Visualization of Differentially Private Histograms

Diane Tam, Dan Zhang, Gerome Miklau

College of Information and
Computer Sciences
UMass Amherst
dztam@umass.edu

Abstract

Prior research D. Zhang (2016) explores the challenges and proposed solution of visualizing differentially private data specifically in 2D. In such multi-dimensional data, it is easy to identify a visual artifact based on image perception. Here, we instead explore the visualization of differentially private data in 1D and whether or not "visual utility" holds for histograms.

Introduction

The datasets and algorithms used for the following experiments come from DPBench M. Hay (2016). Each experiment was run on two datasets, BIDS-ALL and HEPTH, with focus on the H_b, Identity, MWEM, and DAWA algorithms (the latter two being workload-aware). The workloads used are Identity and Prefix1D. The experiments were also tested on two different domains: 4096 and 1024. The general work flow for each experiment included first plotting the original histogram and cumulative density function of the data and then choosing a scale-epsilon pairing for the data generator with a domain of either 4096 or 1024 and then plotting the noisy histogram and cumulative density function outputted by the H_b and Identity algorithms, followed by the same plots for MWEM and DAWA under each of the two workloads. Thus, each experiment under a domain for a choice of scale-epsilon generates 6 noisy plots for visual analysis compared to the original.

For algorithms that produced negative outputs, the noisy data was cleaned up during post-processing via a normalized non-negative rounding by multiplying each positive bin count by a global weight of the [original noisy data sum]/[sum without negative values]. The negative bin counts are then rounded to 0. This method aims to maintain the overall distribution of the plot, while keeping the sum constant. However, as discussed later in the findings, while this works well in terms of visual utility for the histogram plot itself, there are some nuances in the cumulative distribution plot which are affected by this post-processing method.

Experiment Findings

Challenges in Standard Visual Analysis

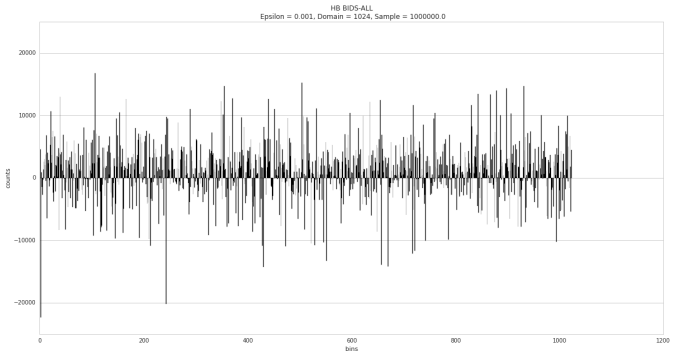

Figure 1: H_b on BIDS-ALL without post-processing

When comparing visual utility, there is no set guideline as to how each user should judge similarity of two plots. The easiest aspect for people to analyze is the distribution and general shape of a plot. There are clear artifacts of negative bin counts if the scale becomes too low, or epsilon is set too high. For example, (Fig. 1) depicts the plot generated from applying H_b to the BIDS-ALL dataset with an $\epsilon = 0.001$, domain = 1024, and sample = 10^6 In an application sense, the negative outputs are meaningless in a histogram of data counts that might be used for statistical analysis, and this is a visual artifact of the differential privacy algorithm.

The axis scales of a plot are another source of a less obvious visual artifact that arises from the noisy algorithm histogram output. In the case of the non-post-processed noisy H_b and Identity plots, or MWEM, the original axis y-scales do not fit the new noisy data scales. The H_b and Identity plots have many negative bin counts that cause the maximum y-axis to increase drastically as well in order to maintain a constant sum. MWEM often captures one high outlier bin count and underestimates lower bin counts for the rest of the data. Ideally, if the distribution of the noisy data was more accurate, the axis scales should remain consistent with the

original. This is the case with the normalized rounding post-processed noisy plots, but we cannot test this for MWEM because of the inherent workload-aware algorithm design.

