***A Compendium of Thoughts: Generative (Diffusion) Differential Privacy***

In this age of digital economy, we heard phrases like “Data is new oil”, “Data Democratization”, “Data monetization” what does this mean is, we are dependent on Data more than ever. Data comes from both internal and external sources and data has no boundaries. Take for example an analyst wants to analyze the data using OpenAI out of curiosity or sloppiness, there is every possibility of a [data leak](https://www.washingtonpost.com/technology/2023/07/13/ftc-openai-chatgpt-sam-altman-lina-khan/) as your query will be retained by OpenAI for retraining, Just like google storing search history for personalization to you and monetization for them. If you are wondering can OpenAI analyze tabular data look no further there is a plugin for excel - [tabulate](https://github.com/openai/tabulate). So, all this is a nightmare scenario for PRIVACY.

If we take a brief tour down the history lane, how to address the problem of privacy, we have started with simple anonymization and turned to more robust statistical techniques like [t-Closeness](https://www.cs.purdue.edu/homes/ninghui/papers/t_closeness_icde07.pdf) to [m-invariance](https://arxiv.org/abs/2306.15371). Differential Privacy is a mathematically rigorous framework that gaurenties privacy parameterized by ε- This is the privacy budget. It measures the strength of the privacy guarantee by bounding and δ - Bounds the probability of the privacy guarantee not holding. This helps to mitigate “linkage attacks” using auxiliary knowledge or data that data curator is not aware. Enterprises take privacy seriously to protect their reputation and to adhere regulation like [HIPPA](https://www.cdc.gov/phlp/publications/topic/hipaa.html#:~:text=The%20Health%20Insurance%20Portability%20and,the%20patient's%20consent%20or%20knowledge.), [CCPA](https://oag.ca.gov/privacy/ccpa) , [GDPR](https://gdpr-info.eu/) etc. using various tools. But in my view, there is very little adoption of differential privacy which is an irony in an increasingly [model and data sharing](https://arxiv.org/abs/2007.07646) world.

There are several Differential privacy libraries - [diffprivlib](https://github.com/IBM/differential-privacy-library) from IBM, [openDP](https://opendp.org/) , Microsoft [smartnoise](https://smartnoise.org/), Google [differential privacy tool](https://github.com/google/differential-privacy) adopted by openminded as [pyDP](https://github.com/OpenMined/PyDP) are popular with python. For [Spark](https://spark.apache.org/) framework you can use [tmlt.analytics](https://docs.tmlt.dev/analytics/latest/index.html). In case of “R” you can use [DiffPriv](https://cran.r-project.org/web/packages/diffpriv/diffpriv.pdf). Most of these libraries provide a way to impute basic statistical functions like histograms, mean, variance, standard deviation, quantiles with differential privacy. Some of them provide hooks into standard model building methods like regression, logistic regression, tree-based which obviously need more than basic statistics. To understand the complete techniques and mathematical treatment you can refer to [mechanisms](https://diffprivlib.readthedocs.io/en/latest/modules/mechanisms.html) for documentation. When it comes to neural networks and DNN, TensorFlow provides [DP-SGD](https://www.tensorflow.org/responsible_ai/privacy/api_docs/python/tf_privacy/compute_dp_sgd_privacy) and PyTorch provides [opacus](https://github.com/pytorch/opacus). For comprehensive practical understanding of differential privacy in ML you can refer to this google paper [DP-fy ML](https://arxiv.org/pdf/2303.00654.pdf).

When you take a close look at DP you will see it is adding noise so can I create synthetic dataset using this? The answer is yes. Which I will dwell into shortly. Here is the repo “[DiffusionDP](https://github.com/prakashmstpt/DiffusionDP/)” which I will be using to explain technical aspects.

Everyone would have come across diffusion models by now, if not they would have heard [DALL-E3](https://openai.com/dall-e-3) and impressive art mashups it can create in conjunction with [ChatGPT](https://chat.openai.com/) (Tempest in as tea cup my favorite using DALL-E3). These are specific types of generative models A cup of coffee with a stormy sky and lightning in the background

Description automatically generatedwhich has roots in [VAE](https://arxiv.org/pdf/1312.6114.pdf), [GAN](https://arxiv.org/abs/1406.2661) and [Score-based](https://arxiv.org/abs/2011.13456). To understand diffusion holistically you can refer to “[Understanding Diffusion Models: A Unified Perspective](https://arxiv.org/abs/2208.11970)”. Score-based models use stochastic differential equation (SDE). Further research developments lead to conditional flow matching which uses ordinary differential equations (ODE). Both are very computationally expensive and will need GPUs for evaluation. The concreate implementation of conditional flow match using PyTorch can be found here in [TorchCFM](https://pypi.org/project/torchcfm/).

