[[1]](#footnote-1)

Aggregate Query and Analysis While Maintaining Personally Identifying Information Privacy

Andrew N. Abbott, [anabbott@smu.edu](mailto:anabbott@smu.edu)*, Joseph Stoffa,* [jstoffa@smu.edu](mailto:jstoffa@smu.edu)*, Nathan Mowat,* [nmowat@smu.edu](mailto:nmowat@smu.edu)

*Abstract*—Many database systems collect personally identifiable information (PII), such as social security numbers, dates of birth, or addresses. Protecting this information from undesired users is vital. It is also important to be able to analyze the data in ways that aggregate user’ information together to do things like disease identification or anomaly detection. Researchers have been able to de-anonymize private information with very little information about data sets by using other publicly available data sets. The goal of this paper is to explore differential privacy methods and to both accurately and securely perform certain types of analysis and aggregation without exposing user’ PII.

# INTRODUCTION

D

atabase administrators are the protectors of an ever- growing amount of data, much of which is customers’ Personally Identifiable Information (PII). All this data is continuously being analyzed, bought, and sold. The task of protecting that PII from theft, abuse or negligence is critical. Despite these risks, analysts must maintain the ability to calculate statistics based on these data sets while still preserving that privacy. To address this problem, this paper seeks to first identify PII and describe some of the methods used to protect PII, then to detail the method to securely protect or anonymize the data by introducing noise to the data without changing the aggregate statistics of the data set known as “differential privacy”. In pursuit of this goal, Cynthia Dwork introduced the concept of differential privacy [1]. Rather than attempting to guarantee absolute protection of individual data, differential privacy reduces the ability of disclosures to identify individuals. Finally, we will seek to answer the question of how much noise or modification is necessary to protect the privacy of the data without significantly changing the statistics used for analysis.

# Research Methodology

We begin by defining PII for the purposes of this paper. Definitions vary across and within domains. Generally, “any information that distinguishes one person from another can be used for re-identifying anonymous data [2]”. This definition provides us with too broad a metric. For our purposes, we will define PII as “any information related to a person that distinguishes one person from another and is not intended for the public.”

Next, we will review the history of differential privacy and discuss some real-world cases where privacy was compromised using differential methods. Parts V and VI cover related research and the use of differential privacy in machine learning applications. After a formal definition of differential privacy we will more on to a test of the functionality.

We will develop a test dataset which we will use to examine the effects of noise on accuracy. This dataset will be sufficient in number of observations and scope without being unnecessarily complex. Unlike a real-life data set it will also provide “factual” responses in order to gauge true accuracy.

The final step is to test different levels of data modification to maximize the usefulness of the resulting statistics while also minimizing the ability to differentiate any individual record. Ideally, the presence or absence of an individual record will not affect the calculation output and thus be unidentifiable. This method involves the modification of the data by introducing noise. In effect, we will be trying to determine the ideal amount of noise needed to provide the required anonymity without significantly altering the statistics produced by the aggregation.

# Brief History of Differential Privacy

Before the term “Differential Privacy” was coined, there were different statistical analyses being done on the population without any consideration of the individuals that were part of the studies. Cynthia Dwork and Adam Smith then decided that this had to change and began trying to answer this question: How does one release statistical analysis results to the public without compromising the privacy of the individual respondents? [3]

Dwork and Smith started with the idea that data was noninteractive, that once some statistical result was published by the curator, the dataset could be destroyed because it wasn’t going to be used again. This was troublesome because the statistics released could be biased based on how the curator determined the privacy needed for the data. If significant privacy was needed, then important data would be withheld, and the statistical inference would be altered.

Then they considered how the statistical analysis would be affected if the data were interactive. In this scenario, the curator would act as a gate between the database and the users. The queries would pass through the curator and be cleaned until they didn’t cause privacy issues. This notion would ideally be better because then only the questions of interest would be answered. A user wouldn’t be able to scrub the dataset of every bit of meaning and analyses.

Dwork ad Adams felt that this was still not an ideal way to bring about privacy so they tried to prove the statement, “access to the statistical database does not help the adversary to compromise the privacy of any individual.” This almost immediately fell flat as this example demonstrates: Say John is two times heavier than the average American. Then as long as someone has access to a database such as a medical database that contains the weights of Americans, if a mean statistic can be drawn from the data, then John’s weight is calculated. The point being that regardless of if John was in that database or not, he would be vulnerable to attack. This is where “Differential Privacy” was born. Instead of avoiding the risk by holding back certain information, the goal was to minimize the risk so that the risk of being inside the database was the same as being outside.

# Real World Examples

Differential Privacy is becoming a much more widely researched subject because of the database manipulations that led to individuals having their identities exposed.

