An Overview of Differential Privacy and Its Application to Machine Learning

Kyrylo Rudavskyy

Ryerson University
CPS 40A/B

May 22, 2020

Contents

1 Introduction

The subject of this report is the theory and application of Differential Privacy with a focus on machine learning. Differential Privacy(DP) - as introduced by Cynthia Dwork in [6] - is a framework that mathematically measures confidentiality loss resulting from database participation. Alternatively, DP can be viewed as a privacy guarantee. Regardless, DP releases database queries privately only under a set of strict assumptions. Specifically, the type and number of database queries must be set for a given anonymization approach to guarantee a certain constant of DP.

This report is composed of three parts. First, a theoretic framework of Differential Privacy will be constructed. This framework will cover the definition, properties, variations, and interpretations of Differential Privacy. Two main variations of DP discussed in this report are pure-DP or $(\varepsilon, \delta = 0) - DP$, and $(\varepsilon, \delta) - DP$. The later form of DP is a relaxation of the original definition. The interpretations of DP are an attempt to confine a somewhat abstract nature of the definition to real-world concepts and metrics.

The second part analyzes the application of DP to Machine Learning(ML). The section will cover theoretic foundations of differentially private machine learning, and demonstrate experimental results. An empirical study was performed to judge the performance of deferentially private machine learning (DPML). Experiments employed both a pure-DP framework and its relaxation via the *delta* parameter. The section ends with a brief survey of open-source software for DP.

The third section covers a case-study of DP used as a countermeasure to a theoretical model inversion attack. Fredrikson et al. [11] reproduced a study where linear regression was employed to create a pharmacogenetic algorithm from a medical database that contains genetic information. The authors then developed a model inversion attack on the regressor and deployed a differentially private version thereof as a countermeasure. They concluded a differentially private linear regression fails to balance privacy and accuracy to be an effective countermeasure.

This report demonstrates the result of applying a relaxed version of DP to Fredrikson's study. A hypothesis was developed that relaxation of DP can achieve Fredrikson's goals. This hypothesis resonates with the thesis of this report that differential privacy can be employed as a valid and feasible privacy-protecting technology in the machine learning context.

2 Differential Privacy

2.1 Definition

What is special about Differential Privacy is the fact that it is a mathematically formal approach. Other paradigms, such as de-identification, rely on a subjective notion of privacy [6]. Removal of personally identifiable information fails to fully protect individuals[6]. Moreover, Dinur and Nissim [4] demonstrated that publishing statistical data with less then $\Omega(n)$ noise will enable a database reconstruction attack. By approaching the subject formally, DP eliminates such risk. Before introducing the formal definition of DP, it is necessary to define two contingent concepts.

First, it is necessary to consider how much an outcome of a query can change when an entry is added or eliminated. Dwork refers to this as the *sensitivity* of a function or query [6]. If the queries that a database owner allows are counting queries, then a maximum change in output is one. For example, how many individuals live in a certain neighbourhood? If an entry is added or removed, then this query can only change by one.

Machine Learning algorithms perform much more sophisticated operations on data. Nevertheless, it is possible to analyze these operations and create a boundary. This will be mentioned in a greater detail below. Formally, *sensitivity* of a function is defined as

Definition 1. ([6]) For a query $f : \mathcal{D} \to R^d$, the L_1 sensitivity of f is

$$\Delta f = \max_{D,D'} ||f(D) - f(D')||_1 = \max_{D,D'} \sum_{i=1}^{d} |f_i(D) - f_i(D')|$$

for all databases D, D' differing by at most one entry.

The next step is to choose an appropriate randomization approach. One way to obfuscate data is to add random noise. However, the noise must conform to the definition of DP.

Definition 2. ([6])A randomized function $\mathcal{M}$ gives ε-differential privacy if for all data sets D and D' differing in at most one row, and all $\mathcal{S} \subseteq Range(\mathcal{M})$,

$$Pr[\mathcal{M}(D) \in \mathcal{S}] \leq exp(\varepsilon) \cdot Pr[\mathcal{M}(D') \in \mathcal{S}]$$

where the probability space in each case is over the coin flips of $\mathcal{M}$.

In other words, the noise added to a query must entail an ε bound on the probability of getting the same query outcome for D, D'. Dwork proposes

to add scaled Laplacian noise in order to achieve DP[6]. That is, given n queries it is necessary to add a random variable Y to each one s.t.

$$Y \sim Lap(\frac{\sum_{i=1}^{n} \Delta f_i}{\varepsilon})$$

It is important to observe that the type and number of queries have a direct effect on the magnitude of Y or the noise being added. The larger the number or sophistication of the queries, the bigger Y will be. This can be seen from the probability distribution function of Y, which is

$$P(x|b) = \frac{1}{2b} exp(-|x|/b)$$

with $\sigma^2 = 2b^2$ [6]. Since $b = (\sum_{i=1}^{n} \Delta f_i)/\varepsilon$ we can observe an inverse relationship between error and ε, and a direct relationship with sensitivity or the number of queries.

This makes intuitive sense because the more questions are asked or if the questions are very sophisticated, the more knowledge is derived from the database. Clearly, knowledge, in this case, is inversely related to privacy. The ε variable simply parameterizes the definition. Finally, it is important to note that once the privacy budget is exhausted, the database must not be queried anymore. Otherwise, the privacy guarantee will not hold.

2.2 Mechanisms

Laplace distribution is not the only source of random variables that can be used within the DP framework. Dwork points out that its main advantage is that it provides low sensitivity and is best suited for counting queries[7, p.33]. In general, the Laplace mechanism is applicable when the usefulness of the noisy query is proportional to the magnitude of the random variable[7, p.37]. For example, when querying the most popular musician in a database, the magnitude of the r.v.'s added to possible answers should not change the order of popularity.

