Differential Privacy for Machine Learning project : Bayesian Differential Privacy for Machine Learning

Yangjiawei Xue [*1] **Clement Preaut** [*1] **Iwan Quemada** [*1]

1. Introduction

Differential privacy is widely used in machine learning to provide privacy guarantees for users' input data. However, in industries, although companies claim to use differential privacy (Dwork, 2006) algorithms to protect users' data, the privacy guarantees that these algorithms provide are actually too loose to efficiently prevent various attacks. This is because to achieve meaningful privacy guarantees in machine learning, companies often need to excessively reduce the accuracy. We can not really expect companies to sacrifice their model quality in favour of privacy. So in the paper of (Triastcyn & Faltings, 2020), Bayesian differential privacy has been proposed to provide much tighter privacy guarantees compared to traditional differential privacy at the same level of accuracy.

Pure ϵ-DP is hard to achieve in many realistic learning settings, the notion of approximate (ϵ, δ)-DP is used in machine learning. It is often achieved as a result of applying the Gaussian noise mechanism. A major step was done for privacy loss with the introduction of the moments accountant (Abadi et al., 2016). These results are adapted to Bayesian mechanism in the article.

[*]Equal contribution [1]Université Paris Dauphine - PSL, Paris, France. Correspondence to: Yangjiawei Xue <yangjiawei.xue@dauphine.eu>, Clement Preaut <clement.preaut@dauphine.eu>, Iwan Quemada <iwan.quemada@dauphine.eu>.

2. The state of the art of the subject tackled by the paper

The paper proposes a novel differential privacy——Bayesian differential privacy (BDP). Instead of treating all data points as equally likely like traditional differential privacy does, Bayesian differential privacy uses a probabilistic approach. The authors of this paper observe that machine learning models are designed and tuned for a particular data distribution. They consider typical scenarios where all sensitive data is drawn from the same distribution. As such, Bayesian differential privacy can calibrate noise to the data distribution and provide much tighter expected guarantees. In the paper, the authors also propose a novel privacy loss accounting method——Bayesian accountant.

3. How they advance the state of the art

Firstly they define Bayesian differential privacy and conduct theoretical analysis. Then they define the Bayesian accountant method. At the end they conduct experiments with four different datasets to demonstrate the good performances of Bayesian differential privacy.

3.1. Definition of Bayesian differential privacy

The definition of Bayesian differential privacy differs with traditional differential privacy in one thing: in Bayesian differential privacy, the different data point x' between two adjacent datasets follows a distribution which depends on the dataset. Similar to traditional differential privacy, there are strong Bayesian differential privacy and Bayesian differential privacy.

Definition 1 (*Strong Bayesian Differential Privacy*). A randomized function $\mathcal{A}$: $\mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$, range $\mathcal{R}$ and outcome $w = \mathcal{A}(\cdot)$, satisfies (ϵ_μ,δ_μ)-strong Bayesian differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$, differing in a single data point $x' \sim \mu(x)$, the following holds:

$$\Pr[L_{\mathcal{A}}(w, D, D') \geq \epsilon_\mu] \leq \delta_\mu, \tag{1}$$

where probability is taken over the randomness of the outcome w and the additional example x'.

Here, $L_{\mathcal{A}}(w, D, D')$ is the privacy loss defined as

$$L_{\mathcal{A}}(w, D, D') = \log \frac{p(w|D)}{p(w|D')} \tag{2}$$

where $p(w|D)$, $p(w|D')$ are private outcome distributions for corresponding datasets.

Note that here the probability is not only taken over the randomness of outcome w, but also taken over the randomness of x'. So the privacy parameters ϵ and δ will depend on the data distribution $\mu(x)$. In consequences, the interpretation of δ_μ will also change compared to traditional differential privacy. In Bayesian differential privacy, in addition to the privacy mechanism failures in the tails of outcome distributions, it also accounts for failures in the tails of data distributions.

Definition 2 (*Bayesian Differential Privacy*). A randomized function $\mathcal{A}$: $\mathcal{D} \to \mathcal{R}$ with domain $\mathcal{D}$ and range $\mathcal{R}$ satisfies (ϵ_μ, δ_μ)-Bayesian differential privacy if for any two adjacent datasets $D, D' \in \mathcal{D}$, differing in a single data point $x' \sim \mu(x)$ and for any set of outcomes $\mathcal{S}$ it holds that :

$$\Pr[\mathcal{A}(D) \in \mathcal{S}] \leq e^{\epsilon_\mu} \Pr[\mathcal{A}(D') \in \mathcal{S}] + \delta_\mu. \tag{3}$$

Similarly to traditional differential privacy, (ϵ_μ, δ_μ)-strong Bayesian differential privacy implies (ϵ_μ, δ_μ)-Bayesian differential privacy. Bayesian differential privacy repeats some basic properties of the classic differential privacy, such as composition, post-processing resilience and group privacy.

