Leveraging Public Data in Practical Private Query Release: A Case Study with ACS Data

Terrance Liu,[1] **Giuseppe Vietri,**[2] **Thomas Steinke,**[3] **Jonathan Ullman,**[4] **Zhiwei Steven Wu,**[5]

[1,5] Carnegie Mellon University, [2] University of Minnesota, [3] Google Brain, [4] Northeastern University

terrancl@cs.cmu.edu, vietr002@umn.edu, web@thomas-steinke.net, jullman@ccs.neu.edu, zstevenwu@cmu.edu

Abstract

It has been shown that for differentially private query release, the MWEM algorithm due to Hardt, Ligett, and McSherry (2012) achieves nearly optimal statistical guarantees. However, running MWEM on high-dimensional data is often infeasible, making the algorithm only applicable to low-dimensional data. In this paper, we study the setting in which the data curator has access to public data that is drawn from a similar—but related—distribution. Specifically, we present MW-Pub, which adapts MWEM to leverage prior knowledge from public samples and scale to high-dimensional data. Empirical evaluation on the American Community Survey (ACS) and the ADULT dataset shows that our method outperforms state-of-the-art methods under high privacy regimes.

1 Introduction

Access to individual-level data has become crucial to many decision making processes as they grow increasingly more data-driven. However, as the collection and distribution of private information becomes more prevalent, controlling for privacy has become a priority for organizations releasing statistics about different populations. Today, differential privacy (Dwork 2006) is the standard by which researchers measure the tradeoff between releasing useful information and protecting privacy, serving as the basis for many applications of privacy protection, including the 2020 U.S. Census release (Abowd 2018).

In this paper, we study statistical query release, an application used by many organizations, such as government agencies and medical institutions, and one of the fundamental problems in privacy research. One notable framework for private query release is to directly release private synthetic data, a sanitized version of the private dataset that answers queries under some privacy guarantees. Private multiplicative weights (Hardt and Rothblum 2010) and MWEM (Hardt, Ligett, and McSherry 2012) are two notable examples of synthetic data algorithms, with the latter having been shown to provide nearly optimal guarantees. However, running MWEM requires maintaining a distribution over the domain of the data universe, which often becomes intractable for real-world problems and has prompted development of new algorithms that avoid this issue while following similar no-regret learning dynamics (Gaboardi et al. 2014; Vietri et al. 2020). In a similar vein, our proposed method, MW-Pub, adapts MWEM to make use of public data, which we define as any samples that pose no privacy concerns. In doing so, MW-Pub not only scales better to higher-dimensional datasets but also achieves lower error by leveraging prior information from the auxiliary public dataset.

Related Work. To motivate the setting of assisting privacy mechanisms with public data, we note that sources of public data are often readily available, such as in the case where individuals voluntarily offer or sell their data. Following this observation, many works have also studied utilizing public data for differential privacy. Avent et al. (2017) for example propose a hybrid search model that combines sensitive private data with public data. Bassily et al. (2020) prove upper and lower bounds for private and public sample complexities in the context of private query release. Similarly, Alon, Bassily, and Moran (2019) prove private and public sample complexities for *semi-private learning* (Beimel, Nissim, and Stemmer 2013), a relaxed notion of differentially private supervised learning in which the training set can be divided into private and public samples. Bassily, Moran, and Nandi (2020) extend this line of research, studying PAC learnability while relaxing the assumption that public and private samples come from the same distribution.

2 Preliminaries

In order to present our problem statement, we begin by defining the following:

Definition 2.1 (Statistical linear query)**.** Given as predicate a linear threshold function ϕ and a dataset D, the linear query $q_\phi : \mathcal{X}^n \to [0, 1]$ is defined by

$$q_\phi(D) = \frac{1}{|D|} \sum_{x \in D} \phi(x)$$

Defining a dataset instead as a distribution A over the domain $\mathcal{X}$, the definition for a linear query q_ϕ then becomes

$$q_\phi(A) = \sum_{x \in \mathcal{X}} q(x)A(x)$$

Definition 2.2 (k-way marginal query)**.** Let the data universe with d categorical attributes be $\mathcal{X} = (\mathcal{X}_1 \times \ldots \times \mathcal{X}_d)$, where each $\mathcal{X}_i$ is the discrete domain of the ith attribute. A k-way marginal query is a linear query specified by attributes $M = \{(a_i)_{i\in[k]} \mid a_1 \neq \ldots \neq a_k \in [d]\}$ and target $y \in (\mathcal{X}_1 \times \ldots \times \mathcal{X}_k)$, given by

$$q_{M,y}(x) = \begin{cases} 1 & : x_{a_1} = y_1 \wedge \ldots \wedge x_{a_k} = y_k \\ 0 & : \text{otherwise} \end{cases}$$

where $x_i \in \mathcal{X}_i$ means the ith attribute of record $x \in \mathcal{X}$. Each marginal has a total of $\prod_{i=1}^{k} |\mathcal{X}_{a_k}|$ queries, and we define a *workload* as a set of marginal queries.

Definition 2.3 (Differential Privacy (Dwork 2006))**.** A randomized algorithm $\mathcal{M} : \mathcal{X}^* \to \mathcal{R}$ satisfies (ε, δ)-differential privacy (DP) if for all databases x, x' differing at most one entry and every measurable subset $\mathcal{S} \subseteq \mathcal{R}$, we have that

$$\Pr[\mathcal{M}(x) \in S] \leq e^{\varepsilon} \Pr\left[\mathcal{M}\left(x'\right) \in S\right] + \delta$$

If $\delta = 0$, we say that $\mathcal{M}$ satisfies ε-differential privacy.

