Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework

Yansheng Wang, Yongxin Tong, Dingyuan Shi
SKLSDE Lab, BDBC, School of Computer Science and Engineering and IRI, Beihang University, China
{arthur_wang, yxtong, chnsdy}@buaa.edu.cn

Abstract

Latent Dirichlet Allocation (LDA) is a widely adopted topic model for industrial-grade text mining applications. However, its performance heavily relies on the collection of large amount of text data from users' everyday life for model training. Such data collection risks severe privacy leakage if the data collector is untrustworthy. To protect text data privacy while allowing accurate model training, we investigate federated learning of LDA models. That is, the model is collaboratively trained between an untrustworthy data collector and multiple users, where raw text data of each user are stored locally and not uploaded to the data collector. To this end, we propose FedLDA, a local differential privacy (LDP) based framework for federated learning of LDA models. Central in FedLDA is a novel LDP mechanism called Random Response with Priori (RRP), which provides theoretical guarantees on both data privacy and model accuracy. We also design techniques to reduce the communication cost between the data collector and the users during model training. Extensive experiments on three open datasets verified the effectiveness of our solution.

1 Introduction

The Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) model is a popular topic model for text-mining. It has widespread adoption in applications such as text classification, tag recommendation, opinion mining etc. (Blei, Ng, and Jordan 2003; Steyvers et al. 2004; Airoldi et al. 2008; Blei 2012), and has been a fundamental building block for many commercial products. For example, LDA is used by Microsoft (Yuan et al. 2015), Baidu (Jiang et al. 2018) and Tencent (Yu et al. 2017) in many of their services.

Accurate LDA models rely on large amounts of text data collected from users' everyday life for training. Reviews of products, tags on movies, microblogs and E-mails are all common sources for data collection. Consequently, improper exploitation of these data by an untrustworthy data collector can easily leak sensitive information and violate user privacy.

To avoid malicious use of text data from an untrustworthy data collector while allowing accurate training of the LDA models, we investigate LDA model training in a *federated learning* setting (McMahan et al. 2017; Yang et al. 2019). That is, the LDA model is collaboratively trained between an untrustworthy data collector and multiple users, where the raw text data of each individual user are locally stored and only privacy-protected intermediate results are transmitted to the data collector for model training. However, to realize such federated LDA learning is non-trivial. It is challenging to decide how and what to communicate between the data collector and the users such that there is guarantee on both data privacy and model accuracy.

In this paper, we propose FedLDA, the first-of-its-kind solution to federated learning of LDA models without assuming any trustworthy data collector. FedLDA quantifies and protects user privacy in the context of local differential privacy (LDP) (Evfimievski, Gehrke, and Srikant 2003), a privacy measure naturally fit for a federated learning setting with no trusted third parties. Central in FedLDA is a novel and practical LDP mechanism that offers theoretical guarantees for both privacy and utility (assessed by errors in model parameters) by exploiting randomized response (RR) techniques (Erlingsson, Pihur, and Korolova 2014). We also devise techniques to reduce the communication cost of learning with FedLDA. The contributions of this work are summarized as follows.

- This is the first work to consider LDA training in a federated setting without a trustworthy data collector, an increasingly important yet largely unexplored problem in many industrial topic mining applications.
- We propose a local differential privacy (LDP) based solution to federated learning of topic models without trustworthy third parties. To the best of our knowledge, we are the first to introduce LDP to federated learning and we design a novel LDP mechanism with theoretical guarantees on both data privacy and model accuracy.
- We evaluate our solution on three open datasets. Experimental results validate the effectiveness of our solution.

In the rest of this paper, we first review basics on LDA and LDP, and then elaborate on the design and performance of our proposed FedLDA. Finally we review related work and conclude this work.

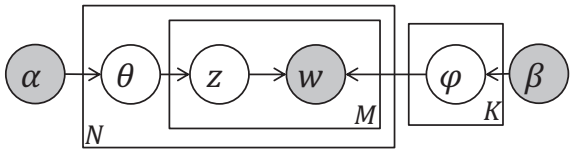


Figure 1: Graphical model for LDA.

2 Preliminaries

Latent Dirichlet Allocation (LDA) Model

LDA is a generative probabilistic model for clustering collections of grouped discrete data such as corpus of documents. It is the most widely used topic model in commercial products (Jiang et al. 2018). Fig. 1 shows an LDA model for text mining. The notations are explained in Table 1. To train an LDA model, we need to infer the posterior distribution of its parameters (document-topic distribution $\boldsymbol{\theta}$ and topic-word distribution $\boldsymbol{\phi}$) from the documents. A popular category of training algorithms is sampling methods such as Gibbs sampling (GS) and Metropolis Hastings (MH) with alias tables (Li et al. 2014). The collapsed Gibbs sampler iteratively samples new topics z_{ij} for the j^{th} word w in the i^{th} document from the following full conditional distribution

$$p(z_{ij} = k|\cdot) \propto (m_{i,k}^{(-ij)} + \alpha_k)\frac{n_{k,word}^{(-ij)} + \beta_{word}}{\sum_{w'} n_{k,w'}^{(-ij)} + \beta_{w'}}$$

until the parameters converge, where $m_{i,k}^{(-ij)}$ is the count of topic k in document i excluding z_{ij}, $n_{k,w}^{(-ij)}$ is the count that word w is assigned with topic k excluding z_{ij}.

