Postprocessing for Iterative Differentially Private Algorithms

Jaewoo Lee JLEE@CSE.PSU.EDU
Penn State University, University Park, PA 16801

Daniel Kifer DKIFER@CSE.PSU.EDU
Penn State University, University Park, PA 16801

Abstract

Iterative algorithms for differential privacy run for a fixed number of iterations, where each iteration learns some information from data and produces an intermediate output. However, the algorithm only releases the output of the last iteration, and from which the accuracy of algorithm is judged. In this paper, we propose a post-processing algorithm that seeks to improve the accuracy by incorporating the knowledge on the data contained in intermediate outputs.

1. Introduction

When designing an iterative algorithm for differential privacy, fully utilizing the information privately learned from data is crucial to the success. Suppose we have a simple algorithm $\mathcal{A}$ that calls the function $\mathcal{K}(D, \theta_{t-1})$ T times in a loop and returns θ_T as the final output, where $D \in \mathcal{X}^N$ is the input dataset and $\theta_t = \mathcal{K}(D, \theta_{t-1})$ is an intermediate output at the t^{th} iteration. A large class of machine learning algorithms, including clustering, classification, and regression, can be written in this form, where $\mathcal{K}(D, \theta_t)$ minimizes some objective function and returns θ_{t+1}. By the composition theorem (Dwork & Roth, 2014), if the function $\mathcal{K}$ satisfies $\frac{\epsilon}{T}$-differential privacy, the algorithm $\mathcal{A}$ becomes ϵ-differentially private. At each iteration t, the algorithm extracts some information θ_t from the given dataset D using the privacy budget of ϵ/T, but it only releases the final output θ_T (thus, the accuracy of the algorithm is largely dependent on the magnitude of noise at the final iteration).

In this paper, we ask the following question: "*Can we improve the accuracy of the final output θ_T by incorporating the knowledge contained in the intermediate outputs $\theta_1, \ldots, \theta_{T-1}$?*" Recent studies have shown that post-processing algorithms that make inferences on the original data from noisy outputs can significantly improve the accuracy of the results (Lee et al., 2015; Hay et al., 2010; Lin & Kifer, 2013). The main source of improvements comes from enforcing *consistency constraints*, a set of (hard) syntactic conditions that hold true for the original data. Inspired by these post-processing algorithms, we view the intermediate outputs $\theta_1, \ldots, \theta_{T-1}$ as soft constraints on our estimates (i.e., the original data). Note that the privacy guarantee of $\mathcal{A}$ is not degraded by this post-processing. as long as it doesn't rely on the randomness of $\mathcal{A}$.

Consider a differentially private algorithm $\mathcal{A}$ that generates a sequence of noisy statistics $\{\tilde{\theta}_1, \ldots, \tilde{\theta}_T\}$ such that $\tilde{\theta}_t = \mathcal{K}(D, \tilde{\theta}_{t-1}) + Y$, where Y is a random variable representing the noise added for privacy. Our goal is to estimate a dataset $\widehat{D}$ from which $\tilde{\theta}_1, \ldots, \tilde{\theta}_T$ are most likely to be generated. Once we have estimated $\widehat{D}$, a new estimator $\hat{\theta}$ can be obtained by repeatedly running $\mathcal{K}$ on $\widehat{D}$ without noise (contrast this to $\tilde{\theta}_T$ produced with noise and using fixed number of iterations). Informally, we try to find $\widehat{D}$ such that $\mathcal{K}(D, \tilde{\theta}_t) \approx \mathcal{K}(\widehat{D}, \tilde{\theta}_t)$ for $t = 1, \ldots, T$. We note that the size of $\widehat{D}$ could be different from that of the original dataset, N; we only require intermediate outputs of $\mathcal{K}$ on both datasets are similar. However, it is still challenging to efficiently explore the space of all possible datasets. To this end, we propose to use MCMC method with carefully designed proposal distribution. The proposed algorithm builds a Markov chain over the space of all possible datasets and makes use of noisy statistics to efficiently propose the next state. Given a dataset D_t, the proposed algorithm samples a new dataset D' and determines whether to accept or reject the dataset by considering the ratio of $\mathbb{P}(\tilde{\theta}_1 \ldots, \tilde{\theta}_T \mid D')$ to $\mathbb{P}(\tilde{\theta}_1, \ldots, \tilde{\theta}_T \mid D_t)$, i.e., Metroplis-Hastings step. While doing so, it keeps track of the best scoring dataset.

