Data Science Individual Project Second Interim Report
Differentially Private Location-based Histogram Publication

1. Introduction

With prevalence of applications collecting data about life patterns of users, it is made possible for analysts to look into both open datasets and private datasets for intelligence. However, it is of prime importance to maintain individual privacy while distributing any of the data and findings. In this project, the problem of releasing one-dimension dataset in its histogram representations under differential privacy policies will be addressed.

Once sufficient privacy enforced, analyst can then publish their findings without risking any sensitive data of both firms and users. Such work will contribute not only to daily life of everyone, but also to development strategies of firms. Taking restaurant reviews and check-in data as an example, analyst can then segment users into different preferences and provide them with suggestions of new restaurants when they are travelling to a new place; analysts can also look into the positioning of their firm and adopt strategies to either consolidate their reputation or mitigating to the desired market segments.

With the possible benefits said, this project covers over a broad domain. By modifying differential private algorithm, our work would be applicable for applications which uses histograms as query answers. Our work would hopefully provide insights on clustering and classification as well. Among all the datasets, our work will be focusing on one-dimension datasets and specifically on datasets containing check-in records.

2. Preliminaries

2.1. Histogram

Given an attribute θ with a set of values V in a one dimensional dataset D, we can aggregate the count (or the frequency), f for each value $v \in V$. With the set of θ and its corresponding count f, a histogram, H can be generated. A histogram H regarding the attribute θ can then be denoted as $H = \{\mathrm{h}_1, \mathrm{h}_2, \dots, \mathrm{h}_n\}$ with h_i representing the count of values of θ covered in the bin i. Under most circumstances, bin i and bin j do not overlap for every $i \neq j$ (i.e. $range(\mathrm{H}_i) \cap range(\mathrm{H}_j) \neq \emptyset$). With such a histogram, we would be able to answer to different range queries.

2.2. Differential Privacy

In this article, we will look into algorithms generating output from databases while enforcing sufficient privacy such that released data. Assume there exists histograms H_1 and H_2 from database D_1 and D_2 which differs by exactly one record. H_1 and H_2 can then be referred as neighbours. Base on these assumptions, differential privacy can then be defined as follows:

Definition 1.[1] A randomised algorithm $\mathcal{A}$ is said to be ϵ-differentially private if for any two neighbour histograms H_1, H_2 and any subset of output value $S \subseteq Range(\mathcal{A})$,

$$\Pr(\mathcal{A}(\mathrm{H}_1) \in S) \leq \exp(\epsilon) \cdot \Pr(\mathcal{A}(\mathrm{H}_2) \in S)$$

where $\epsilon > 0$.

With the parameter ϵ, users can specify the level of privacy to be enforced. The smaller ϵ, the higher privacy protection will be applied. Normally, ϵ will be set as a small value (e.g. $\epsilon < 1$) such that removal of one individual data prior to running the algorithm will not significantly affect the output (i.e. the histogram).

It is often that a series of algorithms $\mathcal{A} = \{\mathcal{A}_1, \mathcal{A}_2, \dots, \mathcal{A}_N\}$ will be run to generate the output. The algorithms will each be ϵ-differentially private, and thus can come up with a privacy budget of $\epsilon = \sum_{i=1}^{N} \epsilon_i$ that will be allocated to each of the algorithm $\mathcal{A}_i$.

In the literatures, one of the most well-developed algorithms for enforcing differential privacy is the Laplace mechanism. This mechanism masks the original data with random noises generated from the Laplace distribution, $Lap(\sigma)$ (i.e. a Laplace distribution with mean 0 and scale σ). The scale is determined based on the concept of a sensitivity of a function f over the histogram.

Definition 2. Denote $f(H)$ as a function of the histogram H which generates a vector in $\mathbb{R}^d$. The Laplace mechanism will then give output of $Lap(H) = f(I) + \boldsymbol{z}$, where $\boldsymbol{z}$ is a d-length vector and that each $z_i \sim Lap(\Delta f/\epsilon)$. The constant Δf is the sensitivity of f and is defined as the maximum difference in f between 2 neighbouring histograms H_1, H_2 (i.e. $\Delta f = max_{H_1,H_2} \|f(\mathrm{H}_1) - f(\mathrm{H}_2)\|_1$.