Visual Artifacts from Plotting Libraries

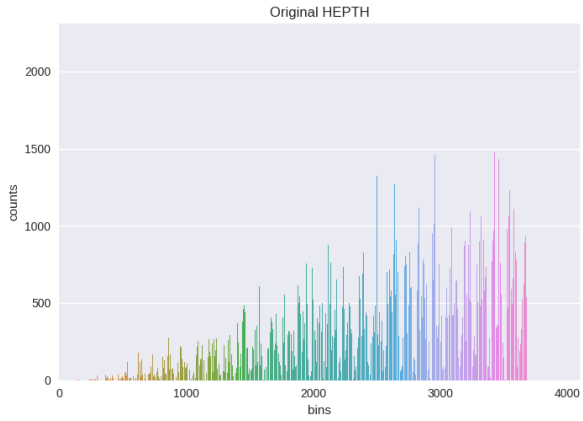

Figure 2: Original HEPTH (10^6), matplotlib2.0.0

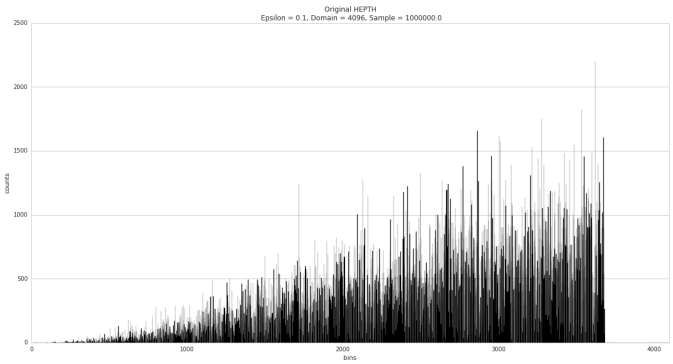

Figure 3: Original HEPTH (10^6), matplotlib1.4.3

When using two different versions of seaborn and matplotlib python libraries to plot these noisy histograms, some vast differences in the visual appearance of the original plots arose. The maximum bins seem to vary slightly between a plot of the original HEPTH data set generated with a domain = 4096 and sample = 10^6 (Fig. 2 and Fig. 3). Therefore, in order to accurately compare noisy histograms to their original counterpart, each plot must be compared to the original plot that was produced by the same version of matplotlib. For this paper, all graphs are plotted with matplotlib version 1.4.3 unless otherwise noted.

Scale Epsilon Visual Exchangeability

Declaring whether or not scale-epsilon exchangeability holds also becomes a very subjective question when referencing the visual utility of the data. Theoretically, changing the scale-epsilon parameter should present the same plot visually, scaled down or up by the correct magnitude. In an experiment comparing the pair (A) $\epsilon = 0.1$, sample = 10^4 and (B) $\epsilon = 0.001$, sample = 10^6 (note the two are scale-epsilon rank equivalent), we should be able to compare the plots generated by each algorithm under otherwise constant circumstances and find them to be visually exchangeable. However, this is not quite the case. It is arguable that in some cases, they are similar to a certain general degree, but not the the level of detail that would aid someone hoping to do statistical analysis with these plots.

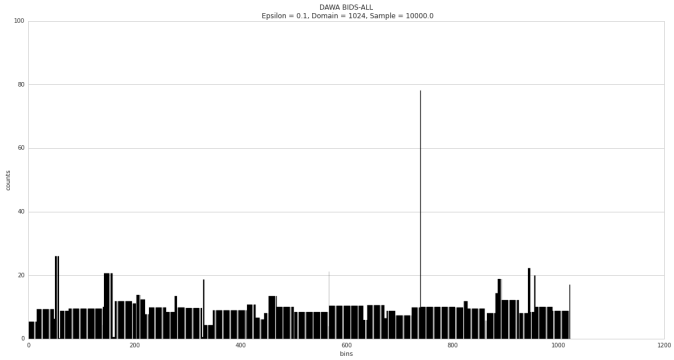

Figure 4: DAWA on BIDS-ALL (A)

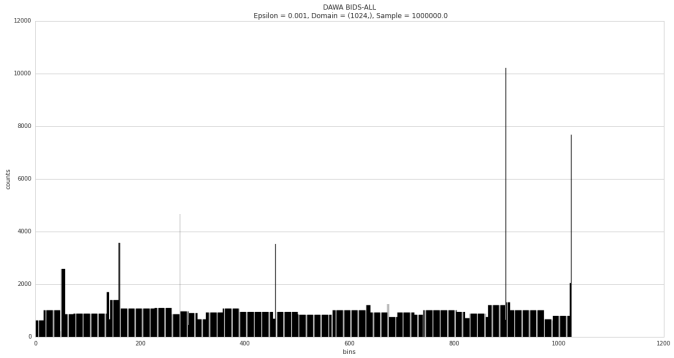

Figure 5: DAWA on BIDS-ALL (B)

For instance, the DAWA plots generated under the Identity workload for the BIDS-ALL dataset appear similar at first glance for overall distribution. However, the y-axis scale of the outliers becomes skewed from experiment (A) to (B). The highest outlier in (Fig. 4) approaches 80, and we would expect the highest outlier in (Fig. 5) to approach 8000, consequently. However, the highest outlier instead passes 10000 in experiment (B). There are also more outliers in experiment (B) that cause a more uniform distribution compared to the drops in experiment (A).

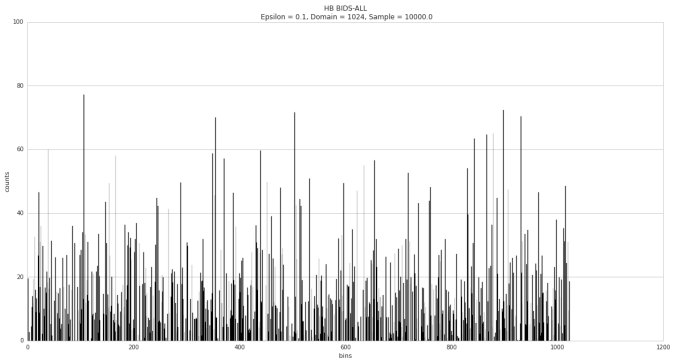

Figure 6: H_b on BIDS-ALL (A)