In this paper, we instantiate this post-processing algorithm in the context of K-means clustering. The contributions of this paper are as follows:

- We propose a general framework for post-processing a sequence of noisy private outputs, which improves the

TPDP 2016 ICML Workshop, New York, NY, USA

Algorithm 1 DP-KMEANS algorithm

Input: data D, # of clusters K, # of iterations T
Initialize $\mathbf{c}_1^{(0)}, \cdots, \mathbf{c}_K^{(0)}$
for $t = 1$ **to** T **do**
  for $j = 1$ **to** K **do**
    $B_j^{(t)} = \left\{ \mathbf{x}_i : j = \arg\min_k \|\mathbf{x}_i - \mathbf{c}_k\|_2^2 \right\}$
    $\tilde{n}_j^{(t)} \leftarrow |B_j^{(t)}| + \mathsf{Lap}\left(\frac{2T}{\epsilon}\right)$
    $\tilde{\mathbf{s}}_j^{(t)} \leftarrow \left(\sum_{\mathbf{x}_i \in B_j} \mathbf{x}_i\right) + \mathsf{Lap}\left(\frac{2T}{\epsilon}\right)^d$
    $\mathbf{c}_j^{(t)} \leftarrow \tilde{\mathbf{s}}_j^{(t)} \Big/ \tilde{n}_j^{(t)}$

accuracy by incorporating intermediate results into the process.

- We applied our framework to K-means clustering problem and introduce an efficient proposal distribution that yields low rejection rate.
- Extensive empirical evaluations on both synthetic and real datasets are provided to validate our proposed approach.

2. Related Works

We discuss differentially private algorithms that can be applied to the K-means problem. The first algorithm is DP-KMEANS introduced in (Blum et al., 2005; McSherry, 2009). Each step of the algorithm is described in Algorithm 1. The algorithm is almost identical to its non-private counterpart, Lloyd's algorithm, with two differences. First, the algorithm takes a positive integer T as input and is only run for T iterations. This is to split the given privacy budget ϵ into each iteration. Second, the centroid update is done by using noisy sum and noisy count. The use of noisy statistics generated by the Laplace mechanism ensures that each update is differentially private.

It is easy to see that the sensitivity of $(n_1^{(t)}, \ldots, n_K^{(t)})$ is 1 as adding or removing one data point can change the size of one cluster by 1. Assuming $\mathcal{X}$ is the unit L_1-ball (i.e., $\|\mathbf{x}_i\|_1 \leq 1$), the sensitivity of $(\mathbf{s}_1^{(t)}, \ldots, \mathbf{s}_K^{(t)})$ is also 1. Therefore, together with the argument of composition theorem, adding $\mathsf{Lap}\left(\frac{2T}{\epsilon}\right)$ to sum and count ensures each iteration satisfies ϵ/T-differential privacy.

GUPT (Mohan et al., 2012) is a general-purpose system that implements the "sample and aggregate" framework (Nissim et al., 2007). Let f be a function on a database. In the context of this work, f is the K-means clustering algorithm, which takes a database as input and returns K centroids. Given a dataset D, GUPT first partitions D into ℓ disjoint blocks, say $T_1, \ldots, T_\ell$, and applies f on each block T_i. The final output of GUPT is computed by averaging the outputs $f(T_i)$ from each block and adding the Laplace noise to the average to ensure privacy.

PrivGene (Zhang et al., 2013) is a genetic algorithm based framework for differentially private model fitting. Starting from a set of randomly chosen solutions, it iteratively improves the quality of candidate solutions. To be specific, the algorithm starts with a candidate parameter set Ω, initialized with random vectors. At each iteration, Ω is enriched by adding offsprings (new candidate parameters), generated using crossover and mutate operations on existing parameters. Then, the algorithm selects and maintains a fixed number of parameters with best fitting scores using exponential mechanism.

Recently, Su et al. proposed EUGkM (Su et al., 2015), a non-interactive grid based algorithm for K-means clustering. The main idea is to divide multi-dimensional space into M rectangular grid cells. For each grid cell, it releases a pair (c_i, n_i) using the Laplace mechanism, where c_i and n_i are the center and the noisy count of data points in the cell, respectively. Note that noise is only added to the count n_i as releasing c_i has no privacy implication. Given a set of pairs $\mathcal{S} = \{(c_i, n_i) : i = 1, \ldots, M\}$, EUGkM considers there are n_i data points at c_i, and it applies (non-private) K-means algorithm on $\mathcal{S}$. They also proposed hybrid method which combines EUGkM with DP-KMEANS.