Theorem 1.[1] For any function $f : H \rightarrow \mathbb{R}^d$, the algorithm set $\mathcal{A}$ is said to be ϵ-differentially private if

$$\mathcal{A}(H) = f(H) + [Lap_1(\Delta f/\epsilon), \dots, Lap_d(\Delta f/\epsilon)]$$

where each $Lap_i(\Delta f/\epsilon)$ are i.i.d. Laplace variables with scale $\sigma = \Delta f/\epsilon$.

As given any database, adding one new record will lead to an increase by exactly 1, the sensitivity constant $\Delta f = 1$. In other words, the mechanism can be simplified as adding random noises of $Lap(1/\epsilon)$, where ϵ is the privacy budget allocated to the Laplace mechanism.

3. Literature Review

There has been a number of new knowledge in the field of differential privacy recently. To be specific, this project is inspired by the work of Zhang et. al. [1] on AHP and its edge over other clustering algorithms. To obtain a broad picture of the latest development, a series of research papers, which cover different histogram settings, dimensionality, and nature of data are reviewed. Apart from that, work focusing on calibrating and benchmarking is also studied.

3.1. Clustering Algorithms in Histogram

The principle behind Zhang et. al. [1] is to minimise Laplace error applied onto the dataset with help of AHP. To come up with the optimal solution, three candidate algorithms are put into comparisons.

The AHP suggested is still, however, questionable. Although the algorithm shows a clear edge over others like *PHP* [11], *GS* [12], *NoiseFirst* [13], *StructureFirst* [13], these comparisons are all made with experimental datasets of scale of 10^3 and 10^4. Whether the algorithm excels in datasets of other scales is still questionable.

Free parameters are also not fully discussed in the research. The work focuses on comparisons of algorithm over different privacy budget. However, the fact that privacy budget can be freely allocated at different stage is not fully exploited, but is preset and hard-coded instead makes the work extendable in the future.

3.2. Node Differential Privacy

In this domain, the general concept towards differential privacy is base on graph and networks. Once differential privacy is enforced, nodes and all its edges should not be distinguishable from other nodes and edges [4].

Day et. al. [4] focuses on graph data representation. In this domain, they have suggested two solutions, that has an edge over the current state-of-art in lower-degree nodes. These are all made possible with their novel graph projection mechanism.

Apart from the reminder on sensitivity, the way Day et. al. compared free tuning parameters can also be applicable on this project given some improvement. They have selected a few values for testing their new algorithm before finalising the preset value. This, however, has restricted the flexibility nature of the free tuning parameter, and thus missed that opportunity to optimise and generalise the algorithm with help of a function that help tune these attributes.

3.3. High Dimension Data

For datasets with higher dimensions, enforcing differential privacy will usually suffer from what is called ‘curse of high dimensionality’ resulting in either heavy computation power requirement, excessive error, or low privacy. The current state-of-art algorithm MWEM is reviewed to be a NP-hard computationally complex algorithm by Hardt et. al. [10].

Chen et. al. [5] proposed to solve the ‘curse’ with a top-down partitioning algorithm on a set-valued dataset. With the novel partition, they yielded a stronger privacy protection with the same amount of relative error. They also suggested that a non-interative data sanitisation can be achieved situationally if underlying data is carefully used. This which would allow users to directly interact with the sanitised dataset without needing a sanitising mechanism as a middleman to modify queries from users.

As their proof for non-interactive data sanitisation is fairly situational, there would still be a big gap of generalising such a sanitisation scheme. On the other hand, their utility is only based on overall amount of relative error. This would have allowed results with low relative error, while having a highly varied error slip through. Further work would be needed to improve the scheme such that error can be kept small and concentrated, and thus, for preserving accuracy of any queries.

In another research, Chen et. al. [3] focuses on solving the problem with a sampling-based inference which allows thicker spread of privacy budget. The solution depends on the conditional independence and the one dimensional association across attributes. With a higher privacy budget at each step, the accumulated noise in the sanitised dataset would be highly reduced.

As the solution depends on the important assumption of conditional independence, its effectiveness would be highly situational, and should be reconsidered before using it as a general approach for enforcing differential privacy.

