On the Intrinsic Differential Privacy of Bagging

Hongbin Liu , **Jinyuan Jia** and **Neil Zhenqiang Gong**
Duke University
{hongbin.liu, jinyuan.jia, neil.gong}@duke.edu

Abstract

Differentially private machine learning trains models while protecting privacy of the sensitive training data. The key to obtain differentially private models is to introduce noise/randomness to the training process. In particular, existing differentially private machine learning methods add noise to the training data, the gradients, the loss function, and/or the model itself. Bagging, a popular ensemble learning framework, randomly creates some subsamples of the training data, trains a base model for each subsample using a base learner, and takes majority vote among the base models when making predictions. Bagging has intrinsic randomness in the training process as it randomly creates subsamples. Our major theoretical results show that such intrinsic randomness already makes Bagging differentially private without the needs of additional noise. Moreover, we prove that if no assumptions about the base learner are made, our derived privacy guarantees are tight. We empirically evaluate Bagging on MNIST and CIFAR10. Our experimental results demonstrate that Bagging achieves significantly higher accuracies than state-of-the-art differentially private machine learning methods with the same privacy budgets.

1 Introduction

Machine learning has transformed various areas such as computer vision, natural language processing, healthcare, and cybersecurity. However, since a model is essentially some aggregate of the training data, the model may reveal rich information about the training data. For instance, with access to a model or only its prediction API, *model inversion* [Fredrikson *et al.*, 2015] can reconstruct (representative) training data of the model, while *membership inference* [Shokri *et al.*, 2017] can predict whether a given data point is among the model's training data or not. As a result, the model may compromise the privacy or confidentiality of the sensitive or proprietary training data such as electronic health records, location traces, and online digital behaviors. Moreover, various countries have passed laws such as General Data Protection Regulation (GDPR) [Voigt and Von dem Bussche, 2017] to regulate and protect data privacy. Therefore, privacy-preserving machine learning that trains models while protecting privacy of the training data is gaining increasing attention in both academia and industry.

(ϵ, δ)-differential privacy [Dwork *et al.*, 2014] has become a de facto standard for privacy-preserving data analytics. Many studies [Hamm *et al.*, 2016; Abadi *et al.*, 2016; Papernot *et al.*, 2016; Papernot *et al.*, 2018; Jordon *et al.*, 2018; Xie *et al.*, 2018] have extended (ϵ, δ)-differential privacy to machine learning. Roughly speaking, a machine learning method satisfies differential privacy if the learnt model does not change much when adding or removing one example in the training data. The key idea of differentially private machine learning is to introduce noise/randomness in the training process. Specifically, existing methods introduce randomness to the training data, the gradients when stochastic gradient descent is used to learn a model, the loss function, and/or the model itself. For instance, Differentially Private Stochastic Gradient Descent (DPSGD) [Abadi *et al.*, 2016] introduces well-calibrated Gaussian noise to the gradient computed from a random batch of the training data in each iteration when using stochastic gradient descent to learn a model. Private Aggregation of Teacher Ensembles (PATE) [Papernot *et al.*, 2016; Papernot *et al.*, 2018] trains multiple teacher models on pre-defined disjoint chunks of the sensitive training data. Then, PATE uses the teacher models to predict labels for examples in a non-sensitive public dataset, aggregates the labels for each example, and adds noise to the aggregated labels to achieve differential privacy. Finally, PATE trains a student model using the non-sensitive dataset with the aggregated labels predicted by teacher models. The student model satisfies (ϵ, δ)-differential privacy.

Our work. Bagging [Breiman, 1996], a popular ensemble learning framework, randomly creates some subsamples of the training data, trains a base model for each subsample using a base learner, and takes majority vote among the base models when making predictions. Bagging has intrinsic randomness in the training process as it randomly creates subsamples. Our major theoretical results have two folds. On one hand, we show that the intrinsic randomness of Bagging already makes it differentially private without the needs of additional noise. In particular, we prove that, for any base learner, Bagging with and without replacement respectively achieves $\big(N \cdot k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^{N \cdot k}\big)$-differential privacy

