algorithm 2 Search approximate nearest neighbors

```
Input: a query short text $s$ and the current document graph $\mathbb{U}_{t-1}$
Set $B = 5$, $E = 5$ and $\mathbb{C} = \emptyset$
Repeat
    Select a node of short text $\widehat{s}_0$ stored in $\mathbb{U}_{t-1}$ randomly
    For $\widehat{t} = 1, 2, \ldots, E$
        $\widehat{s}_{\widehat{t}} = \mathbb{U}_{t-1}(s, \widehat{s}_{\widehat{t}-1})$
        $\mathbb{C} = \mathbb{C} \bigcup \mathbb{U}_{t-1}(\widehat{s}_{\widehat{t}-1})$
    End for
Until $B$ times
If $\exists_s \mathbb{U}_{t-1}$ then $\mathbb{C} = \mathbb{C} \bigcup \mathbb{U}_{t-1}(s)$
Output: $\mathbb{C}(s)$, i.e., the approximate $R$ nearest neighbors of the query $s$ from $\mathbb{C}$
```

Algorithm 3 Update $\mathbb{U}_{t-1}$

```
Input: the document graph $\mathbb{U}_{t-1}$ and approximate nearest neighbors of the current mini-batch $\bigcup_{s\in\Delta_t}\mathbb{C}(s)$
For $s \in \Delta_t$
    If $\exists_s \mathbb{U}_{t-1}$ then $\mathbb{U}_{t-1}(s) \leftarrow \mathbb{C}(s)$
    Otherwise $\mathbb{U}_{t-1} = \mathbb{U}_{t-1} \bigcup \mathbb{C}(s)$
    For $s' \in \mathbb{C}(s)$
        For $s'' \in \mathbb{U}_{t-1}(s')$
            If $\mathrm{dis(s',s)} < \mathrm{dis(s',s'')}$, then replace $s''$ with $s$ in $\mathbb{U}_{t-1}(s')$
        End for
    End for
End for
Output: $\mathbb{U}_t$, i.e., a new graph document by the above updates
```

Update details of $\mathbb{U}_{t-1}$: After the **Search step**, we obtain the approximate R nearest neighbors of all short texts in the mini-batch Δ_t, i.e., $\bigcup_{s\in\Delta_t}\mathbb{C}(s)$. Then, we update $\mathbb{U}_{t-1}$ as follows: **(1)** For any short text $s \in \Delta_t$, if the document graph $\mathbb{U}_{t-1}$ does not store its R neighbors, we incrementally add $\mathbb{C}(s)$ into $\mathbb{U}_{t-1}$, i.e., $\mathbb{U}_{t-1} = \mathbb{U}_{t-1} \bigcup \mathbb{C}(s)$; otherwise we replace its current neighbors $\mathbb{U}_{t-1}(s)$ with the new ones $\mathbb{C}(s)$, i.e., $\mathbb{U}_{t-1}(s) \leftarrow \mathbb{C}(s)$. **(2)** for any neighbor of the mini-batch Δ_t, update its neighbors stored in $\mathbb{U}_{t-1}$ if more closer queries to it are observed. For example, considering a short text $s' \in \mathbb{C}(s)$, $s \in \Delta_t$. if the distance between s' and s is smaller than an existing neighbor of s' stored in $\mathbb{U}_{t-1}$, replace it with the query s. After the above updates, we finally achieve a new document graph $\mathbb{U}_t$

For clarity, the full **Update step** of OLapDMM is outlined in **Algorithm 3**.

Algorithm 4 Optimization for OLapDMM

```
Set model and training parameters, including the number of nearest neighbors $R$, the regularization parameter $\lambda$, the Newton-Raphson learning rate $\rho$, the online average learning rate $\widehat{\rho}_t = \frac{1}{(1000+t)^{0.9}}$ and parameters of the GNNS algorithm i.e., $B$ and $E$.
Initialize $\boldsymbol{\gamma}$ randomly and then expected number counts
Draw the first mini-batch $\Delta_1$ randomly, and initialize a graph document $\mathbb{U}_1$ of $\Delta_1$
For $t = 1, 2, \ldots,$ MaxIter
    If $t \geq 2$
        Draw $\Delta_t$ randomly
        Search approximate $R$ nearest neighbors of $\Delta_t$ using Algorithm 2 given $\mathbb{U}_{t-1}$
        Update the graph document $\mathbb{U}_{t-1}$ using Algorithm 3, leading to a new one $\mathbb{U}_t$
    End if
    Update $\boldsymbol{\gamma}$ of $\widehat{\Delta}_t$ using Eq.8
    While the objective of $\widehat{\Delta}_t$ does not decrease Do
        Update $\gamma$ of $\widehat{\Delta}_t$ using Eq.9
    End While
    Update $\{\mathbb{N}_k^{\boldsymbol{\theta}}, \mathbb{N}_{kv}^{\boldsymbol{\phi}}, \mathbb{N}_k^{\boldsymbol{\phi}}\}$ using Eqs.16, 17 and 18
End for
Compute $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$ using Eqs.10 and 11
```

3.4.3 Summary of OLapDMM optimization

For clarity, the full optimization process of OLapDMM can be summarized as follows: **(1)** Simultaneously initialize the parameters, i.e., R, λ, ρ, $\widehat{\rho}$, B and E, and the document graph $\mathbb{U}_1$ of the first selected mini-batch $\varDelta_1$; **(2)** at each outer iteration t,[3] draw a new mini-batch $\varDelta_t$, and find the approximate R nearest neighbors of $\varDelta_t$ using ***Algorithm 2***; **(3)** update the document graph $\mathbb{U}_{t-1}$ using ***Algorithm 3***, leading to a new one $\mathbb{U}_t$; **(4)** update the variational parameters $\boldsymbol{\gamma}$ of $\varDelta_t$ and all its approximate neighbors, i.e., $\widehat{\varDelta}_t$; **(5)** update the expected number counts $\{\mathbb{N}_k^{\boldsymbol{\theta}}, \mathbb{N}_{kv}^{\boldsymbol{\phi}}, \mathbb{N}_k^{\boldsymbol{\phi}}\}$ using Eqs. 16, 17 and 18.

In summary, we outline the optimization of LapDMM in **Algorithm 4**.

3.4.4 Time complexity of OLapDMM

In this subsection, we discuss the time complexity of OLapDMM. To this end, we divide the OLapDMM optimization into two key parts, i.e., the update of the variational parameter $\boldsymbol{\gamma}$ and the search and update of the document graph $\mathbb{U}$. We now analyze them one by one.

Time complexity of γ update: At each outer iteration of OLapDMM, the $\boldsymbol{\gamma}$ update of a short text requires $O(K\overline{N})$ time (ref. Eq. 8). After drawing a mini-batch of M short texts, we need to update the $\boldsymbol{\gamma}$ of both them and their R neighbors, requiring at most $O(\widehat{M}K\overline{N})$ time,[4] where we define $\widehat{M} = M(R+1)$ for convenience. At each inner iteration, each short text performs the Newton-Raphson update for its corresponding $\boldsymbol{\gamma}$, spending $O(\widehat{T}KR)$ times. Therefore, the inner iteration totally requires $O(\widehat{T}\widehat{M}KR)$ time. In summary, the total time complexity of γ update is $O(T\widehat{M}K\overline{N} + T\widehat{T}\widehat{M}KR)$.

Time complexity of search and $\mathbb{U}$ update: Reviewing ***Algorithm 2***, given a query short text it actually restarts an E-stepsize nearest neighbor expansion (NN-expansion) B times for collecting candidate neighbors $\mathbb{C}$, and then select approximate R nearest neighbors from $\mathbb{C}$. The time costs of NN-expansion and neighbor selection are both proportional to the size of $\mathbb{C}(s)$. Therefore, for each short text the total time cost of neighbor search is $O(BER)$, and further the time complexity of a mini-batch is $O(MBER)$. Reviewing ***Algorithm 3***, its main cost is to update the current mini-batch neighbors' neighbors stored in $\mathbb{U}$. Obviously, this needs $O(MR^2)$ time. In summary, the total time complexity of the search and update of $\mathbb{U}$ is $O(T(MBER + MR^2))$.

Summary and discussion: Following the above analysis, we present that the overall time complexity of OLapDMM is $O(T\widehat{M}K\overline{N} + T\widehat{T}\widehat{M}KR + T(MBER + MR^2))$, outlined in Table 2. Naturally, the time cost of OLapDMM is much less than that of batch LapDMM. **First**, at each outer iteration, OLapDMM only updates the variational parameters of $\widehat{M}$ short texts, instead of the whole corpus of D ones, i.e., $\widehat{M} \ll D$. **Second**, in OLapDMM, the time cost of neighbor search is totally $O(T(MBER + MR^2))$, where the parameters of B, E and R are often very small numbers. Therefore, this step of OLapDMM is much efficient than that of batch LapDMM, requiring expensive $O(D^2)$ time.

Overall, we suggest that OLapDMM can achieve much faster inference than batch LapDMM, and it is practical to learn topics over even massive short texts.

[3] At the first iteration, we can directly obtain R neighbors of $\varDelta_1$ from $\mathbb{U}_1$.

[4] Some of the neighbors may be the same.

Table 3 Summary of the datasets

Dataset	D	V	$\overline{N}$	L
Trec	5952	8392	4.94	6
Snipptes	12,340	30,445	17.5	8
StackOverFlow (SOF)	20,000	17,996	4.93	20
BaiduQA	189,080	26,565	3.94	35
Tweets	10 million	109.345	4.87	–

D: the number of documents. V: the number of unique words. $\overline{N}$: the average document length. L: the number of categories

4 Experiment

In this section, we present the empirical results on various tasks.