In a federated learning setting, the words in each documents are the most private data thus should not be leaked. Also, the parameters $\boldsymbol{\theta}$ and $\boldsymbol{z}$ should be stored and updated locally by each user because they are document-correlated and may contain private information. The untrusted data collector will keep the parameter $\boldsymbol{\phi}$, which is commonly used in other industrial applications and it contains no private information of users. We will explain in next section how to infer all the parameters without directly communicating the topic-word assignments (which will expose the words of a document) between the data collector and the users.

Local Differential Privacy (LDP)

Local Differential Privacy (LDP) (Evfimievski, Gehrke, and Srikant 2003) is a concept for privacy-preserving data collection without assuming a trusted data collector. It naturally meets the privacy requirement of our federated learning setting. An LDP mechanism adds controlled noise into users' private data before sending them to the untrusted data collector, so that certain privacy requirement is fulfilled. Formally, an LDP mechanism should satisfy the property below:

Definition 1 ((ϵ, δ)-LDP)**.** *A randomized algorithm (mechanism) $\mathcal{A} : X \to T$ is (ϵ, δ)-locally differentially private if for any $x, x' \in X$ and $v \in T$, we have*

$$Pr[A(x) = v] \leq e^{\epsilon} Pr[A(x') = v] + \delta$$

Table 1: Summary of symbol notations.

Symbol	Description
D	The collection of documents
N	Number of documents
M	Number of words in each document
K	Number of topics
$\boldsymbol{w}$	$N \times M$ document-word vector
$\mathcal{V}$	The vocabulary set
$\boldsymbol{z}$	$N \times M$ word-topic assignment vector
$\boldsymbol{\phi}$	Topic-word distribution
β	Hyperparameter, $\boldsymbol{\phi_k} \sim Dirichlet(\beta)$
$\boldsymbol{\theta}$	Document-topic distribution
α	Hyperparameter, $\boldsymbol{\theta_i} \sim Dirichlet(\alpha)$

One effective way to design an LDP mechanism is to exploit the randomized response (RR) technique (Warner 1965; Erlingsson, Pihur, and Korolova 2014). The primary idea is that a data collector collects data from users by asking binary questions to users, and each user responds a true answer in a randomized fashion. Specifically, the user flips a coin with a probability of head $1 - \eta$ and only reports the true answer if the coin turns head. RR can estimate unbiased results meanwhile satisfying ϵ-LDP by setting

$$\eta = \frac{1}{1 + e^{\epsilon}} \tag{1}$$

In this work, we design a novel LDP mechanism exploiting RR techniques to protect user data privacy in federated LDA learning while still achieving high model accuracy.

3 FedLDA Overview

Fig. 2 illustrates the workflow of FedLDA, our local differential privacy based solution to federated LDA learning with no trusted data collector. For ease of presentation, we assume N users and each user has only one document d_i. As mentioned before, in a federated learning setting, the document-topic distribution $\boldsymbol{\theta}$ and the latent variable $\boldsymbol{z}$ are stored and updated locally, *i.e.*, user i updates his/her own $\boldsymbol{\theta_i}$, while the untrustworthy data collector aims to infer the topic-word distribution $\boldsymbol{\phi}$. At each iteration during model training, the inference of $\boldsymbol{\phi}$ is partitioned into *local sampling* by each individual user and *global integration* by the data collector.

- *Local Sampling.* At iteration t, each user i will sample new word-topic assignments for all the words in his/her document based on the current topic-word distribution $\boldsymbol{\phi^{(t)}}$ and his/her own document-topic distribution $\boldsymbol{\theta_i^{(t)}}$. Then sampling methods such as GS and MH can be used (we use the parallel implementation of GS as an approximation). After finishing sampling all the topic assignments, user i calculates an updating vector of $\boldsymbol{\phi}$, denoted by $\boldsymbol{U_i^{(t)}}$. The updating vector $\boldsymbol{U_i^{(t)}}$ is then perturbed to protect privacy and then transmitted to the data collector. Algorithm. 1 shows the details of local sampling.
- *Global Integration.* At iteration t, the data collector collects and aggregates $\boldsymbol{U_i^{(t)}}$ from each user and updates $\boldsymbol{\phi}$

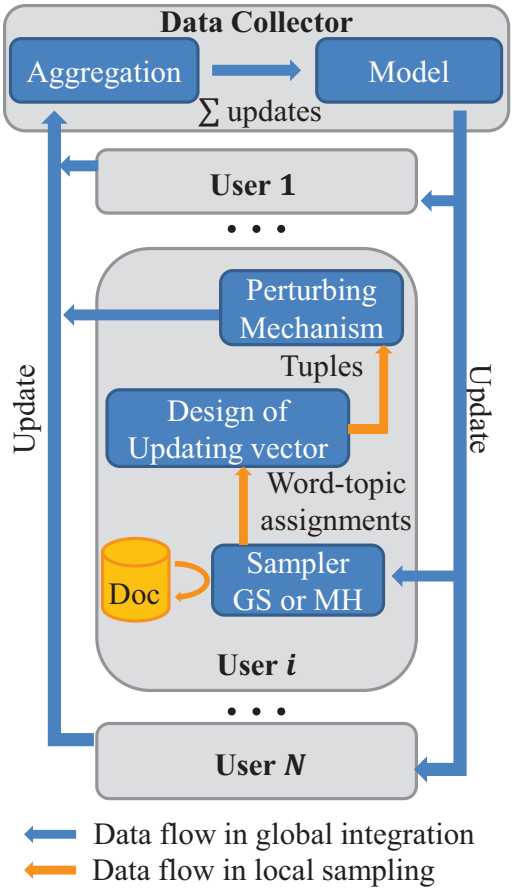


Figure 2: FedLDA overview.

before transmitting it to users for the next iteration. The details of global integration can be found in Algorithm. 2.

