[[1]](#footnote-1)

Aggregate Query and Analysis While Maintaining Personally Identifying Information Privacy

Andrew N. Abbott, *anabbott@smu.edu, MSDS 7330*

*Abstract*—Many electronic record keeping systems necessarily collect personally identifiable information (PII), such as social security numbers, dates of birth, or addresses. Protecting this information from undesired users is important. It is also important to be able to analyze the data in ways that combine user’ information together to do things like intelligent disease cluster identification or anomaly detection. The goal of this project is to describe ways to securely perform certain types of analysis and aggregation without exposing user’ PII.

# INTRODUCTION

D

atabase administrators are the custodians of vast amount of data, much of which may be customer’s Personally Identifiable Information (PII). The task of protecting that PII from theft, abuse or negligence is of vital importance. Analysts and business users of reports and data warehouses are often able to query user’ PII directly, putting customers at risk. Despite these risks, analysts must be able to learn properties of the population while preserving that privacy. To address this problem, this paper seeks to first identify PII, next to describe the various methods used to protect PII, and then detail a method to securely protect or anonymize it to remove the possibility of personal identification.

# Definition

Personally identifiable information (PII) has been defined as “any information about an individual maintained by an agency, including (1) any information that can be used to distinguish or trace an individual’s identity, such as name, social security number, date and place of birth, mother’s maiden name, or biometric records; and (2) any other information that is linked or linkable to an individual, such as medical, educational, financial, and employment information.[1]” While very broad, this definition may not go far enough. It has been shown that “Any information that distinguishes one person from another can be used for re-identifying anonymous data [2].” I would go further to describe PII as any information related to a person that distinguished one person from another and is not generated solely by the enterprise. Breaches involving PII are hazardous to both individuals and organizations. Individual harms may include identity theft, embarrassment, or blackmail. Organizational harms may include a loss of public trust, legal liability, or remediation costs.

# protection Methods

The strategies used for protecting PII include but are not limited to: de-identifying the data, anonymizing the data, and differential privacy [2].

De-identification of data involves removing or obscuring the data such that the remaining data does not identify an individual. De-identified data may be re-identified using a code or algorithm. Hashing and tokenization are two examples of de-identification methods. Anonymization of data takes-de-identification one step further as a code for re-identification does not exist. Removing, suppressing, and generalizing are three forms of anonymizing. While both methods provide some level of privacy without adding fake or removing truthful information, this is not sufficient to prevent re-identification. With the ever-increasing amount of public information available about individuals allows for sophisticated re-identification algorithms to use that information to re-identify anonymized data. Additionally, various approaches to anonymizing data have failed when researchers managed to identify PII by linking two or more separately anonymized databases.

# Differential Privacy

Our goal is aggregate query and analysis while maintaining PII protection. In pursuit of this goal, Cynthia Dwork introduced the concept of differential privacy [3]. Differential Privacy is a framework for formalizing privacy in statistical databases introduced in order to protect against de-anonymization techniques. Rather than attempting to guarantee absolute protection of individual data, differential privacy reduces the ability of disclosures to identify individuals. The presence or absence of an individual record will not affect the calculation output and thus be unidentifiable.

**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),

# Demonstration of Differential Privacy

For example, assume we have a database of medical records where each record is a pair (**Name**, **X**), where **X** is a Boolean denoting whether a person has diabetes or not. For example:

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

Now suppose a user wants to find whether Chandler has diabetes or not. He knows in which row of the database Chandler resides. Now suppose the adversary is only allowed to use a form of query that returns the partial sum of the first i rows of column **X** in the database. In order to find Chandler's diabetes status the adversary executes () and (), then computes their difference. In this example, () = 3 and () = 2, so their difference is 1. This indicates that the "Has Diabetes" field in Chandler's row must be 1 [4].

