PRIVACY PRESERVING LINEAR REGRESSION

Thomas Patton
Department of Computer Science
Case Western Reserve University

I. INTRODUCTION

Linear regression is one of the oldest and most widely-used techniques for analyzing a problem. It provides a straightforward and understandable way to derive structure from problems whose classifications are best expressed as linear combinations of the features. The protection of individual privacy is important for many machine learning models and linear regression certainly falls into these confines.

As has been previously mentioned in this paper, the most popular method of measuring the privacy of an algorithm is ϵ-differential privacy. Although regression is an incredibly common tool, the methods for applying ϵ-differential privacy are few. This is because regression involves solving an optimization problem and analyzing the amount of noise required to ensure private results is difficult *(Yu-Xiang Wang)*. More specifically, attempting to add noise directly to the model weights requires an analysis on the sensitivity of those weights *(Yue Wang et al.)* and is generally not used due to its complexity.

This paper will center around the papers by *Yue Wang et al.* and *Dwork et al.* which seek to use the previously defined *Functional Mechanism* - a framework for enforcing ϵ-differential privacy by peturbing the objective function of the optimization problem rather than the results themselves. This peturbation is created by the sampling of a Laplace random variable and using it as a noise coefficient in the objective function. This noise-injection process was implemented for this paper and evaluated on different data sets. In addition to this, this paper performs a comparative analysis between this Laplacian noise-injection to Gaussian noise-injection for several values of ϵ based on a proposal by *Dwork et al.*. This leads to the conclusion that Laplacian noise and Gaussian noise perform differently on the two datasets and thus both should be investigated for their use on a particular problem.

II. PRELIMINARIES

A. Linear Regression

For consistency, I will use the notation as defined in *Yue Wang et al.*. Let D be a data set with n tuples $t_1, t_2, ...t_n$. Each tuple contains d attributes and a singular class label y. We bound each of these attributes to the domain of $[-1, 1]$ as well as the class label to the domain $[-1, 1]$. Then each tuple can be described as $t_i = (x_{i1}, x_{i2}..., x_{id}, y_i)$. The primary task of linear regression, then, is to find some set of weights $\omega = (\omega_1, ..., \omega_d)$ which minimizes some objective function of the model. The objective function for tuple t_i will be denoted as $f(t_i, \omega)$ and the objective function for the data set will follow as $f_D(\omega) = \sum_{t_i \in D} f(t_i, \omega)$. For this objective function it is common practice to use sum-squared error following the form $f_D(\omega) = \sum_{i=1}^{n}(y_i - x_i^T\omega)^2$. Let us then denote the set of optimal weights, ω^* as $\omega^* = argmin_\omega f_D(\omega) = argmin_\omega \sum_{i=1}^{n} f(t_i, \omega)$ ultimately giving us:

$$\omega^* = argmin_\omega f_D(\omega) = argmin_\omega \sum_{i=1}^{n}(y_i - x_i^T\omega)^2$$

It is then straightforward to determine these optimal ω^* using methods such as gradient descent. However, releasing such a model violates the privacy guarantee as information about ω^* can release information about the corresponding attributes *(Yue Wang et al.)*. To solve this *Yue Wang et al.* use the idea of peturbing the objective function itself rather than ω^* directly as was mentioned previously. This idea is called the *Functional Mechanism* and it follows as stated below.

B. Functional Mechanism

We begin with the idea of ϵ-differential privacy which was stated previously in this paper as:

$$Pr[f(D_1) = O] \leq e^{\epsilon} \cdot Pf[f(D_2) = O]$$

The functional mechanism is a technique which seeks to preserve this relationship by modifying the objective function through adding noise. We can then find the optimal result of this noisy objective function, $\bar{\omega}$ and release that information. Because our objective function can be complicated, we expand it into a polynomial form to make it more simple to work with.

We know from 2.1 that our model holds a weight vector ω which contains d values $(w_1, w_2, ..., w_d)$. Let $\phi(\omega)$ denote a product of $w_1, w_2, ...w_d$ listed as $\phi(\omega) = \omega_1^{c_1} \cdot \omega_2^{c_2} ... \cdot \omega_d^{c_d}$ for $c_1, c_2, ...c_d \in \mathbb{N}$. Now let Φ_j ($j \in \mathbb{N}$) be the set of all products $\omega_1, \omega_2, ..., \omega_d$ with degree j (i.e. the sum of all $c_i = j$). By the Stone-Weierstrass Theorem, any continuous and differentiable function can be written as a polynomial. This means for our objective funciton we can write it of the form:

$$f(t_i, \omega) = \sum_{j=0}^{J} \sum_{\phi \in \Phi_J} \lambda_{\phi t_i} \phi(\omega)$$

The major conclusion from this is that we can rewrite the equation for linear regression as:

$$\begin{aligned} f_D(\omega) &= \sum_{t_i \in D} (y_i - x_i^T \omega)^2 \\ &= \sum_{t_i \in D} y_i^2 - \sum_{j=1}^{d} (2 \sum_{t_i \in D} y_i x_{ij}) \omega_j \\ &+ \sum_{1 \leq j,l \leq d} (\sum_{t_i \in D} x_{ij} x_{il}) \omega_j \omega_l \end{aligned}$$

Here, we can recognize the $\lambda_{\phi t_i}$ as the coefficients of the ω terms e.g. for ω_j, $\lambda_{\phi t_i} = -2 y_i x_i j$ *(Yue Wang et al.)*.