As discussed above, each user only transmits the perturbed updating vector $\boldsymbol{U}_i^{(t)}$ to the data collector during model training. Hence the design of the updating vector and its perturbing mechanism is crucial for the efficiency and effectiveness of our federated topic modeling framework, as we will explain in detail in the next two sections.

Note that although we take LDA as an instance to explain our federated learning framework, our proposed framework is not bounded to LDA and can be easily adapted to other topic models with sampling-based training.

4 Design of Updating Vector

The design of the updating vector $\boldsymbol{U}_i^{(t)}$ mainly considers the efficiency of our solution, particularly the communication cost between each user and the data collector. The updating vector is generated in two steps: *(i)* dense representation and *(ii)* padding & sampling.

Dense Representation

In local sampling (Algorithm. 1), $\boldsymbol{U}_i^{(t)}$ is a sparse vector of size $|\mathcal{V}| \times K$. Directly sending such a large vector leads

Algorithm 1: Local sampling

input : Local word vector $\boldsymbol{w}[i]$,topic assignment $\boldsymbol{z}[i]$,privacy parameter η, δ and $\boldsymbol{\phi}^{(t)}$
output: Updating vector $\boldsymbol{U}_i^{(t)}$

for *word-id* $j = 1, 2, \cdots, M$ **do**
  Sample a new topic $z[i][j]$ for word $w[i][j]$ with GS or MH;
  Update counts of assignments $n_{topic,word}$;
Update $\boldsymbol{\theta}_i^{(t)}$;
Calculate updating vector $\boldsymbol{U}_i^{(t)}$ with dense representation;
Pad $\boldsymbol{U}_i^{(t)}$ to fixed length M ;
for *each sampled tuple* (w, k_1, k_2) *in* $\boldsymbol{U}_i^{(t)}$ **do**
  Perturb (w, k_1, k_2) by Algorithm. 3;
return $\boldsymbol{U}_i^{(t)}$

Algorithm 2: Global integration

input : Privacy parameter η, δ
output: Model parameter $\boldsymbol{\phi}$

Randomly initialize $\boldsymbol{z}$ and the counters;
for *iteration* $t = 1, 2, 3, \cdots$ **do**
  for *user-id* $i = 1, 2, \cdots, N$ **do**
    Collect $\boldsymbol{U}_i^{(t)}$ from each user i by Algorithm. 1 ;
  for *topic-id* $k = 1, 2, \cdots, K$ **do**
    Get $n_{k,word}^{(i)}$ from $\boldsymbol{U}_i^{(t)}$;
    $n_{k,word} \leftarrow \sum_{i=1}^{N} n_{k,word}^{(i)}$;
  Update $\boldsymbol{\phi}^{(t)}$;
return $\boldsymbol{\phi}$

to unacceptable communication cost during model training. Note that the updating vector has at most M non-zero values, where M is the number of words in each document. Hence we propose a dense representation of the updating vector to reduce the communication cost.

Specifically, the non-zero values in $\boldsymbol{U}_i^{(t)}$ stand for the updates of word-topic assignments and each assignment can be replaced by a tuple (w, k_1, k_2), meaning that the topic assignment of word w has been changed from topic k_1 to k_2. We still use $\boldsymbol{U}_i^{(t)}$ to denote the new updating vector and its size is at most $3M$, which is much smaller than $|\mathcal{V}| \times K$. Finally we can calculate the parameter $\boldsymbol{\phi}$ by estimating the frequency of words under each topic in $\boldsymbol{U}_i^{(t)}$ (*i.e.*, $n_{topic,word}$) and then conducting a normalization, *i.e.*,

$$\phi_{topic,word} \leftarrow \frac{n_{topic,word} + \beta_{word}}{\sum_{w'} n_{topic,w'} + \beta_{w'}}$$

Padding & Sampling

Although dense representation of the updating vector significantly saves communication cost, the resulting length of

Algorithm 3: The RRP mechanism

input : A tuple (w, k_1, k_2), $\boldsymbol{\phi}$, $\boldsymbol{\theta}_i$
output: Perturbed tuple (w', k_1, k_2)

1 Sample $x \sim Bernoulli(\eta)$;
2 **if** $x = 0$ **then**
3 $w_1 = w$;
4 **else**
5 Sample $k' \sim \boldsymbol{\theta}_i$;
6 Sample $w' \sim \boldsymbol{\phi}_{k'}$;
7 **if** $w' \in \Phi^r_{k'}$ **then**
8 $w_1 = w$;
9 **else**
10 $w_1 = w'$;
11 **return** (w_1, k_1, k_2)

each user's $\boldsymbol{U}_i^{(t)}$ is different, which violates the requirement of LDP. As a compensation, we apply padding & sampling techniques (Qin et al. 2016; Wang, Li, and Jha 2018) to align the lengths of updating vectors and randomly sample tuples from each vector for the perturbation and uploading later. The padding size is set to M in our solution, as each updating vector has at most M tuples. We pad the vectors whose sizes are smaller than M with dummy tuples. After that, we uniformly sample l ($l \leq M$) tuples (without replacement) from each vector for perturbation.