4.1 Experimental setup

Datasets: For evaluations, we employed five datasets, including four small datasets and one large dataset. Their descriptions are outlined below.

- *Trec*[5] The *Trec* question dataset involves 6 question types of {Abbreviation, Entity, Description, Human, Location, Numeric}. It contains 5952 questions with a vocabulary of 8392 words.
- *Snippets*[6] The *Snippets* dataset was selected from the results of web search transaction using predefined phrases of 8 different domains (Phan et al. 2008), i.e., {Business, Computers, Health, Education, Culture, Engineering, Sports, Politics}. It contains 12,340 research results with a vocabulary of 30,445 words.
- *StackOverFlow*[7] (*SOF*) The original *SOF* dataset was published in Kaggle.com, which collects 3 millions question titles from July 31st, 2012 to August 14, 2012. In our experiments, we use a subset of 20,000 samples from 20 different tags (Xu et al. 2017), i.e., {svn, oracle, bash, apache, excel, matlab, cocoa, visual-studio, osx, wordpress, spring, hibernate, scala, sharepoint, ajax, drupal, qt, haskell, linq, magento}. This subset involves a vocabulary of 17,996 words.
- *BaiduQA*[8] This dataset was collected by Cheng et al. (2014), crawling 189,080 question samples from a popular Chinese Q&A website. The question samples are classified into 35 categories.
- *Tweets* The *Tweets* dataset consists of 10 million Twitter posts crawled from the Internet. It involves a vocabulary of 109.345 words.

After removals of stopwords, the statistics of datasets are summarized in Table 3.

[5] http://cogcomp.cs.illinois.edu/Data/QA/QC/.

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

[7] https://github.com/jacoxu/STC2.

[8] http://zhidao.baidu.com.

Baseline models We selected five state-of-the-art topic models of short texts as baseline methods. Descriptions and model-specific settings are outlined below.

- *DMM* Nigam et al. (2000), Yin and Wang (2014): This is the ancestor method of our models. In the experiment, it is inferred using collapsed variational inference.
- *GPU-DMM* Li et al. (2016a, 2017): This is an extension of DMM with word embeddings. We use the Gibbs sampling code provided by its authors.[9]
- *GPU-PDMM* Li et al. (2017): This model is an extension of GPU-DMM, which allows each short text to be associated with multiple topics using a Poisson prior.
- *BTM* Yan et al. (2013), Cheng et al. (2014): The model directly exploits biterms over the whole corpus. We use the Gibbs sampling code provided by its authors.[10]
- *Latent topic model (LTM)* Li et al. (2018c): This is a LDA-based topic model by adaptively aggregating short texts. The Gibbs sampling is also used to infer it.

For all models, the Dirichlet priors α and β are set to 0.1 and 0.01, respectively. The parameters of baseline models are tuned following the suggestions discussed in their original papers. The basic parameters of LapDMM and OLapDMM are empirically set as: $\widehat{T} = 5$, $R = 9$, $\lambda = 0.1$ and $\rho = 0.1$.

Besides, GPU-DMM, GPU-PDMM and our models require word embeddings. For the English datasets, we employ the pre-trained 100-dimensional *GloVe*[11] word embeddings (i.e., the ones trained on *Wikipedia + Gigaword*), and for the Chinese dataset, i.e., *BaiduQA*, we employ the pre-trained 300-dimensional Chinese word embeddings[12] trained on *Baidu Encyclopedia* (Li et al. 2018b).

4.2 Evaluation of LapDMM

We first compare LapDMM with baseline models on four small datasets, i.e., *Trec*, *Snippets*, *SOF* and *BaiduQA*, across a qualitative document topic visualization task and three quantitative tasks, i.e., topic quality, document clustering and classification.

For clarity, the versions of LapDMM with document graph measured by term frequency and WMD are referred to as LapDMM_T and LapDMM_W, respectively.

4.2.1 Document topic visualization

We empirically evaluate whether LapDMM can estimate more accurate topic representations for documents, i.e., alleviating the sensitivity problem of DMM mentioned in the introduction section. To this end, we illustrate some example inferred topics of documents learnt by DMM and LapDMM_W across *StackOverFlow* (i.e., when $K = 20$).

Specifically, the inferred topics (i.e., the topics with the largest variational parameter) of five example neighboring documents are presented in Table 4. Intuitively, those documents are more likely talking about the topic of “linq”. However, DMM produces that three of them are dominated by other topics, i.e., “excel”, “oracle” and “matlab”. In these cases,

[9] https://github.com/NobodyWHU/GPUDMM.

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

[11] https://nlp.stanford.edu/projects/glove/.

[12] https://github.com/Embedding/Chinese-Word-Vectors.

Table 4 We illustrate several inferred topics of example neighboring documents, where the stopwords are denoted by Italics

Document	Inferred topic	
	DMM	LapDMM$_W$
How do I fill *a* dataset *from a* linq query resultset?	linq (0.32)	linq (0.57)
Return typed datatable *from* linq query	excel (0.33)	linq (0.35)
Updating columns *with the* primary key *using* linq	oracle (0.22)	linq (0.31)
Help required *to* optimize linq query	matlab (0.19)	linq (0.41)
How do i get the min *from a* linq *to* dataset query	linq (0.31)	oracle (0.29)

For each document, the topic with largest variational parameter (in brackets) is presented

that maybe caused by some less discriminative words, e.g., *return*, *setup* and *column*, which are associated with less weights for the topic of "linq", but they equally contribute to the final inferred topics of documents compared with the discriminative word *linq*. In contrast to DMM, the inferred topics by LapDMM$_W$ are more consistent with the ground-truth topic of "linq" in 4/5 documents. The observation indicates that the manifold regularization can alleviate the sensitivity problem of DMM to some extent. According to the update equation of LapDMM, i.e., Eq. 9, the variational parameters of neighboring documents are jointly updated, therefore the larger topics shared by neighboring documents are strengthened by each other. However, a failure case, i.e., the fifth document, also happens, where DMM beats LapDMM$_W$ by achieving the ground-truth topic. The reason is that the inferred topic of each document maybe also affected by inaccurate topic estimations from neighboring documents. We examine all neighbors of the fifth document, and found that about one-third neighbors produce inaccurate topic proportions, so as to validate our analysis. (*PS*: Note that although those five documents are neighbors each other, they may not share all same neighbors.)

4.2.2 Evaluation by topic quality

We present the empirical results on the task of topic quality. We quantitatively measure the quality of topics using the *Topic Coherence (TC)* score, which is computed by counting co-occurrences of their top words (Newman et al. 2010a, b). The intuition is that for any topic, more co-occurrences between its top words, more semantically coherent it is. Towards reproducible evaluations, we compute the *TC* scores using the public *TC* project[13] developed by Roder et al. (2015). We use the setting of "C_V". Specifically, for the English datasets, the topical top word co-occurrences are counted on the default reference corpus, and for the Chinese dataset, i.e., *BaiduQA*, the co-occurrences are counted on an extra reference corpus of 2 millions documents crawled from *Baidu Encyclopedia*.

We present the average *TC* scores of top-10 words of all models in Table 5. Several observations are made below.

[13] https://github.com/AKSW/Palmetto/wiki/Coherences.

Table 5 Results of topic coherence scores (mean±std)

Model	Topic	***Trec***	***Snippets***	***SOF***	***BaiduQA***
DMM	*K=25*	0.44±0.07‡	0.43±0.07‡	0.37±0.08‡	0.45±0.06‡
	K=50	0.46±0.07‡	0.43±0.09‡	0.34±0.08‡	0.44±0.05‡
GPU-DMM	*K=25*	0.47±0.07	0.43±0.06‡	0.38±0.07	0.45±0.06‡
	K=50	0.47±0.07	0.43±0.08‡	0.35±0.06‡	0.45±0.09‡
GPU-PDMM	*K=25*	0.46±0.09	0.44±0.06‡	0.37±0.07	0.46±0.07
	K=50	0.47±0.08	0.45±0.07‡	0.35±0.08‡	0.45±0.07‡
BTM	*K=25*	0.35±0.04‡	0.43±0.06‡	**0.40±0.07**	0.47±0.09
	K=50	0.36±0.06‡	0.45±0.06‡	0.36±0.07	0.46±0.07
LTM	*K=25*	**0.48±0.07**	**0.47±0.07**	0.36±0.08‡	0.47±0.07‡
	K=50	0.48±0.07	0.45±0.08	0.35±0.07‡	0.45±0.08‡
LapDMM$_{\mathrm{T}}$	*K=25*	0.46±0.08	0.45±0.07	0.37±0.07	0.46±0.08
	K=50	0.47±0.08	0.45±0.09	0.36±0.09	**0.47±0.08**
LapDMM$_{\mathrm{W}}$	*K=25*	**0.48±0.06**	0.46±0.08	0.39±0.09	**0.48±0.07**
	K=50	**0.49±0.08**	**0.47±0.08**	**0.37±0.08**	**0.47±0.06**

Best results are highlighted in bold

"‡" means that the gains of both versions of LapDMM are statistically significant simultaneously (paired sample t-test at 0.01 level)

- LapDMM$_{\mathrm{W}}$ performs the best among all models, where it ranks the first in most (i.e., 6/8) settings, indicating that LapDMM$_{\mathrm{W}}$ enables to learn more coherent topics from short texts. For example, the *TC* scores of LapDMM$_{\mathrm{W}}$ are about 0.01~0.04(2~ 9%) higher than those of GPU-DMM.
- LapDMM$_{\mathrm{T}}$ performs quite competitive with GPU-DMM, BTM and LTM, especially the significant gain on *Trec* compared with BTM. This implies that LapDMM$_{\mathrm{T}}$ is also capable of outputting coherent topics.
- We observe that both LapDMM$_{\mathrm{T}}$ and LapDMM$_{\mathrm{W}}$ significantly performs better than DMM, e.g., achieving about 0.01~0.03 and 0.03~0.04 (2~7% and 7~9%) higher *TC* scores on *Snippets* and *BaiduQA*, respectively. Those results raise the fact that the variational manifold regularization can effectively improve the quality of topics learnt from English as well as Chinese short texts.