Although on the surface, the Laplace mechanism seems to have limited utility, in reality, many sophisticated ML algorithms can be constructed from such queries [7, p.33]. One such example is Naive Bayes. The algorithm works by counting the frequencies of different classes and, as such, is suitable for Laplacian perturbation. This, however, sometimes is insufficient.

Queries that can have their utility significantly degraded by the magnitude of the r.v. do not perform well under Laplacian perturbation. Dwork presents as example a bidding process where small deviations can significantly change utility[7, p.37]. Similarly, queries that do not produce real numbers

are not suited for this mechanism. In such cases, Dwork recommends to use the exponential distribution to draw random variables [7, p.37].

The author notes that the **exponential distribution's** probability density function can be scaled by taking into consideration the utility of the answer in addition to the sensitivity. This way, utility-maximizing r.v.'s can be promoted during noise generation. The drawback of this mechanism is that the privacy budget is quickly depleted when the utility of the answer is superpolynomially large with respect to the problem parameters[7, p.37].

My research shows that many distributions can be proven differentially private sources of perturbation. Often it is better to pick a distribution with a specific problem in mind. For example, Dwork demonstrates that Gaussian distribution outperforms the ones above when perturbing second-moment matrices[8]. Similarly, Sheffet demonstrates that random variables drawn from a Wishart distribution outperform Gaussian noise in a multiple regression setting [18]. However, Sheffet notes that the Gaussian approach still works better for single regression. It is interesting to establish a deeper relationship between statistical distributions and problem classes. But this question is delegated for future research.

2.3 Properties

Two main properties of DP are post-processing and composition. **Post-processing** is a formal proposition that proves no matter what further analysis is performed on a differentially private data, the privacy guarantee will not be violated[7, p.19]. This property is important because it creates a sense of finality for DP as a confidentiality measure.

Composition theorems relate the privacy budget of multiple queries to the cumulative budget. In particular, it is proven that k $(\varepsilon, 0) - DP$ queries on raw data will provide a cumulative budget of $k\varepsilon$[7, p.19]. In case the queries have different privacy budgets, then the cumulative one will be equal to their sum[7, p.42]. This permits one to construct higher-level algorithms from simpler ones.

Using the Naive Bayes example again, the algorithm calculates variance and mean of data a number of times equal to the number of features. If a data set contains k features, then the variance and mean queries must be performed with a budget of $\frac{\varepsilon}{2k}$ [14]. The database may not be queried henceforth.

2.4 Relaxation

In practice, DP can be too strong of a privacy guarantee. Some applications may need a slightly weaker guarantee at the benefit of a higher privacy budget. This report will examine one such instance in a later section. On the other hand, an expected number of queries that will break DP is closer to $\sqrt{k}$ rather then k [3]. For these reasons a formal version of relaxed differential privacy was developed. This version relies on an additional parameter called delta.

Definition 3. ([7, p.17])A randomized function $\mathcal{M}$ gives (ε, δ)-differential privacy if for all data sets D and D' differing in at most one row, and all $\mathcal{S} \subseteq Range(\mathcal{M})$

$$Pr[\mathcal{M}(D) \in \mathcal{S}] \leq exp(\varepsilon) \cdot Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta$$

where the probability space in each case is over the coin flips of $\mathcal{M}$.

The δ parameter relates the probability of breaking $\varepsilon - DP$. Specifically, $\varepsilon - DP$ will hold with a probability of at least $1 - \delta$ [7, p.18]. In other words, at most δ of the database will leak. This may be tolerable in certain situations at the benefit of gaining additional privacy budget.

To address the **low likelihood event** of k queries violating $\varepsilon - DP$ Dwork and Roth demonstrate that δ must be less than "any polynomial in the size of the database" [7, p.18]. However, this guidance comes with database-specific caveats and should be employed with care!

The relaxed version of DP obeys pure-DP properties. However, the new parameter necessitates a reformulation of the composition property.

Theorem 1. *(Advanced Composition [7, p.50]) For all $\varepsilon, \delta, \delta' \geq 0$, and k $(\varepsilon, \delta) - DP$ queries the cumulative budget is $(\varepsilon', k\delta + \delta')$ for*

$$\varepsilon' = \sqrt{2k \ln(1/\delta')} \cdot \varepsilon + k\varepsilon(e^{\varepsilon} - 1)$$

where the probability space in each case is over the coin flips of $\mathcal{M}$.

The following corollary will build an inverse relationship than in the theorem above. It gives a per query budget given a fixed cumulative budget.

Corollary 1. (Advanced Composition [7, p.52]) Given $0 < \varepsilon' < 1$, $\delta' > 0$, and a cumulative budget $(\varepsilon', k\delta + \delta')$, then each of k mechanisms must be $(\varepsilon, \delta) - DP$, where

$$\varepsilon = \frac{\varepsilon'}{2\sqrt{2k \ln(1/\delta')}}$$

2.5 Interpretations

One of the challenges with Differential Privacy is comprehension. It is difficult to convey the sense of protection offered by $\varepsilon = 3$ to someone unfamiliar with mathematics. Those that understand the mathematics behind the definition of DP will have a sense of the privacy protection offered. However, this sense is still not sufficient for a rigorous approach to parameter selection.

For these reasons, this section will try to contextualize DP from a Bayesian and an Economic points of view. Within the **Bayesian framework**, ε modulates an adversary's posterior distribution with respect to any given database entry.

Proposition 1. ([3]) Let $\mathcal{M} : \mathcal{X}^n \to \mathcal{R}$ be ε-DP and P' denote an adversary's posterior distribution if $r \leftarrow \mathcal{M}(D')$ is observed instead of $r \leftarrow \mathcal{M}(D)$. Then for any transcript r on data-sets $D,\ D'$

$$P[D|r] \in e^{\pm 2\varepsilon} \cdot P'[D'|r]$$

An **economic interpretation** relates a database participant's cost and an analyst's budget to accuracy, ε, and database size. This argument will be presented for pure-DP, but can be extended to $(\varepsilon, \delta) - DP$. The extension can be performed using the Advanced Composition property.