3.2. Bayesian privacy accounting

During the learning process of the machine learning models, tracking the privacy loss of each iteration is useful. Authors of the article draw inspiration from the moments accountant (Abadi et al., 2016) and develop a general accounting method for Bayesian differential privacy, the *Bayesian accountant*. It allows to provide tight bound on privacy loss. With the definition of Bayesian differential privacy, authors use different mathematical tricks such as Markov's inequality, Chernoff bound, the law of total expectation. Finally they obtain two notable equations. The first one determines how to compute ϵ_μ for a fixed δ_μ (or vice versa) for one invocation of the privacy mechanism :

$$\Pr[L \geq \epsilon_\mu] \leq \mathbb{E}_x[e^{\lambda D_{\lambda+1}[p(w|D)||p(w|D')] - \lambda\epsilon_\mu}]. \tag{4}$$

The second one gives the following theorem :

Theorem 1 (Advanced Composition). Let a learning algorithm run for T iterations. Denote by $w^{(1)}$ . . . $w^{(T)}$ a sequence of private learning outcomes at iterations 1, . . . , T, and $L^{(1:T)}$ the corresponding total privacy loss. Then,

$$\mathbb{E}[e^{\lambda L^{(1:T)}}] \leq \prod_{t=1}^{T} \mathbb{E}_x[e^{T\lambda D_{\lambda+1}(p_t||q_t)}]^{\frac{1}{T}}, \tag{5}$$

where $p_t = p(w^{(t)}|w^{(t-1)}, D)$, $q_t = p(w^{(t)}|w^{(t-1)}, D')$.

This theorem provides an upper bound on the total privacy loss due to computing expectation over the distribution of the same example over all iterations. The bound is found to be tight and straightforward to implement.

By defining the privacy loss of iteration t as $c_t(\lambda, T) = \log \mathbb{E}_x[e^{T\lambda D_{\lambda+1}(p_t||q_t)}]^{\frac{1}{T}}$, the total privacy cost of the learning process is then a sum of the costs of each iteration. We can now relate ϵ and δ parameters of BDP through the privacy cost.

Theorem 2 Let the algorithm produce a sequence of private learning outcomes $w^{(1)}$ . . . $w^{(T)}$ using a

known probability distribution $p(w^{(t)}|w^{(t-1)}, D)$. Then, for a fixed ϵ_μ (and **Corollary 1**):

$$\log \delta_\mu \leq \sum_{t=1}^{T} c_t(\lambda, T) - \lambda\epsilon_\mu \tag{6}$$

$$\implies \epsilon_\mu \leq \frac{1}{\lambda}\sum_{t=1}^{T} c_t(\lambda, T) - \frac{1}{\lambda}\log \delta_\mu \tag{7}$$

This **Theorem 2** (**Corollary 1**) allows us to combine privacy cost and convert to (ϵ_μ, δ_μ) guarantee. During training, the privacy cost $c_t(\lambda, T)$ for each iteration t is computed and then accumulated to compute (ϵ_μ, δ_μ) pair.

The article then describes how to compute the privacy cost by subsampled Gaussian mechanism:

Theorem 3 Given the Gaussian noise mechanism with the noise parameter σ and subsampling probability q, the privacy cost for $\lambda \in N$ at iteration t can be expressed as :

$$c_t(\lambda, T) = \max\{c_t^L(\lambda, T), c_t^R(\lambda, T)\}, \tag{8}$$

where

$$c_t^L(\lambda, T) = \frac{1}{T}\log \mathbb{E}_x[\mathbb{E}_{k\sim B(\lambda+1,q)}[e^{\frac{k^2-k}{2\sigma^2}||g_t - g_t'||^2}]^T] \tag{9}$$

$$c_t^R(\lambda, T) = \frac{1}{T}\log \mathbb{E}_x[\mathbb{E}_{k\sim B(\lambda,q)}[e^{\frac{k^2+k}{2\sigma^2}||g_t - g_t'||^2}]^T] \tag{10}$$

and $B(\lambda, q)$ is the binomial distribution with λ experiments and the probability of success q.