Definition 2.4 (Concentrated Differential Privacy (Bun and Steinke 2016))**.** A randomized mechanism $M : \mathcal{X}^n \to \mathcal{R}$ is ρ-zCDP, if for all neighboring datasets D, D', differing on a single element and for all $\alpha \in (1, \infty)$,

$$\mathrm{D}_\alpha(\mathcal{M}(D) \parallel \mathcal{M}(D')) \leq \rho\alpha$$

where $\mathrm{D}_\alpha((D) \parallel (D'))$ is the Rényi divergence between the distribution of (D) and the distribution of (D').

Problem Statement. We consider a data domain $\mathcal{X} = \{0,1\}^d$ of dimension d and a private dataset $\widetilde{D} \in \mathcal{X}^n$ consisting of the data of n individuals. Our goal is to approximately answer a large class of statistical queries $\mathcal{Q}$ about $\widetilde{D}$. An approximate answer $a \in [0,1]$ to some query $q_\phi \in \mathcal{Q}$ must satisfy $|a - q_\phi(\widetilde{D})| \leq \alpha$ for some accuracy parameter $\alpha > 0$.

In this work, we assume access to a public dataset $\widehat{D} \in \mathcal{X}^m$ with m individuals. This dataset defines a public data domain, denoted by $\widehat{\mathcal{X}} \subset \mathcal{X}$, which consists of all unique rows in $\widehat{D}$. Note that one can think of the dataset $\widehat{D}$ as a distribution over the domain $\widehat{\mathcal{X}}$.

3 Public Data Assisted MWEM

In this section, we revisit the MWEM algorithm and then introduce MW-Pub, which adapts MWEM to take advantage of public data.

3.1 MWEM

MWEM (Hardt, Ligett, and McSherry 2012) is an approach to answering linear queries that combines the multiplicative weights update rule (Hardt and Rothblum 2010) and the exponential mechanism (McSherry and Talwar 2007). MWEM maintains an approximation of the distribution over the data domain $\mathcal{X}$. At each iteration, the algorithm selects the worst approximate query $q_t(\widetilde{D})$ using the exponential mechanism and measures the query with Laplace noise (Dwork et al. 2006). MWEM then improves the approximating distribution using the multiplicative weights update rule. This algorithm can thus be viewed as a two-player game in which a data player updates its distribution A_t using a no-regret online learning algorithm and a query player responds using the exponential mechanism.

Although Hardt, Ligett, and McSherry (2012) show that MWEM achieves nearly optimal theoretical guarantees and performs well empirically across a variety of query classes and datasets, applying MWEM in real-world instances can often be impractical. Maintaining a distribution A over a data domain $\mathcal{X} = \{0,1\}^d$ becomes infeasible when d is large, suffering running time that is exponential in d. Hardt, Ligett, and McSherry (2012) introduce a scalable implementation of MWEM that avoids explicitly tracking A when the query class involves disjoint subsets of attributes. However, while MWEM has running time proportional to $|\mathcal{X}|$ in this special case, it is applicable only to simpler workloads.

3.2 MWEM+PUB

We introduce MW-Pub in Algorithm 1, which adapts MWEM to use a public dataset through the following changes:

The approximating distribution A_t is maintained over the public data domain $\widehat{\mathcal{X}}$ rather than $\mathcal{X}$. Because $|\widehat{\mathcal{X}}|$ is often significantly smaller than $|\mathcal{X}|$, MW-Pub offers substantial improvements in both its running time and memory footprint, allowing it to scale to much more complex query release problems.

A_0 is initialized to the distribution over $\widehat{\mathcal{X}}$ given by $\widehat{D}$. In the standard formulation of MWEM, A_0 is initialized to be a uniform distribution over $\mathcal{X}$. However, Hardt, Ligett, and McSherry (2012) note that in certain cases, it can be beneficial to instead initialize A_0 by performing a noisy count over all rows $x \in \mathcal{X}$. Drawing inspiration from this variation, we instead initialize A_0 to match the distribution of $\widehat{D}$ under the assumption that A_0 provides a better approximation of the distribution of $\widetilde{D}$ than a uniform distribution over $\widehat{\mathcal{X}}$.

In addition, we make the following modifications to both MWEM and MW-Pub:

Permute-and-flip Mechanism. We replace the *exponential mechanism* with the *permute-and-flip mechanism* (McKenna and Sheldon 2020), which like the *exponential mechanism* runs in linear time but whose expected error is never higher.

Gaussian Mechanism. When taking measurements of sampled queries, we add Gaussian noise instead of Laplace noise. The Gaussian distribution has lighter tails, and in settings with a high degree of composition, the scale of Gaussian noise required to achieve some fixed privacy guarantee is lower (Canonne, Kamath, and Steinke 2020). Privacy guarantees for the *Gaussian mechanism* can be cleanly expressed in terms of concentrated differential privacy and the composition theorem given by Bun and Steinke (2016).