Despite the presumption, this piece of work brings a new insight towards efficient spending of privacy budget. Sampling-based inference should be further studied so that it may be applied onto other algorithms to improve the overall accuracy.

Dimitrakakis et. al. [7] looked into sampling inference with posterior sampling in Bayesian network. With little previous differential privacy research in the field of Bayesian inference, this piece provided a building block and a proof for using posterior sampling for enforcing differential privacy.

This piece of work is mostly out of the domain of our project as it mainly focuses on theoretical use of Bayesian inference on high dimension datasets. It however shows that the field of differential privacy is evolving so quick that new scheme with different statistics skills are being looked into.

Gaboardi et. al. [8] proposed an algorithm, named DualQuery, with worst-case complexity in exponential relation against dataset dimension. Despite the exponential increase, they have shown that their algorithm is computationally more concise than the state-of-art by applying the algorithm on a synthetic dataset with more than 500,000 attributes.

Their algorithm, however, only shows edge over in high-dimension queries. In other words, MWEM still remains to be the state-of-art in lower-dimensions queries, while Gaboardi et. al. did not provide the boundary for switching from MWEM to DualQuery for saving computational power.

3.4. Time-series Data in Histogram

In the domain of time-series data, Chen et. al. [2] considered it as an infinite stream of data. Their work focuses on generating differentially private histograms continuously. A sampling based algorithm with a retroactive grouping mechanism is adopted to enhance computational, and spatial efficiency, and thus incurring a smaller delay when publishing histograms.

As timestamps are not the major focus of this project, the work from Chen et. al. is mainly out of scope. However, the use of a Bernoulli sampling scheme may be applicable on this project for getting a more accurate differentially private result while suffering from reduced Laplace noise.

3.5. Differential Privacy without External Noise

While most of the works are focusing on enforcing differential privacy with help of introducing noises into the dataset, Duan [9] worked on answering sum queries without any external noises. Duan pointed out that state-of-art differential privacy techniques is flawed for aggregate queries due to the zero-mean symmetric nature of Laplace distribution. He then argued that if the dataset is sufficiently large, aggregate queries would be private enough itself given that sum of multivariate Gaussian distributions converges when n is large as stated in central limit theorem.

Although aggregate queries are not the main focus of this project, Duan's work has opened a big gap in the field of differential privacy as a possible flaw has been discovered. The work also provides an insight of whether the original data is private enough once the dataset is large enough. If the dataset is, indeed, private enough, publishing an unsanitised anonymised would ensure the most accurate result for various data mining tasks.

3.6. Benchmark and Metrics

While all the above works are focusing on varies algorithms optimising in different scenarios, there are actually limited work on benchmarking algorithms across the table. Hay et. al. [6] suggested DPBench by considering performance of different data-dependent and data-independent algorithms under different scales and sizes.

The work from Hay et. al. adopted a very objective comparison scheme between different algorithms. It also addressed that as the algorithm will provide random noises, it would be biased if comparisons are drawn based on the shape of noise graph across different scale and attributes among different different algorithms. Such standard should also be upheld in this research should a similar comparison be adopted.

4. Dynamic Privacy Budget Allocation and Laplace Noise

With AHP, Laplace noise with the same scale are applied to every bin [1]. The masked count will then go through the threshold function. This would restrict all the cluster means to be in the band of valid value.

However, the state of art will result in a big cluster at the lower threshold and/or upper threshold depend on the threshold function. In order to solve this problem, the Laplace noise applied to each bin would need to be relative to its count such that $\Pr\,(h_i + \ Lap(1/\epsilon_i) < lower\ threshold)$ and $\Pr\,(h_i + \ Lap(1/\epsilon_i) > upper\ threshold)$ can be minimised or reduced.

The optimal way to apply dynamic privacy budget such that it would always be related to count of each bin. This however is not sufficient to prove ϵ-differentially privacy. In order to align with ϵ-differentially privacy, a possible way would be to discretise the privacy budget ϵ into decimal points (e.g. 0.001). An optimal solution can then be found with help of the weak composition theory. This would however only provide an estimate of the optimal solution as we are handling a discretised solution set. While a more accurate solution can be come up by discretising the privacy budget into smaller sections (e.g. 0.00001), it would also make the algorithm a exponentially complex problem.

