Differential Privacy: The way forward?

Pranjal Aggarwal
Dept. of Computer science
Indian Institute of Technology
Delhi, India
cs5190443@iitd.ac.in

Deepanshu Rohilla
Dept. of Computer science
Indian Institute of Technology
Delhi, India
cs5190427@iitd.ac.in

***Abstract*—We are living in a world where a lot of data is being continuously generated and we need a strong technological solution to ensure privacy to individual data and simultaneously ensuring that innovation is not hindered. For this, the idea of differential privacy comes in. Differential privacy is a system for publicly sharing information and while protecting individual information.**

***Index Terms*—differential privacy, server-client system component, formatting, style, styling, insert**

I. Introduction

A major critique of privacy systems is that it hinders with the innovation. This was majorly due to the use of methods like anonymizing the data fields or adding arbitrary noise to the data. Differential privacy is a solution that ensures individual privacy in the system. The idea is to put noise in the data for statistical queries. Since there is an underlying defined framework, it is much more robust than anonymizing. It also provides a framework to critically analyse the tradeoffs between privacy and accuracy in different learning models. (Nguyen, 2019)

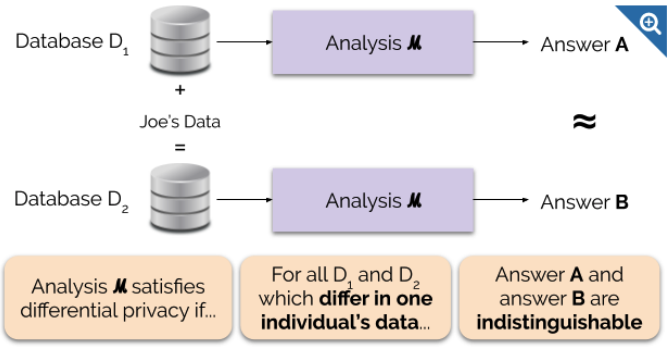

Fig. 1. Dataset based definition of differential privacy.

II. How it works

The section talks about the underlying assumptions and constructs of differential privacy. The underlying idea of differential privacy is to add noise to the answer. Then, we have the task of choosing the noise such that the condition of differential privacy is met. Simultaneously, we do not make the output too much deviated from the actual result that it is no longer useful. We define sensitivity of a function as the amount of change in the function when the input is changed by 1. Given the sensitivity and the ϵ value, we can define the output from differential privacy as

$$F(x) = f(x) + Lap(s/\epsilon)$$

We define neighbouring datasets as datasets that differ at exactly 1 instant. The function satisfying differential privacy has the property that

$$\frac{Pr[F(x) = S]}{Pr[F(x') = S]} \leq e^{\epsilon}$$

For our method, we have used one of the advanced versions of differential privacy (Near & Abuah, 2021). In this method, we have another parameter δ to deal with tolerance in the output. This parameter is used in places where performance is important than privacy. It is an extension of the primitive differential privacy definition. The previous version is one special case of our method in which δ equals zero.

$$Pr[F(x) = S] \leq e^{\epsilon} Pr[F(x') = S] + \delta$$

This δ is kept very small, typically $1/n^2$ or less (where n is the size of dataset). This addition of parameter gives us the flexibility to use approximate differential privacy with different parameter values for different cases.

III. The demonstrative model

This section talks about the steps of building the system that demonstrates the working principles of differential privacy. It also talks about various considerations that needs to be taken care of. The implementation includes asking queries in two ways. The first method is using a server client model to get the output. Another way is using SQL queries to get the desired output.

A. Dataset

We've used 2 different datasets: 1.) Fire Dataset, contatining San Francisco's Fire Department's Call for Service Data, 2.) Covid-19 Dataset: (covid19data, 2021) which is accumulated from official data of covid19india.org over a period of 6 months. It contains patient wise data including symptoms, source, status etc. for all regions in India, of people individuals diagnosed with Covid-19. Both the datasets were cleaned and processed before being used in our implementation. Figure 2 shows some of the fields in the dataset.

