Currently presenting: All Group Members

CS108 Project Option 1

— Team CEK

Bounded-Laplace Mechanism

An algorithm for Differential privacy

Members

Krishaan Patel

Edmund Wong

Chun Ho Wong

Currently presenting: Chun Ho Wong

Goals

a small step toward privacy protection

Differential Privacy (review)

Overview of 3 papers that we studied

<u>Bounded-Laplace Mechanism</u>

- How we Implement it
- How we test it on a dataset

Finally, understand the importance of privacy protection…

Team CEK

Differential Privacy (review)

1. What is Differential Privacy?

   Differential privacy is a framework that adds controlled noise to data analysis to protect individuals' privacy.

2. Ethics

   It balances (We would address this later) the trade-off between privacy and data utility and promotes ethical data practices, particularly in healthcare, finance, and law enforcement domains.

3. Applications

   It has been widely adopted by researchers, companies, and governments to ensure privacy protection while enabling data analysis.

In short, it adds **controlled** noise to data to ensure privacy protection while (still) enabling data analysis (The idea of balancing).

Currently presenting: Chun Ho Wong -> Edmund Wong

3 papers

**these are just the titles..
don't worry**

1. *Differential privacy for public health data: An innovative tool to optimize information sharing while protecting data confidentiality*
2. *Differential privacy in health research: A scoping review*
3. *The Bounded Laplace Mechanism In Differential Privacy*

Currently presenting: Edmund Wong

“Differential privacy in health research: A scoping review”

It is a paper that studies other papers.

- "Differential Privacy in Health Research" emphasizes the importance of protecting sensitive information in health research.
- The review analyzed 50 applications of differential privacy in health research and identified gaps in its usage and evaluation.
- The authors recommend experimenting with differential privacy and sharing case studies to understand its privacy-utility outcomes in real-world implementations.

Reference: https://academic.oup.com/jamia/article/28/10/2269/6333353

"Differential privacy for public health data: An innovative tool to optimize information sharing while protecting data confidentiality"

- Discusses the concept of differential privacy and its effectiveness over current strategies
- Limitations: Current strategies are more prone to re-identification of individual data than differential privacy
- Case study demonstrates how differential privacy protects patient information in public health data sharing
    - Laplace mechanism used to apply noise to COVID-19 dataset
        - Resulted in a 47.5% chance that the person's guess about an individual's COVID-19 status was incorrect

Reference: https://www.sciencedirect.com/science/article/pii/S2666389921002282

“The Bounded Laplace Mechanism In Differential Privacy”

Team CEK

- The “workhorse” of differential privacy
- Applications of numerical data
- The Problem:
  - Can return semantically impossible values
- The Solution:
  - Bounding: addresses the lack of consistency

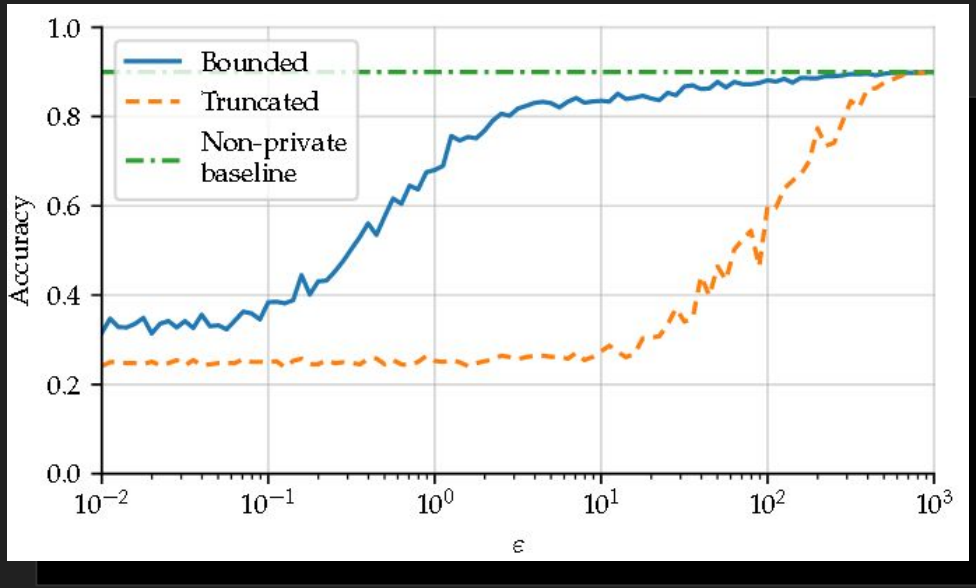

Figure 1. Comparison of accuracy versus epsilon for a differentially private naive Bayes classifier on the Iris dataset (using the bounded and truncated Laplace mechanisms for perturbation of each feature variance).

