International Journal of Approximate Reasoning 112 (2019) 85–104

Contents lists available at ScienceDirect

International Journal of Approximate Reasoning

www.elsevier.com/locate/ijar

Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout

Cuong Ha [a], Van-Dang Tran [b,c], Linh Ngo Van [a], Khoat Than [a,*]

[a] Hanoi University of Science and Technology, No. 1, Dai Co Viet road, Hanoi, Viet Nam
[b] Graduate University for Advanced Studies, Kanagawa, Japan
[c] National Institute for Informatics, Tokyo, Japan

ARTICLE INFO

Article history:
Received 1 October 2018
Received in revised form 25 May 2019
Accepted 28 May 2019
Available online 3 June 2019

Keywords:
Topic models
Short text
Dropout

ABSTRACT

Probabilistic topic models are powerful tools for discovering hidden structures/semantics in discrete data, e.g., texts, images, links. However, on short and noisy texts, directly applying topic models may not work well or face severe overfitting. In this article, we investigate the benefits of dropout for preventing topic models from overfitting. We integrate dropout into several stochastic methods for learning latent Dirichlet allocation (LDA). From extensive experiments on four large-scale datasets, our findings are: (1) dropout helps to prevent overfitting and significantly enhance predictiveness and generalization of LDA on short texts; (2) for long documents, dropout may provide little benefit; (3) dropout can be easily integrated into any learning methods to avoid overfitting for short and noisy text. Furthermore, dropout can be straightforwardly employed in a wide range of topic models. In evidence, we apply dropout to BTM (Biterm topic model), one of the state-of-the-art models for short texts. Our experiments illustrate that BTM with dropout not only remains its good results in term of predictiveness, but also improves the learning time significantly.

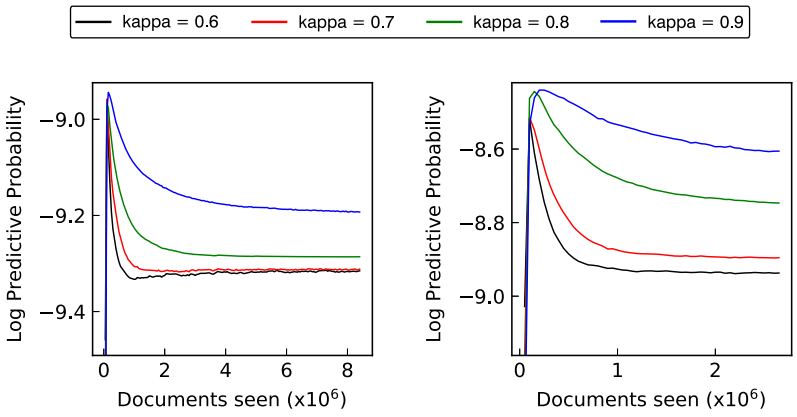
1. Introduction

In the Internet era, billions of people are using the Internet and its services such as websites, social networks, microblogs, instant messages, question & answer forums, news, etc. With the rise in the popularization of users, the number of texts that users generate has been exponentially increasing. For example, according to recent statistics, the number of tweets posted per day by Twitter users is about 500 million. Such data of Internet users are usually in the form of very short texts, but contain rich and useful information. Discovering the topics of the large-scale user-generated short texts from web content or social media is a challenging and promising problem.

Probabilistic topic models have been proven to be effective tools for uncovering the hidden topics of textual corpora. They are commonly based on the assumption that a document is a mixture of topics, where a topic is a probability distribution over the vocabulary. While topic models such as Latent Dirichlet Allocation (LDA) [4] or Hierarchical Dirichlet Processes (HDP) [21] have broad success on news articles and academic papers; they often suffer from bad performance on short texts. Unlike long documents (e.g. carefully edited articles, academic papers), short texts from online social networks are often characterized by a very short length, a large vocabulary, a massive size, and noises. Short and noisy data poses severe

* Corresponding author.
E-mail addresses: nhatcuong94@gmail.com (C. Ha), dangtv@nii.ac.jp (V.-D. Tran), linhnv@soict.hust.edu.vn (L. Ngo Van), khoattq@soict.hust.edu.vn (K. Than).

https://doi.org/10.1016/j.ijar.2019.05.010
0888-613X/



(a) NYT-titles (b) Yahoo

Fig. 1. Overfitting of LDA on short texts: *NYT-titles* which contains news' titles from www.nytimes.com, and *Yahoo* which contains questions from answers.yahoo.com. Log Predictive Probability shows the generalization of a model on unseen data. Higher is better. The curves show the performance of stochastic variational inference [11] for learning LDA, with different choices of the learning rate (κ). (For ease of interpretation, the reader is referred to the colored version of this article.)

challenges for modeling, and thus traditional methods do not work well on short texts. Fig. 1 shows the predictiveness of models learned by LDA on two short text datasets, where stochastic variational inference [11] with different learning rates (κ) is used. LDA suffers overfitting that all the learning curves initially go up but quickly go down as learning from more documents. Those observations have been analyzed in both theory [19] and practice [15,8,13,27,14,6].

A very well-known technique for preventing overfitting is *Dropout* [10,18]. The idea of dropout is straightforward. When training a model, we stochastically drop out some parts of the model in each iteration of the training process. Therefore, instead of only one model, an exponential number of models are trained during the training time and then combined into a single model at test time. It is very powerful for preventing overfitting because it plays as regularization [26]. Dropout works well for various machine learning methods including support vector machine (SVM) [5] and neural networks [18]. Unfortunately, to the best of our knowledge, there is no extensive study about the benefits of dropout for topic models.

In this work, we fulfill that gap to answer the following questions: *Does dropout provide any benefit to topic models? In which situations do dropout help topic models from overfitting?* We take LDA into investigation as it is the core of various models. Four large-scale datasets and five stochastic methods for learning LDA are used for our investigation. From extensive experiments, we found several useful findings:

- Naive use of dropout provides little benefit on long documents.
- However, dropout appears powerful in short text. In particular, dropout can help many learning methods to considerably improve performance on short texts. The improvements sometimes are with a large margin. Overfitting can be easily eliminated.
- Applying dropout to specific topic models, which works well on short texts, does not affect much to the result but improves the learning time significantly.

Although we took LDA into investigations, we believe that those findings are crucial in practice. The reasons are that dropout can be easily employed in a wide range of learning methods for Bayesian models, and that LDA is the core for a large family of topic models [3]. As an illustration, we apply dropout to *Biterm topic model* (BTM) [27,6], one of the state-of-the-art models for short texts, to provide stronger evidences for our findings above. As a side contribution, this work provides a new class of stochastic methods, empowered by dropout, for efficiently learning topic models from short and noisy texts.

The rest of this article is organized as follows. In Section 2, we discuss some related studies. In Section 3, we present some backgrounds briefly. Section 4 presents our dropout approach. We present our experiments in Section 5 with results and observations to demonstrate the benefits of dropout in practice. Finally, Section 6 concludes and figures out some future directions.

2. Related work

Recently, a lot of efforts have been devoted to overcoming challenges in analyzing short texts. On the kind of data, existing models for normal documents are usually not efficient. There have been many approaches to make these models work better for short texts. They can be divided into the following groups:

The first group of researches focuses on reducing the impact of shortness in short texts. Some works [15,8,13] group tweets with the same attribute (hashtags, users) into longer documents before applying LDA for these new documents. In another direction, some researchers try to use additional information from external sources [17,2] or contextualize [20] the short texts. However, these methods require the availability of external sources. Yan et al. [27,6] directly model word co-occurrence pattern by generating pairs of words for each document (called biterm) and then aggregate all the biterms from all documents into one collection. Mai et al. [14] use bag-of-biterms (BoB) to represent documents in order to improve its length, hence BoB makes topic models work better. However, when applying BoB for longer texts or a mixture of short and long texts, the complexity of the model will increase exponentially.

In the second group, researchers proposed novel learning methods for topic models to deal with big data that is a serious problem when the number of short texts has been exponentially increasing. Various methods for inference have been proposed such as variational Bayes (VB) [4], collapsed variational Bayes (CVB) [22], collapsed Gibbs sampling (CGS) [16], Online Maximum a Posteriori Estimation (OPE) [23] and Frank-Wolfe (FW) [24,25]. These approaches enable us to easily work with millions of texts and guarantee on quality or convergence rate. However, the model training with these methods needs a novel regularization technique to make them work efficiently on short texts.

Recently, dropout has been known as an emerging regularization technique used widely in deep learning. It was firstly introduced by Hinton et al. [10] as a way to control overfitting by randomly omitting subsets of features in the training procedure. The authors in [26] analyzed dropout training as a form of adaptive regularization and proved the close connection between dropout training and adaptively balanced L2-regularization. Srivastava et al. [13] show the effectiveness of the dropout technique applying for neural network on supervised learning tasks in vision, speech recognition, document classification and computational biology. The technique achieves the same phenomenon when applying for Support Vector Machines [5]. This idea of dropout can be also extended to Restricted Boltzmann Machines.

3. Background

In this section, we briefly recap some basic backgrounds about topic models and the dropout technique used in neural network. The following notations will be used throughout this article.

$\mathcal{V}$	A vocabulary of V terms, often written as $\{1, 2, \cdots, V\}$.
d	A document represented as a count vector, $d = (d_1, \cdots, d_V)$, where d_v is the frequency of term v.
β_k	A topic which is a distribution over the vocabulary $\mathcal{V}$. $\beta_k = (\beta_{k1}, \cdots, \beta_{kV})^t$, $\beta_{kv} \geq 0$, $\sum_{v=1}^{V} \beta_{kv} = 1$.
λ_{kv}	The variational parameter showing the contribution of term v to topic k.
ϕ_{vk}	The variational parameter showing the probability that term v is generated from topic k.
γ_k	The variational parameter showing the expected contribution of topic k.
$\psi(.)$	The digamma function.
Δ_K	The unit simplex $\Delta_K = \{x \in \mathbb{R}^K : \|x\|_1 = 1, x \geq 0\}$.