4.1. Simple Dynamic Privacy Budget Allocation

To tackle the problems mentioned above, we have developed a novel dynamic privacy budget allocation function which depends on only the sorted rank of the bins instead of the counts of each bin. Such a function would allocate privacy budget linearly according to the sorted rank of a bin.

Definition. When applying dynamic privacy on histogram H, a dynamic privacy budget allocation function is defined as the following:

$$f(i,n) = \frac{n-i}{n \times \frac{n+1}{2}} \qquad for\ 0 \le i \le n-1$$

where n is the number of bins, i as the index of the bin.

To illustrate how the function would allocate budget, we assume an example with 5 bins and a budget $\epsilon = 0.05$ to be spent on Laplace noise masking. With AHP, each bin would get an $\epsilon_i = 0.05$ for $0 \le i \le 4$. This would result in an average $\bar{\epsilon} = \ \epsilon = \ 0.05$. With the novel function, we will obtain the following $f(i,n)$

$$f(i,n) = \begin{cases} \frac{2\times5}{5\times6} = 0.3333 & for\ i = 0 \\ \frac{2\times4}{5\times6} = 0.2667 & for\ i = 1 \\ \frac{2\times3}{5\times6} = 0.2 & for\ i = 2 \\ \frac{2\times2}{5\times6} = 0.1333 & for\ i = 3 \\ \frac{2\times1}{5\times6} = 0.0667 & for\ i = 4 \end{cases}$$

We would then calculate $\epsilon_i = f(i,n) \times n \times \epsilon$ such that we can obtain $\bar{\epsilon} = \epsilon = 0.05$. As an result, we will obtain a series of ϵ_i as shown in the Figure 1. Laplace noise with scale $1/\epsilon_i$ will then be applied onto each bin.

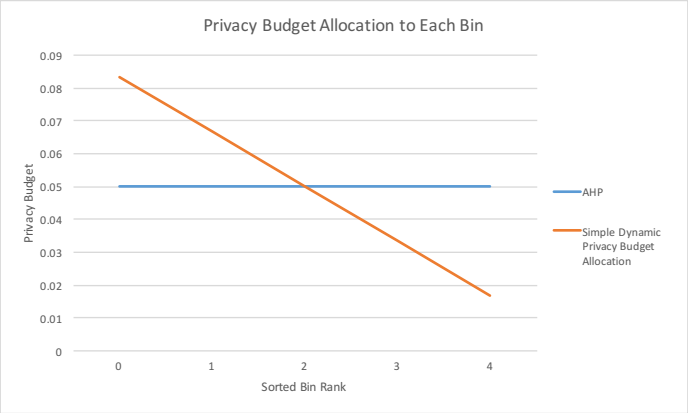

Figure 1 Simple Privacy Budget Allocation 1

From Figure 2, we can see the scale of Laplace noise added to each bin. It is shown that the Simple Dynamic Privacy Budget Allocation is able to apply noises of smaller scales to bins with smaller counts, while applying noises with larger scale to bins with higher counts as a compensation for achieving ϵ-differentially privacy.

4.2. Flexible Dynamic Privacy Budget Allocation

In the previous section, Simple Dynamic Privacy Budget Allocation is introduced. However, from the graph, we can observe that reducing scale of noise at bins with smaller counts would lead to an exponential increase in scale for the bins with higher count. This would make the function applicable only when the histogram of the dataset shows the same trend after sorting.

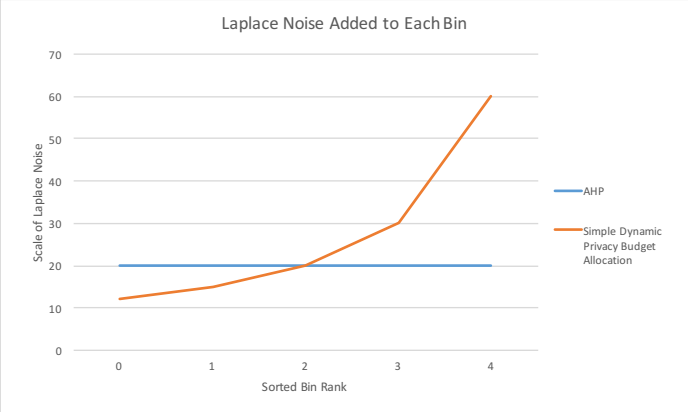

Figure 2 Simple Privacy Budget Allocation 2

In order to make the function more general and flexible, we have come up with the Flexible Dynamic Privacy Budget Allocation which users can specify how diverse the privacy budget should be allocated. This would be specified with a variable δ.