Algorithm 1: MW-Pub

Input: Private dataset $\widetilde{D} \in \mathcal{X}^n$, public dataset $\widehat{D} \in \mathcal{X}^m$, query class $\mathcal{Q}$, privacy parameter $\tilde{\varepsilon}$, number of iterations T.

Let the domain be $\widehat{\mathcal{X}} = \text{supp}(\widehat{D})$.

Let size of the private dataset be $n = |\widetilde{D}|$.

Let A_0 be the distribution over $\widehat{\mathcal{X}}$ given by $\widehat{D}$

Initialize $\varepsilon_0 = \frac{\tilde{\varepsilon}}{\sqrt{2T}}$.

for $t = 1$ **to** T **do**

Sample query $q_t \in \mathcal{Q}$ using the *permute-and-flip mechanism* – i.e.,

$$\Pr[q_t] \propto \exp\left(\frac{\varepsilon_0 n}{2}|q(A_{t-1}) - q(\widetilde{D})|\right)$$

Measure: Let $a_t = q_t(\widetilde{D}) + \mathcal{N}\left(0, 1/n^2\varepsilon_0^2\right)$. (But, if $a_t < 0$, set $a_t = 0$; if $a_t > 1$, set $a_t = 1$.)

Update: Let A_t be a distribution over $\widehat{\mathcal{X}}$ s.t.

$$A_t(x) \propto A_{t-1}(x)\exp\left(q_t(x)\left(a_t - q_t(A_{t-1})\right)/2\right).$$

end for

Output: $A = \text{avg}_{t \le T} A_t$

3.3 Privacy Analysis.

When run with privacy parameter $\tilde{\varepsilon}$, MW-Pub satisfies $\frac{1}{2}\tilde{\varepsilon}^2$-concentrated differential privacy and, for all $\delta > 0$, it satisfies$(\varepsilon(\delta), \delta)$-differential privacy, where $\varepsilon(\delta) = \inf_{\alpha>1} \frac{1}{2}\tilde{\varepsilon}^2\alpha + \frac{\log(1/\alpha\delta)}{\alpha-1} + \log(1 - 1/\alpha) \le \frac{1}{2}\tilde{\varepsilon}^2 + \sqrt{2\log(1/\delta)} \cdot \tilde{\varepsilon}$.

The privacy analysis follows from four facts: (i) Permute-and-flip satisfies ε_0-differential privacy (McKenna and Sheldon 2020), which implies $\frac{1}{2}\varepsilon_0^2$-concentrated differential privacy. (ii) The Gaussian noise addition also satisfies $\frac{1}{2}\varepsilon_0^2$-concentrated differential privacy. (iii) The composition property of concentrated differential privacy allows us to add up these $2T$ parameters (Bun and Steinke 2016). (iv) Finally, we can convert the concentrated differential privacy guarantee into the usual approximate differential privacy (Canonne, Kamath, and Steinke 2020).

4 Experimental Setting

We describe the datasets and benchmarks used to evaluate MW-Pub in our experiments.

4.1 Data

American Community Survey (ACS) We evaluate all algorithms on the 2018 American Community Survey (ACS) 1-year estimates, obtained from the IPUMS USA database (Ruggles et al. 2020). Collected every year by the US Census Bureau, the ACS provides statistics meant to capture the social and economic conditions of households across the United States. Given that the Census Bureau is incorporating differential privacy into 2020 Census release (Abowd 2018) and has plans to incorporate it into the ACS itself after 2025, we believe that the ACS dataset is a natural testbed for privately answering statistical queries in a real-world setting.

For our private dataset $\widetilde{D}$, we use the 2018 ACS for the state of Pennsylvania (PA-18). In addition, we use the 2014 ACS as a validation set for selecting hyperparameters. To select our public dataset $\widehat{D}$, we explore the following:

Selecting across time. We consider the setting in which there exists a public dataset describing our population at a different point in time. Given that privacy laws and practices are still expanding, it is often feasible to identify datasets that were released publicly in the past. Using the 2020 U.S. Census release as an example, one could consider using the 2010 U.S. Census as a public dataset for some differentially private mechanism. For our experiments, we use the 2010 ACS data for Pennsylvania (PA-10) when evaluating on both the validation (PA-14) and test (PA-18) sets.

Selecting across states. Next, we consider the setting in which there exists a public dataset collected concurrently from a different population. In the context of releasing state-level statistics, one can imagine for example that some states have differing privacy laws. In this case, we can identify some dataset for a similar state that has been publicly released. For our experiments, we use 2018 ACS data for Ohio (OH-18), Illinois (IL-18), New York (NY-18), and New Jersey (NJ-18) to evaluate performance on PA-18. We use the same states' data from 2014 (i.e. OH-14, IL-14, NY-14, NJ-14) to evaluate on the validation set.

ADULT We evaluate algorithms on the ADULT dataset from the UCI machine learning dataset repository (Dua and Graff 2017). We construct private and public datasets by sampling with replacement rows from ADULT of size $0.9N$ and $0.1N$ respectively (where N is the number of rows in ADULT). Thus, we frame rows in the ADULT dataset as individuals from some population in which there exists both a public and private dataset trying to characterize it (with the former being significantly smaller).