3.1. Topic models and stochastic learning

A topic model such as LDA often assumes that a corpus is composed of K topics, $\beta = (\beta_1, \beta_2, \ldots, \beta_K)$. Each document $\boldsymbol{d}$ is a mixture of those topics and is assumed to arise from the following generative process:

For the nth word of $\boldsymbol{d}$:

- Draw topic index $z_{dn}|\boldsymbol{\theta}_d \sim Multinomial(\boldsymbol{\theta}_d)$
- Draw word $w_{dn}|z_{dn}, \beta \sim Multinomial(\beta_{z_{dn}})$.

Each topic mixture $\boldsymbol{\theta}_d = (\boldsymbol{\theta}_{d1}, \ldots, \boldsymbol{\theta}_{dK})$ represents the contributions of topics to document $\boldsymbol{d}$, while β_{kj} shows the contribution of term j to topic k. Note that $\boldsymbol{\theta}_d \in \Delta_K, \beta_k \in \Delta_V, \forall k$. Both $\boldsymbol{\theta}_d$ and z_d are latent variables and are local for each document.

The generative process above generally describes a simple topic model, i.e., *Probabilistic latent semantic analysis* (PLSA) [12]. *Latent Dirichlet allocation* (LDA) [4] extensively assumes that θ and β are generated by Dirichlet distributions. More specifically, $\theta \sim Dirichlet(\alpha)$ and $\beta_k \sim Dirichlet(\eta)$ where α and η are hyper-parameters. Fig. 2 presents the graphical model of LDA.

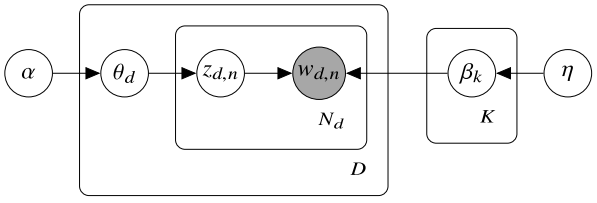


Fig. 2. Graphical model representation of LDA [11].

Algorithm 1: Variational Bayes (VB).

Input: document d, model $\{\lambda, \alpha\}$
Output: ϕ
Initialize ϕ randomly
for $l = 1, 2, \ldots, \infty$ **do**
  $\forall k, \gamma_k := \alpha + \sum_{d_v>0} \phi_{vk} d_v$
  $\forall (k, v), \phi_{vk} \propto exp\psi(\gamma_k).exp[\psi(\lambda_{kv}) - \psi(\sum_t \lambda_{kt})]$

One of the core issues in topic models, such as LDA, is posterior inference. It often refers to the problem of estimating the posterior distribution of latent variables for individual documents d, such as topic indices z or topic proportion θ.

Many approaches have been studied to deal with this estimation problem. Variational Bayes (VB) [4] as in Algorithm 1, collapsed variational Bayes (CVB) [22], CVB0 [1] try to estimate the distribution by maximizing a lower bound of the likelihood $p(d|\alpha, \beta)$. Collapsed Gibbs sampling (CGS) [9,16] try to estimate $p(z|d, \alpha, \beta)$. The Frank-Wolfe algorithm (FW) has been used in [24,25] to obtain sparse solutions to posterior inference, while [23] proposes Online Maximum a Posteriori Estimation (OPE). Both FW and OPE have theoretical guarantees on quality.

Algorithm 2: Online-VB (Stochastic variational inference) for LDA [11].

Input: hyperparameters $K, \eta, \alpha, D, \tau, \kappa$
Output: λ
Initialize λ
for $t = 1, 2, \ldots, T$ **do**
  Collect new data minibatch C
  for *document d in C* **do**
    Do local inference
      $\phi_d \leftarrow VB(d, \lambda, \alpha)$
  Compute intermediate global parameters
    $\forall (k, v), \tilde{\lambda}_{kv} \leftarrow \eta + D/|C| \sum_{dinC} \phi_{dvk} n_{dv}$
  Update the global variational parameters
    $\forall (k, v), \lambda_{kv} \leftarrow (1 - \rho_t)\lambda_{kv} + \rho_t \tilde{\lambda}_{kv}$
  where $\rho_t = (t + \tau)^{-\kappa}$

The main problem in topic models is to discover the hidden structure, i.e. the topics and how they relate to the documents, from an observed collection. Originally, it is solved by iterating between re-analyzing every data point in the dataset and re-estimating its hidden structure. The method is inefficient for large datasets and can quickly become impractical for very large ones because it requires a full pass through the data at each iteration. Hoffman et al. [11] derive a much more efficient algorithm by using stochastic optimization. The method uses less memory and converges faster than the standard approach. Hence it enables topic models to easily work with millions of texts. For LDA, the online learning algorithm can be described generally as in Algorithm 2. It uses Variational Bayes (VB) to do local inference for individual documents and update global variables (topics) in an online manner.

Follow that, Mimno et al. [16] propose to replace the local inference step in Algorithm 2 by Gibbs sampling. Foulds et al. [7] propose SCVB0, which is an online version of the batch algorithm by [1], where local inference for a document is done by CVB0. These algorithms are also known as Online-CGS and Online-CVB0. Than et al. [25,23] employ FW and OPE instead of VB in SVI to form two novel algorithms called Online-FW and Online-OPE respectively.

3.2. Biterm topic models on short texts

Biterm topic model (BTM) [27,6] is one of the state-of-the-art models to work well on short texts. Unlike conventional topic models, we generate all pairs of words (called biterm) for each document and aggregate the biterms of all the documents into one collection. This is then followed by modeling the generative process for this collection instead of each document. In details, suppose that a corpus of N_d documents contains N_B biterms $\mathbf{B} = \{b_1, \ldots, b_{N_B}\}$, where $b_i = (w_{i1}, w_{i2})$, the generative process is described as follows: For each biterm b_i in $\mathbf{B}$:

- Draw topic index $z_i|\boldsymbol{\theta} \sim Multinomial(\boldsymbol{\theta})$
- Draw biterm $(w_{i1}, w_{i2})|z_i, \beta \sim Multinomial(\beta_{z_i})$.

The key idea of BTM is based on two observations. First, if two words co-occur more frequently, they are more likely to belong to the same topic. Second, word co-occurrence (biterm) frequencies in the corpus are more stable and more clearly reveals the correlation between the words than those frequencies at the document level. Therefore, to avoid the sparsity problem at the document level, BTM uses the aggregated word co-occurrence patterns in all the corpus for topic discovery. The graphical model representation of BTM is shown in Fig. 3.

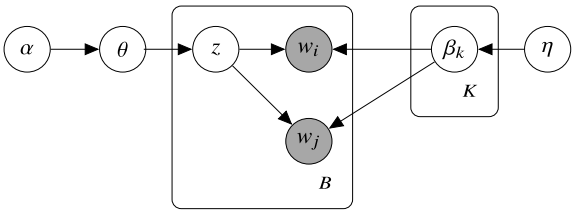
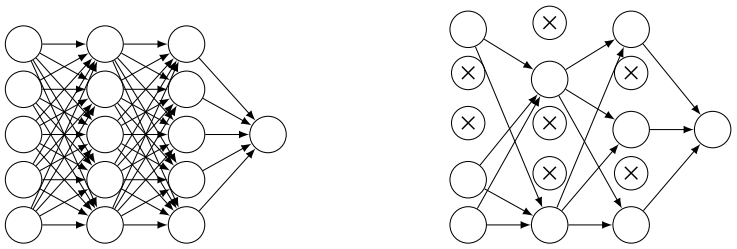


Fig. 3. Graphical model representation of BTM [6].



(a) Standard Neural Net. (b) After applying dropout.

Fig. 4. Illustration of dropout in Neural Networks. Crossed neurons have been dropped out.

3.3. Dropout in Neural Networks

Hilton et al. [10,18] propose dropout as a regularization technique to prevent overfitting in Neural Networks. During training, they drop out neurons along with all their incoming and outgoing connections. Simply, each neuron is retained with a fixed probability p independent of other neurons. This amounts to sampling a "thinned" network from it. Fig. 4 shows an illustration of a thinned network.

For each presentation of each training case, a newly thinned network is sampled by dropout and trained. Training with dropout can be seen as training a collection of 2^n possible thinned networks with extensive weight sharing architectures efficiently, where n is the number of neurons.

At test time, they merely use a single unthinned network whose weights are scaled-down versions of the trained weights so that output at test time has the same scale as the expected output at training time.

Dropout has been shown that it significantly reduces over-fitting and gives major improvements in Neural Networks. In the next section, we discuss how to exploit dropout in topic models, encode it into online learning methods to work with large-scale datasets.

4. Dropout for learning topic models at large scale

"Dropout" refers to dropping out nodes in a topic model. By dropping out a node, we temporarily remove it, along with all of its connections from the model. Ordinarily, we put a fixed probability p to decide whether a node is dropped or not, independent of other nodes.

4.1. Applying dropout to LDA

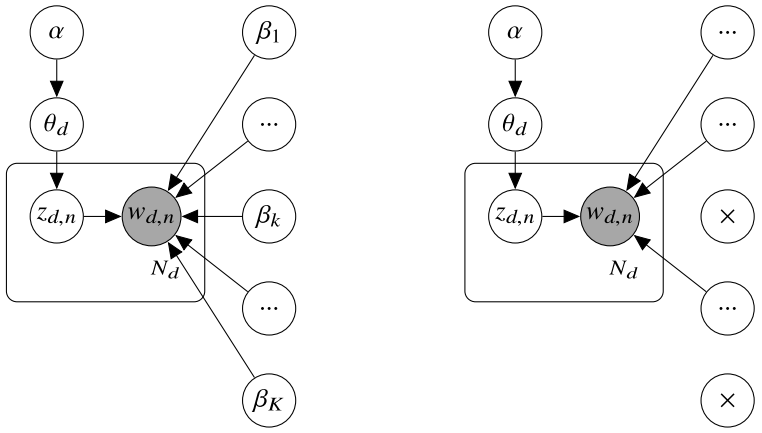
Considering LDA with K topics, each of which is a distribution over V words, we derive two following directions for using dropout: Topic dropout and word dropout.

4.1.1. Topic dropout

Algorithm 3: Topic dropout.

Input: Topic indices list $\{1, 2, \ldots, K\}$, dropout rate p
Output: Retained topic indices list RT
Initialize retained topic indices list RT as an empty list
for $k = 1, 2, \ldots, K$ **do**
  $r_k \sim Bernoulli(p)$
  If $r_k = 0$ then append topic index k to RT

The process of dropping topics is performed in a stochastic manner. In particular, each topic is chosen to be dropped with a probability p, independent of other topics. This is shown in Algorithm 3.