Definition. When applying flexible dynamic privacy budget allocation on histogram H, the dynamic privacy budget allocation function consist of a weight function $v(i,n,\delta)$ and the allocation function $f(i,n,\delta)$:

$$v(i,n,\delta) = \left(\left\lceil\frac{n}{2}\right\rceil + \frac{n-2i-1}{2}\times\delta\right) \qquad for\ 0 \le i \le n-1$$

$$f(i,n,\delta) = \frac{v(i,n,\delta)}{\sum_{i=0}^{n-1} v(i,n,\delta)} \qquad for\ 0 \le i \le n-1$$

where n is the number of bins, i as the index of the bin, and $step$ as decrement of the portion of differential privacy budget allocated at the bin i

To illustrate how the function would allocate budget, we again assume an example with 5 bins and a budget $\epsilon = 0.05$ to be spent on Laplace noise masking. With AHP, each bin would get an $\epsilon_i = 0.05$ for $0 \le i \le 4$. This would result in an average $\bar{\epsilon} = \epsilon = 0.05$. With the novel function, we will obtain the following $f(i,n,\delta)$

$$f(i,n,0) = \begin{cases} 0.2 & for\ i = 0 \\ 0.2 & for\ i = 1 \\ 0.2 & for\ i = 2 \\ 0.2 & for\ i = 3 \\ 0.2 & for\ i = 4 \end{cases} \quad f(i,n,0.5) = \begin{cases} 0.2667 & for\ i = 0 \\ 0.2333 & for\ i = 1 \\ 0.2000 & for\ i = 2 \\ 0.1667 & for\ i = 3 \\ 0.1333 & for\ i = 4 \end{cases} \quad f(i,n,0) = \begin{cases} 0.3333 & for\ i = 0 \\ 0.2667 & for\ i = 1 \\ 0.2000 & for\ i = 2 \\ 0.1333 & for\ i = 3 \\ 0.0667 & for\ i = 4 \end{cases}$$

We would then calculate $\epsilon_i = f(i,n) \times n \times \epsilon$ such that we can obtain $\bar{\epsilon} = \epsilon = 0.05$. As an result, we will obtain a series of ϵ_i as shown in the Figure 3. Laplace noise with scale $1/\epsilon_i$ will then be applied onto each bin.

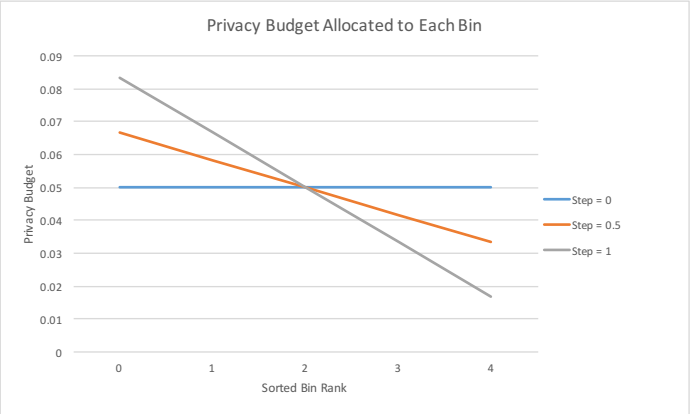

Figure 3 Flexible Dynamic Privacy Budge Allocation 1

As from Figure 4, we can observe that AHP and the Simple Dynamic Privacy Budget Allocation are essentially special cases of the Flexible Dynamic Privacy Budget Allocation where step is 0 or 1 respectively. With the introduction of δ, we can now adjust the privacy budget to suit sorted histograms with different distribution.

Now that the function can replicate the performance of AHP, and provide the flexibility of dynamically allocating privacy budget, it is also very important to prove that adopting such a function will not violate the principle of ϵ-differentially privacy.