3. Postprocessing for K-means

In this section, we describe the proposed post-processing framework in the context of K-means where the algorithm releases a sequence of noisy cluster sums and sizes.

3.1. Inference on Centroids

Given the K initial centroids $\mathbf{c}_1^{(0)}, \ldots, \mathbf{c}_K^{(0)}$ (chosen independent of D), let $\mathcal{S} = \langle \tilde{\theta}_1, \ldots, \tilde{\theta}_T \rangle$ be the sequence of (noisy) outputs generated by running Algorithm 1 for T iterations, where $\tilde{\theta}_t = (\tilde{\mathbf{s}}_1^{(t)}, \ldots, \tilde{\mathbf{s}}_K^{(t)}, \tilde{n}_1^{(t)}, \ldots, \tilde{n}_K^{(t)})$ for $t = 1, \cdots, T$. Notice that (noisy) cluster centroids $\mathbf{c}_1^{(t)}, \ldots, \mathbf{c}_K^{(t)}$ are completely determined by $\tilde{\theta}_t$. We abuse notation and use $\tilde{\theta}_t$ to denote both noisy statistics and K centroids at iteration t. Let $S(D, \theta)$ and $N(D, \theta)$ be the functions that return the sum and the number of data points in each partition determined by the given centroids θ.

Our goal is to make an inference on the cluster centroids $\mathbf{c}_1, \ldots, \mathbf{c}_K$ based on the information $\mathcal{S}$ we learned privately from D. We do this by simulating datasets and evaluating the likelihood of the observed noisy statistics $\mathcal{S}$ under each dataset. Once a dataset that maximizes the likelihood of $\mathcal{S}$ is found, new estimates for the cluster centroids can be derived by running a non-private K-means algorithm (possibly with multiple random restarts) on the

dataset. The log-likelihood is defined by:

$$\begin{aligned}
&\ln \mathbb{P}[\mathcal{S} \mid D] = \ln \mathbb{P}[\tilde{\mathbf{s}}_1^{(1)}, \ldots, \tilde{\mathbf{s}}_K^{(T)}, \tilde{n}_1^{(1)}, \ldots, \tilde{n}_K^{(T)} \mid D] \\
&= \sum_{t=1}^{T} \Big(\ln \mathbb{P}\Big[\tilde{\mathbf{s}}_1^{(t)}, \ldots, \tilde{\mathbf{s}}_K^{(t)} \;\Big|\; S(D, \tilde{\theta}_{t-1})\Big] \\
&\qquad + \ln \mathbb{P}\Big[\tilde{n}_1^{(t)}, \ldots, \tilde{n}_K^{(t)} \;\Big|\; N(D, \tilde{\theta}_{t-1})\Big]\Big) \\
&\propto \sum_{t=1}^{T} \sum_{k=1}^{K} \Big\| \tilde{\mathbf{s}}_k^{(t)} - S_k(D, \tilde{\theta}_{t-1}) \Big\|_1 + \Big\| \tilde{n}_k^{(t)} - N_k(D, \tilde{\theta}_{t-1}) \Big\|_1 ,
\end{aligned}$$

where the subscript k in $S_k(D, \tilde{\theta}_{t-1})$ and $N_k(D, \tilde{\theta}_{t-1})$ represent the sum and number of data points in the k^{th} cluster, respectively.

3.2. Imposing Consistency

The accuracy of noisy output $\mathcal{S}$ can be improved by imposing consistency constraints, using the algorithm proposed in (Lee et al., 2015). Suppose $\hat{\mathbf{s}}_k^{(t)}$ and $\hat{n}_k^{(t)}$ are new estimates for $\tilde{\mathbf{s}}_k^{(t)}$ and $\tilde{n}_k^{(t)}$, respectively. It is clear that they should satisfy the following constraints:

$$\begin{aligned}
&\sum_{k=1}^{K} \hat{\mathbf{s}}_k^{(1)} = \sum_{k=1}^{K} \hat{\mathbf{s}}_k^{(2)} = \ldots = \sum_{k=1}^{K} \hat{\mathbf{s}}_k^{(T)} , \\
&\sum_{k=1}^{K} \hat{n}_k^{(1)} = \sum_{k=1}^{K} \hat{n}_k^{(2)} = \ldots = \sum_{k=1}^{K} \hat{n}_k^{(T)} , \text{ and} \\
&\hat{n}_k^{(t)} \geq 0 \text{ for all } k = 1, \ldots, K \text{ and } t = 1, \ldots, T .
\end{aligned}$$

For clear semantics and better readability, in the following we continue to use the notation $\tilde{\mathbf{s}}_k^{(t)}$ and $\tilde{n}_k^{(t)}$, but they represent the post-processed values.