(Yue Wang et al.) propose in their paper a method to inject noise into this polynomial form using a random sample from the Laplace distribution which is modeled as:

$$Lap(s) = \frac{1}{2s} e^{\frac{-|x-\mu|}{b}}$$

They then prove that by adding a noise factor of $Lap(\frac{2(d+1)^2}{\epsilon})$ that the new objective function $\bar{f}_D(\omega)$ satisfies ϵ-differential privacy.

C. *Other Methods*

This idea of the *Funcitonal Mechanism* is not the only method to ensure ϵ-differential privacy and so other leading techniques will be briefly enumerated. In his paper, *Yu-Xiang Wang* gives an overview of these. Briefly, they are:

- *Sufficient Statistics Peturbation* - Release X^TX and X^Ty differential privately and then output $\omega^* = X\hat{^T}X\bar{^{}}1\hat{X}y$.
- *Subsample and Aggregate* - Subsample many times and apply MLE to each subset. The randomize the aggregation of the results.
- *Posterior Sampling* - Output $\omega^* \sim P(\omega)\alpha e^{-\gamma(F(\omega)+0.5\lambda||\omega||^2)}$
- *Noisy Stochastic Gradient Descent* - Run stochastic gradient descent for a fixed number of iterations with additional Gaussian noise added to the gradient.

III. Methods

A. *Implementation*

For the implementation aspect of this project, the focus was this idea of creating a new, noisy objective function for which to solve. TensorFlow was used to create a “network” with a singular layer with one neuron and a linear activation function. In this way the network created a linear regression model, as the input vector combined linearly to form a binary output. Then a custom objective function was then created which followed the polynomial expansion of linear regression as described in *Section II B.* of this paper. This was done using SciPy by sampling a Laplace random variable and adding it to each of the λ_ϕ values. The parameters for the Laplace function that are specified in *Yue Wang et al.* were used.

B. *Research Extension*

The research extension of this project investigated the use of a different random variable other than that of the Laplace. In Appendix A. of their paper, *Dwork et al.* prove that injection of Gaussian noise satisfies ϵ-differential privacy. However, they only speculate on the performance of Laplace and Gaussian and do not derive any actual results. To my knowledge, there exist no significant papers comparing Laplace noise and Gaussian noise for linear regression problems. To implement this Gaussian noise-injection into this project, SciPy was again used to sample a Gaussian random variable using the parameters specified in *Dwork et al.*. This paper’s research hypothesis is that the Laplace mechanism will outperform the Gaussian mechanism as the scale parameters of each imply the Laplace will on average inject less noise.

C. *Procedure*

The analysis of this implementation focused on the two data sets of this paper: The *Bank Marketing* data set from *Moro et al.* and the *Breast Cancer Diagnostic* dataset from the University of Wisconsin. For more information about these datasets, see the introduction of this full paper. For each noise type (None, Laplace, Gaussian) the model found the optimal linear regression model using the ADAM optimizer from TensorFlow with 10 epochs, a batch size of 64, and a learning rate of 0.001. This was repeated this for some ϵ values from 10^{-3} to 10^3 and the mean and standard deviation 5-fold cross-validated accuracy for each trial was recorded. Additionally, a random seed was fixed to ensure reproducibility between trials.

IV. Results

The primary results of these experiments are displayed in *Table I* and *Figure I.*

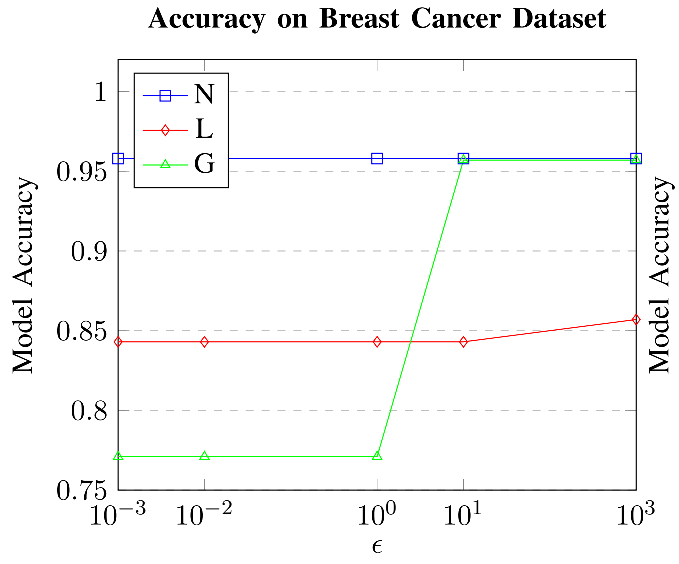
(a) Dataset: Breast Cancer

ϵ	None	Laplace	Gauss
0.001	0.958 ± 0.017	0.843 ± 0.017	0.771 ± 0.157
0.01	0.958 ± 0.017	0.843 ± 0.017	0.771 ± 0.157
1	0.958 ± 0.017	0.843 ± 0.017	0.771 ± 0.157
10	0.958 ± 0.017	0.843 ± 0.017	0.957 ± 0.017
1000	0.958 ± 0.017	0.857 ± 0.154	0.957 ± 0.007