4.2 Benchmarks

In addition to MWEM, we evaluate MW-Pub against the following:

DUALQUERY. Similar to MWEM, DualQuery (Gaboardi et al. 2014) frames query release as a two-player game by reversing the roles of the data and query players. In DualQuery, the query player runs multiplicative weights to update its distribution over queries while the data player outputs a data record as its best response. At each round, the algorithm preserves privacy guarantees by drawing samples from the query distribution using the exponential mechanism. Gaboardi et al. (2014) prove theoretical accuracy bounds for DualQuery that are worse than that of MWEM and show that on low-dimensional datasets where running MWEM is feasible, MWEM outperforms DualQuery. However, DualQuery solves an optimization problem whose space and running time are linear in the number of queries being answered, and given that the number of queries is often significantly smaller than the size of the data universe for high-dimensional datasets,

DualQuery has the advantage of being scalable to a wider range of query release problems.

HDMM. Unlike MWEM and DualQuery, which solve the query release problem by generating synthetic data, the High-Dimensional Matrix Mechanism (McKenna et al. 2018) is designed to directly answer a workload of queries. By representing query workloads compactly, HDMM selects a new set of "strategy" queries that minimize the estimated error with respect to the input workload. The algorithm then answers the "strategy" queries using the Laplace mechanism and reconstructs the answers to the input workload queries using these noisy measurements, solving a ordinary least squares problem to resolve any inconsistencies. With the U.S. Census Bureau deploying HDMM (Kifer 2019), the algorithm offers a particularly suitable baseline for privately answering statistical queries on the ACS dataset.

4.3 Additional Optimizations

Following a remark made by Hardt, Ligett, and McSherry (2012) for optimizing the empirical performance of MWEM, we apply the multiplicative weights update rule using sampled queries q_i and measurements a_i from previous iterations i. However, rather than use all past measurements, we choose queries with estimated error above some threshold. Specifically at each iteration t, we calculate the term $c_i = |q_i(A_t) - a_i|$ for $i \leq t$. In random order, we apply multiplicative weights for all queries and measurements, indexed by i, such that $c_i \geq \frac{c_t}{2}$, i.e. queries whose noisy error estimates are relatively high.

5 Results

In this section, we present our results on the ACS and ADULT datasets, comparing MW-Pub to the benchmark algorithms. Across all experiments, we report the maximum error on a set of statistical queries. Our experiments entail answering a random set of 3 or 5-way marginal queries with varying workload sizes ranging from 512 to 4096. We test performance on privacy budgets $\varepsilon(\delta) \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ and $\delta = \frac{1}{N^2}$, where N is the size of the private dataset.

In all figures, we plot the average of 5 runs at each privacy budget ε and use error bars to represent one standard error. For MW-Pub, we select hyperparameters using the validation set (PA-14). For MWEM and DualQuery, we simply report the best performing 5-run average across all hyperparameter choices. Running HDMM does not require hyperparameter selection. For a complete list of hyperparameters, refer to Appendix 7.2

5.1 ACS (Pennsylvania)

We compare the performance of MW-Pub using the public datasets described in section 4.1. As seen in Figure 2, the ACS data for New Jersey is a poor candidate for a public dataset, despite being a bordering state of Pennsylvania. The maximum error of using the NJ ACS dataset to directly answer queries ($\varepsilon = 0$) is quite high. Moreover, the performance of MW-Pub does not improve, indicating that

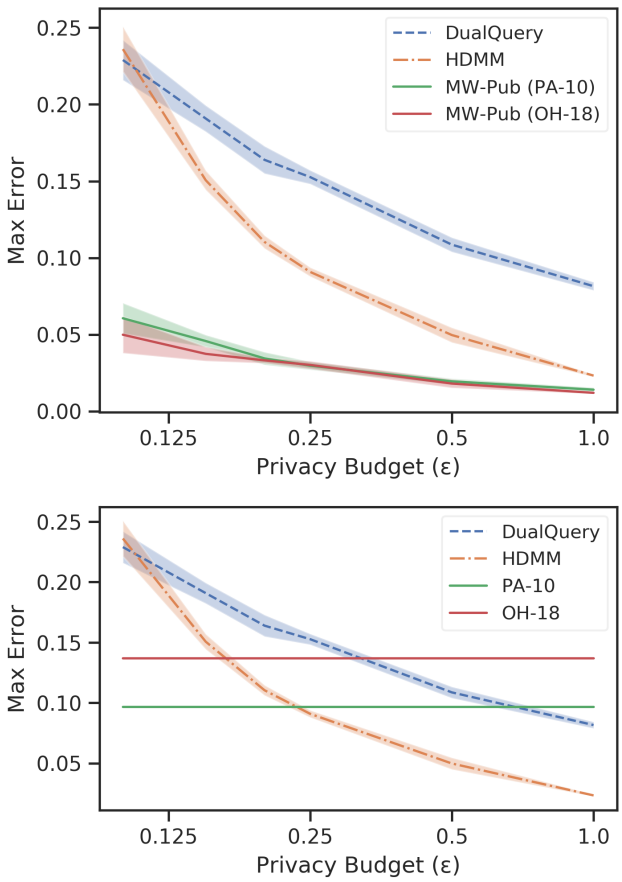
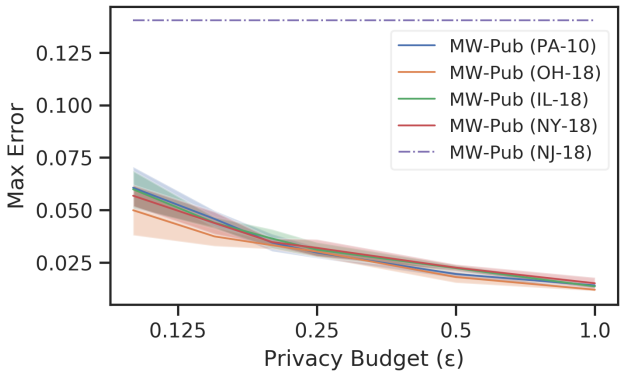