Notice that the original padding and sampling techniques (Qin et al. 2016) only take one sample for aggregation, which will cause significant accuracy drop in the trained LDA model (see experiments). So we relax it by sampling l tuples. l/M is empirically set to 0.7, meaning a saving of communication cost by 30%.

5 Perturbing Mechanism

We now present our perturbing mechanism for each tuple (w, k_1, k_2) in the updating vector $\boldsymbol{U}_i^{(t)}$. The mechanism should be (ϵ, δ)-LDP and allows accurate LDA training.

Limitation of Prior Mechanisms

A naive method to perturb w is to directly apply the kRR (Kairouz, Oh, and Viswanath 2014) mechanism, an LDP mechanism for categorical values leveraging the random response (RR) technique. However, applying the kRR mechanism to our federated topic modeling has two drawbacks.

- *Limited privacy level.* The privacy budget ϵ of the kRR mechanism should satisfy $\epsilon \geq \ln(|\mathcal{V}| - 1) + \ln(1/\eta - 1)$, where $|\mathcal{V}|$ is the dictionary size and η is the same as in Eq.(1). In many real topic modeling applications, $|\mathcal{V}|$ is large (*e.g.*, over $10,000$) and thus a large privacy budget is necessary. Conversely, given a privacy budget, the privacy level achieved is limited.
- *Ineffective for model training.* The kRR mechanism uniformly samples noisy items from the entire item set. In the context of topic modeling, it means some rare words will also be sampled due to the uniformly sampling. Consequently, many meaningless tuples will be generated which may never appear in the sampling process before perturbing. Hence it will introduce too much noise and make the LDA model hard to converge (see experiments).

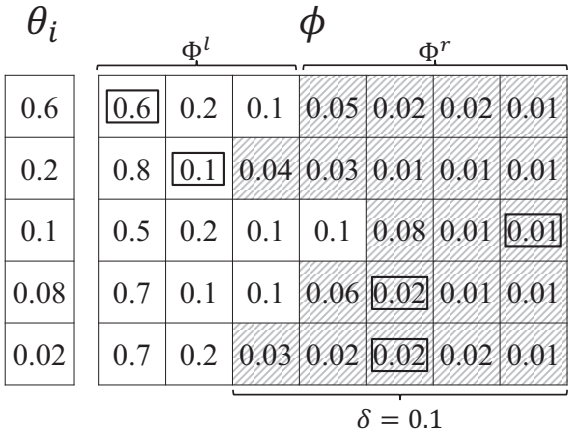


Figure 3: An example of deserted sets (unshaded part of $\boldsymbol{\phi}$) and truncated sets (shaded part of $\boldsymbol{\phi}$) when $\delta = 0.1$. Words are sorted in descending order by their proportions. Numbers with rectangles refer to the same word "cat".

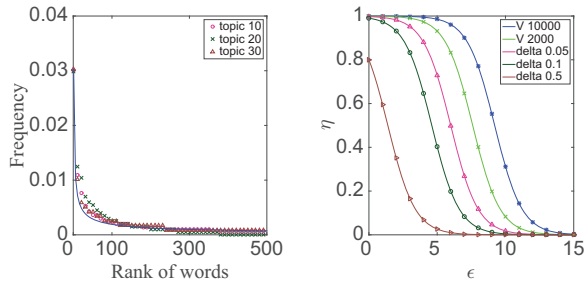
Random Response with Priori (RRP)

To overcome these drawbacks, we devise a novel perturbing mechanism based on RR, called Randomized Response with Priori (RRP). It has the following two advantages.

- RRP requires a privacy budget that is irrelevant to the dictionary size $|\mathcal{V}|$, if we assume the proportions of words under each topic follow the Zipf's law [1] (see Theorem. 1).
- RRP adaptively and non-uniformly samples the noisy terms from a priori drawn from the LDA model. Accordingly, the drop in model accuracy caused by the noise can be effectively reduced.

Algorithm. 3 shows our RRP mechanism. With probability $1 - \eta$ the word in each tuple will not be perturbed (Line 1-3). With probability η, we will first sample a topic k' from the document-topic distribution of that user, and then sample a new word w' from the topic-word distribution of topic k' (Line 5-6). In other words, we generate a new word w' according to the word generation process of LDA but we use the model under training instead. However, there are still a large number of candidate words. So we sort the words by their proportion in descending order and truncate the set of words. Let $\Phi^l_{k'}$ denote the truncated set of words in topic k' and $\Phi^r_{k'}$ denote the deserted set of words. We truncate it according to $\sum_{w \in \Phi^r_{k'}} \phi_{k'}(w) = \delta$. Only if the sampled word w' is from $\Phi^l_{k'}$ (with probability $1 - \delta$) will we replace it (Line 7-10). Finally it will output the perturbed tuple.

[1] https://en.wikipedia.org/wiki/Zipf%27s_law



(a) (b)

Figure 4: Empirical validations of assumptions for Theorem. 1: (a). Illustration of word frequencies by their ranks in ϕ, which approximately follows the Zipf's law; (b).Comparison between RRP with different δ and kRR with different vocabulary sizes.

Fig. 3 illustrates an example, where the numbers with rectangles correspond to the same word "cat". We assume $\delta = 0.1$ and the shaded part of each row in ϕ denotes the deserted set, while the unshaded parts are truncated sets. Suppose the original word in the tuple is "dog" and we fall into the else-branch of Algorithm. 3. And the probability that the perturbed word is "cat" will be $0.6 \times 0.6 + 0.2 \times 0.1 = 0.38$ as it lies in the deserted sets of the last three topics thus will not be sampled from them.

