Survey on Privacy in Machine Learning

Yihan Zhou, Tianyun Yu, Zichuan Wei*

January 3, 2020

Abstract

We conduct a survey on privacy in machine learning. First we show an examples of how people could use machine learning to invade privacy and how machine learning could cause some privacy issues which could never occur before. Then we give the rigorous definition of privacy in mathematics. In the end of this paper, we demonstrate three specific privacy preserving machine learning algorithms and one noise generation method.

1 Introduction

As artificial intelligence and machine learning develops rapidly today, such technology brings people convenience but also cause some problems. With the great power of machine learning, people can gain access to some private or protected data more easily. Some sensitive information hidden in public data could also be extracted by some advanced machine learning algorithm in a way people could hardly imagine in the past. To some extent, the development of machine learning threats everyone's privacy. At least we cannot trust some of the traditional privacy protection mechanisms and we need to be more careful of the data shared or generated online. R. McPherson[1] et al. demonstrated the power of deep learning in recogonition of images protected by various forms of obfuscation. This is an example of potential power of machine learning to invade privacy.

2 Definition of privacy

First of all, we need to define privacy rigorously in mathematics. Privacy in this paper actually means differential privacy, which is the most popular definition of privacy used today in machine learning area. Generally speaking, any change to one example in a database will not reveal any private information, so privacy is achieved. Before we give the formal definition of differential privacy, we need

*Zichuan Wei is not enrolled in this course, but he is interested in this topic and made some contributions to this paper

to know what is a mechanism and the definition of distance of two datasets.
Definition 1 A *mechanism*[2] is a random function takes a dataset D as an input and outputs a random variable $f(D)$.
For example, a function takes the dataset of heights of all students of University of Waterloo and outputs the average height of all students plus the value of a random variable ranging from 1 to 6 is a mechanism.
Definition 2 The *distance*[2] of two datasets D and D' is the minimum number of sample changes required to change D into D'. We denote the distance of two datasets D and D' by $d(D, D')$.
The distance of two datasets measures the discrepancy of two datasets. We can define differential privacy formally next:
Definition 3 A mechanism f gives ϵ-*differential privacy*[4] if for all datasets D and D' that satisfies $d(D, D') = 1$, and all $S \subseteq Range(f)$,

$$Pr(f(D) \in S) \leq e^{\epsilon} Pr(f(D') \in S)$$

Intuitively, this definition gives an upper bound of the probability of the output of the mechanism when only one example gets changed in the dataset. This means the removal of a single example in a dataset will not significantly improve the probability of any possible output of the mechanism. For example, the removal of the height of the student David in the UW student height dataset will not have a noticeable influence of the output of the mechanism defined above, thus the height of David cannot be leaked by this mechanism.

3 Privacy-preserving algorithms

In the last section we defined privacy formally in mathematics. In this section we will introduce two privacy-preserving algorithms.

3.1 simple algorithm

Chaudhuri et al.[3] provides two privay-preserving regularized logistic regression algorithm. The first algorithm adds noise to the classifier obtained by logistic regression, proportional to its sensitivity.

step 1: Compute w , the classifier obtained by regularized logistic regression on the labelled examples (x_1, y_1), . . . ,(x_n, y_n).

step 2: Pick a noise vector η according to the following density function: $h(\eta) \propto e^{-(n\epsilon\lambda/2)||\eta||}$. To pick such a vector, we choose the norm of from the $\Gamma(d, 2/n\epsilon\lambda)$ distribution, and the direction of uniformly at random.

step 3: Output $w^{\star} + \eta$.
Chaudhuri et al.[3] state that the learning performance degrades with decreasing λ and is poor when λ is very small. Thus we here introduce a privacy-preserving approximation to logistic regression, which has better performance bounds for small λ.

3.2 privacy-preserving logistic regression

Chaudhuri et al.[3] provides two privay-preserving regularized logistic regression algorithm. The second algorithm that is based on a new privacy-preserving technique. It requires us to solve a perturbed optimization problem and is not dependent on sensitivity. Note that in their paper, Chaudhuri et al. assumed that each value in the database is a vector with norm value at most one and the best separator pass through the origin. The output of this algorithm is a vector w that makes prediction $sign(w \cdot x)$ on a point x because it is a logistic regression algorithm. This algorithm has two steps and work as follows:

step 1: Picking a random vector b from the density function $h(b) \propto e^{-\frac{\epsilon}{2}\|b\|}$. In the paper Chaudhuri et al. picks the norm of b from the $\Gamma(d, \frac{2}{\epsilon})$ distribution and the direction of b uniformly at random.

step 2: Given examples $x_1, x_2, .., x_n$, with labels $y_1, y_2, ..., y_n$ and a regularization constant λ, computing $w^\star = argmin_w \frac{1}{2}\lambda w^T w + \frac{b^T w}{n} + \frac{1}{n}\sum_{i=1}^{n} log(1 + e^{-y_i w^T x_i})$.

$w^\star$ in step 2 is the separator this algorithm returns. This algorithm requires us to solve a convex linear programming problem very similar to logistic regression convex program, thus they have similar time complexity. In [3], Chaudhuri et al. proved this algorithm is ϵ-differential private. We will not present their proof in this survey. This algorithm is powerful because it can be generalized to more general convex optimization problems and it runs faster than sensitivity-based logistic regression algorithm.