and $\left(N \cdot \ln \frac{n+1}{n+1-N \cdot k}, \frac{N \cdot k}{n}\right)$-differential privacy, where n is the training data size, k is the subsample size, and N is the number of base models. Moreover, we prove that if no assumptions about the base learner are made, our derived privacy guarantees are tight. On the other hand, our theoretical results indicate that Bagging can only provide differential privacy with $\delta \geq 1/n$. According to Dwork and Roth [Dwork *et al.*, 2014], $\delta \geq 1/n$ provides "just a few" privacy guarantee, which is equivalent to protecting the privacy of most training examples while compromising the privacy of just a few training examples. We empirically evaluate Bagging on MNIST and CIFAR10. For instance, Bagging achieves 79.55% testing accuracy on CIFAR10 with privacy budget $\epsilon = 0.2, \delta = 0.18$ (i.e., $k = 10000$ and $N = 1$). With the same privacy budget, DPSGD [Abadi *et al.*, 2016] only achieves 30.63% testing accuracy.

Our main contributions can be summarized as follows:

- We derive the (ϵ, δ)-differential privacy of Bagging.
- We prove our derived (ϵ, δ)-differential privacy of Bagging is tight if no extra assumptions about the base learner are given.
- We empirically compare Bagging with state-of-the-art privacy-preserving machine learning methods on MNIST and CIFAR10.

2 Background and Related Work

In this section, we first review the concept of (ϵ, δ)-differential privacy as well as its composition theorem and post-processing property. Then, we review Bagging [Breiman, 1996] and existing differentially private machine learning approaches.

2.1 Differential Privacy

Differential privacy [Dwork *et al.*, 2014] is defined in terms of *adjacent datasets*. In machine learning, a dataset consists of training examples. We call two training datasets adjacent datasets if there only exists one training example that appears in one dataset but is absent in the other. With the definition of adjacent datasets, we can introduce the definition of (ϵ, δ)-differential privacy as follows:

Definition 1 ((ϵ, δ)-differential privacy [Dwork *et al.*, 2014])**.** *A randomized mechanism $\mathcal{M}: \mathcal{D} \to \mathcal{R}$ satisfies (ϵ, δ)-differential privacy if for all adjacent datasets D, $D' \in \mathcal{D}$, and for all $S \subseteq \mathcal{R}$, it holds that:*

$$\Pr(\mathcal{M}(D) \in S) \leq e^{\epsilon} \cdot \Pr(\mathcal{M}(D') \in S) + \delta, \qquad (1)$$

where the randomness is taking over the mechanism $\mathcal{M}$.

In machine learning, the randomized mechanism $\mathcal{M}$ denotes the algorithm to train a model. (ϵ, δ)-differential privacy formalizes that the learnt model does not change much when adding or removing an arbitrary example in the training data. ϵ and δ quantify the upper bound of observable probability differences between the learnt models conditioned on adjacent datasets. If $\delta = 0$, we say that $\mathcal{M}$ is ϵ-differentially private [Dwork *et al.*, 2014]. Thereby, the additive term δ is considered as the probability at which the ϵ-differential privacy guarantee may be broken.

When several differential privacy mechanisms are composed, the differential privacy guarantee of the composed mechanism is the sum of the privacy guarantees of the individual mechanisms. Formally, differential privacy follows the standard composition theorem [Dwork *et al.*, 2014] below:

Theorem 1 (Composition theorem of (ϵ, δ)-differential privacy [Dwork *et al.*, 2014])**.** *Let $\mathcal{M}_i : \mathcal{D} \to \mathcal{R}_i$ be an (ϵ_i, δ_i)-differentially private algorithm for $i \in [k]$. If $\mathcal{M}_{[k]} : \mathcal{D} \to \prod_{i=1}^{k} \mathcal{R}_i$ is defined to be $\mathcal{M}_{[k]}(\mathcal{D}) = (\mathcal{M}_1(\mathcal{D}), \cdots, \mathcal{M}_k(\mathcal{D}))$, then $\mathcal{M}_{[k]}$ satisfies $\left(\sum_{i=1}^{k} \epsilon_i, \sum_{i=1}^{k} \delta_i\right)$-differential privacy.*

Proof. Please refer to the proof of Theorem 3.16 in Dwork and Roth [Dwork *et al.*, 2014] □

The composition theorem is a standard way to obtain privacy guarantees for repeated application of differentially private algorithms. Besides the composition theorem, (ϵ, δ)-differential privacy also has the following post-processing property:

Proposition 1 (Post-processing [Dwork *et al.*, 2014])**.** *Let $\mathcal{M} : \mathcal{D} \to R$ be a randomized algorithm that is (ϵ, δ)-differentially private. If $f : R \to R'$ is an arbitrary randomized or deterministic mapping. Then $f \circ \mathcal{M} : \mathcal{D} \to R'$ satisfies (ϵ, δ)-differential privacy.*

Proof. Please refer to the proof of Proposition 2.1 in Dwork and Roth [Dwork *et al.*, 2014]. □

The post-processing property ensures that the computation results of a differentially private mechanism can be safely released because any post-processing computation of originally (ϵ, δ)-differentially private algorithm will also be (ϵ, δ)-differentially private. The composition theorem and post-processing property make (ϵ, δ)-differential privacy applicable to analyze complex differentially private algorithms.

2.2 Bagging

Ensemble learning [Dietterich, 2000] tries to combine the base models produced by several learners into an ensemble that performs better than the original base learners. Bagging is a popular ensemble learning framework [Breiman, 1996], which we formally define as follows:

Definition 2 (Bagging (**B**ootstrap **Agg**regat**ing**) [Breiman, 1996])**.** *Given a training dataset D of size n, Bagging generates N subsamples D_i $(i = 1, 2, \cdots, N)$. Each subsample D_i randomly samples k examples from D with or without replacement. Then, Bagging trains a base model on each subsample D_i using a base learner. When predicting the label for a testing example, Bagging takes majority vote among the base models.*

If Bagging creates a subsample by randomly sampling k examples from the training dataset *with replacement*, some examples in the training dataset may be chosen multiple times. If Bagging creates a subsample by randomly sampling k examples from the training dataset *without replacement*, an

example in the training dataset may be selected at most once. In the following parts of this paper, we distinguish these two methods as Bagging *with replacement* and Bagging *without replacement*, respectively. In Section 3, our major theoretical results show that Bagging's intrinsic randomness brought by its re-sampling process satisfies (ϵ, δ)-differential privacy.

2.3 Differentially Private Machine Learning

Many studies [Hamm *et al.*, 2016; Abadi *et al.*, 2016; Papernot *et al.*, 2016; Papernot *et al.*, 2018; Jordon *et al.*, 2018; Xie *et al.*, 2018; Chaudhuri *et al.*, 2011; Kifer *et al.*, 2012; Song *et al.*, 2013; Bassily *et al.*, 2014; Wang *et al.*, 2017; Jordon *et al.*, 2018] have extended (ϵ, δ)-differential privacy to machine learning. Generally speaking, most studies satisfy differential privacy by introducing additive-noise mechanisms. Specifically, existing methods introduce noise/randomness to the training data, the gradients when stochastic gradient descent is used to learn a model, the loss function, and/or the model itself. For instance, Chaudhuri et al. [Chaudhuri *et al.*, 2011] proposed to add noise to the loss function and then minimize the noisy loss function using a standard optimization method. Kifer et al. [Kifer *et al.*, 2012] improved the utility of such loss function based perturbation method. Several other methods [Song *et al.*, 2013; Bassily *et al.*, 2014; Abadi *et al.*, 2016; Wang *et al.*, 2017] proposed to add noise to the gradient in each iteration of gradient descent or stochastic gradient descent. For instance, Abadi et al. [Abadi *et al.*, 2016] proposed DPSGD, which introduces well-calibrated Gaussian noise to the gradient computed from a random batch of the training data in each iteration when using stochastic gradient descent to learn a model. Moreover, they proposed moments accountant, which is a stronger accounting method to track the privacy loss for adding Gaussian noise than the standard composition theorem. Jordan et al. [Jordon *et al.*, 2018] proposed a method for the generator in generative adversarial networks [Goodfellow *et al.*, 2014] to generate synthetic data for training, which provides privacy guarantee for the original training dataset.

Papernot et al. [Papernot *et al.*, 2016; Papernot *et al.*, 2018] developed the PATE [Papernot *et al.*, 2016; Papernot *et al.*, 2018] framework, which trains multiple teacher models on pre-defined disjoint chunks of the sensitive training data and distills the teacher models to a student model using public non-sensitive data in a privacy-preserving way. The student model satisfies (ϵ, δ)-differential privacy. Jordan et al. [Jordon *et al.*, 2019] introduced a variant of PATE to improve the student model's accuracy by dividing the sensitive data several times (rather than just once in PATE) and learning teacher models on each chunk within each division. Note that PATE and its variant [Jordon *et al.*, 2019] divide the sensitive training data to chunks, which is different from Bagging that randomly creates subsamples. Moreover, our results essentially show that if PATE trains the teacher models using randomly created subsamples, then the teacher models (they can be treated as base models in Bagging) already satisfy (ϵ, δ)-differential privacy.