Specifically, since the TC scores measure the topic quality of top topical words, we can discuss the effect of word embeddings that are associated with the semantic information of words. Observations and discussions are given below.

- The three DMM variants with word embeddings, i.e., GPU-DMM, GPU-PDMM and LapDMM$_{\mathrm{W}}$, can perform higher TC scores than DMM in most settings. For example, the performance gains of GPU-PDMM and LapDMM$_{\mathrm{W}}$ are about 0.01~0.02 (2~ 5%) and 0.02~0.04 (5~9%), respectively. The results directly indicate that the word embeddings can effectively improve the quality of topics extracted from short texts. Naturally, the observation is reasonable since the word embeddings are capable of capturing similar words, so as to improve coherent topics.

- LapDMM$_\mathrm{W}$ consistently outperforms LapDMM$_\mathrm{T}$. This further indicates the effectiveness of incorporating word embeddings, since LapDMM$_\mathrm{W}$ employs the word embeddings-based WMD, which can better capture similarities between short texts at the semantic level.
- The version of LapDMM without word embeddings, i.e., LapDMM$_\mathrm{T}$, performs competitive with GPU-DMM and GPU-PDMM, while consistently outperforms DMM. Based on this observation, we consider that the manifold regularization is also effective for improving the topic quality. The reason is that the manifold regularization enables to enrich word co-occurrences from nearest neighboring documents to some extent.

4.2.3 Evaluation by clustering

We evaluate our models by the task of document clustering. To apply topic models for clustering, we refer to each topic as a cluster, and set the number of topics to the true category number of datasets. For models trained by Gibbs sampling, i.e., GPU-DMM, GPU-PDMM, BTM and LTM, each document is assigned to the topic with the largest probability of topic sampling after burn-in iterations; for models trained by collapsed variational inference, i.e., DMM and our LapDMM, each document is assigned to the topic with largest variational parameter.

We employ two popular clustering metrics, i.e., *ACCuracy (ACC)* and *Normalized Mutual Information (NMI)*. Let $\mathbf{Y}$ and $\mathbf{C}$ be the true category set and the prediction cluster set of a given dataset, respectively. Then, the *NMI* score can be computed by:

$$\boldsymbol{NMI}(\mathbf{Y}, \mathbf{C}) = \frac{MI(\mathbf{Y}, \mathbf{C})}{\sqrt{H(\mathbf{Y})H(\mathbf{C})}}, \tag{19}$$

where $MI(\mathbf{Y}, \mathbf{C})$ denotes the mutual information of $\mathbf{Y}$ and $\mathbf{C}$; and $H(\cdot)$ denotes the entropy. Besides, for any document d, let y_d and c_d denote the true category and prediction cluster, respectively. Then, the *ACC* score can be computed by:

$$\boldsymbol{ACC} = \frac{\sum_{d=1}^{D} \mathcal{I}\left(y_d, map\left(c_d\right)\right)}{D}, \tag{20}$$

where $\mathcal{I}(\cdot)$ denotes the indicator function; and $map\left(c_d\right)$ is the mapping function between c_d and y_d, computed by the Hungarian algorithm. Higher values of *NMI* and *ACC* imply better performance.

We independently run each model 10 times, and report the average scores in Table 6. We observe the following comparisons.

- Surprisingly, both LapDMM$_\mathrm{T}$ and LapDMM$_\mathrm{W}$ significantly outperform all baseline models on datasets of *Trec*, *Snippets* and *SOF*, which contain relatively less short texts, i.e., less than 20,000. For example, the performance gains of *NMI* achieve about 0.15~ 0.17 (120~168%) across *Trec* and those of *ACC* are even about 0.23~0.2 (46~54%) across *SOF*. The possible reason is that with small numbers of short texts, the baseline models can be by no means sufficiently trained to discover accurate topical structures of corpora. In this situation, the variational manifold regularization enables to effectively enhance the topic structure discovery by preserving local neighborhood structure of short texts, i.e., linking neighboring ones. We conclude that LapDMM is a very competitive candidate given smaller datasets of short texts.

Table 6 Clustering results of NMI and ACC (mean±std)

Model	Metric	*Trec*	*Snippets*	*SOF*	*BaiduQA*
DMM	*NMI*	0.125±0.06‡	0.526±0.05‡	0.457±0.05‡	0.472±0.04‡
	ACC	0.355±0.05‡	0.698±0.04‡	0.494±0.03‡	0.419±0.03‡
GPU-DMM	*NMI*	0.127±0.04‡	0.544±0.02‡	0.439±0.01‡	0.464±0.02‡
	ACC	0.352±0.03‡	0.723±0.02‡	0.482±0.03‡	0.407±0.04‡
GPU-PDMM	*NMI*	0.133±0.02‡	0.527±0.03‡	0.442±0.02‡	0.469±0.02‡
	ACC	0.362±0.04‡	0.711±0.04‡	0.489±0.03‡	0.411±0.02‡
BTM	*NMI*	0.109±0.05‡	0.521±0.02‡	0.429±0.02‡	0.445±0.01‡
	ACC	0.337±0.04‡	0.683±0.04‡	0.472±0.01‡	0.412±0.03‡
LTM	*NMI*	0.114±0.06‡	0.539±0.02‡	0.442±0.01‡	0.456±0.02‡
	ACC	0.348±0.05‡	0.705±0.05‡	0.498±0.02‡	0.418±0.01‡
LapDMM$_\mathrm{T}$	*NMI*	**0.292±0.02**	0.634±0.04	0.641±0.01	0.482±0.01
	ACC	**0.499±0.05**	0.761±0.06	**0.728±0.02**	**0.461±0.01**
LapDMM$_\mathrm{W}$	*NMI*	0.288±0.04	**0.653±0.01**	**0.645±0.02**	**0.486±0.01**
	ACC	0.484±0.05	**0.793±0.03**	0.710±0.05	0.457±0.03

Best results are highlighted in bold

"‡" means that the gains of both versions of LapDMM are statistically significant simultaneously (paired sample t-test at 0.01 level)

- Our LapDMM$_\mathrm{T}$ and LapDMM$_\mathrm{W}$ also consistently perform better than all baseline models on *BaiduQA*, which contains relatively more short texts, i.e., 189,080. This further indicates the variational manifold regularization is beneficial for short text clustering. However, we found that the performance gain on *BaiduQA* is less than those on other smaller datasets. For example, the *NMI* and *ACC* scores of LapDMM$_\mathrm{W}$ are only about 0.01(2%) and 0.04(9%) higher than those of DMM, respectively. This may imply the affect of manifold regularization gets smaller as the number of short texts increases.
- Unlike the results of topic coherence evaluation, we observe that LapDMM$_\mathrm{T}$ is competitive with LapDMM$_\mathrm{W}$ on the task of clustering. This is an interpretable observation: Note that the *TC* score measures the topic coherence by counting topical top word co-occurrences on big extra reference corpora. while the WMD depends on word embeddings that are trained by exploring the word context information (including word co-occurrences) of extra reference corpora. Therefore the WMD is beneficial for *TC* scores, leading to superior performance for LapDMM$_\mathrm{W}$. In contrast, the variational manifold regularization can directly enhance the clustering by linking neighboring short texts, but is insensitive to the types of neighbors.

In summary, we consider that both versions of LapDMM are capable of achieving superior performance for short text clustering, especially for the datasets with relatively small numbers of short texts. Our empirical results are consistent with the previous study of Cai et al. (2008), where it has shown that the manifold regularization methodology significantly improved the clustering performance of PLSI.

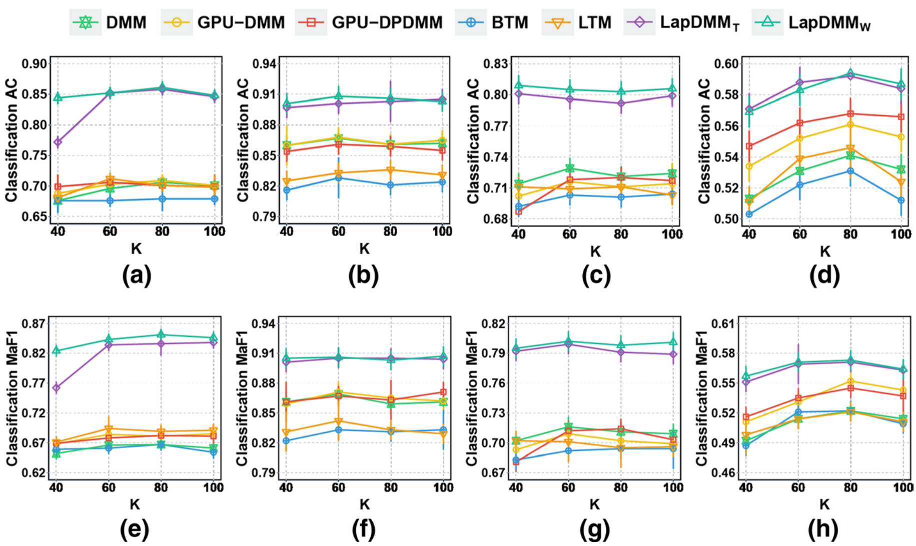

Fig. 2 Classification results of AC and MaF1 with different topic numbers (i.e., K=40, 60, 80, 100) across the datasets of **a**, **e** Trec, **b**, **f** Snippets, **c**, **g** SOF and **d**, **f** BaiduQA

4.2.4 Evaluation by classification

We further evaluate our models by the task of document classification. For each model, we train it on the datasets, and exploit the SW representation described in Li et al. (2016a) as the feature vectors of short texts. Then, we feed those feature vectors into SVMs[14] to train text classifiers. The *ACcuracy (AC)* and *Macro-F1 (MaF1)* are used to measure the classification performance. Higher values of *AC* and *MaF1* imply better performance.