(a) LDA (b) LDA after applying dropout.

Fig. 5. Illustration of *topic dropout* in LDA. Crossed topics have been dropped and will not affect the documents.

Topic dropout allows us to sample a "small" model with fewer topics from the original one. The dropped topics have been integrated out and temporarily have no influence on the document. This is illustrated in Fig. 5.

Topic dropout will affect the generative process of the current document. Let $RT = [k_1, \ldots, k_{K'}]$ denote the topic indices that survive after the dropout. $\boldsymbol{\theta}_{d'} = (\boldsymbol{\theta}_{dk_1}, \ldots, \boldsymbol{\theta}_{dk_{K'}})$ denote the topic mixture of document d.

For the nth word of $\boldsymbol{d}$:

- draw topic index $z_{dn}|\boldsymbol{\theta}_{d'} \sim Multinomial(\boldsymbol{\theta}_{d'})$
- draw word $w_{dn}|z_{dn}, \beta \sim Multinomial(\beta_{z_{dn}})$.

As can be seen, each word in the document is assigned to a topic in RT. Therefore, the inference process of that document will be done using the retained topics.

4.1.2. Word dropout

Word dropout follows the similar manner to topic dropout, the operation of word dropout is shown in Algorithm 4.

Algorithm 4: Word dropout.

Input: Word indices list $\{1, 2, \ldots, V\}$, dropout rate p
Output: Retained words indices list RW
Initialize retained word indices list RW as an empty list
for $v = 1, 2, \ldots, V$ **do**
  $r_v \sim Bernoulli(p)$
  If $r_v = 0$ then append word index v to RW

The difference is that because we use all words' information to do local inference on a document, we only drop words out in updating the global parameters. Due to a few words in a short text, we use its all words to learn its topic distribution θ. This way avoids losing the quality of document representation. When updating topics from all data in a minibatch, the parameters corresponding to the dropped words are not updated.

4.2. Learning LDA with dropout

We show how to modify learning methods to employ dropout in a topic model. More specifically, we apply dropout on several stochastic learning algorithms for LDA, including Online-VB, Online-CVB0, Online-CGS, Online-FW, and Online-OPE.

4.2.1. Topic dropout

When applying topic dropout, a topic model can still be learned in a stochastic way similar to Online-VB [11]. The only difference is that we sample a "thinned" model by dropping out topics and do local inference for a document with the retained ones. In other words, from the original matrix λ, we derive a smaller matrix λ', which has the smaller number of dimensions than λ, by selecting all the rows λ_k in the original matrix λ, which correspond to the topics not dropped. The newly constructed matrix λ' will be used for the local inference. When forming intermediate global parameters, the local variational parameters of that document will not contribute to the dropped topics.

Algorithm 5: Online-VB with topic dropout.

Input: hyperparameters $K, \eta, \alpha, D, \tau, \kappa$, dropout rate p
Output: λ
Initialize the matrix λ
for $t = 1, 2, \ldots, T$ **do**
  Collect new data minibatch C
  Use Algorithm 3 to drop some topics out, and get $RT = \{k_1, \ldots, k_{K'}\}$
  Derive "thinned" matrix λ' from matrix λ: $\lambda' = \{\lambda_k \text{ for } k \in RT\}$
  for *document d in C* **do**
    Do local inference by
      $\phi_d \leftarrow VB(d, \lambda', \alpha)$
  Compute intermediate global parameters
    $\forall(i, v), \tilde{\lambda}_{iv} \leftarrow \eta + D/|C| \sum_{d \in C} \phi_{dvi} n_{dv}$
  Update the global variational parameters
    $\forall(i, v), \lambda_{k_i v} \leftarrow (1 - \rho_t)\lambda_{k_i v} + \rho_t \tilde{\lambda}_{iv}$
  where $\rho_t = (t + \tau)^{-\kappa}$

For simplification, when learning LDA, at each mini-batch, we sample only one model. The model affects all documents in the mini-batch. Consequently, the learning process of the mini-batch will be done on that model while all dropped topics remain unchanged.

Algorithm 5 shows Stochastic variational inference with topic dropout. It is worth noticing that, for each mini-batch, all the local parameters (ϕ_{dv}) are inferred by using the survived topics (λ') after dropout. Only these topics are updated in the mini-batch.

Following the same scheme, one can easily modify the other stochastic learning methods, such as Online-CVB0, Online-CGS, Online-FW, and Online-OPE, to employ topic dropout for LDA.

4.2.2. Word dropout

Algorithm 6: Online-VB with *word dropout*.

Input: hyperparameters $K, \eta, \alpha, D, \tau, \kappa$, dropout rate p
Output: λ
1 Initialize λ
2 **for** $t = 1, 2, \ldots, T$ **do**
3 Collect new data minibatch C
4 Use Algorithm 4 to drop some words out, and get $RW = \{v_1, \ldots, v_{V'}\}$
5 **for** *document d in C* **do**
6 Do local inference by
7 $\phi_d \leftarrow VB(d, \lambda, \alpha)$
8 Compute intermediate global parameters
9 $\forall k, i \in \{1, \ldots, V'\}, \tilde{\lambda}_{ki} \leftarrow \eta + D/|C| \sum_{d \in C} \phi_{dv_i k} n_{dv_i}$
10 Update the global variational parameters
11 $\forall k, i \in \{1, \ldots, V'\}, \lambda_{kv_i} \leftarrow (1 - \rho_t)\lambda_{kv_i} + \rho_t \tilde{\lambda}_{ki}$
12 where $\rho_t = (t + \tau)^{-\kappa}$

Instead of applying topic dropout in the local inference step, we apply word dropout only in the steps of updating global variational parameters. In particular, the local inference process on each mini-batch is done normally. But when updating the global variational parameters, we use the results learned from the local inference to update only parameters related to the words, which are not dropped out.

Algorithm 6 briefly presents how word dropout is employed for learning LDA by Online-VB. Where the lines 4 and 5 are the word dropout step, which samples a set of words RW from the vocabulary (dropout), we perform the local inference normally in lines 6-8, then update global parameters related to the sampled word set RW in lines 9-13. We emphasize that all words in each short text are used for doing local inference in order to keep the quality of original document representation.

4.3. Applying dropout on BTM

As shown in Fig. 3, BTM [27] aggregates all biterms of the corpus into a collection. It is worth observing that topic dropout is not appropriate for BTM. However, the number of biterms is obviously much larger than the number of original words in the corpus. In addition, many biterms, which are constructed from two unrelated words, have no meaning and make the set of biterms noisy. Therefore, word dropout has great potential for improving existing learning methods.

Instead, due to the above observation, we can sample a subset from the collection in the corpus, and use this sampled set of biterms for inference steps in each iteration. Note that BTM does not model each short text, therefore, dropping out

Algorithm 7: Online BTM algorithm with word dropout.

Input: hyperparameters $K, \eta, \alpha, B, \lambda$, dropout rate p
Output: β, θ

1 $\alpha^{(1)} := (\alpha, \ldots, \alpha)$
2 $\forall k \in \{1, \ldots, K\}, \eta_k^{(1)} := (\eta, \ldots, \eta)$
3 **for** $t = 1$ *to* T *do* **do**
4 **for** $i = 1$ *to* N_{iter} **do**
5 Sample a mini-batch $B'^{(t)}$ from $B^{(t)}$ by dropping words out
6 **for** ***each*** *biterm* $b_i = (w_{i,1}, w_{i,2}) \in B'^{(t)}$ **do**
7 Draw topic k
8 Update the counts of biterms in each topic k, $n_k^{(t)}$, and the number of times that each word w assigned to topic k, $n_{w_{i,1}|k}^{(t)}$ and $n_{w_{i,2}|k}^{(t)}$
9 Compute $\alpha^{(t+1)}$ and $\{\eta_k^{(t+1)}\}_{k=1}^K$
10 Update the global variational parameters $\beta^{(t)}$ and $\theta^{(t)}$

words in the collection is similar with word dropout when learning the global variables of LDA. Algorithm 7 illustrates how to apply word dropout for online Gibbs sampling in BTM. We use the same notations as in the original paper [27]. Line 5 is the sampling step that is different to the original method. Note that Online Gibbs sampling spends a great deal of time in comparison with Online-CGS that is a hybrid stochastic variational-Gibbs inference.

4.4. Theoretical analysis

A topic model usually does not work well on short texts. During learning, we iteratively do local inference for a document using all the topics, then use its local variational parameters to update the model. Hence, for a big model, the topic mixture for a short text may memorize all of that text. This may lead to the fact that many of memorized data are the result of sampling noise. Tang et al. [19] showed that poor performance of the LDA is expected when documents are too short, even if there is a very large number of documents.

Model ensemble often improves the predictive performance of machine learning methods. Dropout provides an effective way to use model ensemble in a topic model by combining many different models during learning. At each mini-batch, we sample a "small" model. For example, applying topic dropout in a model with K topics amounts to sample from 2^K possible "small" models. The whole learning method combines several "small" models from all mini-batches.

Regularization is critical for most machine learning models to improve the generalization. Wager et al. [26] describe how dropout can be seen as an adaptive regularizer. Instead of putting explicit constraints on the model parameters or variables, we use dropout as an implicit regularization. In particular, for dropping out topics in a topic model, we limit the solutions space of document's local parameters to a smaller-size vector space that represents the retained topics. That could probably reduce overfitting on short texts, each of which usually contains few topics.

More generally, dropout can be interpreted as a Sparsity-induced regularization. Each document works with a "small" sampled model, which is, in fact, sparse in view of the whole "big" model. That will fit better with the nature of sparse data like short texts. Moreover, a "small" model reduces the chance to just memorize all data. That promises for avoiding over-fitting on short texts modeling.

Working with a "small" model at each mini-batch could also reduce running time for learning topic models, which is crucial for big datasets with millions of documents. Furthermore, this research proposes broader implications for dropout. Because the idea of dropout is simple, it can be applied to a wide range of Bayesian models. We suggest using dropout to deal with extremely sparse data, which have the same nature with short texts.