3.3. Simulation via MCMC

The proposed algorithm makes use of approximate sampling method to find a dataset under which the likelihood of $\mathcal{S}$ is maximized. Using MCMC, it samples datasets from the approximate posterior distribution $\mathbb{P}(D \mid \mathcal{S})$ and evaluates the likelihood, while keeping track of the best solution. The target distribution is

$$\begin{aligned}
\pi(D) = \exp\Bigg(&- \sum_{t=1}^{T} \sum_{k=1}^{K} \epsilon_{\mathcal{S}} \|\tilde{\mathbf{s}}_k^{(t)} - S_k(D, \tilde{\theta}_{t-1})\|_1 \\
&+ \epsilon_{\mathcal{N}} \|\tilde{n}_k^{(t)} - N_k(D, \tilde{\theta}_{t-1})\|_1 \Bigg) ,
\end{aligned}$$

where $\epsilon_{\mathcal{S}}$ and $\epsilon_{\mathcal{N}}$ correspond to the privacy budgets for noisy cluster sums and sizes.[1]

Proposal distribution The Metropolis-Hastings (MH) algorithm requires choice of proposal distribution, and the convergence of Markov chain to its stationary distribution π is greatly dependent on that choice. The use of a proposal distribution that is far from π will have a high rejection rate and result in slow convergence.

It is shown that K-means algorithm can be thought as a special case of Gaussian Mixture Model (GMM), with means equal to centroids and a common covariance set to $\delta \mathbf{I}$ for small $\delta > 0$. Given $\theta_t = (\tilde{\mathbf{s}}_1^{(t)}, \ldots, \tilde{\mathbf{s}}_K^{(t)}, \tilde{n}_1^{(t)}, \ldots, \tilde{n}_K^{(t)})$, our proposal distribution is defined to be a mixture of Gaussians:

$$q(\mathbf{x}) = \sum_{k=1}^{K} \omega_k \mathcal{N}\left(\mathbf{x}; \mathbf{c}_k^{(t)}, \delta \mathbf{I}\right) , \tag{1}$$

where $\omega_k = \tilde{n}_k^{(t)} \Big/ \sum_{i=1}^{K} \tilde{n}_i^{(t)}$ and $\mathbf{c}_k^{(t)} = \tilde{\mathbf{s}}_k^{(t)} / \tilde{n}_k^{(t)}$.

Given the current dataset $D^{(\tau)}$, a new dataset D' is proposed by randomly choosing a data point $\mathbf{x}_i$ from D and replacing it with a new point $\mathbf{x}'$ sampled from the proposal distribution q. The sampling of $\mathbf{x}'$ is done as follows:

(i) choose an integer t randomly from $\{1, 2, \ldots, T\}$.
(ii) sample $z \mid t \sim \mathsf{Cat}\,(K, \omega_1, \ldots, \omega_K)$.
(iii) sample $\mathbf{x}' \mid z, t \sim \prod_{k=1}^{K} \mathcal{N}\left(\mathbf{c}_k^{(t)}, \delta \mathbf{I}\right)^{\mathbb{I}(z=k)}$.

In the above, $\mathsf{Cat}\,(K, \omega_1, \ldots, \omega_K)$ represents the Categorical distribution having possible values in $\{1, \ldots, K\}$, each with probability mass ω_k for $k = 1, \ldots, K$. $\mathbb{I}(z = k)$ is an indicator function whose value is 1 if $z = k$ and 0 otherwise.