3 (ϵ, δ)-Differential Privacy of Bagging

In this section, we first intuitively explain why the randomness of the re-sampling process in Bagging may satisfy differential privacy. Then, we formally derive the (ϵ, δ)-differential privacy of Bagging in different cases and prove the tightness of our derived privacy bounds. Due to the space limitation, we put our detailed proofs in the Supplementary Material.

Roughly speaking, a machine learning method satisfies differential privacy if the learnt model does not change much when adding or removing one example in the training data. Suppose Bagging randomly samples k examples with replacement from the training dataset to create one subsample and trains one base model. When the size of the training dataset is large, Bagging is unlikely to sample the example, which is added or removed from the original training dataset. Specifically, when the size of the training dataset is n, Bagging with replacement would not select that added or removed example with probability of $\left(\frac{n-1}{n}\right)^k$ when creating a subsample. Therefore, adding or removing one example in the training dataset is unlikely to substantially affect the base model, making Bagging differentially private. In the following, we formally analyze the (ϵ, δ)-differential privacy guarantees of Bagging. Table 1 summarizes our derived (ϵ, δ)-differential privacy guarantees for Bagging with/without replacement.

Theorem 2 ((ϵ, δ)-differential privacy of Bagging with replacement when $N = 1$)**.** *Given a training dataset of size n and an arbitrary base learner, Bagging with replacement achieves $(k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^k)$-differential privacy when training one base model, where k is the subsample size.*

Proof. Please refer to Supplementary Material for details. In our proof, we first show the process of subsampling in Bagging with replacement achieves $(k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^k)$-differential privacy. Then we can view Bagging's training of the base model as post-processing of subsampling so that Bagging with replacement achieves $(k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^k)$-differential privacy. □

Note that our results are applicable to any base learner. One way to obtain the differential privacy guarantee of Bagging when training N base models is to apply the standard composition theorem in Theorem 1. In particular, based on the standard composition theorem of (ϵ, δ)-differential privacy, Bagging with replacement achieves $\left(N \cdot k \cdot \ln \frac{n+1}{n}, N \cdot \left(1 - (\frac{n-1}{n})^k\right)\right)$-differential privacy when training N base models. However, this differential privacy guarantee from the standard composition theorem is loose. In the following theorem, we show that Bagging with N base models achieves better differential privacy guarantees than that indicated by the standard composition theorem.

Theorem 3 ((ϵ, δ)-differential privacy of Bagging with replacement when $N>1$)**.** *Given a training dataset of size n and an arbitrary base learner, Bagging with replacement achieves $(N \cdot k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^{N \cdot k})$-differential privacy when training N base models, where k is the subsample size.*

Proof. We first sample $N \cdot k$ examples from the training dataset uniformly at randomly with replacement. Based on the proof of Theorem 2, the sampled $N \cdot k$ examples

Bagging with replacement	$\epsilon = N \cdot k \cdot \ln \frac{n+1}{n}$	$\delta = 1 - (\frac{n-1}{n})^{N \cdot k}$		Bagging without replacement	$\epsilon = \ln \frac{n+1}{n+1-N \cdot k}$	$\delta = \frac{N \cdot k}{n}$

Table 1: Our derived tight (ϵ, δ)-differential privacy guarantees for Bagging. n is the training dataset size, k is the subsample size, and N is the number of base models.

achieve $\big(N \cdot k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^{N \cdot k}\big)$-differential privacy. Then, we can evenly divide the $N \cdot k$ examples to N subsamples and train N base models. Note that we can view training the N base models as post-processing of the $N \cdot k$ examples, which does not incur extra privacy loss based on Proposition 1. Therefore, the entire process achieves $(N \cdot k \cdot \ln \frac{n+1}{n}, 1 - (\frac{n-1}{n})^{N \cdot k})$-differential privacy. □

If Bagging randomly samples k examples from the training dataset without replacement to create each subsample, then Bagging has the following differential privacy guarantee:

Theorem 4 ((ϵ, δ)-differential privacy of Bagging without replacement)**.** *Given a training dataset of size n and an arbitrary base learner, Bagging without replacement achieves $(\ln \frac{n+1}{n+1-N \cdot k}, \frac{N \cdot k}{n})$-differential privacy when training N base models, where k is the subsample size.*

Proof. Please refer to Supplementary Material. □

Next, we show that our derived privacy guarantees of Bagging are tight if no extra assumptions are made on the base learner. More specifically, if no assumptions on the base learner are made, it is impossible to derive a δ that is smaller than ours for Bagging.