5. Empirical evaluation

In this section, we carry out extensive experiments to investigate the effectiveness of dropout for topic models with a variety of stochastic learning methods:

- Online Variational Bayes (Online-VB) which is often known as SVI [11]
- Online Collapsed variational Bayes (Online-CVB0) [7]
- Online Frank-Wolfe (Online-FW) [25]
- Online-OPE [23]
- Online Collapsed Gibbs sampling (Online-CGS) [16]

The two types of datasets are used in these experiments: short texts and normal documents. For short texts, we did experiments for both LDA and BTM, while we did experiments for only LDA for normal documents. The detailed description for each dataset is shown in Table 1:

1. Short texts: We use three large data sets of short texts for evaluation [14]: *Yahoo* Questions crawled from answers.yahoo.com, each document is a question; *Tweets* from Twitter (twitter.com) - each document is the text content of a tweet; *NYT-titles* from The New York Times (www.nytimes.com) - each document is the title of an article. These datasets are preprocessed by tokenizing, stemming, removing stopwords, removing low-frequency words (appear in less than 3 documents) and removing extremely short documents (less than 3 words).
2. Normal texts: We use a large data set of normal texts for evaluation: *Pubmed*[1] consisting of 8.2 millions of medical articles from the Pubmed central. For each corpus, we set aside randomly 1000 documents for testing, and used the remaining for learning.

Table 1
Description of four datasets.

Dataset	Corpus size	Average length per doc	V
Yahoo questions	537,770	4.73	24,420
Twitter	1,485,068	10.14	89,474
NYT-titles	1,684,127	5.15	55,488
Pubmed	7,999,000	89.12	141,044

We used Log Predictive Probability (LPP) [11] to measure the quality of a model which has been learned from the training data. LPP measures the predictiveness and generalization of a model to new data.

For the purpose of comparison, we follow previous studies [11,7,23] for all the parameter settings. In particular, the model parameters for LDA are $K = 100$, $\alpha = 1/K$, $\eta = 1/K$; the learning parameters are $S = |C_t| = 5000$, $\kappa = 0.9$, $\tau = 100$. We will examine the sensitivity of these parameters in Appendix A.1.

5.1. LDA on short texts

In the first group of experiments, we investigate the benefit of dropout for 5 learning methods on short texts. Fig. 6 shows the performance of topic dropout with different values of dropout rate $p \in \{0.1; 0.3; 0.5; 0.7; 0.9\}$. It is clear that Online-CVB0, Online-FW, and Online-VB do not work well for short text. Their learning curves (black lines) fall very soon. The shorter document length is, the more overfitting the LDA model suffers. Indeed, it can be seen that NYT-titles and Yahoo have shorter document length than Twitter and quality of the models learned from NYT-titles or Yahoo by Online-CVB0, Online-FW and Online-VB declines significantly during the training process.

We observe that topic dropout significantly reduces overfitting in three methods: Online-CVB0, Online-FW and Online-VB. For NYT-titles and Yahoo, the increase of dropout rate almost leads to the reduction of overfitting, hence, better performance. The best dropout rate (0.9 for NYT-titles and 0.7/0.9 for Yahoo) can increase the model quality up to 10% in comparison with the original learning methods. However, in Twitter dataset, the highest value of dropout rate ($p = 0.9$) does not make the quality of the learned model better, in comparison with the lower rates ($p = 0.3$ is the best value of dropout rate). The possible reason is that since average document length of Twitter is approximately as twice as that of NYT-titles or Yahoo (described in Table 1), each document in Twitter may be composed of more number of topics. Therefore, dropping more topics in the inference procedure may lose the information of documents.

Online-OPE and Online-CGS do not suffer overfitting. The topic dropout still helps Online-OPE obtain better results in some cases, whereas it reduces the performance of Online-CGS significantly. The reason is that CGS is based on sampling rather than optimizing a specific a function, therefore dropping out topics has a negative impact on learning performance.

Fig. 7 shows how fast the learning methods do. It is clear that the time for the training process with dropout of almost all methods is equal or smaller than the original training. Since a number of topics are dropped out, the inference procedure requires less computing operations, therefore its runtime is reduced.

Fig. 8 illustrates the results of word dropout training when applying for Online-VB on the datasets: NYT-titles, Twitter and Yahoo. We observe the same phenomenon as topic dropout that the overfitting is prevented. However, the increase percentage of performance is about 5% smaller than topic dropout.

5.2. BTM on short texts

In this experiment, we investigate the effectiveness of word dropout for BTM learned by Online-VB and Online Gibbs sampling (Online-GS) on short texts. We set the model parameters: $K = 100$, $\alpha = 1/K$, $\eta = 1/K$, minibatch size $= 5000$. For Online-VB, the learning parameters are $\kappa = 0.9$, $\tau = 100$. For Online-GS, $N_{iter} = 1000$ and $\lambda = 1$. Due to which Online-GS in the original paper spends a great deal of training time, we only run two algorithms through all data one time. We evaluate our method in terms of performance and computational time. First, Table 2 shows the learning time with different values

[1] The datasets are retrieved from http://archive.ics.uci.edu/ml/datasets.

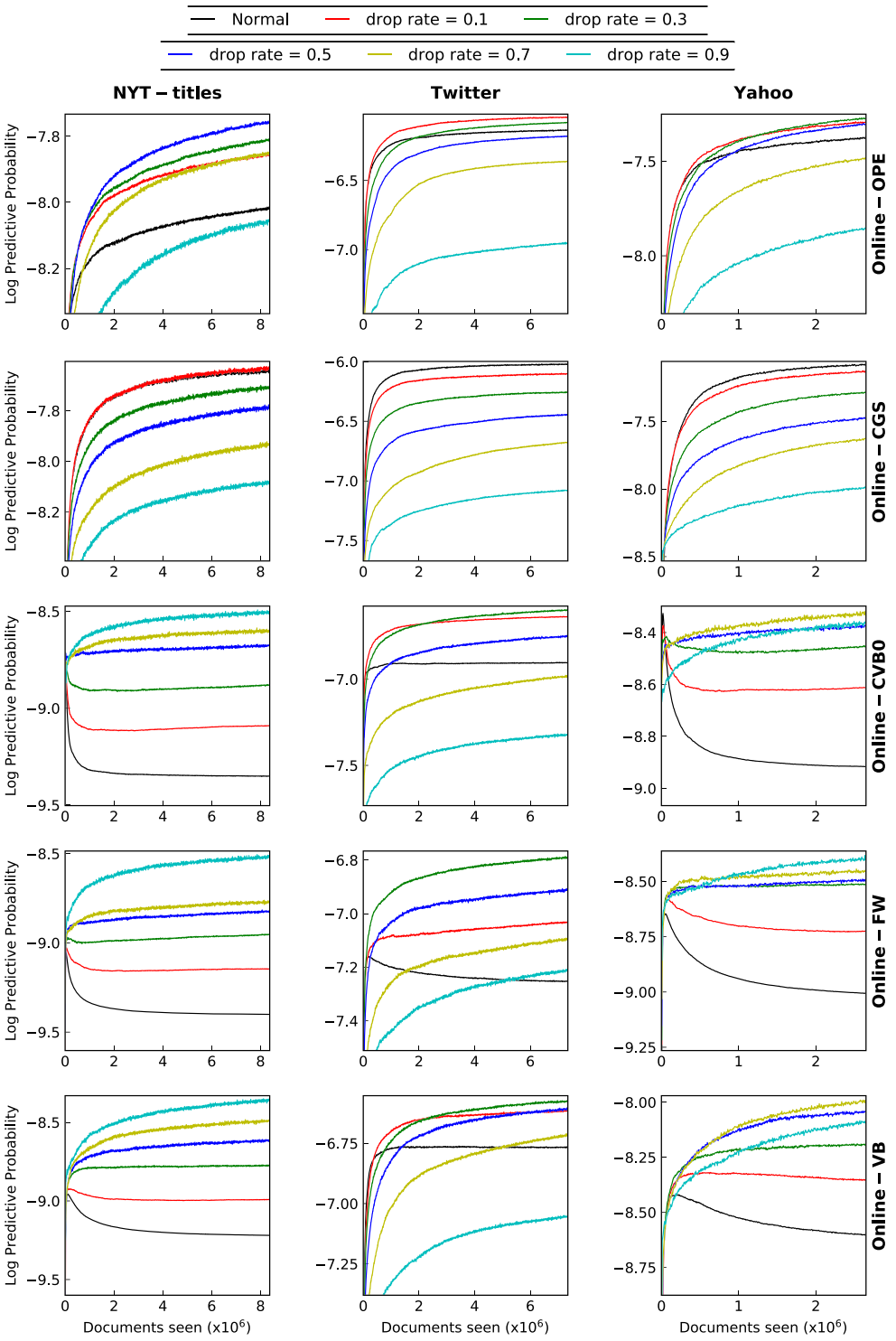


Fig. 6. Predictiveness of the models learned by five different methods: Online-OPE, Online-CGS, Online-CVB0, Online-FW, Online-VB (read in columns from top to bottom), with *topic dropout* on three datasets: NYT-titles, Twitter, and Yahoo (read in rows from left to right). Higher is better. 'Normal' (black curves) are the original learning methods (without dropout). Note that Online-CVB0, Online-FW, and Online-VB often suffer from overfitting.