(b) Dataset: Bank Marketing

	None	Laplace	Gauss
0.001	0.910 ± 0.007	0.901 ± 0.011	0.740 ± 0.353
0.01	0.910 ± 0.007	0.901 ± 0.011	0.740 ± 0.353
1	0.910 ± 0.007	0.901 ± 0.011	0.740 ± 0.353
10	0.910 ± 0.007	0.901 ± 0.011	0.770 ± 0.157
1000	0.910 ± 0.007	0.901 ± 0.011	0.958 ± 0.017

TABLE I: *Average 5-fold cross-validated accuracies with standard deviation*



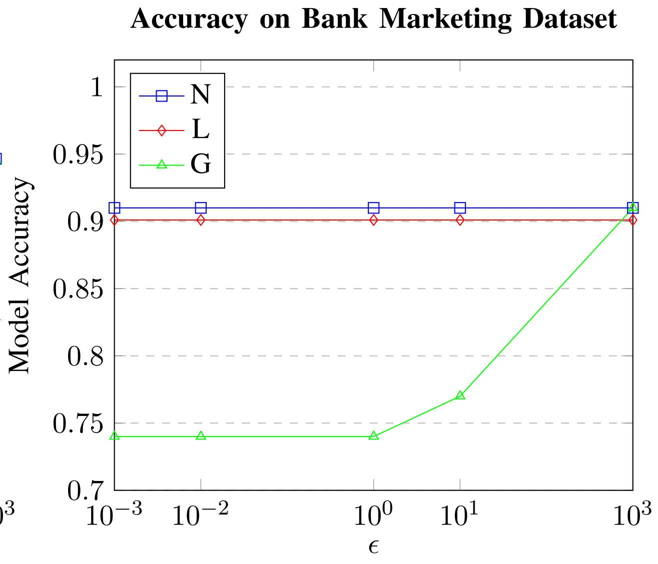


Fig. 1: *Graphs displaying the results of Table I.*

A. Variation of ϵ

The primary takeaway from these figures is that model accuracy increases as we increase the ϵ parameter. Since ϵ does not vary the result for a non-noisy objective function, we see the noisy results increase towards the non-noisy version as ϵ increases. This is intuitive, though, as larger values of ϵ result in proportionally less noise injected into the loss function. This also aligns with the result of *Yue Wang et al.*, though their improvement on the noise injection algorithm allows the convergence of the noisy function to the non-noisy one for lower values of ϵ than what is presented here. Another large takeaway from these figures is that while a very large ϵ was required to converge for the *Breast Cancer Prognosis* dataset, this was not the case for the *Bank Marketing* dataset. A possible reason for this is that the data for the *Bank Marketing* dataset is more linearly separable and thus even a noisy objective function can converge onto this minima.

B. Comparison of Laplace and Gaussian Noise

For the *Breast Cancer Prognosis* dataset, we see that the Gaussian noise injection works better than that of the Laplace, converging to the non-noisy solution at only $\epsilon = 10^1$. Even for the largest $\epsilon = 10^3$ the Laplace function has only just began convergence indicating that an unreasonable privacy budget would be required to get a reasonable solution here. However, for the *Bank Marketing* dataset the

Laplace noise injection model has converged to an answer close to the non-noisy solution for a very low $\epsilon = 10^{-3}$. The Gaussian noise injection model, though, does not begin convergence until 10^{1}. Based on these trials, the major conclusion seems to be that Gaussian noise is more reliable as it tends to converge more reliably. However, Laplacian noise does have potential to converge significantly earlier. This paper recommends using both additive noise mechanisms on a dataset and performing a comparative result to determine which is more appropriate.

V. FUTURE WORK AND CONCLUSIONS

A. Future Work

While this project does make some conclusions about privacy preservation in linear regression, there are many improvements still to be made. In their paper *Yue Wang et al.* enumerate a new noise injection method which works similarly but injects less noise into those features which do not need to maintain privacy. While they only show their algorithm with logistic regression, they do show convergence to the non-noisy objective for ϵ on the order of 10^{0}. Using their improvements with Gaussian noise-injection is a possible future direction.

It is also important to better acknowledge the amount of intrinsic randomness in these results. With better computing power one could perform the same experiments as above but determine average metrics over several different random seeds to gain a better understanding of the relationship between Laplace and Gaussian noise. Furthermore, these techniques could be extended to a wider range of datasets to draw deeper conclusions.

B. Conclusion

This paper has presented a detailed depth study of privacy preserving linear regression. It gave a background of linear regression, differential privacy, and showed how a peturbation of the polynomial objective function could be used to satisfy privacy constraints. It then performed a comparative analysis of Gaussian and Laplacian noise injection on two different datasets with varying values of ϵ to ultimately draw the conclusion that both noise mechanisms can be useful in different situations.