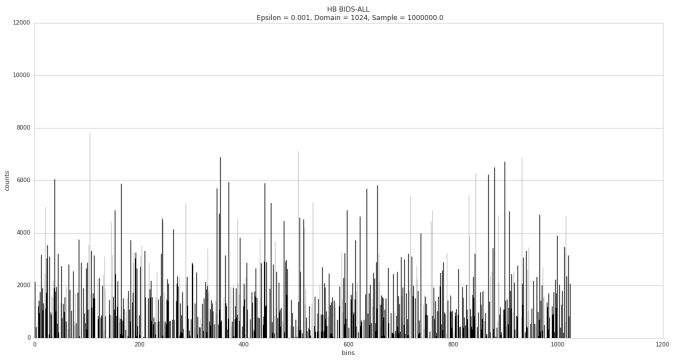

Figure 7: H_b on BIDS-ALL (B)

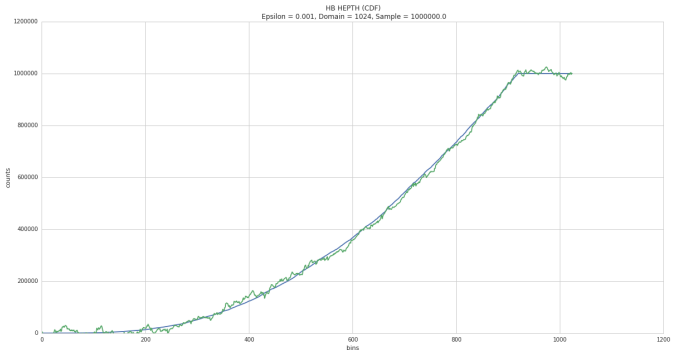

Figure 8: H_b on HEPTH (no post-processing)

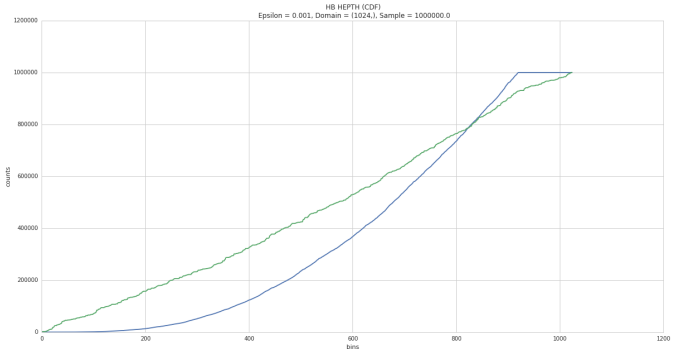

Figure 9: H_b on HEPTH (with post-processing)

For the same experiments (A) and (B), but instead looking at the H_b plots (post-processed to remove all negative counts), these issues seem to be harder to discern. The peaks of most bins lie between 60 and 80 (Fig. 6) for experiment (A), and as theorized, these lie between 6000 and 8000 for experiment (B) (Fig. 7). The general visual utility of the distribution also seems accurate to the naked eye. Again, this degree of accuracy might depend on the level of statistical analysis to be performed on the histogram.

Normalized Non-negative Rounding Visual Effects on CDF

Although the normalized non-negative rounding worked quite well in maintaining visual utility for the original histogram in negative-count generating plots such as H_b and Identity, the plot of the cumulative distribution function suffered visually from this post-processing routine. Take the CDF plot of a non-post-processed H_b noisy output run on the HEPTH dataset ($\epsilon = 0.001$ domain = 1024, sample = 10^6) and its post-processed counterpart run on the same parameters as an example. The green line represented the CDF of the noisy histogram, and the blue represents the CDF of the original histogram.

As expected, the noisy CDF still sums up to the original count of 10^6 samples. However, the non-post-processed CDF line (Fig 8) matches the original line more closely than that of the post-processed CDF line (Fig 9). The post-processed CDF line appears to rise much faster than the original until they intersect, where the CDF then slows down to meet the original. The post-processing gets rid of any negative bin counts, and as a result, allows the CDF to grow much faster than original. This could pose an issue if the statistical analysis of the noisy data were not focused on the distributions of the original data, but rather the growth rate or changes within the data. We would need a better algorithm to satisfy the consistency of both the distribution and deviations for differential privacy.

Conclusion

If the sample size is large, H_b, Identity, and DAWA perform well with respect to visual utility. Overall, the H_b and Identity algorithms were best at maintaining general visual utility after a round of normalized non-negative rounding post-processing of the noisy output independent of scale. As the scale dropped, DAWA was able to reproduce the outliers in the distribution, but unable to maintain visual utility. In general, MWEM failed to maintain visual utility in all cases, even if the sample size was large.

References

D. Zhang, M. Hay, G. M. B. O. (2016). Challenges of visualizing differentially private data.

M. Hay, A. Machanavajjhala, G. M. Y. C. D. Z. (2016). Principled evaluation of differentially private algorithms using dpbench.