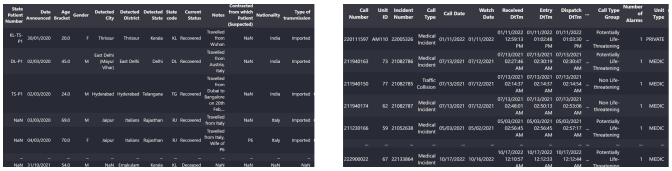

Fig. 2. Samples from Datasets used for Differential Privacy

B. The working principle

The model of differential privacy works on the idea of having a bound on the variation in datasets (while adding individual entries) to define a bound on the output. Apart from this, there is another choice about which type of noise is to be added to the dataset. For that again, there are different noise distributions. For our system, we are using the Laplace mechanism to generate noise.

C. Building the system

We created a server-client system using Gradio (Abid et al., 2019) for frontend and Fastapi for backend. We implemented two models. One of the models was used to make Basic Queries such as mean, median mode on columns, and another to make complicated SQL Queries. For SQL Queries, we used Google's ZetaSQL and for Basic Queries pydp library has been used. (google, 2022) There is a checkbox to enable or disable privacy. There is also a sliding bar to adjust ϵ and δ values for differential privacy based on Laplace Mechanism. (Wilson et al., 2019)

We have given an option to randomly generate SQL queries to focus on the privacy part and not the query. The system picks one of the queries from a set of pre-defined query sets.

IV. Analysis on the server-client system

The main idea of building the server client system was to observe the trend and analyse the trade-off objectively. This section discusses the various outputs and how the output is differentially private. (SELESHI & ASSEFFA, 2018)

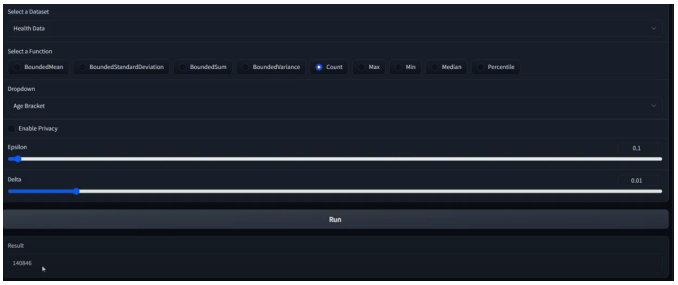

Fig. 3. Output with privacy disabled

The platform can be used to critically analyse the data based on the output with different ϵ and δ values. The images below depicts the output behavious based on different values of ϵ and δ.

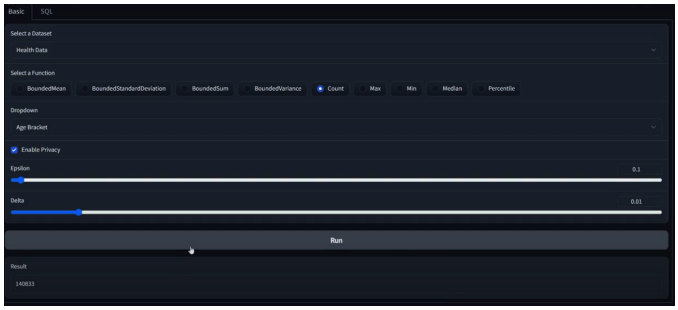

Fig. 4. Output with privacy enabled having low epsilon values

There is a threat to individual privacy in SQL setting. One can ask multiple queries and use the output to pinpoint any individual based on some information about the person. Our system can also be extended to handle SQL queries.

The SQL system has data about covid patients in various parts of India. An SQL query can be designed such that there is only one output. Changing the queries and modifying the ϵ and δ values are one of the methods to ensure privacy for the individual. For example, the figure below shows the output of the person infected with covid in the Indian state of Meghalaya. If someone knows who the individual is, one can get data about his/her age. Thus, for output having single values, the system does not show any data about the individual when the privacy is enabled.