For each dataset, we conduct a 5-fold cross validation evaluation. The average *AC* and *MaF1* scores of 10 independent runs are shown in Fig. 2. The observations are described below.

- First, our LapDMM$_T$ and LapDMM$_W$ significantly outperform all baseline models across the datasets of *Trec* and *SOF*, where the improvements are higher than 0.17 (25%) and 0.09 (13%), respectively. That is, our models can output more discriminative topical representations for short texts, being beneficial for improving classification performance.
- We also observe significant improvements of our models on the datasets of *Snippets* and *BaiduQA*, where their *AC* and *MaF1* scores are about 0.04 (5%) and 0.02 (4%) than those of baseline models, respectively. Those results provide further evidences that the topical representation learnt by our models are more discriminative.
- Besides, we can see that for all models, the *AC* and *MaF1* scores of $K = 60, 80$, and 80 are higher than those of $K = 40$ in most cases. The possible reason is that using more topics can discover more accurate document-level structures to some extent. This

[14] http://scikit-learn.org/.

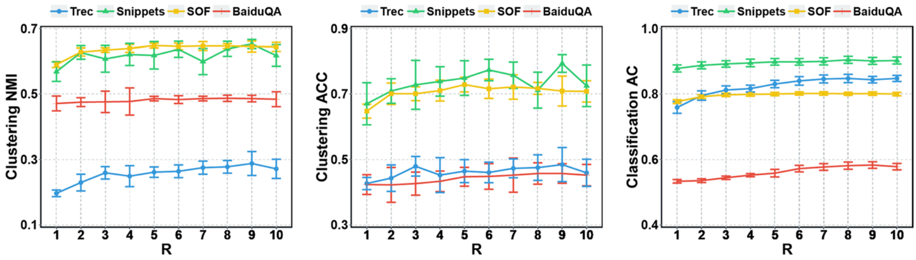

Fig. 3 Evaluation results of different R values

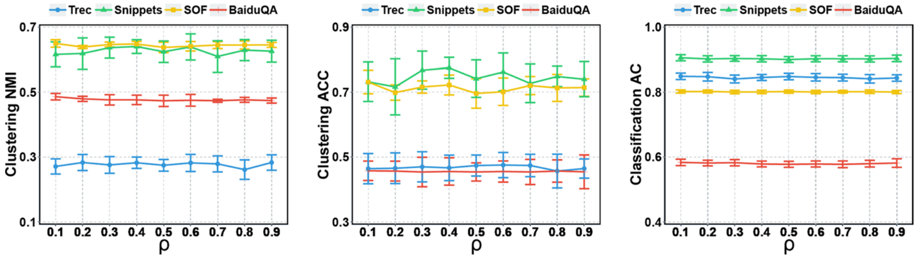

Fig. 4 Evaluation results of different ρ values

empirical result is consistent with the previous study of Li et al. (2016a), where it has also reported better classification scores with more topics.

4.2.5 Parameter analysis of R and ρ

We now empirically evaluate two crucial parameters, including the nearest neighbor number R and learning rate ρ in the inner Newton-Raphson iteration. Towards this goal, we examine the clustering (i.e., *NMI* and *ACC*) and classification (i.e., *AC*) results of LapDMM$_{\mathrm{W}}$ by varying the two parameters' values. In terms of classification, we fix K to 60.

- We first evaluate the impact of different R values over the set $\{1, 2, \cdots, 10\}$. The empirical results are shown in Fig. 3. Roughly speaking, the overall trend is that the performance becomes better as the value of R increases, e.g., the *AC* scores of classification across *Trec*. The best scores are achieved at $R = 8$ and 9 in most cases. The *ACC* scores of clustering seem a bit unsmooth, but LapDMM$_{\mathrm{W}}$ also achieves higher *ACC* values when $R = 8$ and 9. Those results empirically tell us that using more nearest neighbors in the variational manifolds may be helpful. Therefore, we fix $R = 9$ in our experiments, and suggest to set a relatively larger value of R in practical applications.
- Then, we examine the impact of ρ with different values over the set $\{0.1, 0.2, \cdots, 0.9\}$. The experimental results are shown in Fig. 4. Overall speaking, we can observe that the performance gap between different ρ values are unobvious. Therefore, we argue that our

Table 7 Topic coherence scores of online models across the *Tweets* dataset (mean±std)

Model	*K*=50	*K*=100	*K*=150	*K*=200
ODMM	0.39±0.09‡	0.41±0.07‡	0.39±0.08‡	0.41±0.06‡
OGPU-DMM	0.39±0.07	0.40±0.08‡	0.37±0.07‡	0.39±0.06‡
OGPU-PDMM	0.40±0.09‡	0.39±0.07‡	0.37±0.06‡	0.41±0.06‡
OBTM	0.38±0.06‡	0.40±0.06‡	0.39±0.08	0.40±0.07‡
OLTM	0.40±0.08	0.40±0.07‡	0.37±0.08‡	0.41±0.06‡
OLapDMM$_{\mathrm{T}}$	0.41±0.07	0.42±0.06	**0.41±0.07**	0.42±0.06
OLapDMM$_{\mathrm{W}}$	**0.42±0.04**	**0.43±0.07**	0.40±0.07	**0.44±0.06**

Best results are highlighted in bold

"‡" means that the gains of both versions of OLapDMM are statistically significant simultaneously (paired sample t-test at 0.01 level)

models are insensitive to ρ. That is, many efforts on refining ρ are not required, making our models more practical. Besides, we can see that smaller values of ρ perform a little better. In some sense, the learning rate ρ describes the importance degree of the variational manifold regularizer during model training. A smaller value of ρ is safer when we cannot accurately find the nearest neighbors of short texts. We thus suggest $\rho = 0.1$ as the default setting.

4.3 Evaluation of OLapDMM

In this section, we evaluate OLapDMM. For clarity, the versions of OLapDMM with document graph measured by term frequency and WMD are referred to as **OLapDMM**$_{\mathrm{T}}$ and **OLapDMM**$_{\mathrm{W}}$, respectively.

4.3.1 Comparison with online versions of baseline models

We compare OLapDMM on the large dataset of *Tweets*, which contains 10 million short texts. For fairness, we compare OLapDMM against online versions of baseline models with the optimization spirit of mini-batches, and refer to them as **ODMM**, **OGPU-DMM**, **OGPU-PDMM**, **OBTM** and **OLTM**, respectively. In terms of ODMM, it directly follows the optimization of stochastic collapsed variational inference (Foulds et al. 2013). In terms of other baselines using Gibbs sampling, their online versions follow the efficient inference methodology proposed in Yao et al. (2009). For all online versions of models, the mini-batch size M is set to 2^{10}.

Since the *Tweets* dataset is without any category label, we only evaluate OLapDMM across the topic quality task. The *TC* scores with $K = \{50, 100, 150, 200\}$ are shown in Table 7. Overall, we can see that our models perform the best in all cases, indicating the effectiveness of OLapDMM on collections of massive short texts. Several observations are made below.

- Both OLapDMM$_{\mathrm{T}}$ and OLapDMM$_{\mathrm{W}}$ consistently outperform ODMM on different topic numbers. The performance gains over ODMM tell us that the up-to-date docu-