In the definition of the privacy cost, the data distribution is needed. However, it is impossible to know it in real world. We need to estimate the statistics of the distribution.

3.3. Privacy Cost Estimator

The authors propose a method to overestimate with high probability the privacy loss in each iteration, i.e. given a fixed γ, it returns the value that overestimates the true expectation with probability $1 - \gamma$.

In binary case and assuming the data comes from the maximum entropy distribution with the common mean ρ, and assuming a flat prior of ρ, authors get the posterior distribution of ρ, the estimator $\hat{\rho}$ is the $(1 - \gamma)$ quantile of the posterior distribution of ρ. Here γ is the estimator failure probability, $1 - \gamma$ is the estimator confidence.

In continuous case, a m-sample estimator of $c_t(\lambda, T)$ for continuous distributions with existing mean and variance has been proposed:

$$\hat{c}_t(\lambda, T) = \log[M(t) + \frac{F^{-1}(1 - \gamma, m - 1)}{\sqrt{m - 1}}S(t)] \tag{11}$$

where $M(t)$ and $S(t)$ are the sample mean and the sample standard deviation of $e^{\lambda\hat{D}_{\lambda+1}^{(t)}}$, $F^{-1}(1 - \gamma, m - 1)$ is the inverse of the Student's t-distribution CDF at $1 - \gamma$ with $m - 1$ degrees of freedom, and

$$\hat{D}_{\lambda+1}^{(t)} = \max\left\{D_{\lambda+1}(\hat{p}_t||\hat{q}_t), D_{\lambda+1}(\hat{q}_t||\hat{p}_t)\right\}, \tag{12}$$

$$\hat{p}_t = p(w^{(t)}|w^{(t-1)}, B^{(t)}), \tag{13}$$

$$\hat{q}_t = p(w^{(t)}|w^{(t-1)}, B^{(t)}\backslash\{x_i\}). \tag{14}$$

It can be proved that estimator $\hat{c}_t(\lambda, T)$ overestimates $c_t(\lambda, T)$ with probability $1 - \gamma$.

An equivalent estimator can be derived for other classes of distributions.

3.4. Experiments on real datasets

Firstly, authors compare the composition performance of Bayesian accountant with that of moments accountant. The result is that Bayesian accountant provides tighter bound on privacy loss than moments accountant.

Secondly, they compare the performances of Bayesian differential privacy and traditional differential privacy in DP-SGD algorithm with four different datasets. It turns out that Bayesian differential privacy can reach the same accuracy as traditional differential privacy with a much lower ϵ, and Bayesian differential privacy can reduce

significantly the successful rates of attacks. Because there is less noise to add during the learning process, the algorithm runs faster with Bayesian differential privacy.

4. Replication of the experiments

First, we can the results of the article on four datasets figure 1.

Dataset	Accuracy		Privacy	
	Baseline	Private	DP	BDP
MNIST	99%	96%	2.2 (0.898)	**0.95 (0.721)**
CIFAR10	86%	73%	8.0 (0.999)	**0.76 (0.681)**
Abalone	77%	76%	7.6 (0.999)	**0.61 (0.649)**
Adult	81%	81%	0.5 (0.623)	**0.2 (0.55)**

Figure 1. Caption

Our experiments were quite easy to replicate because the article proposes a Github repository with the formula and experiments implemented. We just had to run it in Google Colab by linking it to our drive.

5. The merits and limitations of the proposed approach

The advantages of Bayesian differential privacy is obvious: less noise is needed for comparable privacy guarantees, models train faster and can reach higher accuracy. Nevertheless, this approach is tested on different small datasets. It could be interesting to know if performances of this approach are scalable to larger dataset and to compare performances on different distribution taken from a baseline dataset. That is what will be done in the following part. A critic that we can do about the article is that it is not always clear. For example, authors define in theory what property must fulfill a mechanism to be Bayesian private, but they do not explicitly describe why the mechanism used in experiment is Bayesian. With analysis, we made the link between the mechanism and Bayesian privacy because the mechanism output relies on the batch, which depends exclusively on the data distribution. That is why a Gaussian subsampling mechanism can be considered as Bayesian deferentially private. Besides, the method to compute epsilon for basic DP is not well described in theory, and the method used in the related code leads to a bug.