Theorem 5 (Tightness of δ for Bagging with replacement)**.** *For any $\delta < 1 - \big(\frac{n-1}{n}\big)^{N \cdot k}$, there exists a base learner such that Bagging with replacement cannot satisfy (ϵ, δ)-differential privacy for any ϵ.*

Proof. Our proof is based on constructing a counter-example base leaner that Bagging with replacement cannot achieve (ϵ, δ)-differential privacy for any ϵ when $\delta < 1 - (\frac{n-1}{n})^{N \cdot k}$. Please refer to Supplementary Material for details. □

Theorem 6 (Tightness of δ for Bagging without replacement)**.** *For any $\delta < \frac{N \cdot k}{n}$, there exists a base learner such that Bagging without replacement cannot satisfy (ϵ, δ)-differential privacy for any ϵ.*

Proof. Please refer to Supplementary Material. □

4 Evaluation

We compare Bagging with DPSGD [Abadi *et al.*, 2016] and PATE [Papernot *et al.*, 2018] in different scenarios. We use the open-source implementations of DPSGD and PATE from their authors.

4.1 Experimental Setup

Two cases. Privacy-preserving machine learning aims to train a model while protecting the privacy of a sensitive training dataset. Depending on whether we have access to a public non-sensitive dataset, we consider the following two cases.

- **Case I: No access to a public non-sensitive dataset.** In this case, we only have access to the sensitive training dataset. In this case, PATE is not applicable. Therefore, we compare Bagging and DPSGD in this case.
- **Case II: Access to a public non-sensitive dataset.** In this case, we have access to a public non-sensitive dataset other than the sensitive training dataset. For instance, the sensitive training dataset could be CIFAR10 while the public non-sensitive dataset could be ImageNet. Therefore, we can leverage transfer learning to distill knowledge from the public non-sensitive dataset to boost the accuracy of the privacy-preserving model trained on the sensitive training dataset. In particular, we can first pretrain a model on the public dataset. Then, for DPSGD, we fine-tune the pretrained model on the sensitive training dataset using DPSGD. For Bagging, we fine-tune the pretrained model on subsamples of the sensitive training dataset to obtain the base models. For PATE, we train the teacher models and student model via fine tuning the pretrained model. Note that DPSGD and Bagging do not require the public non-sensitive dataset to have the same distribution as the sensitive training dataset. However, PATE further requires a public dataset that has the same distribution as the sensitive training dataset to train the student model. We call this public dataset *same-distribution public dataset*. Therefore, PATE has stronger assumptions than DPSGD and Bagging.

Datasets and models. We discuss our datasets and models for **Case I** and **Case II** separately.

- **Case I.** We adopt MNIST [LeCun *et al.*, 2010] and CIFAR10 [Krizhevsky *et al.*, 2009] as the sensitive training datasets. On MNIST, we use a simple convolutional neural network with two convolutional layers, each followed by a pooling layer, and a fully connected layer (Table 4 in Supplemental Material shows the details). For CIFAR10, we adopt the VGG16 [Simonyan and Zisserman, 2014] architecture.
- **Case II.** We still adopt MNIST and CIFAR10 as the sensitive training datasets. When MNIST is the sensitive training dataset, we adopt Fashion-MNIST [Xiao *et al.*, 2017] as the public non-sensitive dataset. When CIFAR10 is the sensitive training dataset, we assume ImageNet [Deng *et al.*, 2009] is the public non-sensitive dataset. PATE further requires a small same-distribution public dataset. For this purpose, we select the first 1,000 testing examples of MNIST (or CIFAR10) as the same-distribution public dataset when training PATE's student models on MNIST (or CIFAR10). Note that DPSGD, Bagging, and PATE are all evaluated on the remaining 9,000 testing examples of MNIST (or CIFAR10) in **Case**

Privacy Budget		Accuracy			Parameter Setting	
ϵ	δ	No Privacy	DPSGD	Bagging	σ	k
0.005	0.005	-	16.41%	**90.66%**	200	300
0.008	0.008	-	19.76%	**93.81%**	150	500
0.017	0.017	-	39.73%	**94.60%**	100	1,000
0.083	0.080	-	87.91%	**97.68%**	19	5,000
0.167	0.154	-	91.95%	**98.28%**	8	10,000
-	-	99.1%	-	-	-	-

(a) Comparison results on MNIST in Case I.