Proposition 2. ([3][7, p.21]) Let $f : \mathcal{R} \to \mathbb{R}$ be an individual's cost function of participating in a database and $\mathcal{R}$ be an output space of a privacy mechanism. Then the expected cost to the individual is

$$\mathbb{E}[f(\mathcal{M}(D))] \leq exp(\varepsilon) \cdot \mathbb{E}[f(\mathcal{M}(D'))]$$

Cynthia Dwork observed that the increase in cost to an individual is bounded by a factor of $e^{\varepsilon} \approx (1 + \varepsilon)$ [7, p.21].

Now, following Hsu et al. [13], let $\mathcal{E}$ be a space of harmful events. The events being considered harmful are the ones related to data exposure. Let x_p, x_{np} mark individual's participation in a database. Then, $\forall e \in \mathcal{E}$,

$$Pr[e|x_p] \leq e^{\varepsilon} Pr[e|x_{np}]$$

Moreover, Hsu et al. bound the cost above as

$$e^{-\varepsilon} \mathop{\mathbb{E}}_{r \in \mathcal{R}}[f(r)|x_{np}] \leq \mathop{\mathbb{E}}_{r \in \mathcal{R}}[f(r)|x_p] \leq e^{\varepsilon} \mathop{\mathbb{E}}_{r \in \mathcal{R}}[f(r)|x_{np}]$$

Finally, by defining Marginal Cost to a database participant as

$$MC = \mathbb{E}[f(\mathcal{M}(D))] - \mathbb{E}[f(\mathcal{M}(D'))]$$

and combining with above the authors arrive at

$$MC(\varepsilon) \leq (e^{\varepsilon} - 1) \cdot \mathbb{E}[f(\mathcal{M}(D')]$$

If we denote an analyst's budget as B, accuracy as α, database size as N and an accuracy function as $A(\varepsilon, N)$ Hsu et al. arrive at the following system

$$MC(\varepsilon) \leq (e^{\varepsilon} - 1)N \cdot \mathbb{E}[f(\mathcal{M}(D')] \leq B,\ A(\varepsilon, n) \leq \alpha$$

Thus, if the accuracy of the mechanism and the cost to an individual can be estimated, then it is possible to solve for a range of acceptable epsilons.

The economic framework is essential to implementing DP in the real world. This framework can be adapted to analyze the cost-benefit relationship above from the data owner's point of view. For example, a health ministry may deem the release of certain data prohibitively costly to society. However, with the above framework, this cost is modulated by the ε parameter. Now a new cost-benefit relationship is created that may allow publication. Although this is a Faustian bargain, in situations of dire need, it may prove essential. Such situations are especially easy to imagine during the current COVID-19 pandemic.

The economic framework above rests on the following assumptions:

- an individual is assumed rational in an economic sense
- the model assumes that a study will inevitably take place
- events in $\mathcal{E}$ cannot directly observe database participation

3 Differentially Private Machine Learning

3.1 Motivation

Machine Learning and Differential Privacy share the same fundamental goal: they both aim to see the image on a mosaic but not the individual tiles. Turns out this shared interest in the overall patterns - but not specific points - can be exploited to produce efficient Differentially Private Machine Learning(DPML) algorithms. Indeed, it was demonstrated that DPML algorithms often approach their non-private counterparts in performance [7, p.217].

Dwork and Roth proved that most PAC-learnable functions can be learned privately with the same sample complexity and computational efficiency as their non-private counterparts [7, pp.216-222]. The authors note that parity functions are an exception. Also, they assume the privacy budget is $\varepsilon = \Theta(1)$ w.r.t. the number of samples.

A noteworthy implication of this result is that the private learning outcome is relatively independent of the number of samples. Once we converge to non-private learning, it is impossible to decrease epsilon by increasing the sample complexity. This is useful from an engineering point of view because it focuses the process on ε.

The applications of DPML are numerous, but this paper will primarily consider the following: publication of DP models, and prevention of model-inversion attacks. Going back to the above-mentioned result by Dinur and Nissim, the publication of ML models may lead to partial or full database reconstruction [4].

Zhao et al. identify a number of attacks on ML models that can pose a privacy threat [25]. These attacks will be discussed in section 4. Prevention of such attacks, and other privacy threats, are the motivation behind the development of DPML algorithms. This section will look at some of these algorithms and their empirical performance. The next section will finalize the report with a discussion of DP as a countermeasure to model inversion attacks.

Please note, all the algorithms in the experiments below were benchmarked against their non-private counterparts from scikit-learn [17].

3.2 Naive Bayes

The DP version of the popular ML algorithm was created by Vaidya et al. [20] and implemented by IBM [14]. It perturbs feature means and variances during Bayesian probability calculations. These two operations are the only points of contact with the data.

By adding variables from a Laplacian distribution that is scaled by the privacy budget, the number of queries, and sensitivity, these queries become differentially private. As a result, the output of the algorithm is differentially private as well.

The sensitivity here is bounded by the difference between column maximum and minimum. Scaling the entries to a $[0, 1]$ will bound sensitivity by one. This will save the privacy budget.

Another observation is that the algorithm deploys a number of DP mechanisms equal to the number of features. The privacy budget of each mechanism is scaled according to the composition properties from above. Decreasing the number of these mechanisms will improve the outcome. For Naive Bayes, this can be accomplished by decreasing the number of features.

DP version of Naive Bayes was tested using the IBM library [14]. However, since the library did not contain a relaxed-DP implementation, it was extended to scale the epsilon according to Corollary 1 from section 2.4 above.