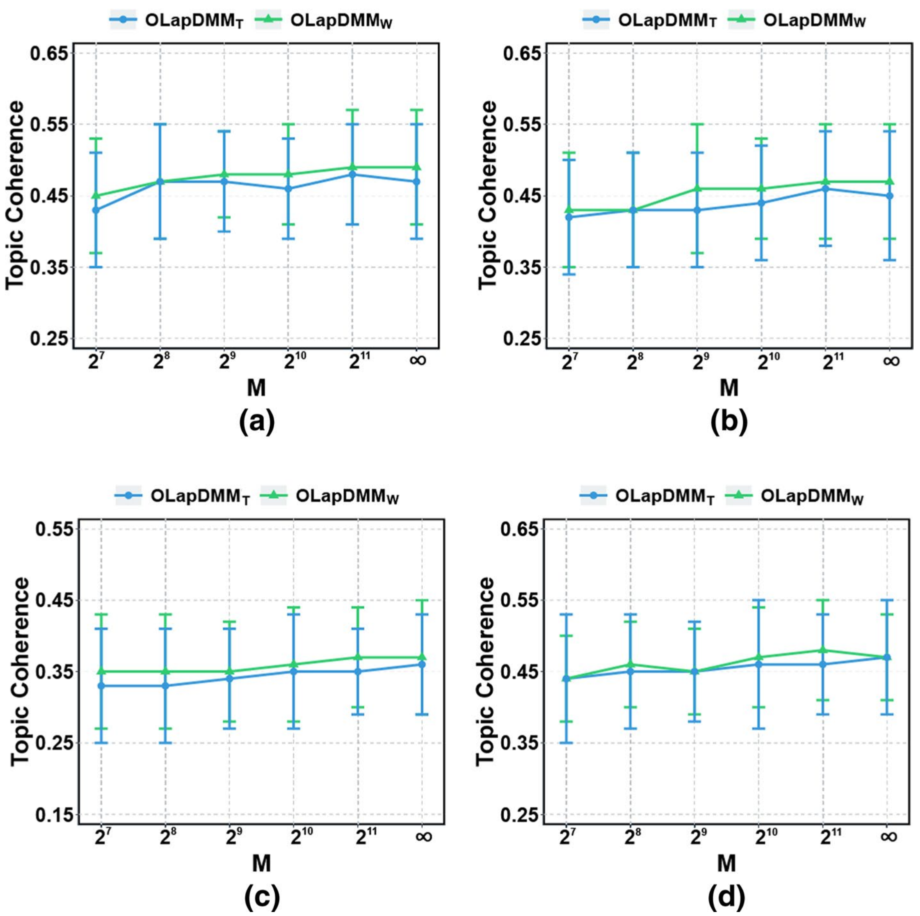

Fig. 5 Comparisons between OLapDMM and LapDMM across the topic coherence scores: **a** Trec, **b** Snippets, **c** SOF and **d** BaiduQA. Here, $M = \infty$ denotes **LapDMM**

ment graph can find high-quality approximate nearest neighbors for short texts, enabling to maintain the effectiveness of the variational manifold regularization.

- The *TC* scores of ODMM are higher than those of other baseline models, i.e., OGPU-DMM, OGPU-PDMM, OBTM and OLTM, in most cases. The possible reason is that ODMM is inferred by stochastic collapsed variational inference (Foulds et al. 2013) that may be more stable than the fast Gibbs sampling (Yao et al. 2009) used in other baseline models.
- OLapDMM$_\mathrm{W}$ can still learn more coherent topics, as well as achieving higher *TC* scores, than OLapDMM$_\mathrm{T}$. This further indicates that the document graph measured by the WMD can enhance the topic coherence, since it involves the semantic distances of short texts in some sense.

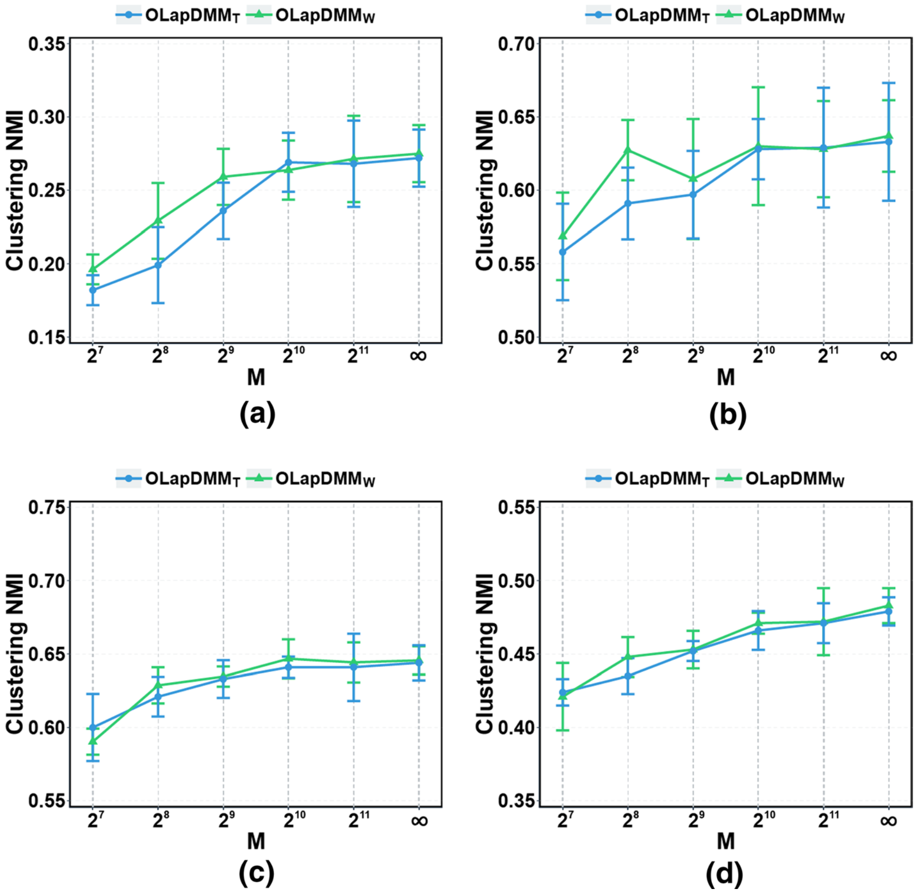

Fig. 6 Comparisons between OLapDMM and LapDMM across the clustering *NMI* scores: **a** Trec, **b** Snippets, **c** SOF and **d** BaiduQA. Here, $M = \infty$ denotes **LapDMM**

4.3.2 Comparison with batch LapDMM

We compare OLapDMM with the batch LapDMM on four small datasets, i.e., *Trec*, *Snippets*, *SOF* and *BaiduQA*, by topic coherence, clustering and classification tasks. For OLapDMM, we vary different values of M (i.e., the mini-batch size) over the set $\{2^7, 2^8, 2^9, 2^{10}, 2^{11}, \infty\}$. Specially, we notice that $M = \infty$ denotes LapDMM.

The results of TC ($K = 50$), clustering *NMI* and classification *AC* scores ($K = 60$) are plotted in Figs. 5, 6 and 7, respectively. Details of observations are outlined below.

- For clustering and classification tasks, the performance gap between OLapDMM and LapDMM is significant when the mini-batch size M is relatively small. For example, in terms of the *Trec* and *BaiduQA* datasets, the *AC* scores of LapDMM$_\mathrm{W}$ are about 0.1 (50%) higher than those of OLapDMM$_\mathrm{W}$ when $M = 2^7$ and 2^8. The reported results indicate that the discriminative power of document-level topical fea-

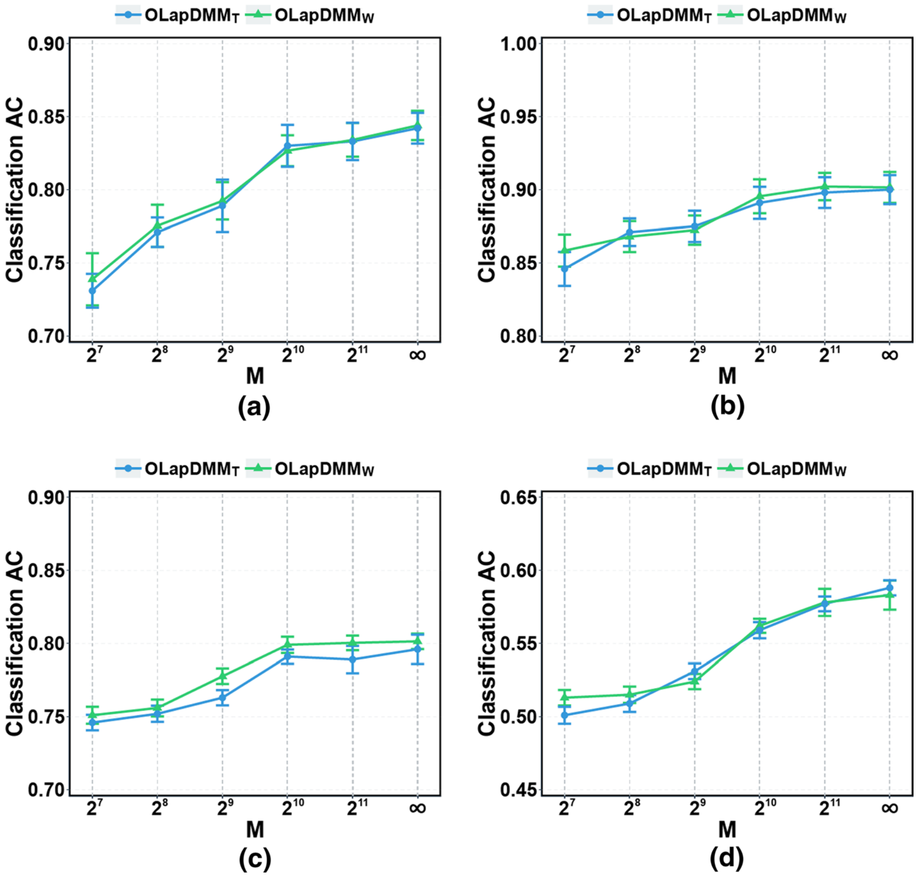

Fig. 7 Comparisons between OLapDMM and LapDMM across the classification *AC* scores: **a** Trec, **b** Snippets, **c** SOF and **d** BaiduQA. Here, $M = \infty$ denotes **LapDMM**

Table 8 Average per-iteration (i.e., outer iteration for LapDMM and OLapDMM) runtime (second) evaluation of OlapDMM and LapDMM across SOF (top section) and BaiduQA (bottom section)

Model	$M = 2^7$	$M = 2^8$	$M = 2^9$	$M = 2^{10}$	$M = 2^{11}$	$M = \infty$
OLapDMM$_T$	0.031	0.085	0.209	0.460	1.024	0.985
OLapDMM$_W$	0.041	0.112	0.253	0.518	1.112	0.985
OLapDMM$_T$	0.024	0.068	0.150	0.414	0.953	8.233
OLapDMM$_W$	0.031	0.081	0.185	0.450	0.996	8.233

Here, $M = \infty$ denotes LapDMM

tures is relatively sensitive to small mini-batch sizes. On the other hand, we observe that the performance trend of TC scores is stable even with small values of M. That implies OLapDMM enables to efficiently output coherent topics.