Privacy Budget		Accuracy			Parameter Setting	
ϵ	δ	No Privacy	DPSGD	Bagging	σ	k
0.02	0.02	-	12.74%	**41.63%**	72	1,000
0.1	0.095	-	15.78%	**59.76%**	11.5	5,000
0.2	0.181	-	27.86%	**65.25%**	4.9	10,000
0.4	0.33	-	40.96%	**70.75%**	2.18	20,000
0.6	0.45	-	46.53%	**73.95%**	1.4	30,000
-	-	80.82%	-	-	-	-

(b) Comparison results on CIFAR10 in Case I.

Table 2: Comparison results in Case I.

Method	ϵ	δ	Accuracy	Parameter Setting
No Privacy	-	-	98.83%	-
DPSGD	0.008	0.008	11.17%	$\sigma = 150$
PATE	0.57	0.008	74.37%	Queries to Aggregator $= 100$
Bagging	0.008	0.008	**90.20%**	$k = 500$
DPSGD	0.017	0.017	14.46%	$\sigma = 100$
PATE	0.66	0.017	78.42%	Queries to Aggregator $= 150$
Bagging	0.017	0.017	**93.59%**	$k = 1,000$
DPSGD	0.08	0.08	81.43%	$\sigma = 19$
PATE	0.63	0.08	83.31%	Queries to Aggregator $= 180$
Bagging	0.08	0.08	**96.87%**	$k = 5,000$

(a) Comparison results on MNIST in Case II.

Method	ϵ	δ	Accuracy	Parameter Setting
No Privacy	-	-	87.90%	-
DPSGD	0.02	0.02	11.97%	$\sigma = 72$
PATE	1.12	0.02	37.8%	Queries to Aggregator $= 100$
Bagging	0.02	0.02	**62.22%**	$k = 1,000$
DPSGD	0.1	0.095	21.08%	$\sigma = 11.5$
PATE	1.02	0.095	40.86%	Queries to Aggregator $= 130$
Bagging	0.1	0.095	**75.67%**	$k = 5,000$
DPSGD	0.2	0.18	30.63%	$\sigma = 4.9$
PATE	1.03	0.18	42.55%	Queries to Aggregator $= 170$
Bagging	0.2	0.18	**79.55%**	$k = 10,000$

(b) Comparison results on CIFAR10 in Case II.

Table 3: Comparison results in Case II.

II. On Fashion-MNIST, we pretrained a convolutional neural network, which has the same architecture as the model in **Case I** for MNIST. Moreover, we adopt the pretrained VGG16 model[1] [Simonyan and Zisserman, 2014] for the ImageNet dataset. DPSGD, PATE, and Bagging fine-tune these pretrained models on the sensitive training dataset following the standard fine-tuning procedure. In particular, we replace the last fully connected layer of a pretrained model as a new one that has the same number of classes as the sensitive training dataset. We then fine-tune the model using a learning rate that is 10 times smaller than that when training from scratch.

Parameter settings. We set training epochs=100 for both Bagging and DPSGD in both **Case I** and **Case II**. We adopt Bagging with replacement and $N = 1$ as the default setting in **Case I** and **Case II**. For PATE [Papernot *et al.*, 2018], we set the number of teachers to be 250, and both teacher and student models are trained for 1,000 epochs. Bagging, DPSGD, and PATE have different ways to control ϵ and δ. Next, we describe how to set their parameters to achieve a target level of ϵ and δ.