Theoretical Analysis

Now we analyze the privacy protection and the error on model parameter ϕ of our FedLDA. Due to limited space, all proofs are provided as supplementary materials.

Theorem 1. *By setting* $\eta = \frac{1}{\delta\delta_0 e^{\epsilon}+1}$*, Algorithm. 3 satisfies* $(\epsilon, 2\delta)$*-LDP, where* $\delta_0 = \delta - (\delta^{-\frac{1}{\gamma}} + 1)^{-\gamma}$*,* $\gamma \geq 1$ *is a constant.*

It means that an untrusted data collector cannot easily distinguish the updates from any two users, thus the privacy of each local document is protected. Note that in each iteration, every user has l tuples to submit. Therefore, by the composition property (Dwork 2006), given the total privacy budget ϵ', the privacy budget ϵ for one tuple should satisfy $\epsilon = \frac{1}{l}\epsilon'$. Also note that we make the assumption in Theorem. 1 that the word proportions satisfy the Zipf's law. This can be verified by Fig. 4a, an illustration of the frequency of words by descending ranking under different topics. We also show the relationship between η and ϵ for different δ in our mechanism and different vocabulary size in kRR in Fig. 4b. We observe that with the same η, the privacy budget for our mechanism is much smaller than kRR, due to that the vocabulary size is often very large in training LDA models.

Theorem 2. *Given a fixed topic, the expected relative error of the model parameter* ϕ_w *after perturbation is bounded by* $O(\eta k^2)$ *where* k *is the rank of* w *by sorting* ϕ_w *in descending order.*

Here we estimate the aggregation of each user's count of words under the same topic by a direct summation. Although it does not result in an unbiased estimation, we can still bound the expected error of model parameters. From the upper bound $O(\eta k^2)$, we find that if k is small, *i.e.*,, the word is very likely to be a keyword in that topic, the relative error is small and proportional to η. But if k is large, the relative error may be large. Nevertheless, it will not largely influence the overall performance of the model, as the parameters with large k are close to 0, and the model users only care about parameters with top ranks, *i.e.*, the keywords. It should also be noticed that the error is likely to accumulate by iterations. Considering that the LDA model performance can be effected by many other random factors such as errors from sampling algorithms which are hard to calculate, we will evaluate the model's final performance empirically. Our experimental results also show that the noise from our perturbing mechanism will slightly influence the overall performance.

6 Experiments

Experimental Setup

Datasets. We use three open datasets: Reviews [2], Emails [3] and Sentiments [4] (Maas et al. 2011). The dataset Emails contains 33,716 spam/non-spam emails with $M = 150$ and $|\mathcal{V}| = 3309$. The dataset Sentiments has 50,000 highly polar movie reviews with positive/negative sentiments, with $M = 150$ and $|\mathcal{V}| = 22574$. We use Reviews and Emails for evaluating our LDP mechanisms. Besides, we also evaluate our approach in two real applications: spam filtering on Emails and sentiment analysis on Sentiments.

Evaluation Metric. We use the perplexity to evaluate the performance of LDA models. The perplexity on a collection of documents D is defined as

$$perplexity = exp(-\frac{1}{NM} \sum_{i=1}^{N} \sum_{w \in d_i} \ln(\sum_{k=1}^{K} p(w|z_k)p(z_k|d_i)))$$

For the two real applications, we evaluate the performance of different methods by Precision, Recall, F1 score and AUC score, which are commonly used in binary classification tasks.