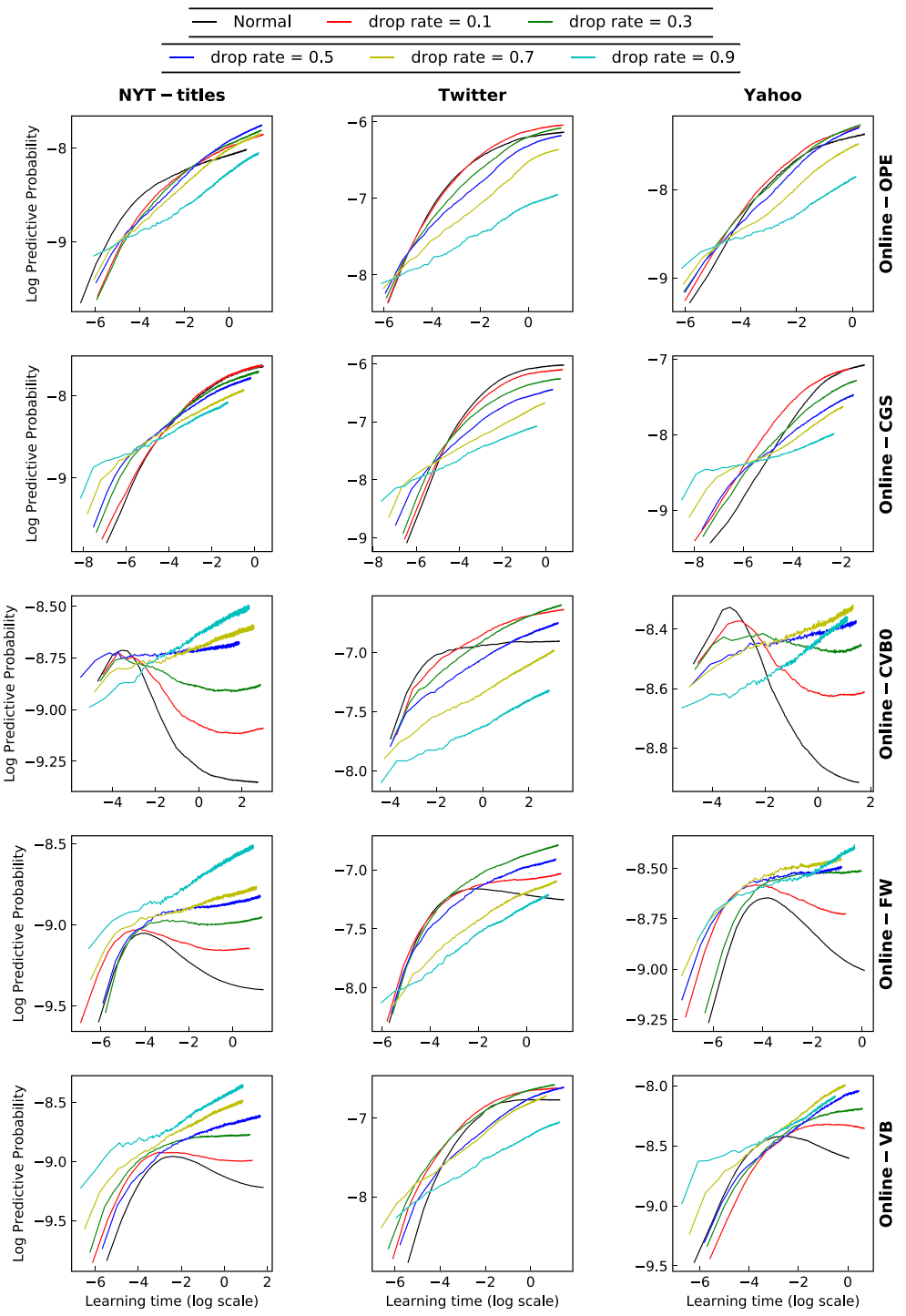


Fig. 7. Predictiveness against learning time (log scale of hours) of five different methods: Online-OPE, Online-CGS, Online-CVB0, Online-FW, Online-VB (read in columns from top to bottom), with *topic dropout* on three datasets: NYT-titles, Twitter, and Yahoo (read in rows from left to right). Higher is better. 'Normal' (black curves) are the original learning methods (without dropout).

of dropout rate $p \in \{0.1; 0.3; 0.5; 0.7; 0.9\}$ when applying word dropout to BTM. Online-GS spends a great deal of time when requiring a lot of iterations. It is obvious that applying word dropout significantly reduces the learning time. Second, the performance of word dropout is shown in Fig. 9. It illustrates that the predictive likelihood increases gradually when data increases; therefore, BTM work well on short texts. Moreover, Online-VB achieves better results than Online-GS. In

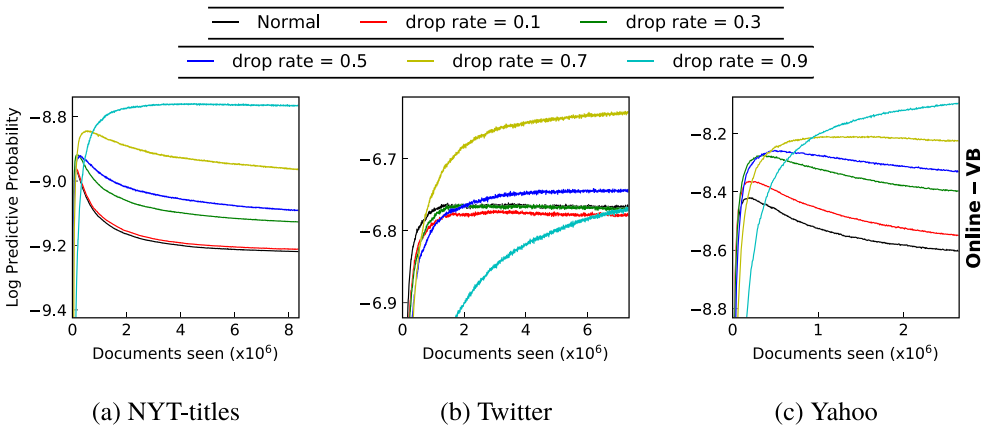


Fig. 8. Predictiveness of models learned by Online-VB with *word dropout*. 'Normal' is Online-VB without dropout.

Table 2
Learning time (s) of BTM on short texts with word dropout.

Dataset	Methods	Drop rate					
		Normal	$r = 0.1$	$r = 0.3$	$r = 0.5$	$r = 0.7$	$r = 0.9$
NYT-titles	Online-VB	3224	3050	2503	1870	1189	880
	Online-GS	90219	85610	71573	56306	39515	15900
Twitter	Online-VB	3224	3050	2503	1870	1189	880
	Online-GS	404573	357891	288019	213525	134510	46840
Yahoo	Online-VB	448	418	372	328	274	223
	Online-GS	32042	27544	21763	16729	19533	7565

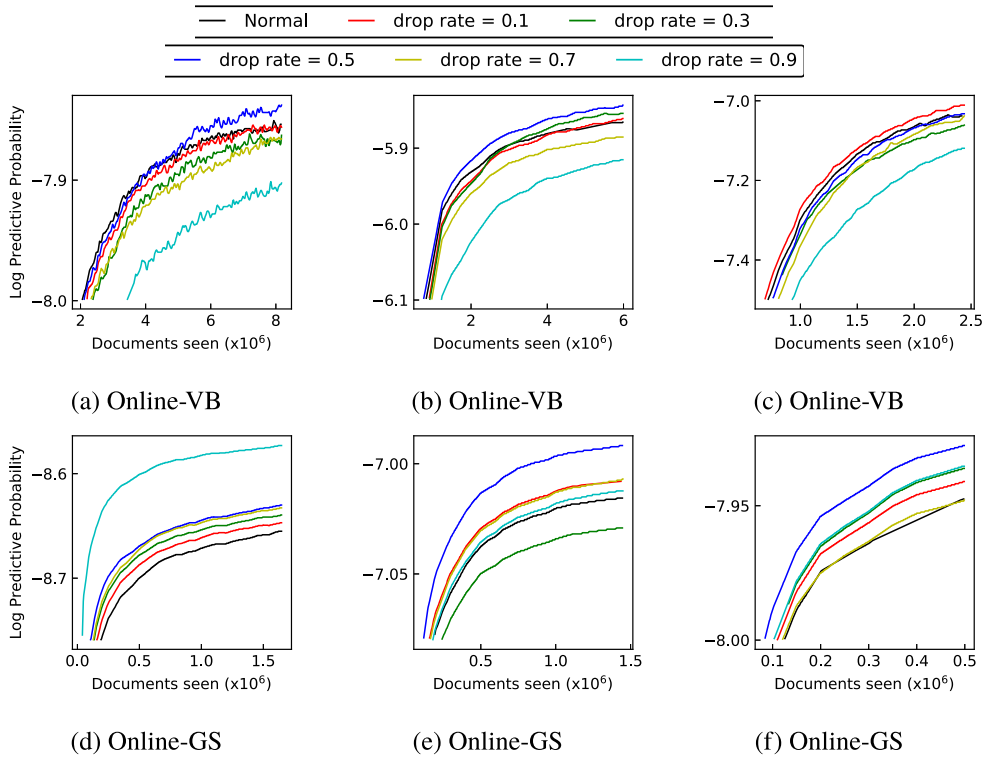


Fig. 9. Predictiveness of BTM learned by Online-VB, and Online-GS with *word dropout* on three datasets: NYT-titles (a, d), Twitter (b, e), and Yahoo (c, f). Higher is better. 'Normal' (black curves) are the original learning methods (without dropout).

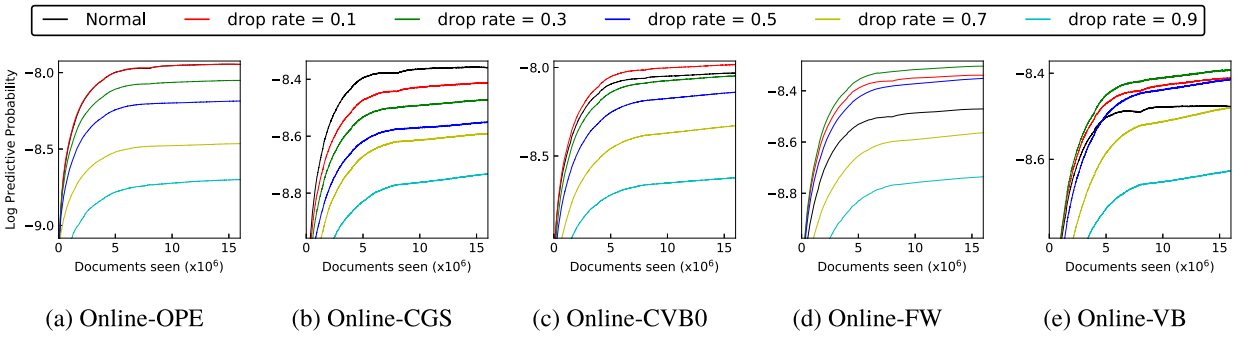


Fig. 10. Predictiveness of the models learned by different methods with *topic dropout* on Pubmed. We observe that dropout often leads to degradation in predictiveness. The larger the drop rate is, the deeper the degradation seems to be.