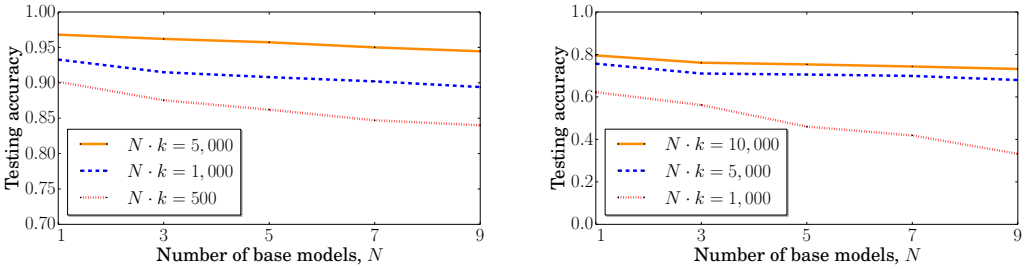
- Bagging. Given the training dataset size n and subsample size k, we can calculate the privacy budget (ϵ, δ) of Bagging based on Table 1.
- DPSGD [Abadi *et al.*, 2016]. We vary the standard deviation σ of the Gaussian noise used by DPSGD to achieve a target level of ϵ and δ.
- PATE [Papernot *et al.*, 2018]. For PATE with the Confident-GNMax aggregation mechanism [Papernot *et al.*, 2018], threshold T and noise parameter σ_1 are used for privately checking consensus of the teachers' predictions. Gaussian noise standard deviation σ_2 is used for the usual max-of-Gaussian [Papernot *et al.*, 2018]. Following the authors of PATE, we set T$= 200$, $\sigma_1 = 150$, and $\sigma_2 = 40$. We then vary the number of queries answered by the Confident-GNMax aggregator to achieve a target privacy budget ϵ and δ. We found that PATE cannot reach the small ϵ we set for Bagging and DPSGD, so we relax its ϵ to be around 1, which is much larger than the ϵ used by Bagging and DPSGD. In other words, we give additional advantages for PATE.

[1]https://github.com/keras-team/keras-applications/blob/master/keras_applications/vgg16.py

4.2 Experimental Results

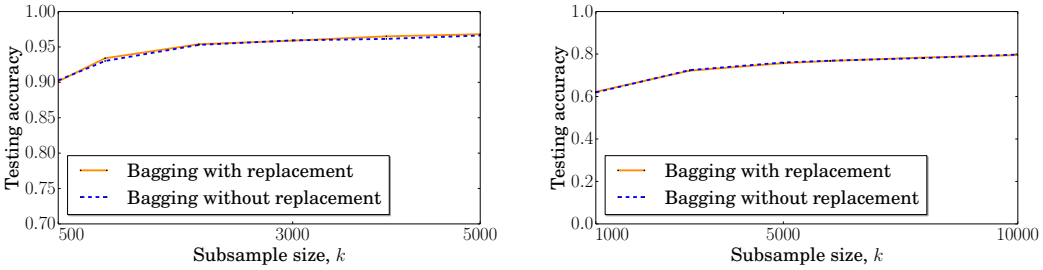
We first compare Bagging with DPSGD and PATE. Then, we evaluate different variants of Bagging.

Comparison results in Case I. Table 2a and 2b show the testing accuracies of DPSGD and Bagging (with replacement and $N = 1$) for different privacy budgets in Case I on MNIST and CIFAR10, respectively. The column "No Privacy" corresponds to models without privacy guarantees. First, we observe that Bagging achieves significantly higher testing accuracies than DPSGD under the same privacy budget. Second, increasing the subsample size k in Bagging is equivalent to decreasing the Gaussian noise scale in DPSGD, which provides weaker privacy guarantees and trains models with higher accuracies.



(a) MNIST (b) CIFAR10

Figure 1: Testing accuracy of Bagging when training N base models, where the privacy budget $N \cdot k$ is fixed.



(a) MNIST (b) CIFAR10

Figure 2: Bagging with replacement vs. Bagging without replacement.

Comparison results in Case II. Table 3a and 3b respectively show the testing accuracies of DPSGD, PATE, and Bagging for different privacy budgets in **Case II** on MNIST and CIFAR10, respectively. We have two observations. First, Bagging achieves significantly higher testing accuracies than DPSGD and PATE, even if PATE has weaker privacy guarantees. Second, via comparing Table 2b and Table 3b, we observe that Bagging achieves better accuracies in **Case II** than **Case I** for CIFAR10 when the public non-sensitive dataset is ImageNet, which means that transferring knowledge from a public non-sensitive dataset does improve accuracy of the model trained on the sensitive training dataset. DPSGD also achieves higher accuracies in **Case II** for CIFAR10 when the privacy budgets are larger than some threshold (e.g., 0.02). However, we didn't observe such accuracy improvement for MNIST in **Case II** when the public non-sensitive dataset is Fashion-MNIST. In fact, based on Table 2a and 3a, testing accuracies of DPSGD and Bagging may even decrease in **Case II**. We suspect the reason may be that the pretrained model for Fashion-MNIST is much simpler than that for ImageNet, which does not extract meaningful features.