UCI Car Evaluation data set was used with approx. 2000 entries [5]. The set contains six categorical features describing the cars, and a categorical independent variable with an expert evaluation of desirability. For the purpose of this test, the class of "acceptable" cars is the target variable. In addition, every row of the features matrix was divided by its $l2$ norm. (Please zoom in for better figure examination)

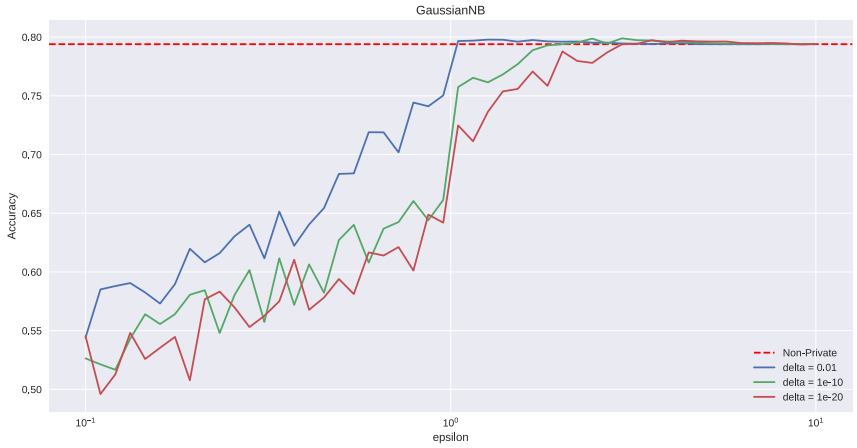


Figure 1: Differentially Private Naive Bayes Performance

Delta parameter at 10^{-20} is so small that it can be used to approximate pure-DP. When $\delta = 10^{-10}$ the overly strict privacy guarantee mentioned above is removed. This delta setting is very close to the low-likelihood bound from section 2.4. One can see that this slight adjustment helps the classifier converge to the non-private version much quicker. In this case, already at $\varepsilon = 1$, the performance is acceptable. Finally, when the probability of pure-DP holding is $\geq 99\%$ at $\delta = 0.01$, the classifier performance improves dramatically.

The set was trained using the default scikit-learn train/test split of 25% testing set [17]. Moreover, the classifier was trained over a 100 iteration and the mean was reported for each epsilon. It is worth noting that training performance is very volatile at low epsilon values. In addition, the volatility was present in the pure-DP version as well.

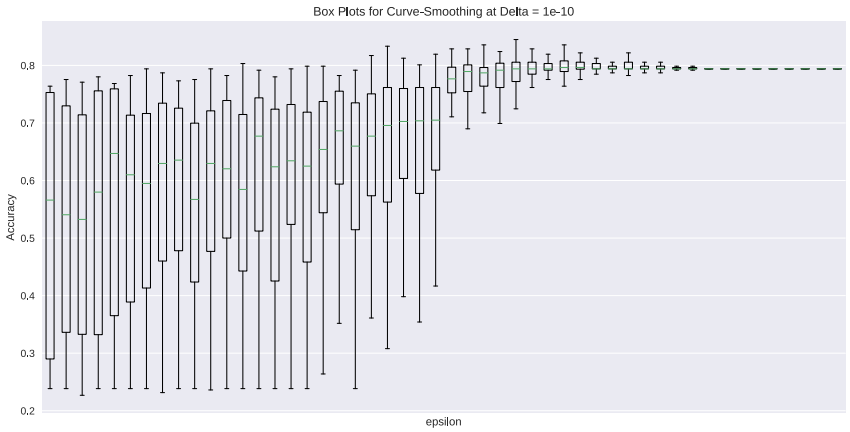


Figure 2: DPNB Volatility

3.3 Linear Regression

Dwork et al. [8], and Imtiaz and Sarwate [15] developed algorithms for a differentially private release of 2nd-moment matrices. These algorithms are called Analyze Gauss and SN Algorithm, respectively. They build matrix $A = [X|\mathbf{y}]$, which is a concatenation of the features matrix and the target variable. Laplacian and Wishart noise, respectively, is scaled, added to the product $A^T A$ and released.

The scaling of the distributions is proportional to sensitivity and epsilon, as discussed above. The delta parameter is factored according to advanced composition. Once noisy $A^T A$ is released, it is trivial to extract $X^T X$ and $X\mathbf{y}$.

It is notable that these algorithms almost release a noisy version of the original data. Moreover, due to DP post-processing property, DP guarantee will persist no matter what analysis or manipulation is performed on the matrix.

While Dwork was a pioneer in this field, SN Algorithm demonstrated superior performance [15]. SN Algorithm was tested on three datasets: Car Evaluation from above, California Housing (housing) [21][16] and Cover Type (covtype) [5]. California Housing contains approx. 20,000 entries and 8 numeric attributes. Cover Type is a very large set with half a million entries, 54 features, and 7 target classes. Only the last target class was used.

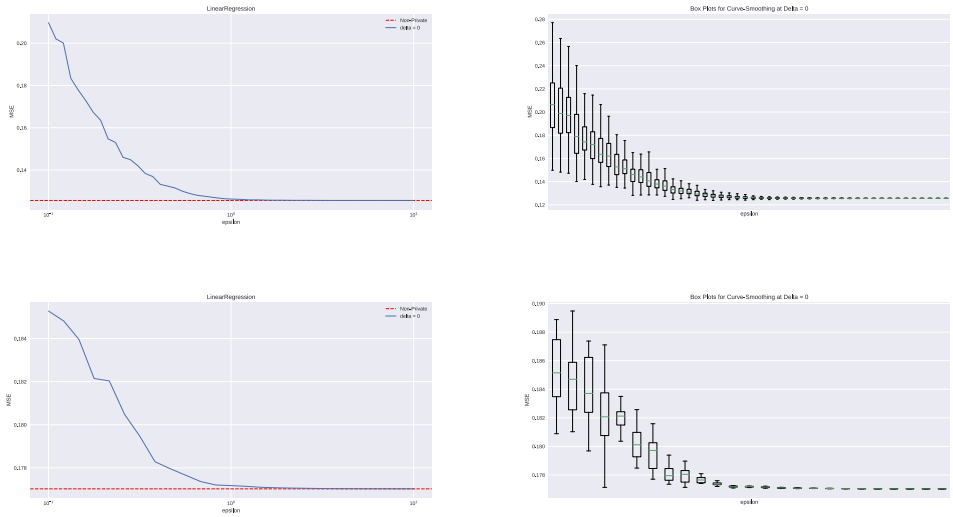
Housing and covtype set had their features scaled by subtracting mean and dividing by standard deviation. In all experiments here, centring was