As another example, assume we have a set ***S*** consisting of all 0s except for one 1. The average value of ***S*** is therefore 0.01. If consists of the first 99 0’s of ***S***, then an average of would be 0 and identify the 100th element of **S**. Differential privacy introduces randomized noise into the calculations in order to address this type of situation. Different randomization mechanisms are used including Laplace, exponential, and posterior sampling. I will use R 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. The resulting estimates of the means ***S*** and should be indistinguishable. As Figure 1 shows, as epsilon decreases, the relative difference between the two estimates decreases. There is a tradeoff however, as epsilon gets stricter, the estimates also become poorer. The tradeoff is evident in Figure 2. This is an extreme example of course, containing all 0’s and one 1 intended to demonstrate functionality.

Figure 1

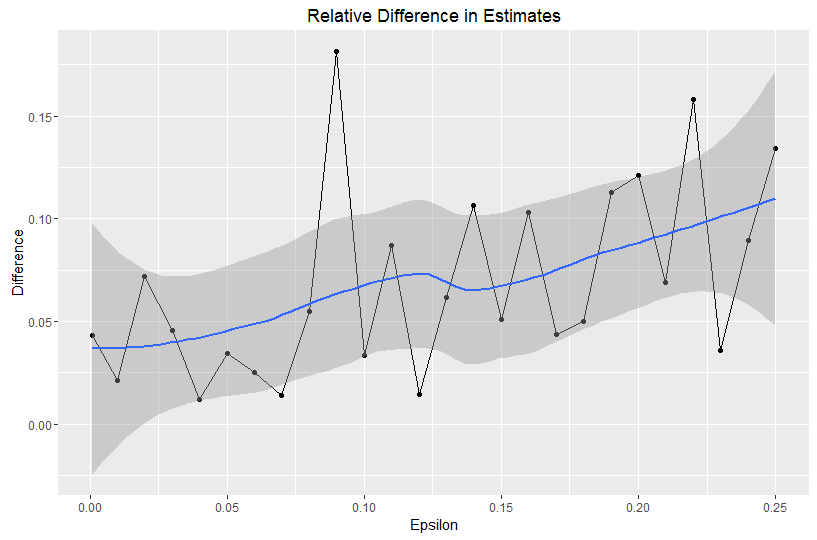
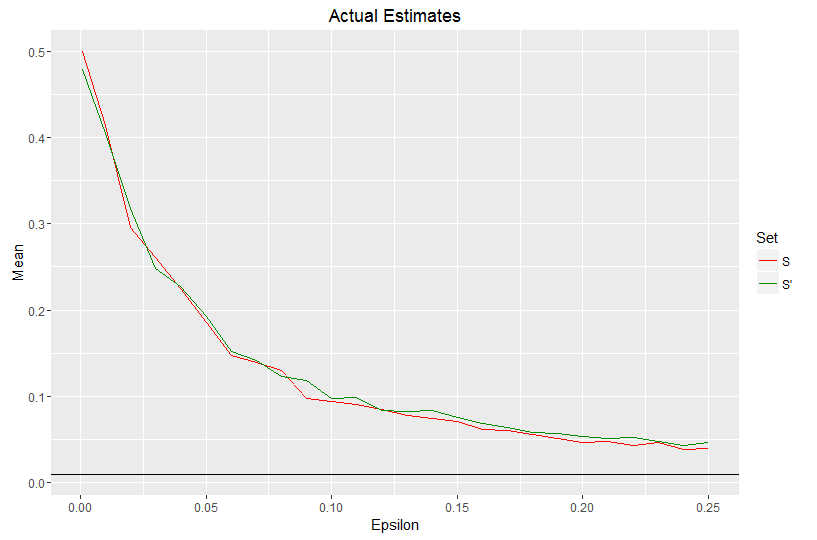