To learn the score function or the vector field instead of neural networks a novel gradient boosting method was introduced in a recent paper “[Generating and Imputing Tabular Data via Diffusion and Flow-based Gradient-Boosted Trees](https://arxiv.org/abs/2309.09968)”. This is promising in many fronts one it uses [XGBoost](https://xgboost.readthedocs.io/) which is the workhorse of data scientist and has been a de facto standard in the industry. Second since this is a tree-based method it can gracefully handle both continuous and categorical data. Third it is a bit memory intensive rather than compute intensive so can achieve similar results with or without GPUs as supported by XGBoost. Another main advantage as the paper suggest is to impute missing data. The concrete implantation of this is done in [ForestDiffusion](https://pypi.org/project/ForestDiffusion/).

To understand the effectiveness of the ForestDiffusion I have created a [python notebook](https://github.com/prakashmstpt/DiffusionDP/blob/main/StudentGen.ipynb) comparing with diffprivlib. For this exercise we use “[Predict student’s dropout and academic success](https://archive.ics.uci.edu/dataset/697/predict+students+dropout+and+academic+success)” UCI dataset which has 36 features and around 4.4K records. I want you to get first idea of privacy budget ε. I have used ε =1 for the Target and ε = 0.01 for the grade and visualize histogram to understand the impact via [k-l divergence](Kullback–Leibler%20divergence%20%20Wikipedia%20https:/en.wikipedia.org%20›%20wiki%20›%20Kullback–Leibler_d...). You can play with ε in the notebook.

A graph of red and blue bars

Description automatically generated A graph of a bar graph

Description automatically generated

When ε=1 it is just enough i.e. balancing privacy with noise as you can see the histograms match. This is the default setting of ε. If ε is low, you are favoring more privacy, and it can deviate from the distribution more than intended. Now question is what happens when ε is more than one for that instance it can take infinity theoretically and we should expect exact same data as original, and noise will completely cancel out.

Now we will turn to ForestDiffusion it takes several hyperparameters which include XGBoost parameters which I will not mention user can refer to documentation. Some of the key ForestDiffusion parameters are

n\_t – number of noise levels/sampling steps. For performance you can increase this.

duplicate\_K – number of noises per sample. You should decrease this if you want reduce memory footprint.

bin\_, cat\_, int\_ indexes to specify index arrays of input data types.

diffusion\_type can take “flow” for ODE and “vp” for SDE.

We will compare how ForestDiffusion performed with same variables on a generated dataset.

A graph with red and blue bars

Description automatically generated A graph of different colored bars

Description automatically generated

I used default parameter for n\_k =50, duplicake\_K=100 and diffusion\_type=” flow”. You can vary hyperparameters to fit the data as required.

ForestDiffusion model can be used a classification and regression. These methods have been implemented in the library. There is an interesting [paper](https://arxiv.org/abs/2309.16779) suggesting how generative models make better classifiers.

CDO and CSO are in a constant tussle in trying to balance out the need to provide cutting edge innovative AI tools with adequate security. You can get a glimpse of how industry is coping from [ycombinator new feed](https://news.ycombinator.com/item?id=35330438). CDO should encourage the adoption of differential privacy in their enterprise to address the concerns of CSO. In my view enterprises should provide clean room approach isolating innovation area for employs to innovate using synthetic data all the time. The importance of synthetic data is underscored in the [NIST blog](https://www.nist.gov/blogs/cybersecurity-insights/differentially-private-synthetic-data) which you can refer. Industry leader Nvidia has identified [synthetic data as one of the 2024 trends](https://blogs.nvidia.com/blog/ai-in-financial-services-survey-2024/?ncid=so-link-401627-vt09&=&linkId=100000235363209) in Financial industry. So generative model can become an integral part of enterprise to enhance productivity and privacy.