performed before normalization. Also, feature rows were divided by their norms. The target variable is binary for covtype and did not require scaling since it is less or equal to one. The target vector of the housing set was divided by its maximum. Ten iterations were performed on the housing and covtype sets instead of 100. SN Algorithm used was developed by IBM [14].

This scaling was chosen for the following reason. The two algorithms have sensitivity bounded by the maximum norm of a row. Scaling the rows to ensure unit norms decreases the sensitivity. Since Wishart noise is scaled similarly to Laplace, low sensitivity reduces noise.

Experimental results in fig. 3 show that a differentially private linear regression converges very quickly to its non-private counterpart. The convergence occurs at values of ε close to one without relaxation. This implies a high privacy budget.

Both algorithms focus on the case of single regression. Sheffet [18] utilized additive and inverse Wishart noise to create a 2nd-moment matrix. The author empirically demonstrated that his algorithm outperformed AG in a multiple regression setting. However, in a single regression setting AG outperformed Sheffet. The same conclusion can be extended to AdaSSP.



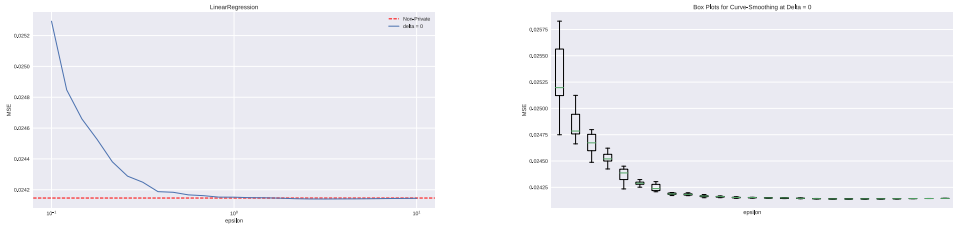


Figure 3: Cars(top), Housing(middle), Covtype(bottom) SN Algorithm

IBM implementation performs a least-squares regression on perturbed $X^TX, X\mathbf{y}$ from the SN Algorithm. This implementation, however, lacks a theoretical discussion. For this reason, an additional DP linear regression algorithm was implemented.

The algorithm - AdaSSP - was published by Wang [22]. The author took a unique approach to estimate the penalty coefficient of a Ridge Regression. Their contribution lies in making the choice of the coefficient an adaptive process. The choice is data-dependent and obeys differential privacy. Additionally, the author proves a bound on the error.

Wang deduced the error as follows [22]. Let $F(\theta) = \min_\theta \frac{1}{2}\|\mathbf{y} - X\theta\|^2$. Let $\hat{\theta}$ be a private estimator and θ^* be a non-private solution. Note that θ^* is the optimal solution of F above. Then the error $F(\hat{\theta}) - F(\theta^*)$ with probability $1 - \beta$ (machine learning accuracy) and with constant factors removed is

$$\max\left(\frac{\sqrt{d\log\frac{1}{\delta}}\|\mathcal{X}\|^2\|\theta^*\|^2}{\varepsilon}, \frac{d^2\log\frac{1}{\delta}\|\theta^*\|^2}{\alpha n\varepsilon^2}\right) \text{ and } \alpha = \lambda_{min}(X^TX)\frac{d}{n\|\mathcal{X}\|^2}$$

where d is the number of features, n is the number of entries, $\|\mathcal{X}\|$ (assumed equal to one) is the maximum among row norms of the data domain, and $\|\theta^*\|$ is the Euclidean norm.

Similar to the other two algorithms, AdaSSP has sensitivity bounded by the maximum l2 norm of a row. This applies to the feature matrix and the target vector. For this reason, the same scaling was applied as before. The norms were constrained to ≤ 1.

Results in fig. 4 demonstrate the effect of δ relaxation on DP Linear Regression. As before, the introduction of the delta parameter significantly improves performance.

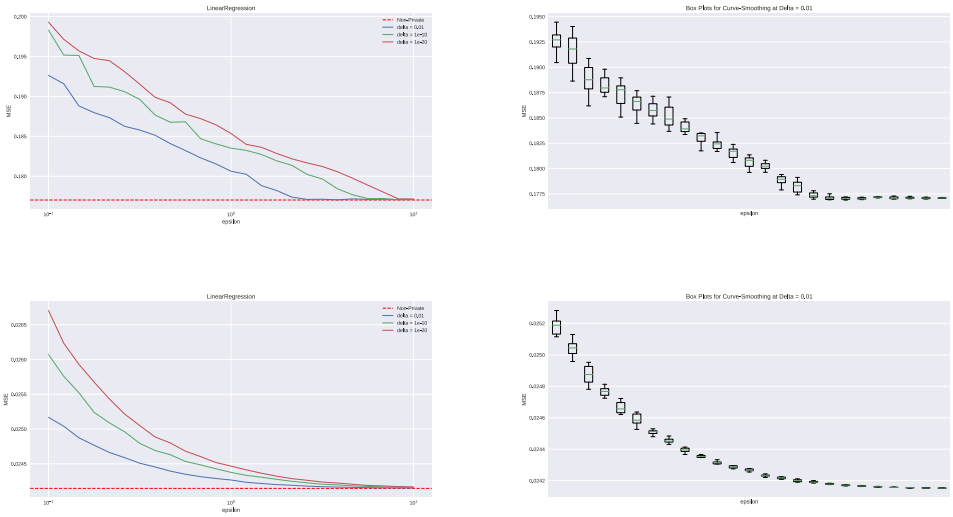


Figure 4: California Housing(top) and Covtype(bottom) AdaSSP

3.4 Open-Source Tools

To finish off the DPML discussion, a brief survey open-source DP projects will be presented. Since its inception in 2006 [9], differential privacy gained interest in industry and government. Several major companies implemented differential privacy in their products [2][12]. However, the biggest project yet to employ DP is the US Census [1].