Figure 2



# 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 current methods of protecting PII and the remaining dangers associated with them I chose to further investigate the method known as differential privacy. I 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. *This definition is the GAO expression of an amalgam of the definitions of PII from OMB Memorandums 07-16 and 06-19. GAO Report 08-536, Privacy: Alternatives Exist for Enhancing Protection of Personally Identifiable Information, May 2008, http://www.gao.gov/new.items/d08536.pdf*
2. A. Narayanan, V. Shmatikov. (2010, June). Privacy and Security, Myths and Fallacies of “Personally Identifiable Information”. *Communications of the ACM*. [Online]. *53(6)*, pp. 24-26. Available: https://www.cs.utexas.edu/~shmat/shmat\_cacm10.pdf
3. C. Dwork, “Differential privacy,” in ICALP, 2006, pp. 1-12.
4. https://en.wikipedia.org/wiki/Differential\_privacy
5. Handbook for Safeguarding Sensitive Personaly Identifiable Information, DHS 2012.
6. Guide to Protecting the Confidentiality of Personally Identifiable Information (PII), NIST Special Publication 800-122, 2010.

Appendix

library(ggplot2) # For the graphs at the bottom of the code

library(reshape2) # For the melt at the bottom of the code

set.seed(345345)

# Function to create noise from Laplace distrbution

rlaplace <- function(n,sigma) {

if(sigma<=0) {

numeric(n)

}

rexp(n,rate = 1/sigma) - rexp(n,rate = 1/sigma)

}

# Ensures x is between 0 and 1

collar = function(x) {

pmax(0, pmin(1, x))

}

# Play N rounds of the differential privacy game

run\_exp = function(n, threshold, Nruns, epsilon) {

# The noise parameter.

sigma = (20/epsilon)/Nruns

# The Universe is two sets, S and S'

s = numeric(n);

sprime = numeric(n);

sprime[1] =1

sets = list(s, sprime)

# this is run inside the "private world"

estimate\_s = collar(mean(s) + rlaplace(Nruns, sigma))

estimate\_prime = collar(mean(sprime) + rlaplace(Nruns, sigma))

# this is what the adversary sees

outcome\_s = estimate\_s >= threshold

outcome\_prime = estimate\_prime >= threshold

#probability >= threshold

ps = mean(outcome\_s)

pprime = mean(outcome\_prime)

# epsilon-dp means abs(log(ps/pprime)) < epsilon

logratio = abs(log(ps/pprime))

data.frame(mean\_s = mean(estimate\_s),

mean\_prime = mean(estimate\_prime),

logratio = logratio)

}

n = 100;

threshold=1/200;

Nruns = 1000

epsilon = seq(from=0, to=0.25, by=0.01)

epsilon[1] = 0.001 # can't do zero

runframe = NULL

for(eps in epsilon) {

run = cbind(run\_exp(n, threshold, Nruns, eps), epsilon=eps)

runframe = rbind(run, runframe)

}

runframe$diff = with(runframe, abs(mean\_s-mean\_prime)/pmax(mean\_s, mean\_prime))

# As epsilon gets stricter (smaller), the relative difference in the

# estimates of the means of S and S' gets smaller, as you would hope.

# the trend is clear.

ggplot(runframe, aes(x=epsilon, y=diff)) +

geom\_point() + geom\_line() + geom\_smooth() +

ggtitle("Relative Difference in Estimates") + ylab("Difference") + xlab("Epsilon")

# However, as epsilon gets stricter, the estimates also become poorer,

# relative to the actual set means (0 and 0.01, respectively)

estimateframe = melt(runframe, measure.vars=c("mean\_s", "mean\_prime"), variable.name="set", value.name="mean\_value")

ggplot(estimateframe, aes(x=epsilon, y=mean\_value, color=set)) + geom\_line() +

geom\_hline(yintercept = 0.01) + scale\_color\_manual(values=c("red", "green4"), name="Set", labels=c("S","S'")) +

ggtitle("Actual Estimates") + ylab("Mean") + xlab("Epsilon")

1. Proposal submitted September 21, 2016

   A. N. Abbott is a student at Southern Methodist University, Dallas, TX 75275 USA (e-mail: anabbott@smu.edu). [↑](#footnote-ref-1)