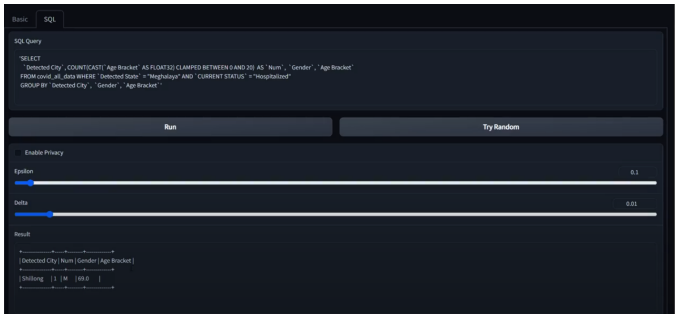

Fig. 5. Output with privacy disabled for the query with single entry

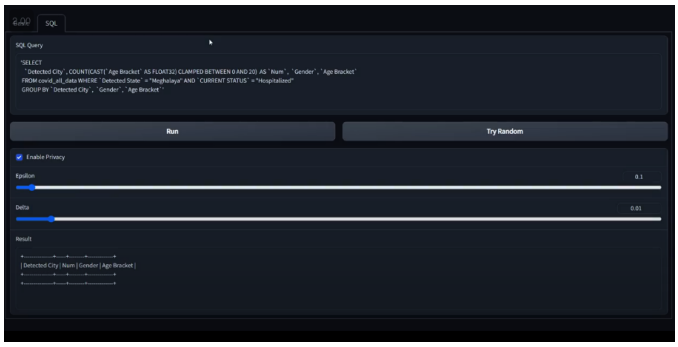

Fig. 6. Output with privacy enabled for the query with single entry

In all the other cases, we can modify the query to get the output and add noise to the system to ensure differential

privacy. This method is quite similar to the one used in the server-client setting.

Fig. 7. Output with privacy enabled for the query with multiple entry

V. Differential Privacy in Machine Learning Models

A. Why Machine Learning?

Recent development of Large Language models have shown human level performance on text related tasks (Brown et al., 2020), (Chowdhery et al., 2022). These models are trained on trillions of words, and as an artifact can often remember as high as 1% of the text they saw at the train time (Carlini et al., 2021). While the knowledge of the worldly data is somewhat desired, the side-effect is that the model can remember personally identifiable information, which it can then generate to unintended user. While one can argue that the models was trained on public data, and thus is not doing any more harm, we must remember that often sensitive data gets leaked into public web, it is mostly difficult to be searched and indexed by search engines. Overall privacy leak is a serious issue, with some of big companies trying to tackle this problem (Carlini, 2022). To tackle this issue, differential privacy can be very useful. In differential privacy, the model is trained on the original data with all the sensitive content, but the learning algorithm is modified in such a manner, that the model cannot remember exact data. However, the cost of the privacy is decrease in accuracy and performance of such models. Unfortunately, there has been very limited research by academia and corporates analyzing this domain, and in this report we aim to give a brief overview on the method currently used, and conduct experiments on small datasets(due to computational limitations) to show the trade-off between the two.

B. Algorithm

One of the first and commonly used algorithm for differential privacy in Machine Leanring is 'DP-SGD (Differentially-Private Stochastic Gradient Descent)' (Song, Chaudhuri, & Sarwate, 2013). The main idea is to take the gradients of each batch and add normal noise to it. Then apply these new gradients to update the model weights. This ensures, that the model doesn't have exact information about what was present in the data, but on multiple batches, it would still be able to learn the required probability distributions. Mathematically,

$$Grads = \mathcal{G} + \alpha * \mathcal{N}(\mu, \sigma^2) \tag{1}$$

Now the question is how to choose the value of α. If we choose too little value, noise will be too low and privacy would be lost, and if we increase it too much, data distribution cannot be learned. In general, α is set equal to the largest norm of gradient in a minibatch. Further to ensure outliers do not influence the model too much(they will have higher gradients and are more likely to be memorised), the gradients are clipped to a maximum value. Since this was introduced in 2013, many improvements have been proposed which are mathematically rigorous and are out of scope for this report. Moreover, machine learning models require fast training and inference speeds, which require series of software level optimizations and parallelization. Thus instead of implementing the method from scratch, we use Meta AI's Opacus library (Yousefpour et al., 2021) and PyTorch to train machine learning models, and do analysis on accuracy vs privacy(epsilon).

C. Results

We next train a classical Naive Bayes and a deep learning Convolutional Neural Network on two different benchmark datasets. While these datasets are not very related to privacy, it is a standard starting point in machine learning. Moreover, training on large datasets is computationally infeasible for us. Nonetheless, the results demonstrated on small datasets are expected to scale to large models and datasets as well, and infact it is more likely, the trade-off between accuracy and privacy would only decrease.