6. Improvements

First we replicated the results of Bayesian-DP on CIFAR10, then we wanted to test Bayesian-DP on only 2 classes of CIFAR10. We took 2 classes that are similar and then 2 classes that are very different in distribution. We can measure this distribution thanks to figure 2. TSNE is a technique used to project the result of last classification layer of a model into a 2D-graph. In this way, we can see that some classes are clearly closer than others. It gives intuitive results in sense that, for example in figure 2, car is close to truck in a tighter way than cat, which is visually verified too. Secondly in the same way we will test the same process but on 2 distant classes. Then we wanted to see in what extend a Bayesian-DP mechanism is scalable. We tested it on CIFAR100 to see if it presents a benefit for the same type of data as CIFAR10, but in a more extended distribution.

For all our experiments, to train a predictive model, we first download the VGG16 pretrained model and we change the last layer dimension from 1000 to the number of class to predict. Then we start a training on the corresponding dataset by unlocking the weight update only on a few last layers. It allows students to shift the learning from ImageNet classes to CIFAR classes and keep the pre-trained VGG16 performance and feature extraction quality at the same time.

6.1. CIFAR10

We saw the results of the Bayesian-DP figure 3, obtain with batch size of 512, a learning rate of 0.001 and a training over 2 epochs. First of all, the accuracy decrease when the privacy is more

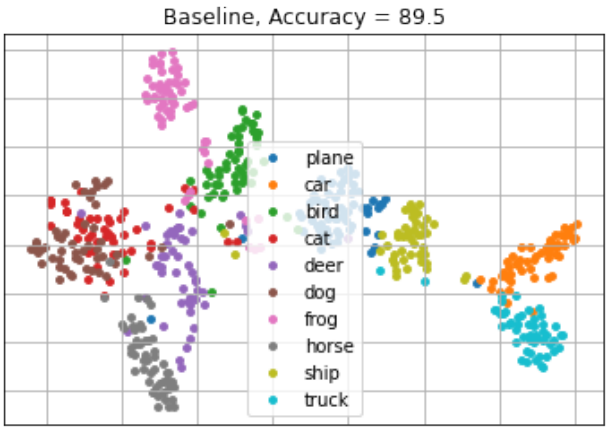

Figure 2. Distribution of data according to TSNE.

strong. Secondly, the best trade off in accuracy and privacy is the one given by the article : 73% of accuracy with $\epsilon_\mu = 0.76$.

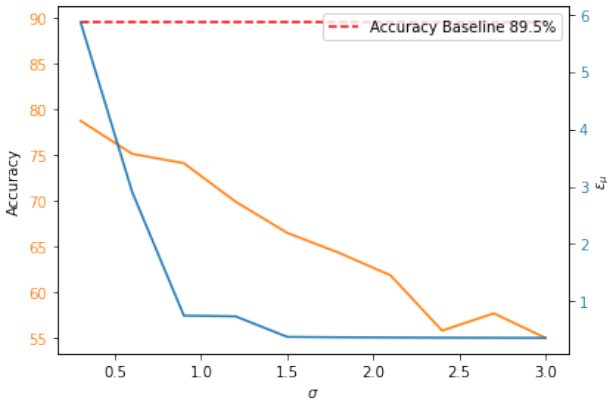

Figure 3. Evolution of the Accuracy (orange) and the ϵ_μ privacy (blue, for $\delta_\mu = 10^{-10}$) in function of the noise σ for CIFAR10.

Finally, for study the impact of the distribution of the dataset within the Bayesian-DP mechanism, we wanted to saw the distribution of CIFAR10 with an embedding representation using t-SNE algorithm on the output of the model and we see the results **??**. In conclusion we choose 'car' and 'truck' for study a close distribution and 'car' and 'cat' for a far one.

6.2. "CIFAR2"

We see our results figure 4 for similar classes and 5 for distant classes. In this experiment, we show the "best" trade-off between accuracy and privacy, which is around $\sigma = 1.2$. We can see, in case of distant classes distribution, we can obtain more privacy with lower loss of accuracy than the case of similar classes, which is because when we add noise on gradient the model separate the classes with more difficulties. But in both case, Bayesian-DP give us strong private privacy and keep good performance.