The first real event began with Netflix when they offered the public some prize money to improve their recommendation system for their movies [4]. Netflix released a subset of their database for the public to work with, but they anonymized it by removing any identifying features. [A simple way to anonymize data is to remove the individual’s name, their phone number, and their home address. It is important to also remove anything else that will specifically identify them, such as a student ID.] Two researchers by the names of Arvind Narayanan and Vitaly Schmatikov were able to link Netflix’s training dataset with the IMDB dataset and were able to figure out the identities of a portion of the individuals. [5]

Some sensitive information about the Massachusetts governor was also exposed. Latanya Sweeney, the director of the data privacy lab at Harvard [6], was able to link together the Massachusetts Group Insurance Commission (GIC)

medical database with the voter registration records. The GIC was an anonymized data source, but, by combining records and databases, she could leak the medical records of the governor.

# Previous and Related Research

Dwork published a procedure for differential privacy which has served as the basis for most other research in the field. Two of Dwork’s key contributions are the concepts of ε known as the *privacy budget* and *sensitivity*. The privacy budget limits the number of queries a user can execute before they exceed the budget and therefore can begin to compromise the data’s confidentiality [1]. Sensitivity “measures how much a query amplifies the distance between two inputs” or more plainly it is the max number of queries that can be made on two databases that differ by only one row (referred to as *neighbors*). Both terms are critical to the discussion of differential privacy data aggregation optimization. Dwork introduced the concept of adding noise to a dataset in the form of a Laplace distribution, referred to as the Laplace mechanism and the subject of examples in this paper.

Two competing mechanisms for differential privacy of linear aggregate data are the Matrix Mechanism [8] and the Low- Rank Mechanism [9]. G. Yuan et al, state that while theoretically sound the Matrix Mechanism is unpractical and, according to their tests, produces no more accuracy then more naïve methods of differential privacy [12].

Apart from applying differential privacy to linear and batch aggregation there has also been increased interest in its application to other areas as well. L. Fan et al and E. Shi et al have both made noteworthy contributions to differential privacy of aggregations of time-series data [7, 10]. L. Fan et al have provided compelling evidence that it is possible to provide differential privacy of highly correlated time series data through the exposition of their “FAST” method on three distinct datasets. What makes L. Fan et al’s work unique is its focus on real-time data whereas previous work is more suited to analyzing a time-series as a whole after it has been recorded. Recently in the arena of machine learning, M. Ababi et al have attempted to develop techniques for conducting analysis while maintaining “modest” privacy budgets [11].

# Differential Privacy and Machine Learning

When it comes to doing data analyses on data sets and maintaining privacy, machine learning goes hand in hand with differential privacy. As a general rule, the goal of machine learning is to look at the data and come up with a simple rule that explains the dataset. This rule is designed to explain the current dataset, and any new data that is added without ever being dependent on the value of a single point. In a way, this is in line with differential privacy because differential privacy algorithms are trying to pull out a statistical inference from a population without exposing any particular individual. Both machine learning and differential privacy and looking for ways to describe the distributional properties of the population and not the PII of anyone specifically.

Since both machine learning and differential privacy have similar structure and goals, it only makes sense to combine them, thus, “Differentially Private Machine Learning” was born [17]. Under this umbrella there are a few questions that are commonly asked, noted by Cynthia Dwork and Aaron Roth in their paper, “The algorithmic Foundations of Differential Privacy”: “When is it possible to privately perform machine learning?” and “How many additional samples are required to privately learn, compared to the number of samples that are already required to learn without privacy” [17]. It turns out that not that many more observations are required for privacy as for non-private learning. In differentially private machine learning, if our goal is find a hypothesis that has an error rate within alpha of the optimal error, a database of size n in the equation below will be needed.

n ≥ O ( max ( log |C| εα , log |C| α2 ))

(*definition of terms below)*

# Definition

**Definition:** A randomized function K gives ε- differential privacy if for all data sets and differing on at most one element, and all S ⊆ Range(K),

Table 1

|  |  |
| --- | --- |
| Symbol | Definition |
| *K* | The function K() is the mechanism that adds noise to the dataset |
| ε | “Epsilon” is the amount of noise added to the dataset |
| & | and are datasets, differing on at most one element |
| *n* | Sample size of the database |
| *O* | Optimal function |
| ε | error |
| *C* | C is for class |
| α | The distance in between errors in the private machine learning hypothesis and non-private |

# Demonstration of Differential Privacy

A clear example is found on the Wikipedia page for Differential Privacy [4]. Assume we have a database of medical records where each record contains a name and an indicator **X** denoting whether a person has diabetes. For example:

Table 2

|  |  |
| --- | --- |
| **Name** | **Has Diabetes (X)** |
| Ross | 1 |
| Monica | 1 |
| Joey | 0 |
| Phoebe | 0 |
| Chandler | 1 |

Now suppose an adversary wants to determine if Chandler has diabetes or not and the adversary knows in which row of the database Chandler’s record is. If the adversary can use a form of query that returns the sum of the first i rows of column **X** in the database, to find Chandler's diabetes status, the adversary can execute () and (), then compute the difference. In this example, () = 3 and () = 2, so the difference is 1. This indicates that the **X** field in Chandler's row must be 1 [4].