Figure 1: **[2018 ACS-PA]** Max error on 3-way marginals across privacy budgets $\varepsilon \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ where $\delta = \frac{1}{N^2}$ and the workload size is 4096. **Top:** We compare MW-Pub to the benchmark algorithms. **Bottom:** We evaluate the max error when using the public datasets to answer queries directly ($\varepsilon = 0$).

the support $\widehat{\mathcal{X}}$ is insufficient for improving the approximating distribution A_t any further. On the other hand, we observe that when using our other choices for public datasets, MW-Pub performs much better, with the algorithm converging to approximately the same error as ε approaches 1.0.

Next, we compare MW-Pub using the best performing public datasets selected across time (PA-10) and across states (OH-18) to the benchmark algorithms described in section 4.2. We present the following observations:

MW-Pub outperforms all benchmark algorithms. In the high privacy regime in which ε is small, the benchmark algorithms have high maximum errors. For example, Figure 1 shows that for $\varepsilon < 0.25$, it is better to directly answer queries using the 2010 ACS data for Pennsylvania, rather than use DualQuery or HDMM. Running MW-Pub improves upon the initial error of using the public datasets, outperforming the benchmark algorithms across all privacy budgets that we evaluated on.



Public Dataset	Initial Error
PA (2010)	0.0967
OH (2018)	0.1370
IL (2018)	0.1669
NY (2018)	0.3002
NJ (2018)	0.3389

Figure 2: **[2018 ACS-PA]** Max error on 3-way marginals at a workload size of 4096. **Top:** Comparison of max errors for $\varepsilon \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ and $\delta = \frac{1}{N^2}$ using different public datasets. Note that MW-Pub performs similarly across all the public datasets except NJ-18 (dashed line). **Bottom:** Table comparing the initial error of each public dataset ($\varepsilon = 0$).

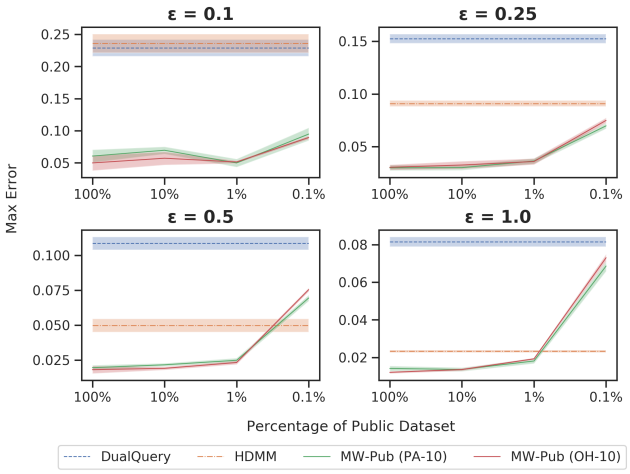


Figure 3: **[2018 ACS-PA]** Max error on 3-way marginals while varying the fraction of the public dataset used, where $\varepsilon \in \{0.1, 0.25, 0.5, 1\}$, $\delta = \frac{1}{N^2}$, and workload size is 4096.

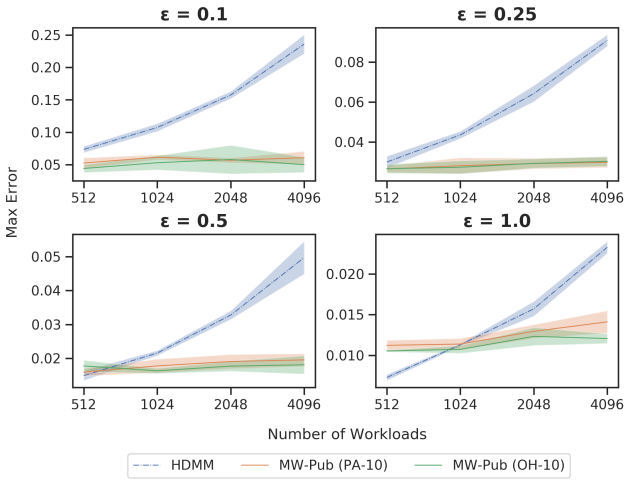


Figure 4: **[2018 ACS-PA]** Comparison of MW-Pub against HDMM on 3-way marginals while varying the workload size. ($\delta = \frac{1}{N^2}$).

MW-Pub **performs well even when the public dataset is reduced to** 1% **of its original size.** In Figure 3, we plot the performance of MW-Pub using different public dataset sizes. Reducing the 2010 ACS-PA and 2018 ACS-OH datasets to only 1% of their original sizes yields no significant performance loss, with MW-Pub still outperforming all benchmarks. However, further decreasing the public dataset size dramatically increases the error. We attribute this increase to (1) $\widehat{\mathcal{X}}$ provides an insufficient support and (2) reducing $\widehat{D}$ induces too much sampling error to make our initialization for A_0 strong enough to outperform HDMM.