- The *NMI* and *AC* scores of OLapDMM consistently become higher as the size of mini-batch M increases. The result is reasonable since using larger values of M can

Table 9 Average per-iteration (i.e., outer iteration for OLapDMM) runtime (s) comparisons between OLapDMM and online versions of baseline methods across Tweets

Model	$M = 2^7$	$M = 2^8$	$M = 2^9$	$M = 2^{10}$	$M = 2^{11}$	$M = 2^{12}$
ODMM	0.002	0.003	0.006	0.013	0.022	0.045
OGPU-DMM	0.004	0.009	0.013	0.021	0.039	0.071
OGPU-PDMM	0.013	0.024	0.041	0.075	0.139	0.255
OBTM	0.006	0.013	0.028	0.055	0.101	0.193
OLTM	0.039	0.080	0.174	0.339	0.653	1.369
OLapDMM$_\mathrm{T}$	0.032	0.078	0.212	0.446	1.014	2.084
OLapDMM$_\mathrm{W}$	0.052	0.127	0.273	0.535	1.091	2.313

accurately update the global variable, and simultaneously find more precise nearest neighbors, being beneficial for the variational manifold regularization.

- OLapDMM becomes even competitive with LapDMM given larger mini-batches. For example, we can observe that the *NMI* scores of OLapDMM are very close to those of LapDMM when $M = 2^{10}$ and 2^{11}. Therefore, we argue that OLapDMM can be a promising candidate to LapDMM for real-world applications, where fast inference is required.

4.3.3 Efficiency evaluation

In this subsection, we compare the runtime between OLapDMM, batch LapDMM and online versions of baseline methods. We run each method until 10000 (outer) iterations when $K = 20$, and present the average per-iteration (i.e., outer iteration for LapDMM and OLapDMM) runtime (in seconds).

- As shown in Table 8, we present the runtimes of OLapDMM and LapDMM[15] on two smaller datasets of *SOF* and *BaiduQA*. In contrast to LapDMM, as expected we observe that OLapDMM performs more efficient when the mini-batch size (i.e., M) is relatively small. On the dataset of *BaiduQA*, OLapDMM is about 20 and 10 faster than LapDMM when $M = 2^{10}$ and 2^{11}, while achieving very competitive performance on clustering and classification tasks as the results shown in Sect. 4.3.2. On the dataset of *SOF*, the runtime of OLapDMM is a bit higher than that of LapDMM, since in LapDMM we directly utilize the pre-computed document graph. Additionally, we can observe that OLapDMM$_\mathrm{W}$ is always less efficient than OLapDMM$_\mathrm{T}$. That is because computing WMD between short texts is more computationally expensive.
- Then, we compare OLapDMM with online versions of baselines on *Tweets*, where results are presented in Table 9. Overall speaking, we can observe that all those online versions are computationally efficient; our OLapDMMs spend more runtimes for each iteration but they are still practical in real applications. Again, we observe that OLapDMM$_\mathrm{W}$ is less efficient than OLapDMM$_\mathrm{T}$ because of the computationally expensive WMD in approximate nearest neighbor search and document graph update.

[15] For the versions of LapDMM, we don't consider the runtime of the offline document graph construction.

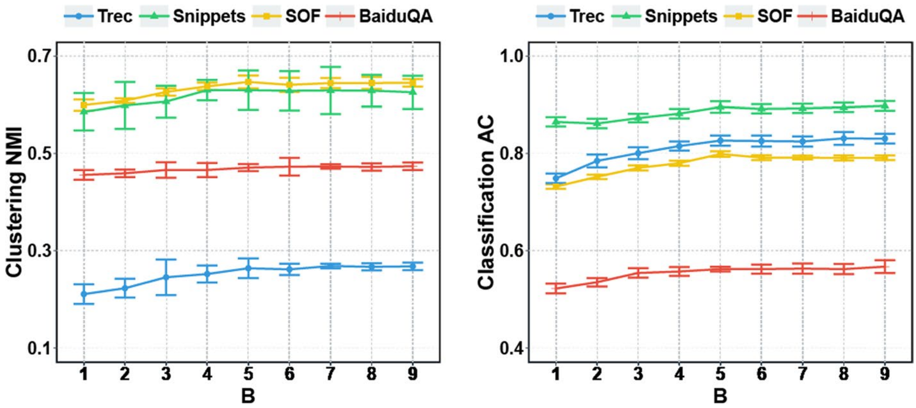

Fig. 8 Evaluation results of different B values of OLapDMM_W

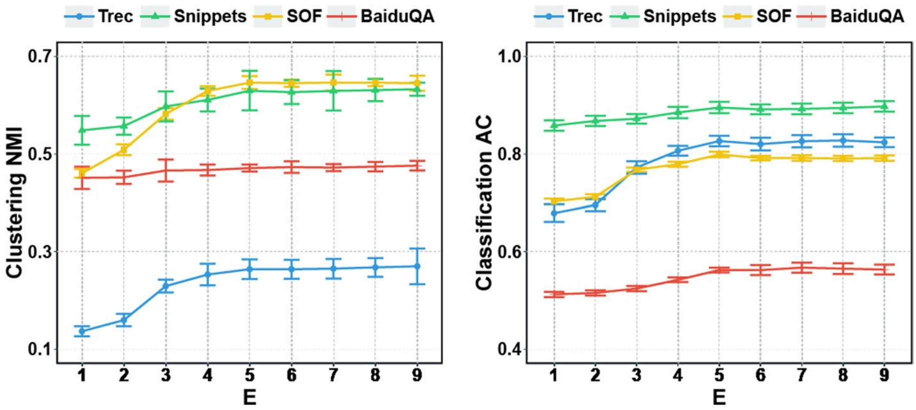

Fig. 9 Evaluation results of different E values of OLapDMM_W

4.3.4 Parameter analysis of B and E

In addition, we empirically evaluate the two parameters of B and E in the up-to-date document graph construction of OLapDMM. We also examine them using clustering *NMI* and classification *AC* scores (i.e., when $K = 60$) across four small datasets, i.e., *Trec*, *Snippets, SOF* and *BaiduQA*. In this evaluation, the mini-batch size M is fixed to 2^{10}. Since the performance trends of different versions of OLapDMM are similar, we only present the results of OLapDMM_W.

For each of the two parameters, we vary its value over the set $\{1, 2, \ldots, 9\}$ by fixing the other one as 5. The experimental results are plotted in Figs. 8 and 9, respectively. Broadly speaking, we can observe that the performance trends of B and E are similar, and simultaneously straightforward to understand. Reviewing **Algorithm 2**, the parameters control the repeated number as well as iterative number for searching approximate nearest neighbors, therefore larger values naturally tend to achieve more accurate search results, being benefit for

OLapDMM$_W$. The observable result is consistent with the analysis, where on all the datasets, the *NMI* and *AC* scores of OLapDMM$_W$ roughly become higher as the values of B and E increase. In contrast, the parameter E is more sensitive to smaller values, e.g., the performance deteriorates when $E = 1, 2$. The possible reason is that given a random initialization, it is intractable to compute accurate approximate nearest neighbors with fewer iteration numbers, i.e., search steps, from the current document graph. Since the performance tends to be stable when $B, E \geq 5$, we fix them to 5 in our experiments.

5 Conclusion and future work

In this article, we investigate how to effectively learn topics from short texts that are extremely sparse. To this end, we extend DMM by incorporating the variational manifold regularization into its variational objective, leading to a novel topic model, namely LapDMM. The manifold constraints can link nearest short texts, so as to spread topical signals among them. We exploit term frequency and WMD to construct the document graph that stores the nearest neighbors of short texts, where the WMD can measure the semantic distances between short texts. To handle collections of massive short texts, we develop an online version of LapDMM, namely OLapDMM, with the spirit of stochastic optimization with mini-batches. Carrying this implications, we exploit an up-to-date document graph, which can efficiently find approximate nearest neighbors of short texts. Extensive experiments are conducted to evaluate LapDMM and OLapDMM on real-world datasets, which demonstrate that LapDMM significantly outperforms the state-of-the-art baselines on the tasks of topic quality, document clustering and classification across small datasets, and OLapDMM works well on collections of massive short texts.

In real-world applications, the huge volume of short texts as well as short text streams, e.g., social media posts, is still a primary problem for knowledge mining from short texts using topic modeling or any other methodology. Our OLapDMM can be applied as an alternative solution for efficiently modeling massive short texts or even stream data, however, it is still a "basic" model without touching the rich available side information, e.g., time stamp, rates with item comments, citation relationships of paper titles, *etc.* For example, in social media, emerging bursty topics related to some important events or issues often appear by certain time slices (Diao et al. 2012; Yan et al. 2015); texts, such as research papers (i.e., paper titles), refer to citation relationships, giving text document networks (Chang and Blei 2009; Zhang et al. 2013). In the future works, we are going to pay more attention on investigations with side information of short texts as well as further problems and topics caused by them.