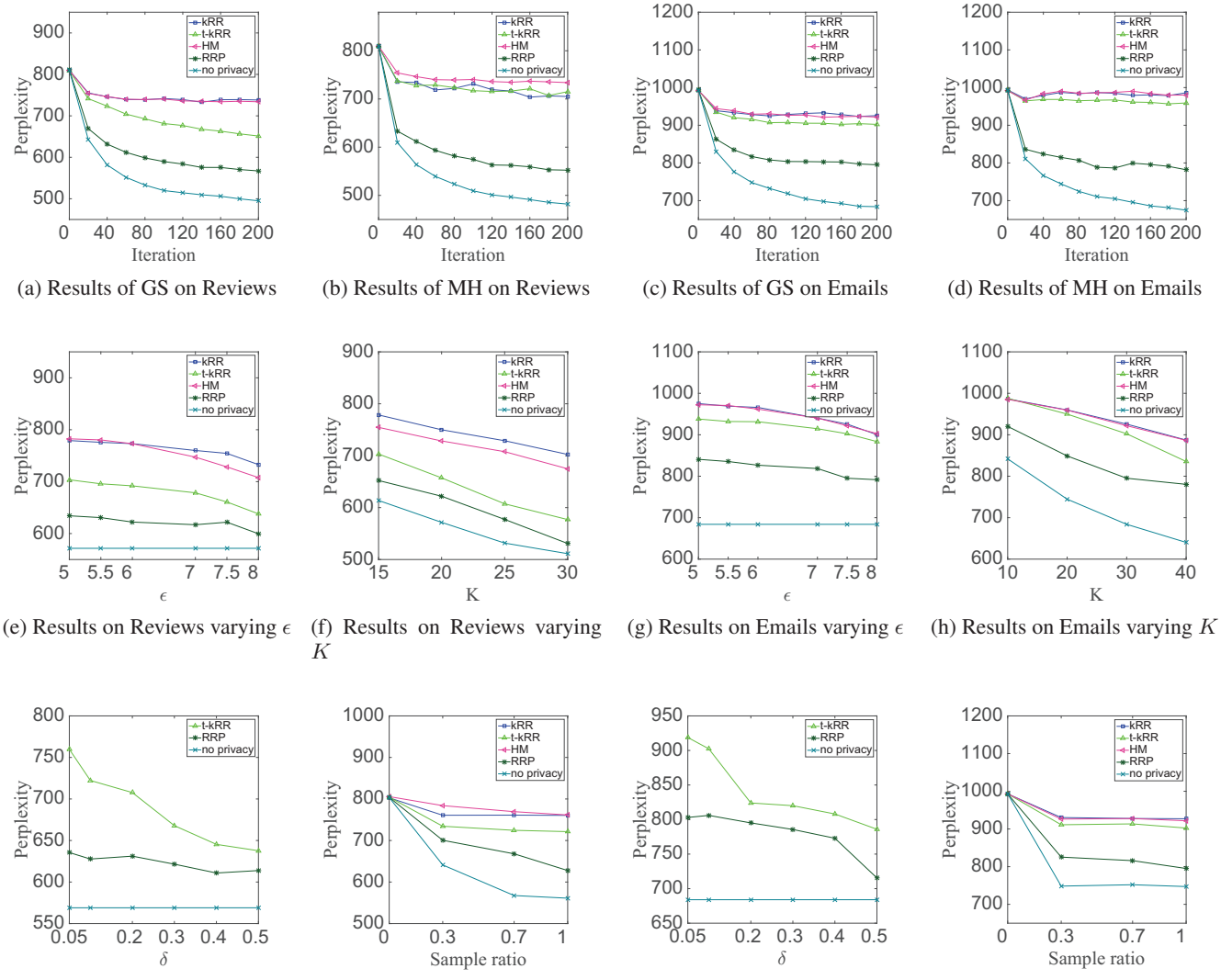
Baselines. We compare our RRP with kRR (Kairouz, Oh, and Viswanath 2014) and HM (Wang et al. 2019) using the same privacy budget ϵ. For fair comparison, we also compare with truncated kRR (t-kRR) which truncates the vocabulary set with the same δ based on kRR. In the real applications, we compare FedLDA with tradition LDA that does not have privacy-preserving techniques.

Parameter settings. We randomly sample 1K, 5K and 3K instances respectively from Reviews, Emails and Sentiments for evaluation. The default ϵ is 7.5 for all datasets and the default K is 20 for Reviews, 30 for Emails and 50 for

[2] https://www.kaggle.com/snap/amazon-fine-food-reviews

[3] https://www.kaggle.com/uciml/sms-spam-collection-dataset

[4] http://ai.stanford.edu/ amaas/data/sentiment/



(a) Results of GS on Reviews (b) Results of MH on Reviews (c) Results of GS on Emails (d) Results of MH on Emails

(e) Results on Reviews varying ϵ (f) Results on Reviews varying K (g) Results on Emails varying ϵ (h) Results on Emails varying K

(i) Results on Reviews varying δ (j) Results on Reviews varying sample ratio (k) Results on Emails varying δ (l) Results on Emails varying sample ratio

Figure 5: Experimental results on Reviews and Emails

Sentiments. The two real applications are both binary classification problem and we use logistic regression to solve the problems based on the LDA model parameters. We split training data and test data by $4:1$ for logistic regression and train both data for 100 iterations with the same solver.

Quantitative Evaluation

Impact of sampling algorithms. Fig. 5a, Fig. 5b, Fig. 5c, and Fig. 5d show the convergence using GS and MH as the sampling algorithm. On both datasets, MH converges faster than GS. RRP outperforms the other three baselines significantly for both sampling algorithms and the final results are very close to no-privacy. It can converge as fast as original GS or MH but may fall into sub-optimal values so the final results are slightly damaged.

Impact of ϵ. Fig. 5e and Fig. 5g show the impact of the privacy budget ϵ. We observe that with larger ϵ, *i.e.*, smaller η and lower level of privacy protection, the perplexity decreases for all the methods and RRP is still the best approach compared with baselines.

Impact of K. Fig. 5f and Fig. 5h show the impact of the number of topics K. With larger K, the perplexity of all the approaches decreases, which is in line with common sense. RRP still performs the best and is also close to no-privacy.

Impact of δ. Fig. 5i and Fig. 5k show the impact of δ. We find that with a larger δ, *i.e.*, a larger failure probability, the performance of t-kRR improves significantly. But for RRP, the results change slightly, which means that RRP is robust with δ. This is reasonable as we rank the words by frequency and the deserted set already contains the majority of words even if δ is small. In real applications, δ is expected to be small (less than 0.1), and the performance of our approaches

Table 2: Performance on real applications

		LDA	FedLDA $_{7.5}$	FedLDA $_{5.0}$
SF	Precision	0.868	0.781	0.736
	Recall	0.708	0.767	0.760
	F1 score	0.780	0.774	0.748
	AUC score	0.798	0.771	0.738
SA	Precision	0.777	0.774	0.761
	Recall	0.814	0.776	0.766
	F1 score	0.795	0.775	0.764
	AUC score	0.794	0.778	0.767

is good enough in such cases.