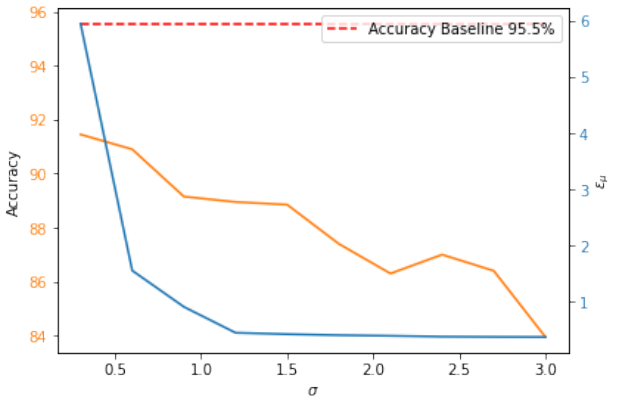

Figure 4. Similar distribution ('car' and 'truck') : Evolution of the Accuracy (orange) and the ϵ_μ privacy (blue, for $\delta_\mu = 10^{-10}$) in function of the noise σ for CIFAR2 with similar classes.

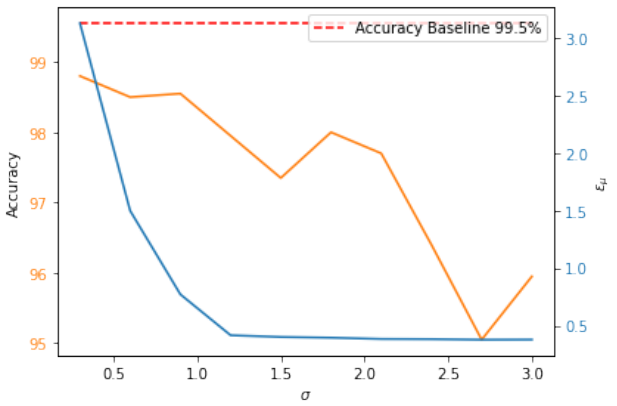

Figure 5. Different distribution ('car' and 'cat') : Evolution of the Accuracy (orange) and the ϵ_μ privacy (blue, for $\delta_\mu = 10^{-10}$) in function of the noise σ for CIFAR2 with distant classes.

6.3. CIFAR100

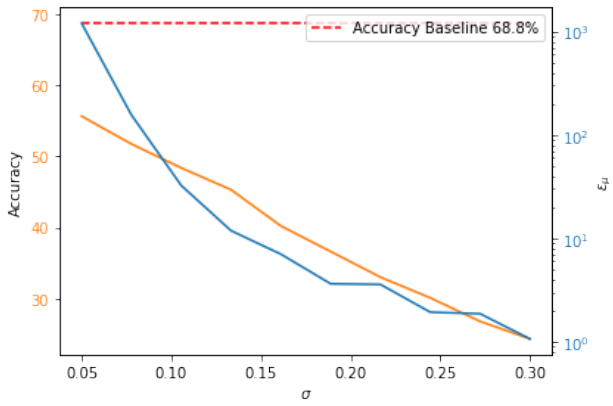

Figure 6. Evolution of the Accuracy (orange) and the ϵ_μ privacy (blue, for $\delta_\mu = 10^{-10}$) in function of the noise σ for CIFAR100.

Figure 6 is the same principle of accuracy and ϵ according to the noise variance σ, as for CIFAR10.

As usual, we would like to take the best trade-off between accuracy and privacy. Here the problem is that if we want an ϵ under 1, that is to say a relatively low probability of attack success, we fall under 30% of accuracy, which is not applicable in practice. And for accuracy over 50% the privacy reach an order of 10^2, so the privacy guarantee is null. So the mechanism is not scalable for larger distribution so not applicable in practice.

7. Conclusion

This paper showed the interest of adapting noise according to the distribution of data we want to protect. It proved several complex theorems and proposed a complete experiment setup. We saw in the improvement part how we can calibrate the noise to see if we can get a good trade-off between accuracy and privacy in dataset distribution variations. For larger distribution we would have to think about more robust mechanism if we want to get benefit from Bayesian differential privacy.

References

Abadi, M., Chu, A., Goodfellow, I., McMahan, H. B., Mironov, I., Talwar, K., and Zhang, L. Deep Learning with Differential Privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 308–318, October 2016. doi: 10.1145/2976749.2978318. URL http://arxiv.org/abs/1607.00133. arXiv:1607.00133 [cs, stat].

Dwork, C. Differential privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, pp. 1–12. Springer Verlag, July 2006. ISBN 3-540-35907-9. URL https://www.microsoft.com/en-us/research/publication/differential-privacy/.

Mironov, I. Renyi Differential Privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pp. 263–275, August 2017. doi: 10.1109/CSF.2017.11. URL http://arxiv.org/abs/1702.07476. arXiv:1702.07476 [cs].

Triastcyn, A. and Faltings, B. Bayesian differential privacy for machine learning. In *International Conference on Machine Learning*, pp. 9583–9592. PMLR, 2020.