Appendix

In this "Appendix", we derive the optimal form of variational distribution $q(z|\gamma)$, i.e., Eq. 7, in detail. Toward this goal, we first describe the joint distribution of a short text collection S and topic assignments z with Dirichlet priors α and β:

$$
\begin{aligned}
p(S,z|\alpha,\beta) &= \int\int p(S,\boldsymbol{z},\boldsymbol{\theta},\boldsymbol{\phi}|\alpha,\beta)d\boldsymbol{\theta}d\boldsymbol{\phi} \\
&= \int\int \mathbf{Dir}(\theta|\alpha)\prod_{k=1}^{K}\mathbf{Dir}\left(\boldsymbol{\phi}_k|\beta\right)\prod_{d=1}^{D}p\left(z_d|\boldsymbol{\theta}\right)\prod_{n=1}^{N_d}\boldsymbol{\phi}_{z_d w_{dn}}d\boldsymbol{\theta}d\boldsymbol{\phi} \\
&= \int\int \mathbf{Dir}(\theta|\alpha)\prod_{k=1}^{K}\theta_k^{\widehat{N}_k}\prod_{k=1}^{K}\mathbf{Dir}\left(\boldsymbol{\phi}_k|\beta\right)\prod_{v=1}^{V}\boldsymbol{\phi}_{kv}^{N_{kv}}d\boldsymbol{\theta}d\boldsymbol{\phi} \\
&= \left(\frac{\prod_{k=1}^{K}\Gamma\left(\widehat{N}_k+\alpha\right)}{\Gamma(D+K\alpha)}\frac{\Gamma(K\alpha)}{\prod_{k=1}^{K}\Gamma(\alpha)}\right)\left(\prod_{k=1}^{K}\frac{\prod_{v=1}^{V}\Gamma\left(N_{kv}+\beta\right)}{\Gamma\left(N_k+V\beta\right)}\frac{\Gamma(V\beta)}{\prod_{v=1}^{V}\Gamma(\beta)}\right) \\
&\propto \left(\frac{\prod_{k=1}^{K}\Gamma\left(\widehat{N}_k+\alpha\right)}{\Gamma(D+K\alpha)}\right)\left(\prod_{k=1}^{K}\frac{\prod_{v=1}^{V}\Gamma\left(N_{kv}+\beta\right)}{\Gamma\left(N_k+V\beta\right)}\right)\times\mathbf{const},
\end{aligned}
\tag{21}
$$

where $\Gamma(\cdot)$ denotes the Gamma function; $\widehat{N}_k$ is the number of documents assigned to topic k; N_{kv} and N_k are the number of word v assigned to topic k and total number of words assigned to topic k, respectively.

The Optimum of Variational Distribution $q(\boldsymbol{z}|\boldsymbol{\gamma})$: Since a mean-filed form of $q(\boldsymbol{z}|\boldsymbol{\gamma})$ (ref. Eq. 3) is used, we can independently optimize the variational distribution of each document d, i.e., $q(z_d|\boldsymbol{\gamma}_d)$. Following Bishop (2006), its optimum $q(z_d|\boldsymbol{\gamma}_d)$, holding all other variational distributions fixed, can be presented by:

$$
\begin{aligned}
q(z_d|\boldsymbol{\gamma}_d) &\propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\log p(S,z|\alpha,\beta)\right]\right) \\
&\propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\log p(S,z|\alpha,\beta)-\log p(S^{\neg d},z^{\neg d}|\alpha,\beta)\right]\right) \\
&\propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\log\frac{p(S,z|\alpha,\beta)}{p(S^{\neg d},z^{\neg d}|\alpha,\beta)}\right]\right)
\end{aligned}
\tag{22}
$$

where the superscript "$\neg d$" means the corresponding variables and counts with document d excluded. Additionally, the second line of Eq. 22 holds, because the expectation $\mathbb{E}_{q^{\neg d}(z|\boldsymbol{\gamma})}\left[\log p(S^{\neg d},z^{\neg d}|\alpha,\beta)\right]$ is a constant by fixing $q^{\neg d}(\boldsymbol{z}|\boldsymbol{\gamma})$. This is inspired by collapsed Gibbs sampling (Griffiths and Steyvers 2004), enabling to simplify the computation.

By combing Eq. 21 with Eq. 22, we reach the optimal variational probability of document d assigned to topic k (i.e., $\boldsymbol{\gamma}_{dk}$):

$$\begin{aligned}
\gamma_{dk} &\overset{\Delta}{=} q(z_d = k) \propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\gamma)}\left[\log \frac{p(S, z|\alpha, \beta)}{p(S^{\neg d}, z^{\neg d}|\alpha, \beta)}\right]\right) \\
&\propto \exp\left(\mathbb{E}_{q^{\neg d}(z|\gamma)}\left[\log\left(\frac{\Gamma\left(\widehat{N}_k + \alpha\right)}{\Gamma\left(\widehat{N}_k^{\neg d} + \alpha\right)} \frac{\prod_{v=1}^{V} \Gamma\left(N_{kv} + \beta\right)}{\prod_{v=1}^{V} \Gamma\left(N_{kv}^{\neg d} + \beta\right)} \frac{\Gamma\left(N_k^{\neg d} + V\beta\right)}{\Gamma\left(N_k + V\beta\right)}\right)\right]\right) \\
&= \exp\left(\mathbb{E}_{q^{\neg d}(z|\gamma)}\left[\log\left(\widehat{N}_k^{\neg d} + \alpha\right) + \sum_{v \in d} \sum_{n=1}^{N_{dv}} \log\left(N_{kv}^{\neg d} + \beta + n - 1\right)\right.\right. \\
&\qquad \left.\left. - \sum_{n=1}^{N_d} \log\left(N_k^{\neg d} + V\beta + n - 1\right)\right]\right),
\end{aligned} \tag{23}$$

where N_{dv} is the number of times word v occurring in document d. The third line of Eq. 23 follows the fact $(m > n)$:

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

Acknowledgements This research was partially supported by National Key R&D Program of China (2018YFB0804202) and the National Natural Science Foundation of China (NSFC) [Nos. 61876071, 61806035, U1936217] (Grant Nos. 61602204, 61732008, 61725203, U1936217).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

Asuncion, A., Welling, M., Smyth, P., & Teh, Y. W. (2009). On smoothing and inference for topic models. In *Conference on uncertainty in artificial intelligence* (pp. 27–34).

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Blei, D. M. (2012). Probabilistic topic models. *Communications of the ACM, 55*(4), 77–84.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research, 3,* 993–1022.

Cai, D., Mei, Q., Han, J., & Zhai, C. (2008). Modeling hidden topics on document manifold. In *ACM conference on information and knowledge management* (pp. 911–920).

Cai, D., Wang, X., & He, X. (2009). Probabilistic dyadic data analysis with local and global consistency. In *International conference on machine learning* (pp. 105–112).

Chang, J., & Blei, D. M. (2009). Relational topic models for document networks. In *International conference on artificial intelligence and statistics* (pp 81–88).

Cheng, X., Yan, X., Lan, Y., & Guo, J. (2014). BTM: Topic modeling over short texts. *IEEE Transactions on Knowledge and Data Engineering, 26*(12), 2928–2941.

Chi, J., Ouyang, J., Li, X., & Li, C. (2018). Empirical study on variational inference methods for topic models. *Journal of Experimental & Theoretical Artificial Intelligence, 30*(1), 129–142.

Cuturi, M. (2013). Sinkhorn distances: Lightspeed computation of optimal transport. In *Neural information processing systems* (pp. 2292–2300).

Diao, Q., Jiang, J., Zhu, F., & Lim, E. P. (2012). Finding bursty topics from microblogs. In *Annual meeting of the association for computational linguistics* (pp. 536–544).

Du, J., Jiang, J., Song, D., & Liao, L. (2015). Topic modeling with document relative similarities. In *International joint conference on artificial intelligence* (pp. 3469–3475).

Foulds, J., Boyles, L., DuBois, C., Smyth, P., & Welling, M. (2013). stochastic collapsed variational bayesian inference for latent Dirichlet allocation. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 446–454).

Griffiths, T. L., & Steyvers, M. (2004). Finding scientific topics. *National Academy of Sciences of the United States of America, 101*(Suppl. 1), 5228–5235.

Hajebi, K., Abbasi-Yadkori, Y., Shahbazi, H., & Zhang, H. (2011). Fast approximate nearest-neighbor search with *k*-nearest neighbor graph. In *International joint conference on artificial intelligence* (pp. 1312–1317).

Hoffman, M. D., Blei, D. M., & Bach, F. (2010) Online learning for latent Dirichlet allocation. In *Neural information processing systems* (pp. 856–864).

Hoffman, M. D., Blei, D. M., Wang, C., & Paisley, J. (2013). Stochastic variational inference. *Journal of Machine Learning Research, 3,* 1303–1347.

Hofmann, T. (1999). Probabilistic latent semantic indexing. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 50–57).

Hong, L., & Davison, B. D. (2010). Empirical study of topic modeling in Twitter. In: *Proceedings of the first workshop on social media analytics* (pp. 80–88).

Hu, W., Zhu, J., Su, H., Zhuo, J., & Zhang, B. (2017). Semi-supervised max-margin topic model with manifold posterior regularization. In *International joint conference on artificial intelligence* (pp. 1865–1871).

Huh, S., & Fienberg, S. E. (2010). Discriminative topic modeling based on manifold learning. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 653–662).

Huh, S., & Fienberg, S. E. (2012). Discriminative topic modeling based on manifold learning. *ACM Transactions on Knowledge Discovery from Data, 5*(4), 20.

Kusner, M. J., Sun, Y., Kolkin, N. I., & Weinberger, K. Q. (2015). From word embeddings to document distances. In *International conference on machine learning* (pp. 957–966).