3.3 privacy-preserving naive Bayes learning

Given a set of training samples, Bayes predictor classify a test sample $X = \{x_1, x_2, ..., x_n\}$ (x_i's are features of sample) using the most probable value $y_{map} \in \{y_1, y_2, ...\}$ for the given features, calculated using
$y_{map} = argmax_{y_i \in Y}(P(y_i|x_1, x_2, ..., x_n))$. By Bayes theorem,
$y_{map} = argmax_{y_i \in Y}(P(x_1, x_2, ..., x_n)P(y_i))$. To make calculations simple, Bayes predictor assumes that all features X are conditionally independent given label Y and we arrive at $y_{map} = argmax_{y_i \in Y}(P(y_i)\prod_{j \in \{1,...,n\}} P(x_j|y_i))$
The conditional probabilities $P(x_j|y_i)$ above need to be estimated using training samples, and the formulas are different for categorical features and numerical features. For categorical features $A \in X$ with m possible values $a_1, ..a_m$, $P(A = a_k|y_i) = \dfrac{l_{kj} + 1}{l + 2}$ where l_{kj} is the number of samples where $Y = y_i$ and $A = a_k$ and l is simply the number of samples where $Y = y_i$, the extra 1 on the numerator and and 2 on denominator are added to account for some value of the feature never appeared in the training samples, known as additive smoothing. For numerical features, we make another assumption that the distribution for the feature follows a normal distrubition to simplify the problem. We estimate the mean(μ) and standard deviation(σ) for the feature A using the sample mean and sample standard deviation in our training data, then $P(A = a|y_i)$ using the probability distribution function:

$$P(A = a|y_i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{a-\mu^2}{2\sigma^2}}$$

In 2013, Vaidya et al. [6] proposed a ϵ-*differential privacy* preserving algorithm for naive Bayes and the idea of the algorithm is to calculate the sensitivity based on the training samples and then add noise to the training samples according to Laplacian mechanism and compute the model described above.
For categorical features, the sensitivity of l_{kj} is 1 for all pairs of ¡a_k, y_i¿. For numerical features, Vaidya et al. introduce a third assumption that values of each feature are bounded by [$lower_j$, $upper_j$], if the bound covers most of the normal distribution then both assumptions on numerical features are approximately satisfied. Then the sensitivity for the mean(μ) is calculated using the formula $\frac{(upper_j - lower_j)}{l+1}$ and the sensitivity for the standard deviation(σ) is calculated $\sqrt{l * (upper_j - lower_j)/(l+1)}$ where l is the number of samples where $Y = y_i$, same as above.

4 Structure of ODO (Our Data, Ourselves) Protocol

Cynthia Dwork et al. [5] provide efficent distributed protocols for generating shares of random noise, secure against malicious participants.
Consider a database that is a collection of rows. A query is a function f mapping rows to the interval [0, 1]. The true answer to the query is the value obtained by applying f to each row and summing the results. By responding with an appropriately perturbed version of the true answer, privacy can be guaranteed. Cynthia Dwork et al.[5] provide efficient methods allowing the parties holding their own data to act autonomously and without a central trusted center, while simultaneously preventing malicious parties from interfering with the utility of the data.

1. **Share Summands:** On query f, the holder of d_i, the data in row i of the database, computes $f(d_i)$ and shares out this value using a $non-malleable\ verifiable\ secret\ sharing\ scheme$, $i = 1, ..., n$. The bits are represented as 0/1 values in $GF(q)$, for a large prime q. We denote this set $0, 1_{GF(q)}$ to make the choice of field clear.

2. **Verify Values:** Cooperatively verify that the shared values are legitimate (that is, in $0, 1_{GF(q)}$, when f is a predicate).

3. **Generate Noise Shares:** Cooperatively generate shares of appropriately distributed random noise.

4. **Sum All Shares:** Each participant adds together all the shares that it holds, obtaining a share of the noisy sum $\sum_i f(d_i)$ + **noise**. All arithmetic is in $GF(q)$.

5. **Reconstruct:** Cooperatively reconstruct the noisy sum using the reconstruction technique of the verifiable secret sharing scheme.

The main technical work is in Step 3. Cynthia Dwork et al. consider two types of noise, Gaussian and scaled symmetric exponential. In the latter distribution the probability of being at distance $|x|$ from the mean is proportional to $exp(|x|/R)$, the scale R determining how flat the distribution will be. In our case the mean will always be 0. Naturally, we must approximate these distributions using finite-precision arithmetic. The Gaussian and exponential distributions will be approximated, respectively, by the Binomial and Poisson distributions.

5 Conclusion

In this paper, we explored the ability of machine learning on privacy invasion, and then we introduced three basic privacy preserving machine learning algorithm. Finally we learned a distributed noise generation method. Privacy preserving machine learning would be increasingly popular as machine learning grows rapidly, and worth take a deeper look in the future. Our recommendation is concerning further interplay between differential privacy and machine learning and upper bounds of loss functions for differentially private algorithms.

6 Acknowledgement

We would like to thank Dr.Poupart for his enlightening machine learning lectures.

References

[1] R. McPherson, R. Shokri and V. Shmatikov, *Defeating Image Obfuscation with Deep Learning* arXiv:1609.00408v2, 2016.

[2] Z. Ji, Z.C. Lipton and C. Elkan, *Differential Privacy and Machine Learning: a Survey and Review* arXiv:1412.7584v1, 2014.

[3] K. Chaudhuri and C. Monteleoni, *Privacy-preserving logistic regression* Advances in Neural Information Processing Systems, 2009.

[4] C. Dwork, *Differential privacy* Encyclopedia of Cryptography and Security (2nd Ed.), pages 338340, 2011.

[5] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov and M. Naor, *Our data, ourselves: Privacy via distributed noise generation* International Conference on the Theory and Applications of Cryptographic Techniques, pages 486503, 2006.

[6] J. Vaidya, A. Basu, B. Shafiq and Y. Hong *Differentially Private Naive Bayes Classification* Proceedings - 2013 IEEE/WIC/ACM International Conference on Web Intelligence, 2013.