Compared to HDMM, MW-Pub **scales well with respect to workload size.** We compare the performance of MW-Pub and HDMM, our strongest performing benchmark, across different workload sizes. Figure 4 shows that although the maximum error of HDMM grows significantly as we increase the number of 3-way marginal queries, the maximum error of MW-Pub remains relatively stable. Our experiments suggest that, MW-Pub may be a more suitable algorithm when the goal is release large workloads of queries.

5.2 ADULT

We compare MW-Pub against the benchmark algorithms on ADULT in which we construct public and private partitions from the original dataset. We evaluate on 3-way marginal queries with the maximum workload size of 286. Because both the public and private partitions come from the same distribution, the public partition itself already approximates the distribution of the private partition well. Consequently, we conduct additional experiments by sampling from ADULT according to the attribute *sex* with some bias. Specifically, we sample females with probability $r + \Delta$ where $r \approx 0.33$ is the proportion of females in the ADULT dataset. We observe in Figure 5 that an unbiased sample for the public partition ($\Delta = 0$) achieves very low error across all privacy budgets. In addition, even when we take biased samples where $\Delta \in \{-0.05, -0.1\}$, MW-Pub performs well. However, in the case where the public dataset is extremely biased and is comprised almost entirely of males ($\Delta = -0.3$), the performance of MW-Pub deteriorates, with HDMM outperforming it at $\varepsilon = 1$.

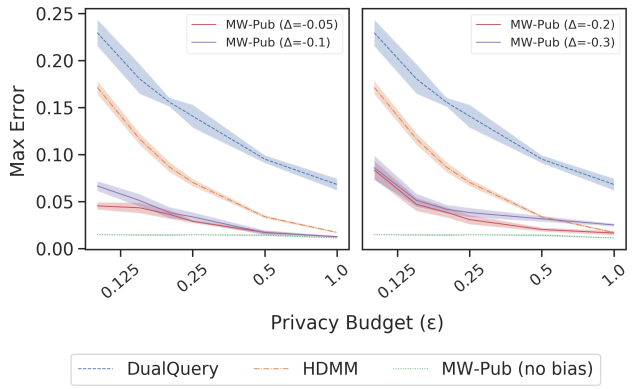


Figure 5: **[ADULT]** Max error on 3-way marginals across privacy budgets $\varepsilon \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ where $\delta = \frac{1}{N^2}$ and the workload size is 286 (maximum). Each public dataset is constructed by sampling from a public partition with some bias Δ over the attribute *sex* (labeled as MW-Pub (Δ)), i.e. rows with the attribute *sex*='Female' are sampled with probability $r + \Delta$ where $r \approx 0.33$ is the true proportion of females in the ADULT dataset.

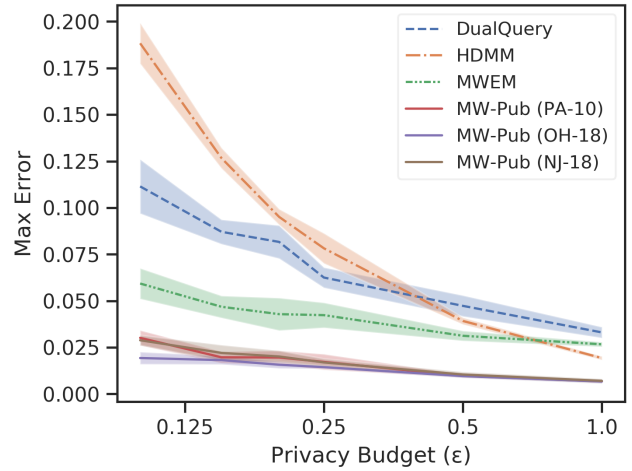


Figure 6: **[2018 ACS (reduced)-PA]** Max error on 5-way marginals with the maximum workload size (3003) across privacy budgets $\varepsilon \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ and $\delta = \frac{1}{N^2}$.

5.3 Ablation studies

To understand how MW-Pub improves upon MWEM, we run additional experiments that compare the two algorithms. Note that because of the drawbacks described in section 2, running MWEM requires data domains that are reasonably small. As a result, we run these experiments on a reduced version the ACS dataset, which we denote as ACS (reduced), by selecting attributes that take on fewer values. Given that the total number of binary attributes is significantly decreased, we are able to run all experiments with 5-way marginals at the maximum workload size of 3003.

For our public datasets, we use PA-10, OH-18, and NJ-18 and present results in Figure 6. When compared to DualQuery, HDMM, and MWEM, the performance of MW-Pub on ACS (reduced) is similar to experiments using the full ACS data, with MW-Pub outperforming all three benchmarks. We note however that for 5-way marginal queries on this reduced set of attributes, using NJ-18 as a public dataset also outperforms HDMM, which was not true in our previous set of experiments. In addition we observe that MWEM outperforms HDMM and DualQuery, further supporting that MWEM can achieve strong performance in cases where it is feasible to run the algorithm.