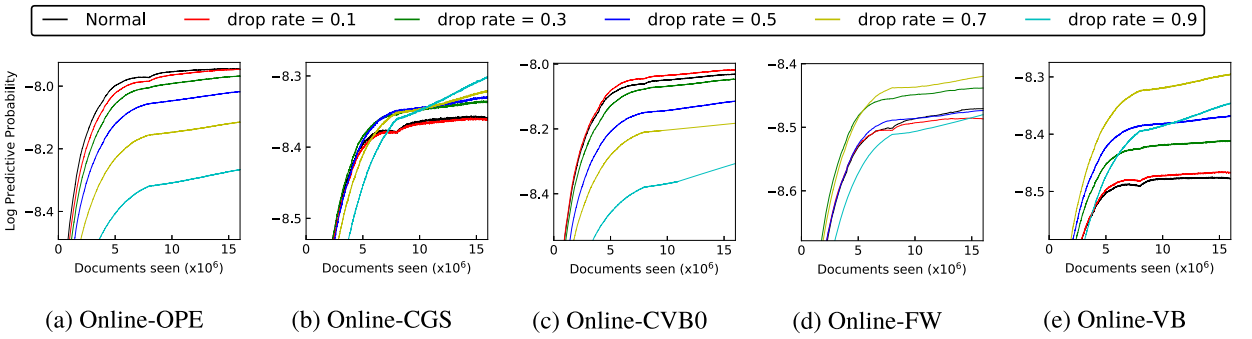


Fig. 11. Predictiveness of the models learned by different methods with *word dropout* on Pubmed.

particular, as can be seen from Fig. 9, utilizing dropout with appropriate drop rate would improve the performance in all cases.

5.3. Normal texts

In this test, we carry out dropout on a dataset of long texts (Pubmed) with the results shown in Fig. 10 and Fig. 11. Even though the learning methods do not suffer overfitting, dropout still helps Online-VB, Online-FW and Online-CVB0 to obtain better performance in the cases of small dropout rates. In normal texts, because of having longer size, each document should be a mixture of more topics. As a result, the smaller dropout rate (0.1 or 0.3) is better for topic models and the benefits of dropout training are not as significant as that in short texts. In particular, the highest increase in predictiveness when applying topic dropout on Pubmed is about 1%.

5.4. Discussion

From our investigation, we observe the following:

- *For short text:* topic dropout makes a big improvement for Online-VB, Online-CVB0, and Online-FW. However, both topic dropout and word dropout do not help much to improve Online-OPE and Online-CGS. Note that the inference for individual documents is deterministic for Online-VB, Online-CVB0, and Online-FW. In contrast, Online-OPE uses stochastic optimization while Online-CGS use Gibbs sampling to do inference for individual documents. It means Online-OPE and Online-CGS stochastically do inference for individual documents. Further, all of those 5 methods use the same strategy with [11] to update the global variable (λ). As a consequence, we conclude that the core inference routines might be the main reason why dropout has different effects on the five learning methods. Dropout might have a significant impact when integrated into the learning methods which do inference deterministically.
- **Conjecture:** we conjecture that dropout will make a significant impact if there are some deterministic inference parts in a learning algorithm that has to work with extremely sparse data or short text.
- *For long text:* dropout might not provide a clear benefit as we do not observe improvement clearly. Some inappropriate drop rates even worsen the performance of the original methods. The reason might be that long documents contain

much information, and do not pose difficulty for the learning methods. Dropout may take the risk of losing information from long documents.

- *Dropout rate:* From experimental point of view, we leave the control of this dropout rate to programmers with the following suggestions for topic dropout and word dropout. First, for topic dropout, intuitively, the shorter each document is, the less number of topics it should be assigned, hence, the more topics should be dropped out when doing local inference. In other words, the bigger the average length of documents is, the bigger the dropout rate should be. In evidence, on normal text dataset "Pubmed", the more topics we drop, the worse performance we get. Moreover, on short text datasets including Twitter, NYT-titles and Yahoo, the average document length of Twitter is approximately as twice as that of NYT-titles or Yahoo. As expected, we observe that dropping more topics (dropout rate p is 0.9 or 0.7) has better performance on NYT-titles and Yahoo, but worse performance on Twitter. Dropout rate on Twitter should be set smaller than those on the two datasets. Second, for word dropout, we have the same phenomenon as topic dropout. If each document has more words, the results of inference on that document should have an influence on more global parameters. Therefore, we suggest that dropout rate should be bigger for short text corpus having smaller average length.

6. Conclusion

In this work, we investigated the different strategies to employ dropout to help topic models to avoid overfitting. The dropout method on both topics and words has been evaluated with a variety of stochastic inference methods on 4 large real-world datasets. The results show that although a wrong selection of the drop rate may cancel the advantages of the dropout, if this parameter is well selected, dropout can prevent topic models from overfitting and significantly improve the generalization. Therefore, our work suggests to use dropout as an efficient technique to deal well with short text.

Declaration of Competing Interest

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

Acknowledgement

This research is funded by the Office of Naval Research Global (ONRG) under Award Number N62909-18-1-2072, and Air Force Office of Scientific Research (AFOSR), Asian Office of Aerospace Research & Development (AOARD) under Award Number 17IOA031.

Appendix A

A.1. Parameter sensitivity

In this subsection, we investigate the sensitivity of the parameters (batch size S, the number K of topics, κ, and τ) to the effectiveness of the learning methods with dropout. We use the setting of ($K = 100$, $S = 5000$, $\kappa = 0.9$, $\tau = 100$) as the core, and then each parameter is changed in experimental scenario while the remaining parameters are fixed. Figs. 12, 13, 14, and 15 illustrate the effectiveness of five methods with topic dropout (drop rate $p = 0.5$) when batch size S, the number K of topics, κ, and τ vary respectively.

Overall, the experiments show that the five methods with dropout can avoid overfitting. It is rare to encounter the phenomenon in which the results of these methods increase sharply to reach at a peak in a few initial minibatches, afterwards decrease gradually in the remaining minibatches.

Fig. 12 shows that all the five methods relatively stabilize the predictive ability as batch size is varied, while the number of topics (Fig. 13) noticeably influences the effectiveness of these methods. One reason is that the amount of data in each minibatch is large enough to form topics obviously. Moreover, it is reasonable that the number of topics is an important parameter to affect the quality of methods. Especially, when S or K increases, almost the methods achieve better results on the grounds that they are provided more data or larger topic space respectively in order to model data explicitly. Furthermore, the influence of both S and K on online OPE is slighter than the other methods.

Regarding κ and τ which are the components of learning rate ρ, Fig. 15 illustrates that all the five methods are slightly influenced by τ (except $\tau = 1$), while κ (Fig. 14) makes online CVB0 and Online-FW to change noticeably more than the remaining methods. It is clear that the amount of information of each minibatch in short texts, which is not large, contributes a minor rate to the model. Therefore, τ should be set a large value. $\tau = 1$ makes these methods get the low effectiveness, however, when $\tau > 50$ those methods achieve consistently better results. $\kappa = 0.6$ is probably suitable for five methods to obtain better results, albeit Online-CVB0 and Online-FW only have low effectiveness in several initial minibatches.

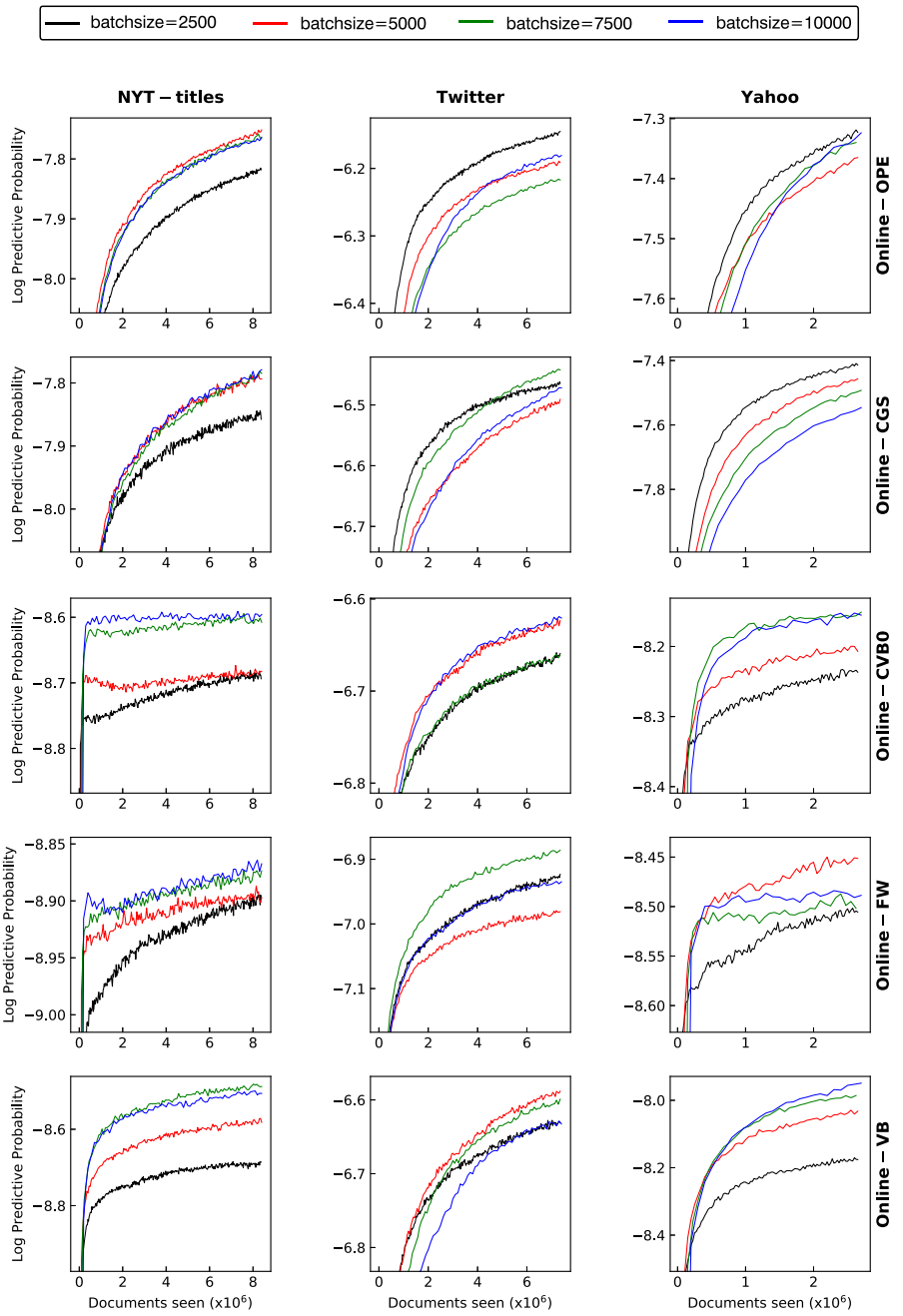


Fig. 12. Sensitivity of batch size to five methods: Online-OPE, Online-CGS, Online-CVB0, Online-FW, Online-VB (read in columns from top to bottom), on three datasets: NYT-titles, Twitter, and Yahoo (read in rows from left to right).