# Test Dataset

To test and to demonstrate differential privacy in practice assume we have a set ***S*** consisting of one thousand 0’s. The average value of ***S*** is therefore 0. If consists of the first 999 0’s of ***S*** with the final record changed to a 1, then an average of would be 0.001 and identify the 1000th element of .

# Introducing Differentially Private Data Aggregation

Differential privacy introduces randomized noise into the calculations to address this type of situation. Different randomization mechanisms are used including Laplace, exponential, and posterior sampling. I will use Python to illustrate the application of differential privacy to the example just described. The code uses the Laplace mechanism and may be found in the appendix. Laplace distributions are similar to Normal distributions but fall off slower, spreading out the noise a bit more than a Normal distribution would as seen in Figure 1.

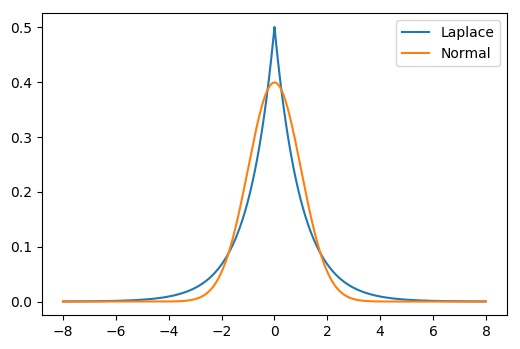


Figure 1

The resulting estimates of the means ***S*** and should be indistinguishable. As Figure 2 shows, as epsilon, the amount of noise parameter, increases, the relative difference between the two estimates increases.

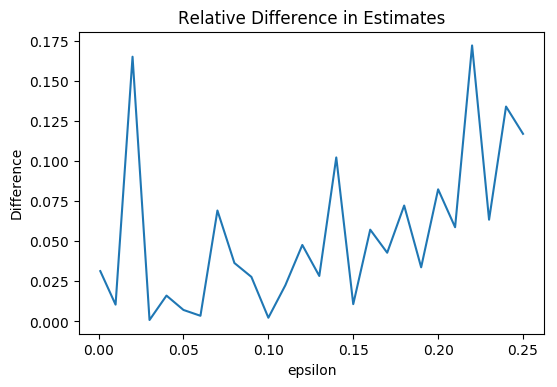


Figure 2

There is a tradeoff however, as epsilon gets smaller, the estimates become poorer. The tradeoff is evident in Figure 3 which shows the estimates of the means for ***S*** and as well as the true mean of . The important question to consider is how much accuracy to sacrifice to achieve privacy. This is an extreme example of course, containing all 0’s and one 1 intended to demonstrate functionality.

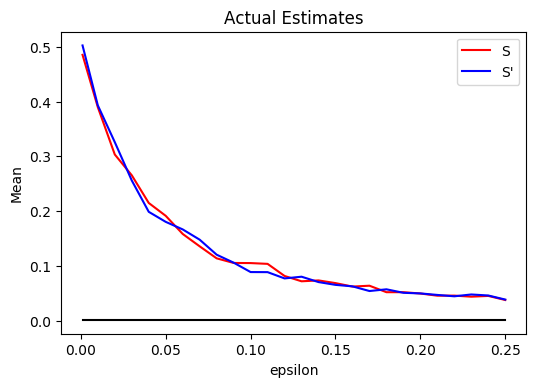


Figure 3

# Conclusion

The importance of discovering useful information from populations contained in databases we must perform queries and calculations. While doing so it is equally important to protect personally identifiable information from adversarial users. Having reviewed the concept of differential privacy and related research, we created a simple test dataset to demonstrate both the concept and the trade-off involved. When using differential privacy, the amount of noise introduced into the calculation affects both the privacy and the accuracy of the calculations. We found this method to be able to protect PII not by removing or obscuring information but by adding noise to the calculations themselves, thus making any one element’s presence not affect the calculation outcome. There are numerous examples of differential privacy protections in use today. Some of those include the United States Census Bureau, Google’s RAPPOR, and Apple’s new iOS 10.