D. IRIS Dataset

We train a simple Gaussian Naive Bayes model on Iris Dataset, where the task is given certain characteristics of flower, the model needs to predict the correct flower. We train two different models, one with differential privacy enabled, and other without. For the former, we evaluate the model for different values of epsilon, and report the results in Figure 8.

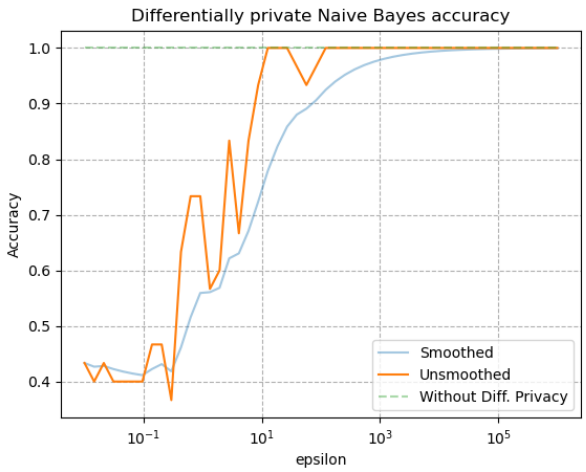

Fig. 8. Trade-off between Accuracy and Differential Privacy

From the figure, it is evident, that even with high epsilon value of 10, the drop in accuracy is roughly 22%. This is significantly high, but the primary reasons are smaller models, and small dataset. Nonetheless, the results demonstrate, that even classical algorithms, which are often known to be sensitive to small changes in probability distribution can perform decently well with differential privacy.

E. CNNs on MNIST Handwriting dataset

We train a 3 layer CNN model on the handwriting dataset with and without differential privacy. Results are reported in figure 9. The task of mnist dataset is to predict the digit given an image of it.

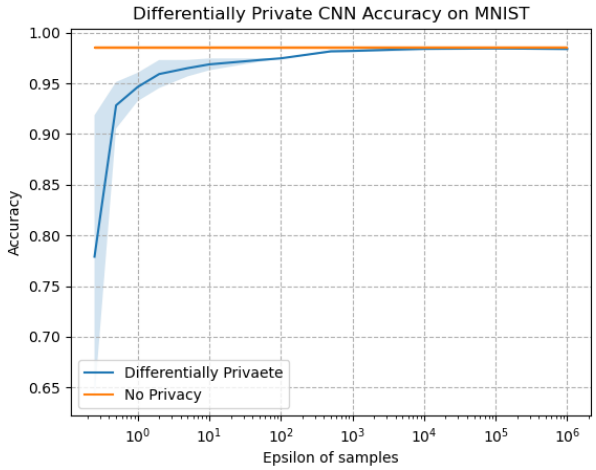

Fig. 9. Trade-off between Accuracy and Differential Privacy

The shaded region in the figure indicates the standard deviation bounds of the accuracy over a set of 3 runs. We observe that even with high differential privacy (epsilon=1), accuracy drop is only 3%. This is much better than the previous dataset, which is most likely because of a better model and also because of larger dataset(50,000 training points).

F. Discussion

Thus large companies, will have to decide on optimizing the trade-off, and find a sweet spot. Policy changes by government can also help to accelerate the process of companies adapting differential privacy. Further, research in this area has been significantly less than that in other domains of machine learning, and the methods should be scaled to larger datasets and models. Certainly, the tradeoff reported here, is the maximum as with increasing data sizes, and improved algorithms, the drop in performance will become less over time.

VI. Advantages of Differential Privacy

A. Mathematically models the trade-off of privacy v/s accuracy

Generally in the context of privacy, the trade-off of privacy v/s accuracy is compared at ethical and administrative level. (Valput, 2020). The discussion is mostly qualitative in nature. The method, however important, lacks implementability and gives very little insights on how to quantify the values. The method of differential privacy caters to this issue and provides a quantitative way to measure the trade-off of privacy v/s accuracy.

B. Helps in sharing data without privacy concerns

This point concerns with the main idea why differential privacy is so famous among businesses. It gives them an automated way to ensure compliance with the laws and simultaneously share data without the concern of individual data coming out. (of Standards & Technology, 2022)

C. Easy to control the parameters by the corporates

The main issue with manual methods of ensuring privacy is the concern of ensuring uniform treatment and difficulty in propagating the changes across different teams globally. The method of differential privacy solves all these concerns. the distribution of noise can be set appropriately based on the tolerance of uniform treatment. The issues of propagating changes is not there in the system by design.