Reference: https://journalprivacyconfidentiality.org/index.php/jpc/article/view/715/690

Currently presenting: Krishaan Patel

The Algorithm

It’s getting technical!

The Laplace Mechanism
(we promise to be easy)

The Laplace Mechanism

- What is it?
  - Technique for adding noise to data
- How the Laplace Mechanism approach works:
  - Laplace Distribution
  - Determines privacy budget & sensitivity of the target query
    - Greater sensitivity results in more noise being added
  - Adds random noise to the output of functions being computed on a dataset
- Value:
  - Simple and effective way to ensure privacy in the dataset while still allowing for meaningful data analysis.

Team CEK

The Inverse Cumulative Distribution Function

- How it works:
  - Laplace has PDF and CDF
  - Probability of random value less than or equal to a certain value
  - CDF calculated by taking the inverse
- The Dataset:
  - Iris Data Set
  - Popular dataset containing features about flower sepal & petals.

The inverse cumulative distribution function is given by

$$F^{-1}(p) = \mu - b\,\mathrm{sgn}(p - 0.5)\,\ln(1 - 2|p - 0.5|).$$

Figure 2. CDF Equation

Iris Data Set

Download: Data Folder, Data Set Description

Abstract: Famous database; from Fisher, 1936

Data Set Characteristics:	Multivariate	**Number of Instances:**	150	**Area:**
Attribute Characteristics:	Real	**Number of Attributes:**	4	**Date Donated**
Associated Tasks:	Classification	**Missing Values?**	No	**Number of W**

Figure 3. Iris Data Set Overview

Code

The inverse cumulative distribution function is given by

$$F^{-1}(p) = \mu - b\ \mathrm{sgn}(p - 0.5)\ \ln(1 - 2|p - 0.5|).$$

Figure 4. CDF Equation

```python
def Bounded_Laplace_Algorithm(original_data, loc, scale, lower, upper, flag):
    if (flag==1):
        noise = np.random.laplace(scale=scale)
    else:
        mu = loc #assign mu to the location parameter

        b = scale #assign b to the scale parameter

        # transform from [uniform distribution] into [Laplace distribution]
        uniform_transform = np.random.uniform(low=0.0, high=1.0, size=1) # Generate the
        p = uniform_transform #assign p to to the uniform random value beteen 0 and 1

        inverse_CDF_noise = mu - b * np.sign(p - 0.5) * np.log(1 - 2 * np.abs(p - 0.5))

        #apply the bounding restrictions to the new method
        bounded_noise = np.clip(inverse_CDF_noise, -lower, upper)

        noise = bounded_noise

    return original_data + noise
```

Figure 5. Code snippet from our implementation, Bounded Laplace Algorithm using CDF Equation (See Figure 4)

Results and Discussion

- 80% / 20% train-test split
- Determined accuracy using Naive Bayes Classifier
- Non-Private Baseline
- Differentially Private accuracy scales with epsilon
- Observations
  - Smaller epsilon reflects higher privacy protection
  - Trade-off between privacy and accuracy
  - Optimal epsilon depends on the situation