Impact of k and N on Bagging. Figure 1 shows the testing accuracy of Bagging in **Case II** as we train more base models. We fix $N \cdot k$ in each curve in the graphs, so each curve has the same privacy budget independent of the number of base models. We observe that, given the same privacy budget, Bagging has lower accuracies when training more base models. The reason is that, given the same privacy budget, training more base models means that each base model is trained using less examples and thus less accurate.

Bagging with vs. without replacement. Figure 2 shows the testing accuracy of Bagging with vs. without replacement in **Case II** as we vary the subsample size k, where $N = 1$. The same subsample size k has very close privacy budgets (ϵ, δ) for Bagging with and without replacement, according to Table 1. Our results show that with or without replacement has negligible impact on Bagging.

5 Conclusion

In this work, we study the intrinsic (ϵ, δ)-differential privacy of Bagging. Our major theoretical results show that Bagging's intrinsic randomness from subsampling already makes Bagging differentially private without the needs of additional noise. We derive the (ϵ, δ)-differential privacy guarantees for Bagging with and without replacement. Moreover, we prove that if no assumptions about the base learner are made, our derived privacy guarantees are tight. We empirically evaluate Bagging on MNIST and CIFAR10. Our experimental results demonstrate that Bagging achieves significantly higher accuracies than state-of-the-art differentially private machine learning methods with the same privacy budget.

Acknowledgements

We thank the anonymous reviewers for insightful reviews. This work was supported by NSF grant No. 1937786.

References

[Abadi *et al.*, 2016] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016.

[Bassily *et al.*, 2014] Raef Bassily, Adam Smith, and Abhradeep Thakurta. Private empirical risk minimization: Efficient algorithms and tight error bounds. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 464–473. IEEE, 2014.

[Breiman, 1996] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[Chaudhuri *et al.*, 2011] Kamalika Chaudhuri, Claire Monteleoni, and Anand D Sarwate. Differentially private empirical risk minimization. *Journal of Machine Learning Research*, 12(Mar):1069–1109, 2011.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[Dietterich, 2000] Thomas G Dietterich. Ensemble methods in machine learning. In *International workshop on multiple classifier systems*, pages 1–15. Springer, 2000.

[Dwork *et al.*, 2014] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[Fredrikson *et al.*, 2015] Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *CCS*, 2015.

[Goodfellow *et al.*, 2014] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[Hamm *et al.*, 2016] Jihun Hamm, Yingjun Cao, and Mikhail Belkin. Learning privately from multiparty data. In *International Conference on Machine Learning*, pages 555–563, 2016.

[Jordon *et al.*, 2018] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Pate-gan: Generating synthetic data with differential privacy guarantees. 2018.

[Jordon *et al.*, 2019] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. Differentially private bagging: Improved utility and cheaper privacy than subsample-and-aggregate. In *Advances in Neural Information Processing Systems*, pages 4325–4334, 2019.

[Kifer *et al.*, 2012] Daniel Kifer, Adam Smith, and Abhradeep Thakurta. Private convex empirical risk minimization and high-dimensional regression. In *Conference on Learning Theory*, pages 25–1, 2012.

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[LeCun *et al.*, 2010] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.

[Papernot *et al.*, 2016] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[Papernot *et al.*, 2018] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

[Shokri *et al.*, 2017] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 3–18. IEEE, 2017.

[Simonyan and Zisserman, 2014] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[Song *et al.*, 2013] Shuang Song, Kamalika Chaudhuri, and Anand D Sarwate. Stochastic gradient descent with differentially private updates. In *2013 IEEE Global Conference on Signal and Information Processing*, pages 245–248. IEEE, 2013.

[Voigt and Von dem Bussche, 2017] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 2017.

[Wang *et al.*, 2017] Di Wang, Minwei Ye, and Jinhui Xu. Differentially private empirical risk minimization revisited: Faster and more general. In *Advances in Neural Information Processing Systems*, pages 2722–2731, 2017.

[Xiao *et al.*, 2017] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[Xie *et al.*, 2018] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739*, 2018.