Because privacy falls under the cyber-security umbrella, it is prudent to use open-source technology in order to avoid security-by-obscurity and verify integrity. Possibly, for this reason, or simply seeking bigger adoption, a number of big IT companies released software for DP. At this point, Google and IBM host open-source repositories on GitHub.

Google created a tool to privately query aggregate statistics - count, sum, mean, variance, std, min, max, med - on a numeric database [1]. This library can be implemented in Java or C++. It is an advanced tool that allows to dynamically discount the privacy budget as the queries arrive. This project resolved the major issue of combining multiple entries that belong to a single identifier. Although such separation is a standard practice in databases, it does complicate private analysis significantly.

IBM chose to provide ready-to-use differentially private machine learning

[1]https://github.com/google/differential-privacy

algorithms[2]. In essence, this is a private extension of the popular scikit-learn library[3]. Private algorithms are subclasses of their non-private counterparts. Both are implemented in Python.

For someone used to working with scikit-learn, the transition to private data analysis will be straight forward. Currently, the library offers k-means, naive Bayes, linear and logistic regression algorithms. Additionally, it offers a number of useful private data analysis tools, such as histograms, statistics and scalers.

Both projects describe their software production-ready. The projects are well documented and easy to integrate. For those contemplating private data analysis, the on-boarding process should be straightforward. The next section will cover DP as a countermeasure to ML attacks accompanied by a case-study.

4 Attacks on ML Algorithms

4.1 Overview

Zhao et al. [25] identify three types of privacy attacks in machine learning setting: (1) membership inference, (2) training data extraction, and (3) model extraction. The author describes (1) as follows. Membership inference attacks work by providing a trained model with input and using the prediction output to determine whether the input record was part of the training data. If the database used for training is sensitive in nature - police records, medical histories, etc. - then the attack poses a significant threat. This attack is possible by exploiting the fact that a model will process new data differently from training data [25].

Tramer et al. demonstrated a model extraction attack that aims to steal model parameters[19]. According to the author, this attack works on a wide array of machine learning models and does not need to know the algorithm in advance. All an adversary needs is the ability to query the model repeatedly for a prediction. The authors tried this type of attack on cloud-provided models. However, this attack can be extended to any setting with required model access.

Training data extraction or model inversion aims to extract parts or all of the training set. According to Zhao et al. [25], this attack works in white-box and black-box settings. In a black-box setting, the adversary does not have access to model parameters, while in the white-box, they do. The black-box

[2]https://github.com/IBM/differential-privacy-library
[3]https://scikit-learn.org

attack works similarly to the model extraction attack. In fact, model extraction can be employed as a precursor [25]. The attack is deployed by probing and creating a meta-classifier [25]. A white-box attack was demonstrated by Fredrikson et al. [11] and will be discussed in detail below.

Two common DP approaches to thwarting such attacks are based on deployment location: input and output [25]. DP can be applied at the input by creating synthetic data, which is costly on large data sets [25]. Alternatively, DP can be applied to the output by making sure the ML model produces differentially private results [25] [11]. This type of countermeasure will be closely examined below.

4.2 Case Study

Fredrikson et al. [11] deployed DP on the output layer as a countermeasure to model inversion attack. The subject of their study was a pharmacogenetic algorithm developed from a genomics database [10] [23]. Fredrikson did not directly perturb algorithm predictions. Instead, they chose to apply DP to the objective function of the regression problem. The authors conclude that DP is an ineffective countermeasure to this type of attack.

According to Fredrikson, the privacy budget needed to effectively thwart the attack decreases model accuracy too much. However, I replicated their work with a relaxed definition of DP and was able to provide the epsilon needed for the desired privacy/accuracy budget.

The original pharmacogenetic study and the subsequent work by Fredrikson used linear regression to derive an estimate of appropriate initial warfarin dosage. Warfarin is a blood thinner and requires very careful dosing[10]. The dosage is increased gradually, but treatment success depends on the correctness of the initial dose [11]. In order to increase initial precision, the authors trained an estimator on a database containing genetic, medical and warfarin dosage information.

A wide variety of DPLR algorithms was developed, as noted in the sections above. However, an algorithm with very high utility is not necessarily the best choice for the purpose of model inversion prevention. Wu et al. [24] tried to solve Fredrikson's problem by maximizing algorithm utility. They were able to achieve the desired performance at $\varepsilon = 0.1$ while Fredrikson achieved it only at $\varepsilon = 20$. Nevertheless, Wu et al. report an unsatisfactory privacy protection result. The author explains this as follows:

> it suffices to observe that DP is a property of the learning"process", while MI attack is on the "result" of the process. It is thus valid that the process satisfies a strong privacy guarantee, while the

> result has some other concerns. In the following, we give a "lower bound"result, which shows that, as long as the optimal solution of the learning problem is susceptible to MI attacks, improving DP/utility trade off will give effective MI attacks "eventually." [24, p.17]

For this reason, I chose an algorithm that matches Fredrikson's performance and relied solely on DP relaxation.

Non-private and private versions of the estimator were created. This involved the reproduction of ML effort in the original study [10] and using the same data [23]. Genetic types were imputed according to the algorithms provided in the appendix from [10]. Irrelevant columns were removed, and categorical data was binarized. Finally, the 'Subject Reached Stable Dose of Warfarin' variable was taken as an indication of the correct dosage.