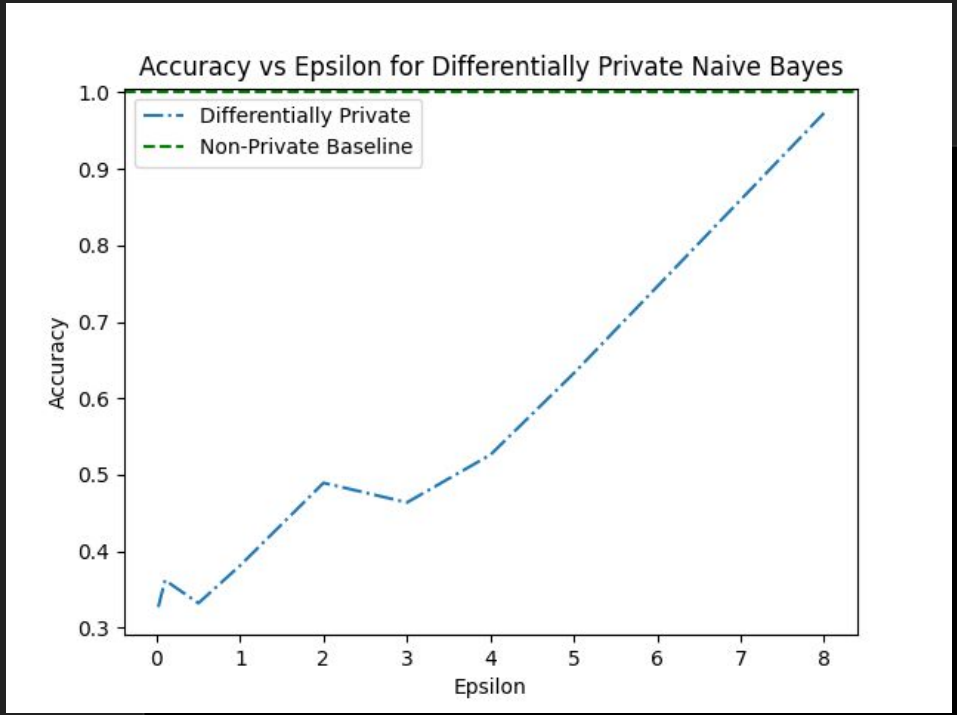

Figure 6. Comparison of accuracy versus epsilon for a differentially private naive Bayes classifier on the Iris dataset (using the Bounded Laplace Mechanism for perturbation of each feature's mean).

Reflection

- Check List

  [✓] To understanding Bounded-Laplace Mechanism

  [✓] Then..Implement it

  [✓] Then..Test it on a dataset

- Take-Home Messages

  [✓] While we are demonstrating this algorithm with a flower dataset, it is typically deployed in datasets containing sensitive human information such as personal health records, income information, and other similar data.

Currently presenting: All Group Members

Question?

Please allow 5-7 business days for a response.

-THANK YOU!-

FAQs

Team CEK

Can I apply X algorithm to **every** dataset?

Differential privacy is not magic that can apply to everything, researchers must be carefully evaluated for each use case to balance the benefits of privacy protection with the potential loss of accuracy in the analysis.

Is Bounded-Laplace Mechanism good?

It is good and easy to use (even implement!) once you understand how it alters your dataset and how to determine the optimal parameters for your specific dataset.

How to find the optimal parameters?

To find the optimal parameters, a simple approach is to loop through a random subset of your dataset and try all reasonable options, recording the best parameter set.

Currently presenting: All Group Members

-THANK YOU!-

#DATA NOT FOR SALE

Currently presenting: All Group Members

Optional material for An Enhanced understanding of the algorithm

Code

Team CEK

```
# ==========================BEGINNING OF Testing If Bounded_Laplace_Algorithm Works============================================================
# The following method is to find the optimal parameters for the BLA.
testing_data = iris_data_df.values.tolist() # We want to perform testing on all rows
scale_testing_values = [0.1, 0.5, 1]
bounds_testing_values = [(0, 1), (0, 10), (-1, 1), (-0.5, 0.5)]

# Set None to begin our testing
best_so_far_result = None
best_so_far_params = None

# Main loop for testing, this is a nested nested loop we would test all combination of testin value, epsilon, and bound to find a good combination.

test_data = calculate_mean(original_data, "sepal.length")

for scale in scale_testing_values:
    for bound in bounds_testing_values:
        lower, upper = bound #set lower and upper to be the current bound.
        result = Bounded_Laplace_Algorithm(test_data, loc=0, scale=scale, lower=lower, upper=upper, flag=2)  # apply function to the subset we just created
        print(f"Calculating: Scale: {scale}, Bound: {bound}, Result: {result}")

        if best_so_far_result is None or np.min(result) < np.min(best_so_far_result):  # since we looking for the smallest value
            best_so_far_params = (scale, bound) # record down what loop id we in
            best_so_far_result = result   # update best so far
            print("New record! Best so far has been updated! ^")


# AT THE END # Print the best out of the best
print(f"The Best Parameters are: Scale: {best_so_far_params[0]}, Bound: {best_so_far_params[1]}, Result: {best_so_far_result}")
print("\n")
print(" ==============================END OF Testing If Bounded_Laplace_Algorithm Works==============================\n")
# ==========================END OF Testing If Bounded_Laplace_Algorithm Works============================================================
```

Figure 6. Code snippet from our implementation, finding optimal parameters