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*(6755), 788–791.

Li, C., Chen, S., Xing, J., Sun, A., & Ma, Z. (2018a). Seed-guided topic model for document filtering and classification. *ACM Transactions on Information Systems, 37*(1), 1–37.

Li, C., Duan, Y., Wang, N., Zhang, Z., Sun, A., & Ma, Z. (2017). Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems, 36*(2), 1–30.

Li, C., Li, X., Ouyang, J., & Wang, Y. (2020). Semantics-assisted Wasserstein learning for topic and word embeddings. In *IEEE international conference on data mining*.

Li, C., Wang, H., Zhang, Z., Sun, A., & Ma, Z. (2016a). Topic modeling for short texts with auxiliary word embeddings. In *International ACM SIGIR conference on research and development in information retrieval* (pp. 165–174).

Li, S., Zhao, Z., Hu, R., Li, W., Liu, T., & Du, X. (2018b). Analogical reasoning on Chinese morphological and semantic relations. In *Annual meeting of the association for computational linguistics* (pp. 138–143).

Li, X., Li, C., Chi, J., & Ouyang, J. (2018c). Short text topic modeling by exploring original documents. *Knowledge and Information Systems, 56*(2), 443–462.

Li, X., Li, C., Chi, J., Ouyang, J., & Li, C. (2018d). Dataless text classification: A topic modeling approach with document manifold. In *ACM international conference on information and knowledge management* (pp. 973–982).

Li, X., Ouyang, J., & Zhou, X. (2016b). Sparse hybrid variational-gibbs algorithm for latent Dirichlet allocation. In *SIAM international conference on data mining* (pp. 729–737).

Li, X., Wang, Y., Zhang, A., Li, C., Chi, J., & Ouyang, J. (2018e). Filtering out the noise in short text topic modeling. *Information Sciences, 456,* 83–96.

Li, X., Zhang, A., Li, C., Guo, L., Wang, W., & Ouyang, J. (2019a). Relational biterm topic model: Short text topic modeling using word embeddings. *The Computer Journal, 62*(3), 359–372.

Li, X., Zhang, J., & Ouyang, J. (2019b). Dirichlet multinomial mixture with variational manifold regularization: Topic modeling over short texts. In *AAAI conference on artificial intelligence*.

Liang, S., Ren, Z., Zhao, Y., Ma, J., Yilmaz, E., & Rijke, M. D. (2017a). Inferring dynamic user interests in streams of short texts for user clustering. *ACM Transactions on Information Systems, 36*(1), 1–37.

Liang, S., Yilmaz, E., Shen, H., Rijke, M. D., & Croft, W. B. (2017b). Search result diversification in short text streams. *ACM Transactions on Information Systems, 36*(1), 1–35.

Lu, H. Y., Xie, L. Y., Kang, N., Wang, C. J., & Xie, J. Y. (2017). Don't forget the quantifiable relationship between words: Using recurrent neural network for short text topic discovery. In: *AAAI conference on artificial intelligence* (pp. 1192–1198).

Mehrotra, R., Sanner, S., Buntine, W., & Xie, L. (2013). Improving LDA topic models for microblogs via Tweet pooling and automatic labeling. In: *International ACM SIGIR conference on research and development in information retrieval* (pp. 889–892).

Mei, Q., Cai, D., Zhang, D., & Zhai, C. (2008). Topic modeling with network regularization. In *International conference on World Wide Web* (pp. 101–110).

Mikolov, T., Tau Yih, W., & Zweig, G. (2013). Linguistic regularities in continuous space word representations. In *Annual conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 2292–2300).

Newman, D., Karimi, S., & Cavedon, L. (2010a). External evaluation of topic models. In: *Australasian document computing symposium* (pp. 11–18).

Newman, D., Lau, J. H., Grieser, K., & Baldwin, T. (2010b). Automatic evaluation of topic coherence. In *Annual conference of the North American chapter of the association for computational linguistics* (pp. 100–108).

Nigam, K., Mccallum, A. K., Thrun, S., & Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning, 39*(2), 103–134.

Phan, X. H., Nguyen, L. M., & Horiguchi, S. (2008). Learning to classify short and sparse text & web with hidden topics from large-scale data collections. In *International conference on World Wide Web* (pp. 91–100).

Quan, X., Kit, C., Ge, Y., & Pan, S. J. (2015). Short and sparse text topic modeling via self-aggregation. In *International joint conference on artificial intelligene* (pp. 2270–2276).

Roder, M., Both, A., & Hinneburg, A. (2015). Exploring the space of topic coherence measures. In *International conference on web search and data mining* (pp. 399–408).

Sato, I., & Nakagawa, H. (2012). Rethinking collapsed variational Bayes inference for LDA. In *International conference on machine learning*.

Shi, T., Kang, K., Choo, J., & Reddy, C. K. (2018). Short-text topic modeling via non-negative matrix factorization enriched with local word-context correlations. In *International world wide web conference* (pp. 1105–1114).

Sridhar, V. K. R. (2015). Unsupervised topic modeling for short texts using distributed representations of words. In *Annual conference of the North American chapter of the association for computational linguistics: Human language technologies* (pp. 192–200).

Teh, Y. W., Newman, D., & Welling, M. (2006). A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation. In *Advances in neural information processing systems* (pp. 1353–1360).

Wang, M., Fu, W., Hao, S., Tao, D., & Wu, X. (2016a). Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, M., Liu, X., & Wu, X. (2015a). Visual classification by l1-hypergraph modeling. *IEEE Transactions on Knowledge and Data Engineering*.

Wang, X., & McCallum, A. (2006). Topics over time: A non-Markov continuous-time model of topical trends. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 424–433).

Wang, Y., Lin, X., Wu, L., & Zhang, W. (2015b). Effective multi-query expansions: Robust landmark retrieval. In *ACM multimedia* (pp. 79–88).

Wang, Y., Lin, X., Wu, L., & Zhang, W. (2017). Effective multi-query expansions: Collaborative deep networks for robust landmark retrieval. *IEEE Transactions on Image Processing, 26*(3), 1393–1404.

Wang, Y., Lin, X., Wu, L., Zhang, W., Zhang, Q., & Huang, X. (2015c). Robust subspace clustering for multi-view data by exploiting correlation consensus. *IEEE Transactions on Image Processing, 24*(11), 3939–3949.

Wang, Y., Wu, L., Lin, X., & Gao, J. (2018). Multiview spectral clustering via structured low-rank matrix factorization. *IEEE Transactions on Neural Networks and Learning Systems, 29*(10), 4833–4843.

Wang, Y., Zhang, W., Wu, L., Lin, X., Fang, M., & Pan, S. (2016b). Iterative views agreement: An iterative low-rank based structured optimization method to multi-view spectral clustering. In *IJCAI* (pp. 2153–2159).

Weng, J., Lim, E. P., Jiang, J., & He, Q. (2010). Twitterrank: Finding topic-sensitive influential Twitters. In *International conference on web search and data mining* (pp. 261–270).

Xin, W., Jiang, Z., Shu, J., He, W., Lim, E. P., Yan, H., & Li, X. (2011). Comparing twitter and traditional media using topic models. In *European conference on information retrieval* (pp. 338–349).

Xu, J., Xu, B., Wang, P., Zheng, S., Tian, G., Zhao, J., & Xua, B. (2017). Self-taught convolutional neural networks for short text clustering. *Neural Networks, 88,* 22–31.

Yan, X., Guo, J., Lan, Y., & Cheng, X. (2013). A biterm topic model for short texts. In *International conference on world wide web* (pp. 1445–1456).

Yan, X., Guo, J., Lan, Y., Xu, J., & Cheng, X. (2015). A probabilistic model for bursty topic discovery in microblogs. In *AAAI conference on artificial intelligence*.

Yan, X., Guo, J., Liu, S., Cheng, X., & Wang, Y. (2013). Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *SIAM international conference on data mining* (pp. 749–757).

Yao, L., Mimno, D., & McCallum, A. (2009). Efficient methods for topic model inference on streaming document collections. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 937–946).

Yin, J., & Wang, J. (2014). A Dirichlet multinomial mixture model-based approach for short text clustering. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 233–242).

Zhang, A., Zhu, J., & Zhang, B. (2013). Sparse relational topic models for document networks. In *European conference on machine learning and knowledge discovery in databases* (pp. 670–685).

Zhang, M. L., & Zhou, Z. H. (2014). A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering, 26*(8), 1819–1837.

Zhao, W. X., Zhang, W., He, Y., Xie, X., & Wen, J. R. (2018). Automatically learning topics and difficulty levels of problems in online judge systems. *ACM Transactions on Information Systems, 36*(3), 1–33.

Zuo, Y., Wu, J., Zhang, H., Lin, H., Wang, F., Xu, K., & Xiong, H. (2016a). Topic modeling of short texts: A pseudo-document view. In *ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2105–2114).

Zuo, Y., Zhao, J., & Xu, K. (2016b). Word network topic model: A simple but general solution for short and imbalanced texts. *Knowledge and Information Systems, 48*(2), 379–398.

Authors and Affiliations

Ximing Li[1,2] · Yang Wang[3,4] · Jihong Ouyang[1,2] · Meng Wang[3,4]

✉ Yang Wang
yangwang@hfut.edu.cn

Ximing Li
liximing86@gmail.com

Jihong Ouyang
ouyj@jlu.edu.cn

Meng Wang
eric.mengwang@gmail.com

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

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

[3] School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China

[4] Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei, China