A.2. Quality of learned topics

We next want to see how good the topics have been learned from short text. NYT-titles is used as the dataset, while LDA and BTM are used as the base models in this evaluation. After learning, we extract some topics and visualize top 10 words in each topic to evaluate the quality. Tables 3 and 4 show some of those topics. It can be seen in Table 3, in comparison with conventional LDA, the topics learned by applying the dropout technique have less ambiguous words (highlighted in the tables), which seem inappropriate in the topics. Meanwhile, as shown in Table 4, applying the dropout technique to BTM does not change much the quality of learned topics. This is reasonable since BTM has a good performance on short texts.

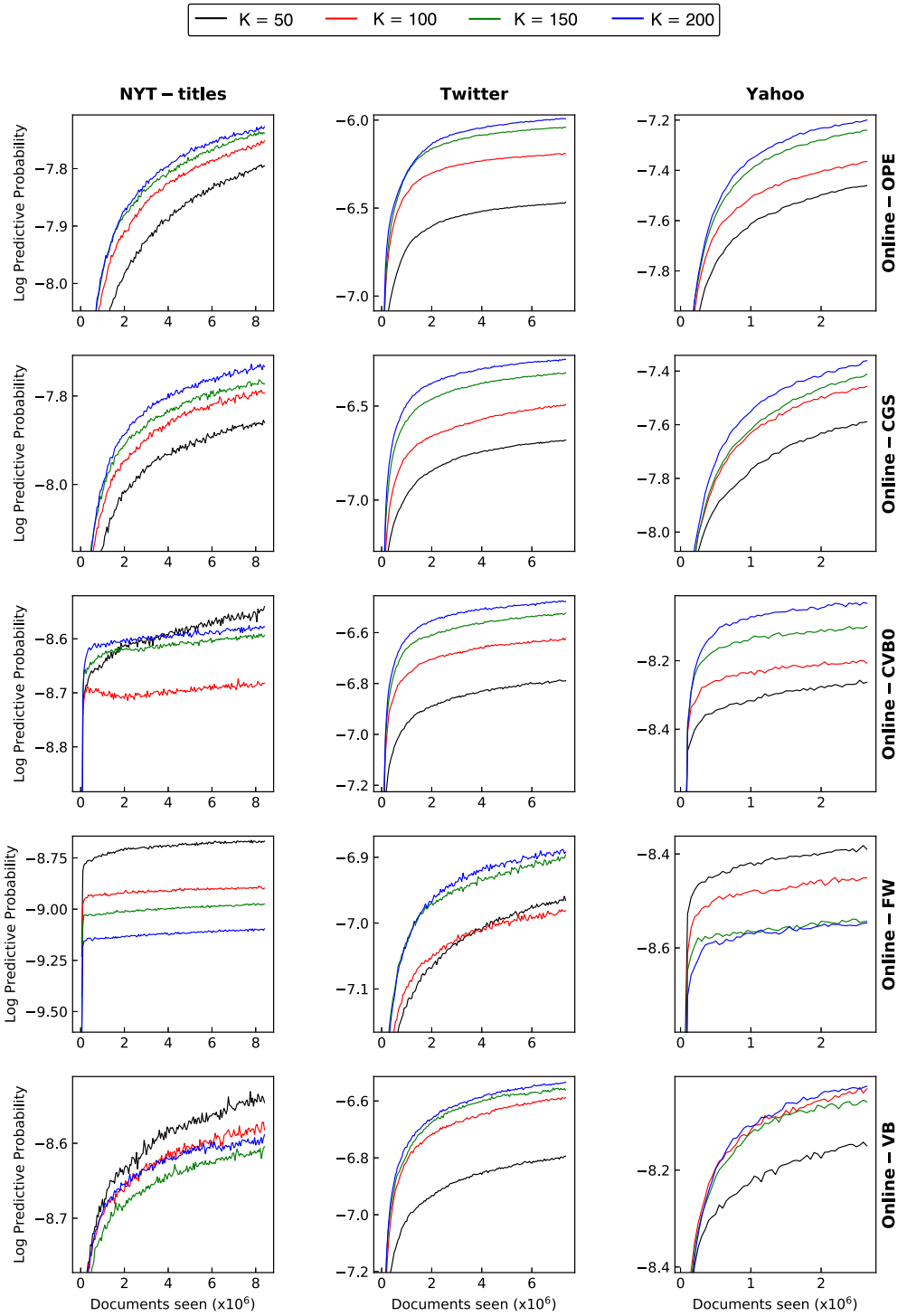


Fig. 13. Sensitivity of the number K of topics to five methods: Online-OPE, Online-CGS, Online-CVB0, Online-FW, Online-VB (read in columns from top to bottom), on three datasets: NYT-titles, Twitter, and Yahoo (read in rows from left to right).

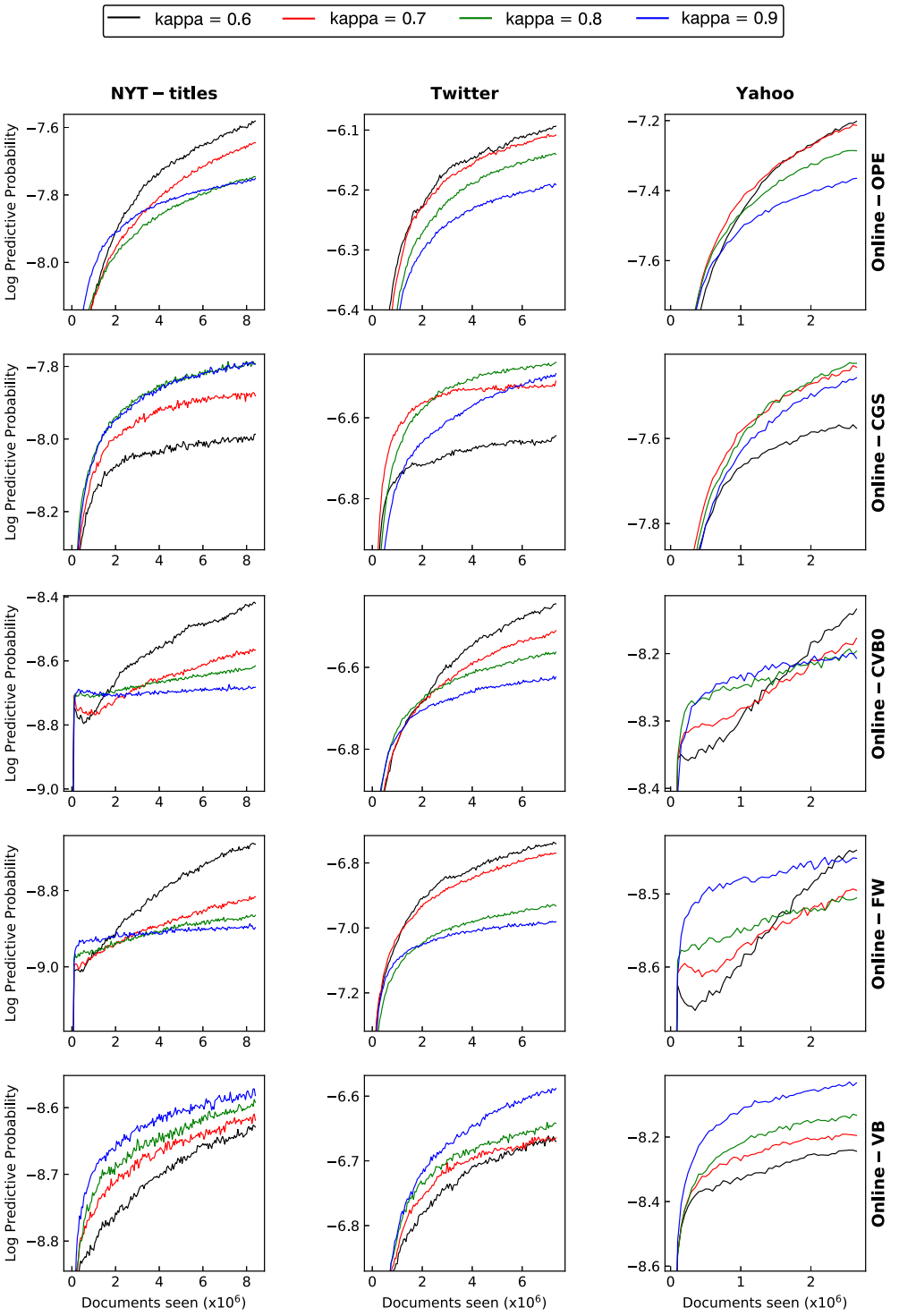


Fig. 14. Sensitivity of κ to five methods: Online-OPE, Online-CGS, Online-CVB0, Online-FW, Online-VB (read in columns from top to bottom), on three datasets: NYT-titles, Twitter, and Yahoo (read in rows from left to right).