Assumes we have two neighbour histograms H_1, H_2. $\Pr(\mathcal{A}(\mathrm{H}_1) = \alpha)$ can be calculated as follows

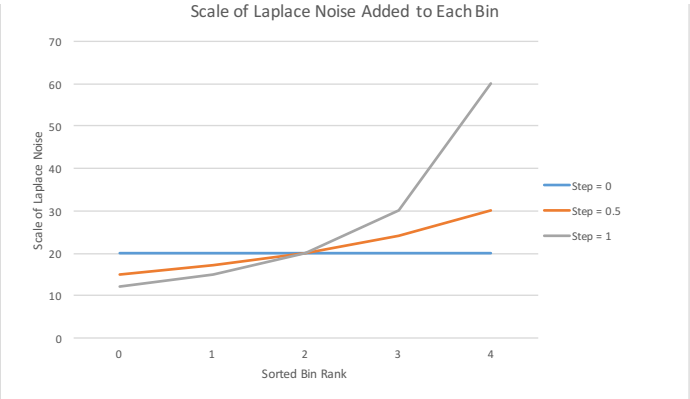

Figure 4 Flexible Dynamic Privacy Budge Allocation 2

$$\begin{aligned}
\Pr(\mathcal{A}(\mathrm{H}_1) = \alpha) &= \prod_{i=0}^{n-1} \Pr\left(h_{1,i} + Lap\left(\frac{1}{\epsilon_i}\right) = \alpha_i\right) \\
&= \prod_{i=0}^{n-1} \frac{\epsilon_i}{2} \exp\left(-|\alpha_i - h_{1,i}|\epsilon_i\right) \\
&= \prod_{i=0}^{n-1} \frac{1}{2} \frac{v(i,n,\delta) \times n \times \epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \exp\left(-|\alpha_i - h_{1,i}|\epsilon_i\right) \\
&= \left(\frac{n\epsilon}{2}\right)^n \exp\left(\sum_{i=0}^{n-1} -|\alpha_i - h_{1,i}|\epsilon_i\right) \prod_{i=0}^{n-1} v(i,n,\delta) \\
&= \left(\frac{n\epsilon}{2}\right)^n \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} -|\alpha_i - h_{1,i}|v(i,n,\delta)\right) \times \prod_{i=0}^{n-1} v(i,n,\delta) \\
&= \left(\frac{n\epsilon}{2}\right)^n \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} -|\alpha_i - h_{1,i}|v(i,n,\delta)\right) \times \prod_{i=0}^{n-1} v(i,n,\delta)
\end{aligned}$$

Repeat the process, and we will obtain the following for H_2

$$\begin{aligned}
\Pr(\mathcal{A}(\mathrm{H}_2) = \alpha) &= \prod_{i=0}^{n-1} \Pr\left(h_{2,i} + Lap\left(\frac{1}{\epsilon_i}\right) = \alpha_i\right) \\
&= \prod_{i=0}^{n-1} \frac{\epsilon_i}{2} \exp\left(-|\alpha_i - h_{2,i}|\epsilon_i\right) \\
&= \prod_{i=0}^{n-1} \frac{1}{2} \frac{v(i,n,\delta) \times n \times \epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \exp\left(-|\alpha_i - h_{1,i}|\epsilon_i\right) \\
&= \left(\frac{n\epsilon}{2}\right)^n \exp\left(\sum_{i=0}^{n-1} -|\alpha_i - h_{2,i}|\epsilon_i\right) \prod_{i=0}^{n-1} v(i,n,\delta) \\
&= \left(\frac{n\epsilon}{2}\right)^n \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} -|\alpha_i - h_{2,i}|v(i,n,\delta)\right) \times \prod_{i=0}^{n-1} v(i,n,\delta) \\
&= \left(\frac{n\epsilon}{2}\right)^n \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} -|\alpha_i - h_{2,i}|v(i,n,\delta)\right) \times \prod_{i=0}^{n-1} v(i,n,\delta)
\end{aligned}$$

$$\frac{\Pr(\mathcal{A}(\mathrm{H}_1)=\alpha)}{\Pr(\mathcal{A}(\mathrm{H}_2)=\alpha)} = \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} -|\alpha_i - h_{1,i}| v(i,n,\delta)\right) / \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} -|\alpha_i - h_{2,i}| v(i,n,\delta)\right)$$