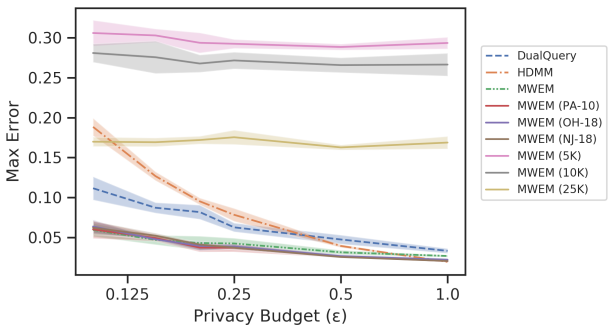


Figure 7: **[2018 ACS (reduced)-PA]** Max error on 5-way marginals across privacy budgets $\varepsilon \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ where $\delta = \frac{1}{N^2}$ and workload size is 3003 (maximum). We run MWEM while maintaining a distribution over the domains of our public datasets $\widehat{D} \in \{\text{PA-10}, \text{OH-18}, \text{NJ-18}\}$ (labeled as MWEM ($\widehat{D}$)) rather than the entire universe. In addition, we sample C rows from the data domain and run MWEM with $C \in \{5K, 10K, 25K\}$ (labeled as MWEM (C)).

Next recall from section 3 that that MW-Pub makes two modifications to MWEM: (1) MW-Pub maintains a distribution over the public domain rather than the entire data domain, and (2) MW-Pub initializes its approximating distribution to the distribution of the public dataset. To understand how each modification impacts the performance of our algorithm, we evaluate MW-Pub against MWEM and summarize our experiments and analysis as the following:

(1) To evaluate the impact of maintaining a distribution over the public data domain, we run MW-Pub using only this first modification with public datasets PA-10, OH-18, and NJ-18. In other words, we run MWEM over a reduced support. As a separate baseline, we also run MWEM using supports of varying sizes sampled uniformly from the data domain. We present the performance of these algorithms in Figure 7 alongside the performance of our benchmark algorithms. The support size of our three public datasets are each approximately 2.5K, which is significantly smaller than that of the data domain, which is approximately 100K. However, we observe that running MWEM over these public data domains yields nearly identical performance to running MWEM over the entire data domain. On the other hand, when using a random sample of the data domain at roughly 2x, 4x, and 10x the size of our public data domains, MWEM

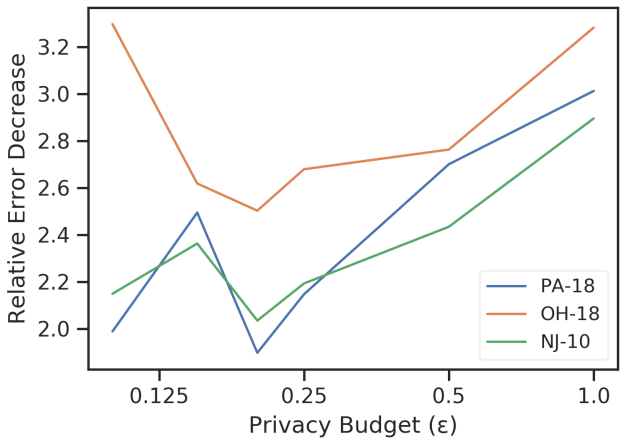


Figure 8: **[2018 ACS (reduced)-PA]** Relative decrease in max error (**higher** is better) of MW-Pub vs. MWEM initialized with a uniform distribution over the public data domain. For example, a relative decrease of 2.5 means that the max error of MW-Pub is 2.5x lower. Experiments are run across privacy budgets $\varepsilon \in \{0.1, 0.15, 0.2, 0.25, 0.5, 1\}$ where $\delta = \frac{1}{N^2}$. We evaluate over 5-way marginal queries at the maximum workload size of 3003.

performs very poorly. Therefore, we conclude that for this set of attributes and queries, our public datasets offer sufficient supports with significantly reduced dimensionality.

(2) We evaluate how the initialization of A_0 in MW-Pub affects performance by comparing it to a variant of MW-Pub that does not incorporate this modification. In other words, we again run MWEM where A_0 is initialized to to a uniform distribution over the *public data* domain. We observe in Figure 8 that initializing to the distribution of the public dataset rather than a uniform distribution significantly improves performance across all privacy budgets, decreasing the max error by a factor of approximately 2 to 3.

6 Conclusion and Future Work

In this paper, we introduced MW-Pub, an extension to MWEM that leverages prior knowledge from public data and dramatically reduces its running time and memory requirements. We empirically evaluate our method on the 2018 ACS dataset for Pennsylvania and show that there exists a number of choices for public datasets that allow MW-Pub to outperform state-of-the-art benchmark algorithms. In addition, we run experiments on ADULT and a reduced version of the ACS dataset to better understand our proposed algorithm. For future work, we hope to formally characterize how properties of the public dataset affect the final accuracy of MW-Pub.

7 Appendix

7.1 Data

Attributes for our experiments on ACS, ACS (reduced), and ADULT:

- **ACS:** ACREHOUS, AGE, AVAILBLE, CITIZEN, CLASSWKR, DIFFCARE, DIFFEYE, DIFFHEAR, DIFFMOB, DIFFPHYS, DIFFREM, DIFFSENS, DIVINYR, EDUC, EMPSTAT, FERTYR, FOODSTMP, GRADEATT, HCOVANY, HCOVPRIV, HINSCAID, HINSCARE, HINSVA, HISPAN, LABFORCE, LOOKING, MARRINYR, MARRNO, MARST, METRO, MIGRATE1, MIGTYPE1, MORTGAGE, MULTGEN, NCHILD, NCHLT5, NCOUPLES, NFATHERS, NMOTHERS, NSIBS, OWNERSHP, RACAMIND, RACASIAN, RACBLK, RACE, RACOTHER, RACPACIS, RACWHT, RELATE, SCHLTYPE, SCHOOL, SEX, SPEAKENG, VACANCY, VEHICLES, VET01LTR, VET47X50, VET55X64, VET75X90, VET90X01, VETDISAB, VETKOREA, VETSTAT, VETVIETN, VETWWII, WIDINYR, WORKEDYR
- **ACS (reduced):** DIFFEYE, DIFFHEAR, EMPSTAT, FOODSTMP, HCOVPRIV, HINSCAID, HINSCARE, OWNERSHP, RACAMIND, RACASIAN, RACBLK, RACOTHER, RACPACIS, RACWHT, SEX
- **ADULT:** sex, income>50K, race, relationship, marital-status, workclass, occupation, education-num, native-country, capital-gain, capital-loss, hours-per-week, age

In addition, we discretize the following continuous variables into various bins sizes:

- **ACS:** AGE
- **ACS (reduced):** AGE
- **ADULT:** capital-gain, capital-loss, hours-per-week, age

7.2 Hyperparameters

We report hyperparameters used across all experiments in Table 1.

Table 1: Hyperparameter selection for experiments on all datasets.

Method	Parameter	Values
MW-Pub	T	300, 250, 200, 150, 25, 100, 75, 50, 25, 10, 5
MWEM	T	300, 250, 200, 150, 25, 100, 75, 50, 25, 10, 5
DualQuery	samples η	500 250 100 50 5 4 3 2

References

Abowd, J. M. 2018. The U.S. Census Bureau Adopts Differential Privacy. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, 2867. doi:10.1145/3219819.3226070. URL https://doi.org/10.1145/3219819.3226070.

Alon, N.; Bassily, R.; and Moran, S. 2019. Limits of private learning with access to public data. *arXiv preprint arXiv:1910.11519* .

Avent, B.; Korolova, A.; Zeber, D.; Hovden, T.; and Livshits, B. 2017. {BLENDER}: Enabling local search with a hybrid differential privacy model. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, 747–764.

Bassily, R.; Cheu, A.; Moran, S.; Nikolov, A.; Ullman, J.; and Wu, Z. S. 2020. Private Query Release Assisted by Public Data. *arXiv preprint arXiv:2004.10941* .

Bassily, R.; Moran, S.; and Nandi, A. 2020. Learning from mixtures of private and public populations. *arXiv preprint arXiv:2008.00331* .

Beimel, A.; Nissim, K.; and Stemmer, U. 2013. Private learning and sanitization: Pure vs. approximate differential privacy. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 363–378. Springer.

Bun, M.; and Steinke, T. 2016. Concentrated Differential Privacy: Simplifications, Extensions, and Lower Bounds. In *Proceedings of the 14th Conference on Theory of Cryptography*, TCC '16-B, 635–658. Berlin, Heidelberg: Springer.

Canonne, C. L.; Kamath, G.; and Steinke, T. 2020. The Discrete Gaussian for Differential Privacy. In *NeurIPS*. URL https://arxiv.org/abs/2004.00010.

Dua, D.; and Graff, C. 2017. UCI Machine Learning Repository. URL http://archive.ics.uci.edu/ml.

Dwork, C. 2006. Differential Privacy. In *33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006)*, volume 4052 of *Lecture Notes in Computer Science*, 1–12. Springer Verlag. ISBN 3-540-35907-9. URL https://www.microsoft.com/en-us/research/publication/differential-privacy/.

Dwork, C.; McSherry, F.; Nissim, K.; and Smith, A. 2006. Calibrating Noise to Sensitivity in Private Data Analysis. In *Proceedings of the 3rd Conference on Theory of Cryptography*, TCC '06, 265–284. Berlin, Heidelberg: Springer.

Gaboardi, M.; Arias, E. J. G.; Hsu, J.; Roth, A.; and Wu, Z. S. 2014. Dual query: Practical private query release for high dimensional data. In *International Conference on Machine Learning*, 1170–1178.

Hardt, M.; Ligett, K.; and McSherry, F. 2012. A simple and practical algorithm for differentially private data release. In *Advances in Neural Information Processing Systems*, 2339–2347.

Hardt, M.; and Rothblum, G. N. 2010. A multiplicative weights mechanism for privacy-preserving data analysis. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, 61–70. IEEE.

Kifer, D. 2019. Consistency with External Knowledge: The TopDown Algorithm. http://www.cse.psu.edu/~duk17/papers/topdown.pdf.

McKenna, R.; Miklau, G.; Hay, M.; and Machanavajjhala, A. 2018. Optimizing error of high-dimensional statistical queries under differential privacy. *PVLDB* 11(10): 1206–1219.

McKenna, R.; and Sheldon, D. 2020. Permute-and-Flip: A new mechanism for differentially private selection. *arXiv preprint arXiv:2010.12603* .

McSherry, F.; and Talwar, K. 2007. Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, 94–103. IEEE.

Ruggles, S.; et al. 2020. IPUMS USA: Version 10.0, DOI: 10.18128/D010. *V10. 0* .

Vietri, G.; Tian, G.; Bun, M.; Steinke, T.; and Wu, Z. S. 2020. New Oracle-Efficient Algorithms for Private Synthetic Data Release. *arXiv preprint arXiv:2007.05453* .