MH Algorithm Given $D^{(\tau)}$, the proposed dataset D' of next state $(\tau + 1)$ is accepted with probability

$$A(D^{(\tau)}, D') = \min\left\{ \frac{\pi(\mathcal{S} \mid D')}{\pi(\mathcal{S} \mid D^{(\tau)})} \frac{q(\mathbf{x} \mid \mathbf{x}')}{q(\mathbf{x}' \mid \mathbf{x})}, 1 \right\} .$$

Without loss of generality, suppose a data point $\mathbf{x}$ is removed from the i^{th} cluster and a new data point $\mathbf{x}'$ is added to the j^{th} cluster at time τ. Then we have

$$\begin{aligned}
&\ln \pi(\mathcal{S} \mid D') - \ln \pi(\mathcal{S} \mid D^{(\tau)}) \\
&= -\sum_{t=1}^{T} \|\tilde{\mathbf{s}}_i^{(t)} - S_i(D', \tilde{\theta}_{t-1})\|_1 - \|\tilde{\mathbf{s}}_i^{(t)} - S_i(D^{(\tau)}, \tilde{\theta}_{t-1})\|_1 \\
&\quad + \|\tilde{\mathbf{s}}_j^{(t)} - S_j(D', \tilde{\theta}_{t-1})\|_1 - \|\tilde{\mathbf{s}}_j^{(t)} - S_j(D^{(\tau)}, \tilde{\theta}_{t-1})\|_1 \\
&\quad + |\tilde{n}_i^{(t)} - N_i(D', \tilde{\theta}_{t-1})| - |\tilde{n}_i^{(t)} - N_i(D^{(\tau)}, \tilde{\theta}_{t-1})| \\
&\quad + |\tilde{n}_j^{(t)} - N_j(D', \tilde{\theta}_{t-1})| - |\tilde{n}_j^{(t)} - N_j(D^{(\tau)}, \tilde{\theta}_{t-1})| .
\end{aligned}$$

We note that $S(D', \tilde{\theta}_{t-1})$ and $N(D', \tilde{\theta}_{t-1})$ can be calculated from $S(D^{(\tau)}, \tilde{\theta}_{t-1})$ and $N(D^{(\tau)}, \tilde{\theta}_{t-1})$, respectively. The MH correction term is given by

$$\frac{q(\mathbf{x})}{q(\mathbf{x}')} = \frac{q(\mathbf{x} \mid \mathbf{z}, t) q(\mathbf{z} = i) q(t)}{q(\mathbf{x}' \mid \mathbf{z}', t') q(\mathbf{z}' = j) q(t')} = \frac{\tilde{n}_i^{(t)} \mathcal{N}(\mathbf{x}; \mathbf{c}_i^{(t)}, \delta \mathbf{I})}{\tilde{n}_j^{(t')} \mathcal{N}(\mathbf{x}'; \mathbf{c}_j^{(t')}, \delta \mathbf{I})} .$$

[1] For simplicity, we assume $\epsilon_{\mathcal{S}} = \epsilon_{\mathcal{N}}$.

Table 1. Datasets

DATASETS	SIZE N	DIMENSION d	K
S1	5,000	2	15
TIGER	16,281	2	2
GOWALLA	107,021	2	5
IMAGE	34,112	3	3
ADULT (NUMERIC)	48,842	6	5
LIFESCI	27,733	10	3

The initial state $D^{(0)}$ is initialized with K centroids at the last iteration. It consists of $\tilde{n}_k^{(T)}$ data points at $\mathbf{c}_k^{(T)}$ for $k = 1, \ldots, K$.

4. Experiments

In this section, the performance of the proposed post-processing algorithm is evaluated over both synthetic and real datasets. We note that our goal is not to develop a better private algorithm for K-means; rather, we seek to improve the accuracy of iterative differentially private algorithms in general by taking intermediate results into account. Given K partitions $\beta_1, \ldots, \beta_K$ and their centroids $\mathbf{c}_1, \ldots, \mathbf{c}_K$, the quality of clustering is measured by the sum of squared distance between data points and their nearest centroids, within cluster sum of squares (WCSS); it is the objective function of K-means problem.

$$WCSS = \sum_{k=1}^{K} \sum_{\mathbf{x} \in \beta_k} \|\mathbf{x} - \mathbf{c}_k\|_2^2$$

For the experiments, we used 6 external datasets. For all datasets, the domain of each attribute is normalized to $[-1, 1]$ and then projected onto L_1-ball. The characteristics of datasets used in our experiments are summarized in Table 1. For each dataset, we run DP-KMEANS (DPKM) and the proposed method (MCMC) 10 times and report the averaged WCSS. The performance of K-means algorithm is largely dependent on the choice of initial centroids, it is important to carefully select them. As in (Su et al., 2015), K initial centroids are chosen independent of data such that pairwise distance between centroids are greater than some given constant.

Throughout the experiments, the number of iterations T for DPKM is fixed to 5. For the proposed algorithm, the length of Markov chain is fixed to 30,000 and the value of δ, the variance of Gaussian component in the proposal distribution, is set to 0.001.