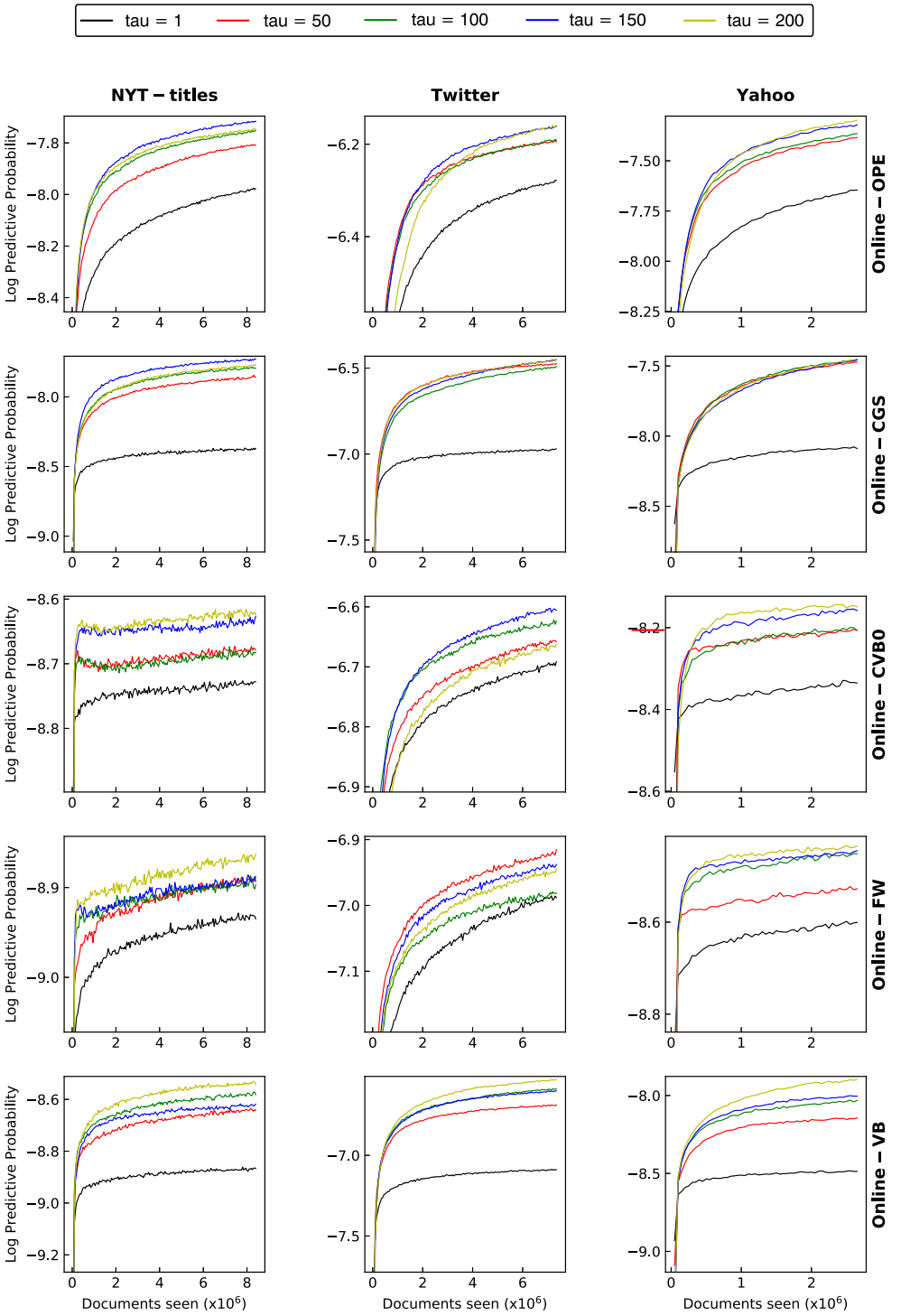


Fig. 15. Sensitivity of τ to five methods: Online-OPE, Online-CGS, Online-CVB0, Online-FW, Online-VB (read in columns from top to bottom), on three datasets: NYT-titles, Twitter, and Yahoo (read in rows from left to right).

Table 3
Top words of some learned topics in LDA on NYT-titles. Shaded words in a topic seem to be the intruders to that topic.

Online FW				Online VB			
Topic 1 (Business)		Topic 2 (Politics)		Topic 1 (Business)		Topic 2 (Politics)	
Normal	Dropout ($p = 0.5$)	Normal	Dropout ($p = 0.5$)	Normal	Dropout ($p = 0.5$)	Normal	Dropout ($p = 0.5$)
$	$	world	house	world	world	rise	bill
chief	million	obama	obama	europe	job	price	return
million	sale	big	senate	business	business	trade	senate
arm	rise	europe	bill	news	cut	bill	u.n.
continue	stock	gain	campaign	service	profit	senate	official
control	profit	britain	plan	britain	europe	free	today
pact	price	quit	health	rebel	china	french	vote
risk	fall	small	democrat	policy	america	support	clinton
deny	buy	paul	clinton	agree	build	gold	tax
history	market	replace	vote	land	fall	crime	capital

Table 4
Top words of some learned topics in BTM by Online-GS on NYT-titles. Shaded words in a topic seem to be the intruders to that topic.

Topic 1 (Business)		Topic 2 (Politics)		Topic 3 (Music)		Topic 4 (Sport)	
Normal	Dropout ($p = 0.5$)	Normal	Dropout ($p = 0.5$)	Normal	Dropout ($p = 0.5$)	Normal	Dropout ($p = 0.5$)
$	$	time	state	music	music	time	time
million	million	state	president	review	review	sport	sport
buy	buy	president	north	art	concert	win	world
billion	deal	political	carolina	opera	jazz	cup	cup
deal	company	reagan	mr.	business	play	game	art
pay	unit	campaign	reagan	meet	art	world	city
unit	business	press	political	concert	work	show	east
sell	pay	mr.	lead	work	dance	team	game
plan	sell	college	secretary	play	hall	title	play
company	world	lead	campaign	s	pop	big	real

References

[1] A. Asuncion, M. Welling, P. Smyth, Y.W. Teh, On smoothing and inference for topic models, in: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, AUAI Press, 2009, pp. 27–34.
[2] S. Banerjee, K. Ramanathan, A. Gupta, Clustering short texts using Wikipedia, in: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, 2007, pp. 787–788.
[3] D.M. Blei, Probabilistic topic models, Commun. ACM 55 (2012) 77–84.
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent Dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[5] N. Chen, J. Zhu, J. Chen, B. Zhang, Dropout training for support vector machines, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27–31, 2014, Québec City, Québec, Canada, 2014, pp. 1752–1759.
[6] X. Cheng, X. Yan, Y. Lan, J. Guo, BTM: topic modeling over short texts, IEEE Trans. Knowl. Data Eng. 26 (2014) 2928–2941.
[7] J. Foulds, L. Boyles, C. DuBois, P. Smyth, M. Welling, Stochastic collapsed variational Bayesian inference for latent Dirichlet allocation, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '13, ACM, New York, NY, USA, 2013, pp. 446–454.
[8] C.E. Grant, C.P. George, C. Jenneisch, J.N. Wilson, Online topic modeling for real-time Twitter search, in: TREC, 2011.
[9] T.L. Griffiths, M. Steyvers, Finding scientific topics, Proc. Natl. Acad. Sci. 101 (2004) 5228–5235.
[10] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R.R. Salakhutdinov, Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint, arXiv:1207.0580, 2012.
[11] M.D. Hoffman, D.M. Blei, C. Wang, J.W. Paisley, Stochastic variational inference, J. Mach. Learn. Res. 14 (2013) 1303–1347.
[12] T. Hofmann, Unsupervised learning by probabilistic latent semantic analysis, Mach. Learn. 42 (2001) 177–196.
[13] L. Hong, B.D. Davison, Empirical study of topic modeling in Twitter, in: Proceedings of the First Workshop on Social Media Analytics, SOMA '10, ACM, New York, NY, USA, 2010, pp. 80–88.
[14] K. Mai, S. Mai, A. Nguyen, N. Van Linh, K. Than, Enabling hierarchical Dirichlet processes to work better for short texts at large scale, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2016, pp. 431–442.
[15] R. Mehrotra, S. Sanner, W. Buntine, L. Xie, Improving LDA topic models for microblogs via tweet pooling and automatic labeling, in: Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '13, ACM, New York, NY, USA, 2013, pp. 889–892.
[16] D. Mimno, M. Hoffman, D. Blei, Sparse stochastic inference for latent Dirichlet allocation, in: 29th Annual International Conference on Machine Learning, 2012.
[17] P. Schönhofen, Identifying document topics using the Wikipedia category network, Web Intell. Agent Syst.: Int. J. 7 (2009) 195–207.
[18] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J. Mach. Learn. Res. 15 (2014) 1929–1958.
[19] J. Tang, Z. Meng, X. Nguyen, Q. Mei, M. Zhang, Understanding the limiting factors of topic modeling via posterior contraction analysis, in: ICML, 2014, pp. 190–198.
[20] J. Tang, M. Zhang, Q. Mei, One theme in all views: modeling consensus topics in multiple contexts, in: Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2013, pp. 5–13.

[21] Y.W. Teh, M.I. Jordan, M.J. Beal, D.M. Blei, Hierarchical Dirichlet processes, J. Am. Stat. Assoc. 101 (2006) 1566–1581.
[22] Y.W. Teh, D. Newman, M. Welling, A collapsed variational Bayesian inference algorithm for latent Dirichlet allocation, in: NIPS, vol. 6, 2006, pp. 1378–1385.
[23] K. Than, T. Doan, Guaranteed inference in topic models, arXiv preprint, arXiv:1512.03308, 2015.
[24] K. Than, T.B. Ho, Fully sparse topic models, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, in: LNCS, vol. 7523, Springer, 2012, pp. 490–505.
[25] K. Than, T.B. Ho, Inference in topic models: sparsity and trade-off, arXiv preprint, arXiv:1512.03300, 2015.
[26] S. Wager, S. Wang, P.S. Liang, Dropout training as adaptive regularization, in: C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, K.Q. Weinberger (Eds.), Advances in Neural Information Processing Systems, vol. 26, Curran Associates, Inc., 2013, pp. 351–359.
[27] X. Yan, J. Guo, Y. Lan, X. Cheng, A biterm topic model for short texts, in: Proceedings of the 22nd International Conference on World Wide Web, WWW '13, ACM, New York, NY, USA, 2013, pp. 1445–1456.