$$= \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} v(i,n,\delta)\left(|\alpha_i - h_{2,i}| - |\alpha_i - h_{1,i}|\right)\right)$$

$$\leq \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} \sum_{i=0}^{n-1} v(i,n,\delta)\left(|h_{2,i} - h_{1,i}|\right)\right)$$

$$\leq \exp\left(\frac{\epsilon}{\sum_{i=0}^{n-1} v(i,n,\delta)} n v(0,n,\delta)\right)$$

$$\leq \exp(\epsilon)$$

And thus, from Definition 1, we have proved that this mechanism is of ϵ-differentially private.

5. Test and Experiment

In this section, only Flexible Dynamic Privacy Allocation function will be compared with AHP as the Simple Dynamic Privacy Allocation function is just a special case of the more advanced one with $\delta = 1$. Mainly 4 datasets will be used for comparison between the algorithms, namely TIST, TSMC, and ubicomp. To cohere with AHP, we will look into two aspects when measuring utility and errors, namely KLD for data distribution, and MSE for range queries.

As the testing data is still being generated, only the MSE value for range queries for AHP is reported here.

5.1. Range Queries

As a standard, AHP has scored an average MSE of 834 across all the records for TIST.

6. Bibliography

1. Xiaojian Zhang, Rui Chen, Jianliang Xu, Xiaofeng Meng, Yingtao Xie. "Towards Accurate Histogram Publication under Differential Privacy." *2014 SIAM International Conference on Data Mining*, 2014: 9.
2. Rui Chen, Yilin Shen, Hongxia Jin. "Private Analysis of Infinite Data Streams via Retroactive Grouping." *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, 2015: 1061-1070.
3. Rui Chen, Qian Xiao, Yu Zhang, Jianliang Xu. "Differentially Private High-Dimensional Data Publication via Sampling-Based Inference." *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015: 129-138.
4. Weiyen Day, Ninghui Li, Min Lyu. "Publishing Graph Degree Distribution with Node Differential Privacy." *Proceedings of the 2016 International Conference on Management of Data*, 2016: 123-138.
5. Rui Chen, Benjamin C. M. Fung, Li Xiong. "Publishing Set-Valued Data via Differential Privacy." *Proceedings of the 37th International Conference on Very Large Data Bases*, 2011: 1087.
6. Michael Hay, Ashwin Machanavajjhala, Gerome Miklau, Yan Chen, Dan Zhang. "Principled Evaluation of Differentially Private Algorithms using DPBench." *Proceedings of the 2016 International Conference on Management of Data*, 2016: 139-154.
7. Christos Dimitrakakis, Blaine Nelson, Zuhe Zhang, Aikaterini Mitrokotsa, Benjamin I. P. Rubinstein. "Differential Privacy for Bayesian Inference through Posterior Sampling." *Journal of Machine Learning Research* 18, no. 11 (2017): 1-39.
8. Marco Gaboardi, Emilio Jesú s Gallego Arias, Justin Hsu, Aaron Roth, Zhiwei Steven Wu. "Dual Query: Practical Private Query Release for High Dimensional Data." *Proceedings of the 31st International Conference on Machine Learning*, 2014: 1170-1178.

9. Duan, Yitao. "Differential Privacy for Sum Queries without External Noise." *Proceedings of the ACM Conference on Information and Knowledge Management*, 2009: 1517-1520.
10. Moritz Hardt, Katrina Ligett, Frank Mcsherry. "A Simple and Practical Algorithm for Differentially Private Data Release." *Advances in Neural Information Processing Systems 25*, 2012.
11. G. Acs, C. Castelluccia, R. Chen. "Differentially private histogram publishing through lossy compression." *Proceedings of ICDM*, 2012: 1-10.
12. Papadopoulos, G. Kellaris and S. "Practical differential privacy via grouping and smoothing." *Proceedings of VLDB Endow* 6, no. 5 (2013): 301-312.
13. J. Xu, Z. Zhang, X. Xiao, and G. Yu. "Differentially private histogram publicaiton." *Proceedings of ICDE*, 2102: 32-43.