Impact of sample ratio l/M. Fig. 5j and Fig. 5l show the impact on l/M, which is the sample ratio in padding and sampling process. If the ratio equals 0, it means that we only sample one record, which is proved to be defective in the figures. The performance will improve with a larger sample ratio, but the improvement is less obvious with the ratio approaching 1. This verifies that with a ratio less than 1 (*e.g.*, 0.7), we can achieve similar results meanwhile the communication cost can be decreased by 30%.

Summary of results. For the model performance of the trained LDA models, RRP always outperforms the baselines in the converged perplexity and is also performs very close to LDA without privacy-preserving techniques. The results verify that FedLDA achieves competitive model performance while the data privacy is effectively protected at the same time.

Application-oriented Evaluation

Table 2 shows the results on spam filtering (SF) and sentiment analysis (SA) applications. We implement FedLDA with RRP $\epsilon = 7.5$ and $\epsilon = 5$ respectively. On spam filtering, we observe that the precision of FedLDA is lower than LDA but the recall is higher. The reductions of AUC are at most 2.7% if $\epsilon = 7.5$ (*i.e.*, 5% of the words will be perturbed in each iteration) and are at most 5% if $\epsilon = 5$ (*i.e.*, 40% of the words will be perturbed in each iteration). On sentiment analysis, the difference is even smaller, with only 1.6% of reduction to AUC if $\epsilon = 7.5$, which verifies that our approach is still effective and will not bring large damage to performance in real applications.

7 Related Work

Our work is related to the following categories of research.

Latent Dirichlet Allocation Models. The Latent Dirichlet Allocation (LDA) (Blei, Ng, and Jordan 2003) model and its extensions (Wang and McCallum 2006; Blei and McAuliffe 2007; Liu et al. 2011) are most widely adopted topic models in industrial applications. Sampling methods such as Gibbs sampling (GS) and Metropolis Hastings (MH) with alias tables (Li et al. 2014) are widely used for parameter estimation in LDA due to their easy adaptation to distributed and parallel platforms. Representative algorithms like WrapLDA (Chen et al. 2016), LightLDA (Yuan et al. 2015) and LDA* (Yu et al. 2017) have shown promising performance in industrial-scale applications. Our solution also uses sampling methods such as GS and MH so as to be suited for distributed and parallel topic modeling applications. Distributed LDA (Newman et al. 2009) is also related to our work. The new challenges are that a federated LDA has to deal with privacy-induced noise added to the data and cannot split or aggregate sensitive data and results freely as in a distributed setting. The novelty of our solution is to ensure the effectiveness and efficiency of (distributed) LDA in presence of such privacy constraints.

Local Differential Privacy. Differential privacy (DP) (Dwork 2006) is a popular privacy-preserving technique, which perturbs the original data by delicately injecting noise. Local Differential Privacy (LDP) (Evfimievski, Gehrke, and Srikant 2003) is a more strict measurement as the data collector is assumed to be untrustworthy. The randomized response (RR) technique (Warner 1965) is one of the most basic and representative mechanisms that satisfy LDP (Erlingsson, Pihur, and Korolova 2014). Based on RR techniques, RAPPOR (Erlingsson, Pihur, and Korolova 2014; Kairouz, Bonawitz, and Ramage 2016) is proposed for user data collection in Google Chrome. As extensions of tradition RR, kRR (Kairouz, Oh, and Viswanath 2014) is designed for collecting categorical data while LDPMiner (Qin et al. 2016) and LoPub (Ren et al. 2018) can deal with multidimensional data. Apart from frequency estimation, some other work focuses on mean estimation for numeric data with LDP (Duchi, Jordan, and Wainwright 2014; Ding, Kulkarni, and Yekhanin 2017). A recent work (Wang et al. 2019) proposes a hybrid mechanism that works for both categorical and numerical data and we have also implemented it in our experiments (*i.e.*, HM) for comparison.

Federated Learning. Federated learning (FL) (McMahan et al. 2016; 2017) was first proposed by Google for large-scale collaborative machine learning with privacy-preserving mechanisms among android users. A more general definition and taxonomy of federated machine learning is proposed in (Yang et al. 2019). The biggest challenge in FL is how to satisfy the strict privacy-preserving requirements when training the model. Some solutions consider collaborative machine learning under the secure multiparty computation (SMC) framework (Bonawitz et al. 2017; Cheng et al. 2019) and apply encryption protocols to satisfy the privacy requirements. These solutions are specially designed for gradient-based machine learning algorithms and are usually computation- and communication-heavy. Some other work (Park et al. 2016; McMahan et al. 2018) are based on traditional DP to reduce the extra overhead but a trusted data collector is required. To the best of our knowledge, we are the first to introduce LDP in federated learning and assume no trustworthy third party.

8 Conclusion

In this paper, we propose FedLDA, the first solution to federated LDA learning without assuming any trustworthy data collector. We devise a novel local differential privacy mechanism which provides theoretical guarantees on both user privacy and model accuracy. Extensive experiments on three

open datasets show that FedLDA can preserve the data privacy of users while allowing accurate training of LDA models. We envision our work as a first step to effective and practical federated topic modeling in real-world applications.

Acknowledgments

We are grateful to anonymous reviewers for their constructive comments. This work is partially supported by the National Science Foundation of China (NSFC) under Grant No. 61822201 and U1811463. Yongxin Tong is the corresponding author of this paper.

References

Airoldi, E. M.; Blei, D. M.; Fienberg, S. E.; and Xing, E. P. 2008. Mixed membership stochastic blockmodels. In *NIPS*, 33–40.