Figure 1 shows the performance of our post-processing algorithm for different values of ϵ. The value of ϵ ranges from 0.05 to 1.0. The proposed algorithm improves the accuracy of the final clusterings on all datasets, except on the Lifesci dataset. On S1 and Tiger datasets, huge improvements in WCSS were observed when $\epsilon = 0.05$.

Acknowledgement

This research was supported by NSF grant 1228669.

References

Blum, Avrim, Dwork, Cynthia, McSherry, Frank, and Nissim, Kobbi. Practical privacy: The sulq framework. In *PODS*, 2005.

Dwork, Cynthia and Roth, Aaron. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4), 2014.

Hay, Michael, Rastogi, Vibhor, Miklau, Gerome, and Suciu, Dan. Boosting the accuracy of differentially private histograms through consistency. *Proc. VLDB Endow.*, 3 (1-2):1021–1032, September 2010. ISSN 2150-8097.

Lee, Jaewoo, Wang, Yue, and Kifer, Daniel. Maximum likelihood postprocessing for differential privacy under consistency constraints. In *KDD*, 2015.

Lin, Bing-Rong and Kifer, Daniel. Information preservation in statistical privacy and bayesian estimation of unattributed histograms. In *SIGMOD*, 2013.

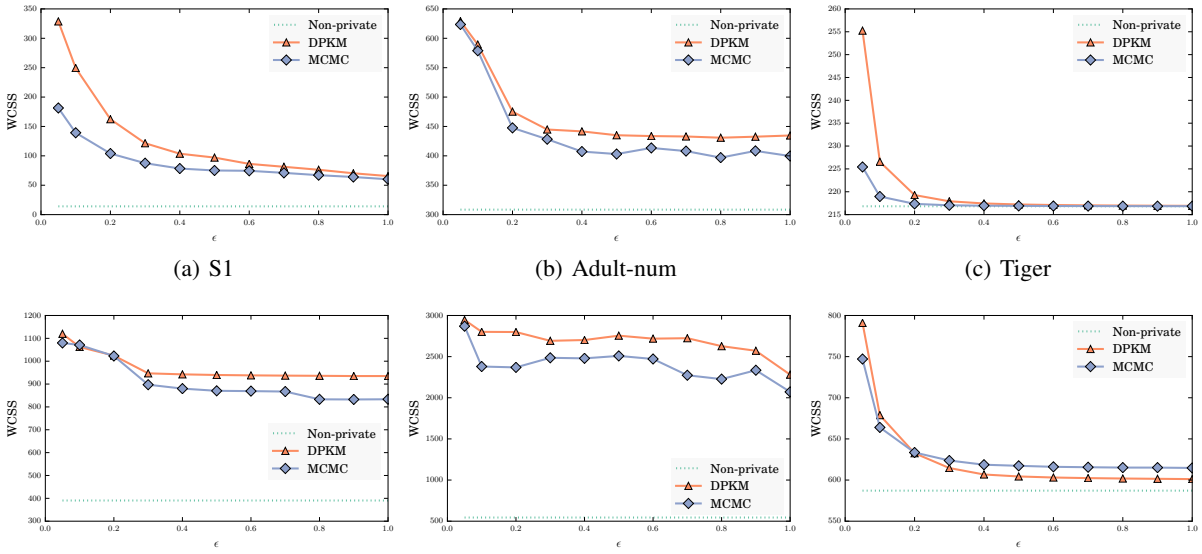
McSherry, Frank D. Privacy integrated queries: An extensible platform for privacy-preserving data analysis. In *SIGMOD*, 2009.

Mohan, Prashanth, Thakurta, Abhradeep, Shi, Elaine, Song, Dawn, and Culler, David. Gupt: Privacy preserving data analysis made easy. In *SIGMOD*, 2012.

Nissim, Kobbi, Raskhodnikova, Sofya, and Smith, Adam. Smooth sensitivity and sampling in private data analysis. In *STOC*, 2007.

Su, D., Cao, J., Li, N., Bertino, E., and Jin, H. Differentially Private k-Means Clustering. *ArXiv e-prints*, April 2015.

Zhang, Jun, Xiao, Xiaokui, Yang, Yin, Zhang, Zhenjie, and Winslett, Marianne. Privgene: Differentially private model fitting using genetic algorithms. In *SIGMOD*, 2013.



(a) S1 (b) Adult-num (c) Tiger

(d) Image (e) Gowalla (f) Lifesci

Figure 1. WCSS by varying ϵ