The preprocessing was similar to Fredrikson's. The first step was to centre the data in order to avoid wasting the privacy budget on intercept fitting. Also, the data was normalized so that the row norms of the features matrix and the target vector were equal to one. This was necessary to bound the function sensitivity, which decreases the noise.

Data centring was performed as follows. Dependent variables were scaled by subtracting each one's mean and dividing by standard deviation. Then each row of the feature matrix was divided by its norm. The target variable was processed differently.

I subtracted the mean from each target entry - similar to features - but divide by the maximum. This was done in order to maintain the norm - defined as a maximum of row norms - of dependent variables and the independent variable at one. This step is in accordance with the private algorithm that was selected, which is AdaSSP from sections above [22]. The target was processed differently from Fredrikson because a different DPLR algorithm was used. As a result, a non-private estimator was produced with a mean relative error of 36% vs Fredrikson's 15%.

The error can be improved with better feature selection, preprocessing, and genetic marker imputation. Nevertheless, I made an assumption that error improvement of the non-private estimator will reflect in the private versions. Basing on this I decided to continue the experiment and observe the deviation of private from non-private estimators while comparing it with Fredrikson's. Besides the error rate, my estimator mimicked Fredrikson's very closely.

The second part of fig. 5 shows that at $\varepsilon = 5$ a private estimator with $\delta = 0.01, 0.001$ has an error approximately 10-15% higher than a non-private one. Per Fredrikson, this is on the boundary of the right privacy/accuracy

balance.

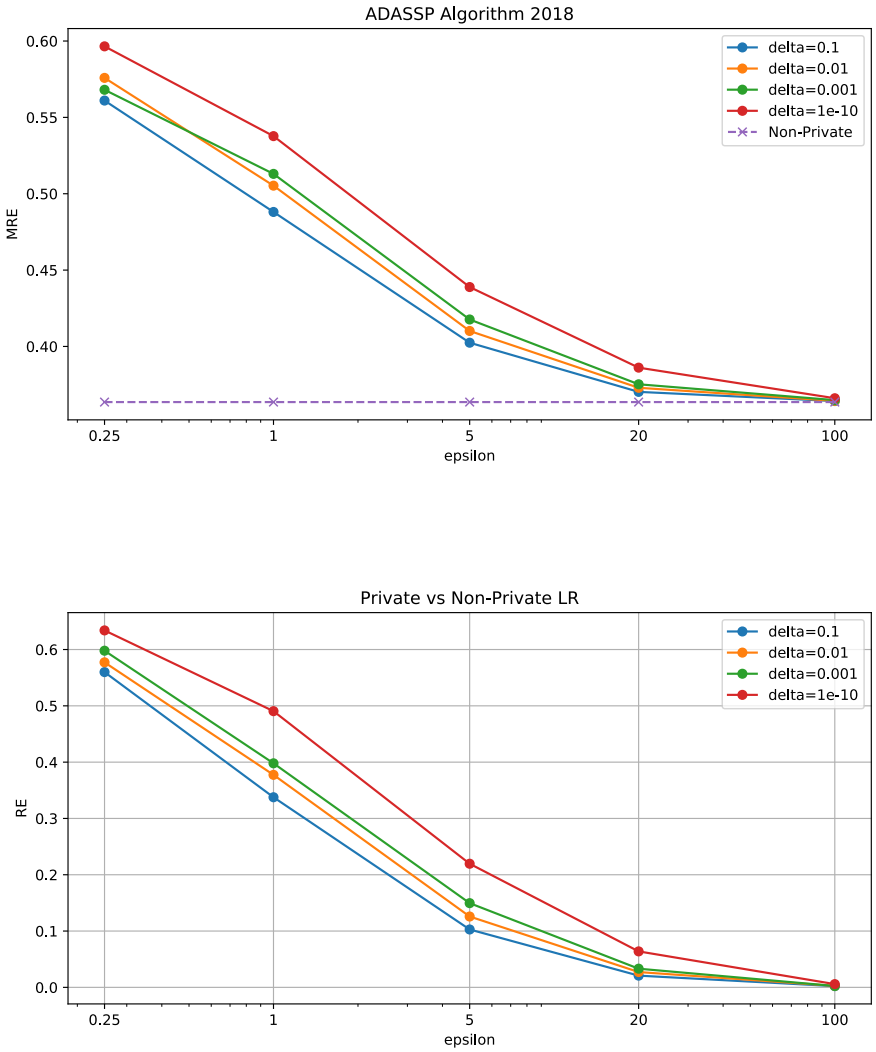


Figure 5

The δ parameter is indeed very high. While $\varepsilon = 5$ DP is expected to hold with at least 99% and 99.9% probability, a number of patients will experience a data exposure event. On the other hand, such an outcome may be justifiable in a medical setting or by considering the economic framework mentioned above. Regardless, this result demonstrates that by taking into account algorithm utility as third axes of DP - along with ε, δ parameters - it is likely possible to create an estimator that will stop the model inversion attack with a low delta parameter. By low delta, I am referring to an equivalent of

the low-likelihood bound mentioned in Section 2.4, which here is at approx. 1e-10.

On a final note, it is important to consider a possible weakness in Fredrikson's analysis. The columns of the feature matrix show very strong linear dependence. This is not surprising given how closely demographic features are related to people's genetic information. This creates two problems.

First, high linear dependence reduces the performance of the AdaSSP algorithm [22]. The second problem is that given the demographic background, it is easy to infer genetic information regardless of the privacy measures [25]. Since the aim of Fredrikson's attack is to extract genetic data using other information from the database, this problem could prove the reason behind apparent DP failure in Fredrikson's study.