Blei, D. M., and McAuliffe, J. D. 2007. Supervised topic models. In *NIPS*, 121–128.

Blei, D. M.; Ng, A. Y.; and Jordan, M. I. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022.

Blei, D. M. 2012. Probabilistic topic models. *Commun. ACM* 55(4):77–84.

Bonawitz, K.; Ivanov, V.; Kreuter, B.; Marcedone, A.; McMahan, H. B.; Patel, S.; Ramage, D.; Segal, A.; and Seth, K. 2017. Practical secure aggregation for privacy-preserving machine learning. In *CCS*, 1175–1191.

Chen, J.; Li, K.; Zhu, J.; and Chen, W. 2016. Warplda: a cache efficient O(1) algorithm for latent dirichlet allocation. *PVLDB* 9(10):744–755.

Cheng, K.; Fan, T.; Jin, Y.; Liu, Y.; Chen, T.; and Yang, Q. 2019. Secureboost: A lossless federated learning framework. *CoRR* abs/1901.08755.

Ding, B.; Kulkarni, J.; and Yekhanin, S. 2017. Collecting telemetry data privately. In *NIPS*, 3574–3583.

Duchi, J. C.; Jordan, M. I.; and Wainwright, M. J. 2014. Privacy aware learning. *J. ACM* 61(6):38:1–38:57.

Dwork, C. 2006. Differential privacy. In *ICALP*, 1–12.

Erlingsson, Ú.; Pihur, V.; and Korolova, A. 2014. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *CCS*, 1054–1067.

Evfimievski, A. V.; Gehrke, J.; and Srikant, R. 2003. Limiting privacy breaches in privacy preserving data mining. In *PODS*, 211–222.

Jiang, D.; Song, Y.; Lian, R.; Bao, S.; Peng, J.; He, H.; and Wu, H. 2018. Familia: A configurable topic modeling framework for industrial text engineering. *CoRR* abs/1808.03733.

Kairouz, P.; Bonawitz, K.; and Ramage, D. 2016. Discrete distribution estimation under local privacy. In *ICML*, 2436–2444.

Kairouz, P.; Oh, S.; and Viswanath, P. 2014. Extremal mechanisms for local differential privacy. In *NIPS*, 2879–2887.

Li, A. Q.; Ahmed, A.; Ravi, S.; and Smola, A. J. 2014. Reducing the sampling complexity of topic models. In *SIGKDD*, 891–900.

Liu, Z.; Zhang, Y.; Chang, E. Y.; and Sun, M. 2011. PLDA+: parallel latent dirichlet allocation with data placement and pipeline processing. *TIST* 2(3):26:1–26:18.

Maas, A. L.; Daly, R. E.; Pham, P. T.; Huang, D.; Ng, A. Y.; and Potts, C. 2011. Learning word vectors for sentiment analysis. In *ACL*, 142–150.

McMahan, H. B.; Moore, E.; Ramage, D.; and y Arcas, B. A. 2016. Federated learning of deep networks using model averaging. *CoRR* abs/1602.05629.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 1273–1282.

McMahan, H. B.; Ramage, D.; Talwar, K.; and Zhang, L. 2018. Learning differentially private recurrent language models. In *ICLR*.

Newman, D.; Asuncion, A. U.; Smyth, P.; and Welling, M. 2009. Distributed algorithms for topic models. *J. Mach. Learn. Res.* 10:1801–1828.

Park, M.; Foulds, J. R.; Chaudhuri, K.; and Welling, M. 2016. Private topic modeling. *CoRR* abs/1609.04120.

Qin, Z.; Yang, Y.; Yu, T.; Khalil, I.; Xiao, X.; and Ren, K. 2016. Heavy hitter estimation over set-valued data with local differential privacy. In *CCS*, 192–203.

Ren, X.; Yu, C.; Yu, W.; Yang, S.; Yang, X.; McCann, J. A.; and Yu, P. S. 2018. Lopub: High-dimensional crowdsourced data publication with local differential privacy. *IEEE TIFS* 13(9):2151–2166.

Steyvers, M.; Smyth, P.; Rosen-Zvi, M.; and Griffiths, T. L. 2004. Probabilistic author-topic models for information discovery. In *SIGKDD*, 306–315.

Wang, X., and McCallum, A. 2006. Topics over time: a non-markov continuous-time model of topical trends. In *SIGKDD*, 424–433.

Wang, N.; Xiao, X.; Yang, Y.; Zhao, J.; Hui, S. C.; Shin, H.; Shin, J.; and Yu, G. 2019. Collecting and analyzing multidimensional data with local differential privacy. In *ICDE*, 638–649.

Wang, T.; Li, N.; and Jha, S. 2018. Locally differentially private frequent itemset mining. In *IEEE SP*, 127–143.

Warner, S. L. 1965. Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association* 60(309):63–69.

Yang, Q.; Liu, Y.; Chen, T.; and Tong, Y. 2019. Federated machine learning: Concept and applications. *ACM TIST* 10(2):12:1–12:19.

Yu, L.; Cui, B.; Zhang, C.; and Shao, Y. 2017. Lda*: A robust and large-scale topic modeling system. *PVLDB* 10(11):1406–1417.

Yuan, J.; Gao, F.; Ho, Q.; Dai, W.; Wei, J.; Zheng, X.; Xing, E. P.; Liu, T.; and Ma, W. 2015. Lightlda: Big topic models on modest computer clusters. In *WWW*, 1351–1361.