# References

|  |  |
| --- | --- |
| [1] | C. Dwork, "Differential Privacy," *ICALP,* pp. 1-12, 2006. |
| [2] | A. Narayanan and V. Shmatikov, "Privacy and Security, Myths and Fallacies of "Personally Identifiable Information"," *Communications of the ACM,* pp. 24-26, 2010. |
| [3] | C. Dwork and A. Smith, "Differential Privacy for Statistics: What we Know and What we Want to Learn," *Journal of Privacy and Confidentiality,* pp. 1325-154, 2009. |
| [4] | "Wkipedia: Differential Privacy," [Online]. Available: https://en.wikipedia.org/wiki/Differential\_privacy. [Accessed 20 April 2017]. |
| [5] | A. Narayanan and V. Shmatikov, "Robust De-anonymization of Large Sparse Datasets," *IEEE Symposium on Security and Privacy,* pp. 111-125, 2008. |
| [6] | "Wikipedia: Latanya Sweeney," [Online]. Available: https://en.wikipedia.org/wiki/Latanya\_Sweeney. [Accessed 20 April 2017]. |
| [7] | C. Li, "Optimizing Linear Queries Under Differential Privacy," *Dissertations 805,* 2013. |
| [8] | G. Yuan, Z. Zhang, M. Winslett, X. Xiao, Y. Yang and Z. Hao, "Low-Rank Mechanism: Optimizing Batch Queries under Differential Privacy," *VLDB,* 2012. |
| [9] | L. Fan and L. Xiong, "Adaptively Sharing Time-series with Differential Privacy," *CoRR,* 2012. |
| [10] | E. Shi, H. T. H. Chan, E. Rieffel, R. Chow and D. Song, "Privacy-preserving Aggregation of Time-series Data," *Network & Distributed System Security Symposium,* 2011. |
| [11] | M. Adabi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar and L. Zhang, "Deep Learning with Differential Privacy," *ACM Conference on Computer Communication Security,* 2016. |
| [12] | Department of Homeland Security, Handbook for Safeguarding Sensitive Personally Identifiable Information, 2012. |
| [13] | NIST, "Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)," 2010. |
| [14] | W. Yuxiang, "Differential Privacy: A Short Tutorial," [Online]. Available: https://www.cs.cmu.edu/~yuxiangw/docs/Differential %20%Privacy.pdf. [Accessed 1 March 2017]. |
| [15] | N. Zumel, "Win-Vector Blog," October 2015. [Online]. Available: http://www.win-vector.com/blog/2015/10/a-simpler-explanation-of-differential-privacy/. [Accessed 1 March 2017]. |
| [16] | M. Kusner, j. Gardner, R. Garnett and K. Weinberger, "Differentially Private Bayesian Optimization," in Proceedings of the 32nd International Conference on Machine Learning, 2015. |

[17] C. Dwork and A. R., “The Algorithmic Foundations of Differential Privacy,” www.cis.upenn.edu. [Online]. Available: https://www.cis.upenn.edu/~aaroth/Papers/privacybook.pdf. [Accessed: 20-Apr-2017].

Appendix

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

from scipy.stats import laplace

np.random.seed(123123)

def collar(x):

return np.maximum(0, np.minimum(1, x))

def run\_exp(n, T, Nruns, epsilon):

sigma = (20/epsilon)/Nruns

S = np.zeros(n)

Sprime = np.zeros(n)

Sprime[0] = 1

estimate\_S = collar(S.mean() + laplace.rvs(size = Nruns, scale = sigma))

estimate\_Sprime = collar(Sprime.mean() + laplace.rvs(size = Nruns, scale = sigma))

r = {'mean\_S': estimate\_S.mean(), 'mean\_Sprime': estimate\_Sprime.mean(), 'epsilon': eps}

return r

n = 1000

T = 1/200

Nruns = 1000

epsilon = np.arange(0, 0.26, 0.01)

epsilon[0] = 0.001

runframe = []

for eps in epsilon:

run = []

run = run\_exp(n, T, Nruns, eps)

run['diff'] = (np.abs(run['mean\_S'] - run['mean\_Sprime'])) / np.maximum(run['mean\_S'], run['mean\_Sprime'])

runframe.append(run)

df = pd.DataFrame(runframe)

plt.plot(df['epsilon'], df['diff'])

plt.xlabel('epsilon')

plt.ylabel('Difference')

plt.title('Relative Difference in Estimates')

plt.show()

df2 = df.sort(columns = ['epsilon'])

plt.plot(df2['epsilon'], df2['mean\_S'], color = 'red', label = 'S')

plt.plot(df2['epsilon'], df2['mean\_Sprime'], color = 'blue', label = 'S\'')

plt.plot([0.001, 0.25], [0.001, 0.001], color = 'black')

plt.legend(loc = 'best')

plt.xlabel('epsilon')

plt.ylabel('Mean')

plt.title('Actual Estimates')

plt.show()

1. [↑](#footnote-ref-1)