D. Automated tools for monitoring

With more and more research in the area, there are a lot of automated tools that have come up to deal with the detection of privacy breach in the system. There are also automated ways to find the extent of privacy. This can be used to find the optimal value of the parameters for different dataset and application setting.

E. Easy to lay down regulation

Since the method is an objective evaluation and automated operation, it becomes very easy for the government to lay down the rules for differential privacy. It involves specifying the parameter values and permissible noise behaviour based on different datasets and operational setting. For example, in aadhaar related issues, (Agrawal, Banerjee, & Sharma, 2017) differential privacy can be used.

VII. Code

We provide our codebase for client-server and training and evaluating machine learning models at https://github.com/Pranjal2041/SIL802-DP.

VIII. Conclusion

The scale in which the digital products operate, it is very difficult to manually monitor all the content that is uploaded on various platforms. There is a strong need to have frameworks that can help the corporates and the government in automating the process of fair content moderation. The differential privacy model is one such way to ensure that all the beneficiaries involved are on the same page. It can also helps to bridge the gap among different bodies and act as a communicating framework to address various tradeoff issues coming up.

Acknowledgment

We would like to thank Dr. Aaditeshwar Seth for providing us with the opportunity to work on this topic. We also thank him for overseeing the work and giving valuable feedback on the work done. We would also like to extend our gratitude to the students of SIL802 group for constructive discussions on the subject matter.

References

Abid, A., Abdalla, A., Abid, A., Khan, D., Alfozan, A., & Zou, J. Y. (2019). Gradio: Hassle-free sharing and testing of ml models in the wild. *ArXiv*, *abs/1906.02569*.

Agrawal, S., Banerjee, S., & Sharma, S. (2017). *Privacy and security of aadhaar: A computer science perspective.* Retrieved from https://www.cse.iitd.ac.in/~suban/reports/aadhaar.pdf

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., ... Amodei, D. (2020). Language models are few-shot learners. *ArXiv*, *abs/2005.14165*.

Carlini, N. (2022). *Privacy considerations in large language models.* Retrieved from https://ai.googleblog.com/2020/12/privacy-considerations-in-large.html

Carlini, N., Tramèr, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., ... Raffel, C. (2021). Extracting training data from large language models. In *Usenix security symposium.*

Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., ... Fiedel, N. (2022). Palm: Scaling language modeling with pathways. *ArXiv*, *abs/2204.02311*.

covid19data. (2021). *Covid19-india api.* Retrieved from https://github.com/covid19india/data

google. (2022). *Differential privacy.* Retrieved from https://github.com/google/differential-privacy

Near, J. P., & Abuah, C. (2021). *Variants of differential privacy.* Retrieved from https://programming-dp.com/ch8.html

Nguyen, A. (2019). Understanding differential privacy. Retrieved from https://towardsdatascience.com/understanding-differential-privacy-85ce191e198a

of Standards, N. I., & Technology. (2022). *Differential privacy blog series.* Retrieved from https://www.nist.gov/itl/applied-cybersecurity/privacy-engineering/collaboration-space/focus-areas/de-id/dp-blog

SELESHI, B., & ASSEFFA, S. (2018). *A case study on differential privacy.* Retrieved from https://www.diva-portal.org/smash/get/diva2:1113852/FULLTEXT01.pdf

Song, S., Chaudhuri, K., & Sarwate, A. D. (2013). Stochastic gradient descent with differentially private updates. *2013 IEEE Global Conference on Signal and Information Processing*, 245-248.

Valput, D. (2020). *Ethical aspects of artificial intelligence, part 2/2: Differential privacy.* Retrieved from https://datascience.aero/ethical-aspects-ai-differential-privacy/

Wilson, R. J., Zhang, C. Y., Lam, W., Desfontaines, D., Daniel, Simmons-Marengo, & Gipson, B. (2019). Differentially private sql with bounded user contribution. *ArXiv*.

Yousefpour, A., Shilov, I., Sablayrolles, A., Testuggine, D., Prasad, K., Malek, M., ... Mironov, I. (2021). Opacus: User-friendly differential privacy library in pytorch. *ArXiv*, *abs/2109.12298*.