5 Conclusion and Future Work

This report presented a theoretical framework of differential privacy. The definition is a mathematically rigorous construct. Its properties make it immune to post-publication attacks and enable modular algorithm construction. A variety of perturbation techniques can be employed to match specific problem classes. Privacy parameters can be chosen using economic or Bayesian interpretations. Finally, the original definition can be relaxed to adjust the privacy budget either to the problem constraints or to remove unlikely DP failure events.

The nature of DP makes it compatible with ML algorithms. The performance was demonstrated empirically on DP versions of Naive Bayes and Linear Regression. In addition, the impact of relaxation was demonstrated. While DP can produce a low but potentially valuable performance at very low ε settings, care must be taken due to observed volatility. A study of this volatility and possible solutions is planned for future work. DPML discussion ended with an overview of open-source tools.

Finally, a brief survey of DP countermeasures to ML attacks was presented, and a relaxed version of DP was considered as a countermeasure to a model inversion attack. The later was performed on a linear regression estimator created for a real-world medical study. The result was too close to the boundary of an acceptable privacy/performance trade-off to make a definitive conclusion. Nevertheless, the countermeasure study performed here demonstrates a possible attack vector against DP. For this reason, it is planned to revisit this research with a wider variety of algorithms and a full reproduction of the underlying study by Fredrikson et al. [11]

References

[1] John M Abowd. "The us census bureau adopts differential privacy". In: *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018, pp. 2867–2867.

[2] Apple. *Learning with Privacy at Scale - Apple*. URL: https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html.

[3] Mark Bun. "A Teaser for Differential Privacy". In: (2017). URL: https://www.cs.princeton.edu/~smattw/Teaching/521fa17lec22.pdf.

[4] Irit Dinur and Kobbi Nissim. "Revealing Information While Preserving Privacy". In: *Proceedings of the Twenty-Second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*. PODS '03. San Diego, California: Association for Computing Machinery, 2003, 202–210. ISBN: 1581136706. DOI: 10.1145/773153.773173. URL: https://doi.org/10.1145/773153.773173.

[5] Dheeru Dua and Casey Graff. *UCI Machine Learning Repository*. 2017. URL: http://archive.ics.uci.edu/ml.

[6] Cynthia Dwork. *A firm foundation for private data analysis*. English. 2011.

[7] Cynthia Dwork and Aaron Roth. *The Algorithmic Foundations of Differential Privacy*. English. Vol. 9. 3-4. Hanover, Massachusetts: now, 2014;2013; pp. 211–407. ISBN: 1551-305X.

[8] Cynthia Dwork et al. "Analyze gauss: optimal bounds for privacy-preserving principal component analysis". English. In: ACM, 2014, pp. 11–20. ISBN: 0737-8017.

[9] Cynthia Dwork et al. "Calibrating noise to sensitivity in private data analysis". In: *Theory of cryptography conference*. Springer. 2006, pp. 265–284.

[10] "Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data". English. In: *The New England journal of medicine* 360.8 (2009). Copyright - Copyright © 2009 Massachusetts Medical Society. All rights reserved; Last updated - 2017-10-31; CODEN - NEJMAG, pp. 753–764. URL: http://ezproxy.lib.ryerson.ca/login?url=https://search-proquest-com.ezproxy.lib.ryerson.ca/docview/223913197?accountid=13631.

[11] Matthew Fredrikson et al. "Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing". English. In: *Proceedings of the . USENIX Security Symposium. UNIX Security Symposium* 2014 (2014), pp. 17–32.

[12] Google. *Learning statistics with privacy, aided by the flip of a coin.* 2014. URL: https://security.googleblog.com/2014/10/learning-statistics-with-privacy-aided.html.

[13] Justin Hsu et al. "Differential Privacy: An Economic Method for Choosing Epsilon". English. In: IEEE, 2014, pp. 398–410. ISBN: 1063-6900.

[14] IBM. *IBM/Differential-Privacy-Library.* 2018. URL: https://github.com/IBM/differential-privacy-library.

[15] Hafiz Imtiaz and Anand D. Sarwate. "Symmetric matrix perturbation for differentially-private principal component analysis". English. In: IEEE, 2016, pp. 2339–2343.

[16] R. Kelley Pace and Ronald Barry. "Sparse spatial autoregressions". English. In: *Statistics and Probability Letters* 33.3 (1997), pp. 291–297.

[17] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[18] Or Sheffet. *Private Approximations of the 2nd-Moment Matrix Using Existing Techniques in Linear Regression.* 2015. arXiv: 1507.00056 [cs.DS].

[19] Florian Tramèr et al. "Stealing Machine Learning Models via Prediction APIs". In: *25th USENIX Security Symposium (USENIX Security 16).* Austin, TX: USENIX Association, Aug. 2016, pp. 601–618. ISBN: 978-1-931971-32-4. URL: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/tramer.

[20] Jaideep Vaidya et al. "Differentially Private Naive Bayes Classification". English. In: vol. 1. IEEE Computer Society, 2013, pp. 571–576. ISBN: 9780769551456;0769551459;

[21] Pantelis Vlachos. *StatLib.* Department of Statistics, at Carnegie Mellon University. 1989. URL: http://lib.stat.cmu.edu.

[22] Yu-Xiang Wang. "Revisiting differentially private linear regression: optimal and adaptive prediction & estimation in unbounded domain". In: *arXiv preprint arXiv:1803.02596* (2018).

[23] Michelle Whirl-Carrillo et al. "Pharmacogenomics knowledge for personalized medicine". In: *Clinical Pharmacology & Therapeutics* 92.4 (2012), pp. 414–417.

[24] Xi Wu et al. “Revisiting differentially private regression: Lessons from learning theory and their consequences”. In: *arXiv preprint arXiv:1512.06388* (2015).

[25] Jingwen Zhao, Yunfang Chen, and Wei Zhang. “Differential Privacy Preservation in Deep Learning: Challenges, Opportunities and Solutions”. English. In: *IEEE Access* 7 (2019), pp. 48901–48911.