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. To address this shortcoming, we study the setting in which the data curator has access to public data. Specifically, we present MWEM+PUB, which adapts MWEM to scale to high-dimensional data by exploiting public samples. 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 statistical information 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 for 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 that 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 proposals for new algorithms that circumvent this issue while following similar no-regret learning dynamics (Gaboardi et al. 2014; Vietri et al. 2020). In a similar vein, our proposed method, MWEM+PUB, adapts MWEM to make use of public data, which we define as any samples that pose no privacy concerns, in order to make the algorithm scalable to higher-dimensional datasets.

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 when individuals voluntarily offer or sell their data. Following this observation, many works have also studied utilizing public data for differential privacy. Bassily et al. for example prove upper and lower bounds for private and public sample complexities in the context of private query release. Similarly, Alon, Bassily, and Moran 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 extend this line of research, studying PAC learnabiliy while relaxing the assumption that public and private samples come from the same distribution.

2 Preliminaries

Definition 2.1 (Differential Privacy (DP))**.** 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 2.2 (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 2.3 (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.

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 MWEM+PUB , which adapts MWEM to take advantage of public data.

MWEM. The multiplicative weights exponential mechanism (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 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 introduce a scalable implementation of MWEM that avoids explicitly tracking A when the query class involves disjoint subsets of attributes. While MWEM has running time proportional to $|\mathcal{X}|$ in this special case, it becomes applicable only to simpler workloads.

MWEM+PUB. To overcome its shortcomings, we introduce MWEM+PUB in Algorithm 1, which adapts MWEM using 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 $|\mathcal{X}|$, MWEM+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 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 of $\widehat{\mathcal{X}}$.

Algorithm 1: MWEM+PUB

Input: Private dataset $\widetilde{D}$, public dataset $\widehat{D}$ with public domain $\widehat{\mathcal{X}}$, query class $\mathcal{Q}$, privacy parameters ε, δ

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

Initialize ε_0, T, s.t. $\varepsilon = \sqrt{2T\ln(1/\delta)}\varepsilon_0 + T\varepsilon_0(e^{\varepsilon_0} - 1)$

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

Sample a query $q_t \in \mathcal{Q}$ using the *exponential mechanism* with epsilon value $\varepsilon_0/2$ and the score function:

$$S_t(\widetilde{D}, q) = |q(A_{t-1}) - q(\widetilde{D})|$$

Measure: Let $a_i = q_t(\widetilde{D}) + \text{Lap}(\varepsilon_0/2)$

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

$$A_t(x) \propto A_{t-1}(x)\exp\left(q_t(x)\left(a_i - q_t(A_{t-1})\right)\right)$$

end

4 Experimental Setting

We describe the datasets and benchmarks used to evaluate MWEM+PUB in our experiments.

4.1 Data

American Community Survey (ACS) We evaluate all algorithms on the 2018 American Community Survey (ACS), obtained from the IPUMS USA database (Ruggles et al. 2020). Collected every year on the United States Census Bureau, the ACS provides statistics meant to capture the social and economic conditions of households across the country. Given that the Census Bureau is incorporating differential privacy into 2020 Census release (Abowd 2018), we believe the ACS 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. For selecting 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 ACS data for Pennsylvania from 2010.

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), Michigan (MI), Illinois (IL), and New Jersey (NJ).

ADULT In addition, we evaluate all algorithms on the ADULT dataset from the UCI machine learning dataset repository (Dua and Graff 2017). We randomly split ADULT into private and public datasets in a 10:1 ratio. 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.

4.2 Benchmarks

We evaluate MWEM+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. 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 added benefit 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.

5 Results

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

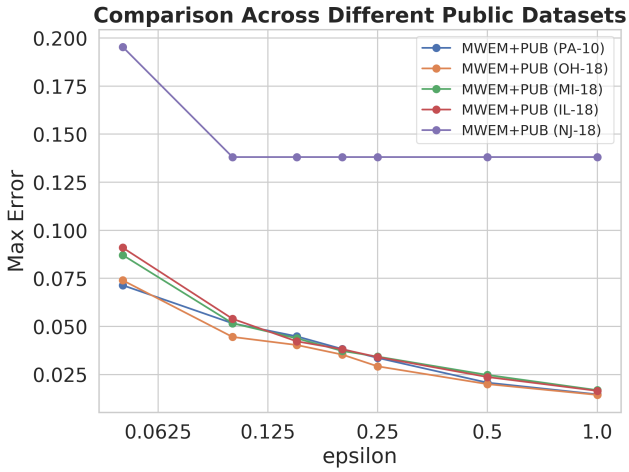
5.1 ACS (Pennsylvania)

We compare the performance of MWEM+PUB using the public datasets described in section 4.1. As seen in Figure 1, 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 ($\epsilon = 0$) is quite high. Moreover, the performance of MWEM+PUB does not improve for $\epsilon > 0.1$, possibly indicating that 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, MWEM+PUB performs much better, with the algorithm converging to approximately the same error as ϵ approaches 1.0.

Next, we compare MWEM+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:

MWEM+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 2 shows that for $\epsilon < 0.25$, it is better to directly answer queries using the 2010 ACS data for Pennsylvania, rather than use DUALQUERY or HDMM. Running MWEM+PUB improves upon the initial error of using the public datasets, outperforming the benchmark algorithms across all values of $\epsilon \leq 1.0$.

MWEM+PUB performs well even when the size of the public dataset is reduced to less than 1%. In Figure 3, we plot the performance of MWEM+PUB using different public dataset sizes. Reducing the 2010 ACS-PA and 2018 ACS-OH datasets to only 10% of their original sizes yields no significant loss in performance. At $\epsilon = 0.5$, MWEM+PUB outperforms all benchmarks even when using just 0.5% of the public dataset. 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 useful.



Public Dataset	Initial Error
PA (2010)	0.0967
OH (2018)	0.1370
MI (2018)	0.1653
IL (2018)	0.1599
NJ (2018)	0.3389

Figure 1: **Top:** Comparison of max errors using different public datasets to run MWEM+PUB on the ACS (PA 2018), fixing $\epsilon = 0.5$ and $\delta = \frac{1}{N^2}$. We evaluate on 3-way marginals at a workload size of 2048. Results for additional values of ϵ can found in Figure 9. **Bottom:** Table comparing the initial error of each public dataset ($\epsilon = 0$).

Compared to MWEM+PUB, HDMM scales poorly with respect to workload size. We compare the performance of MWEM+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 MWEM+PUB remains relatively stable. Our experiments suggest that while HDMM is designed to answer queries for high-dimensional datasets, MWEM+PUB may be a more suitable algorithm when the goal is release large workloads of queries.

5.2 ADULT

Next, we evaluate MWEM+PUB against the benchmark algorithms on the ADULT dataset. In our experimental setting, both the public and private datasets are sampled uniformly without replacement from the overall dataset. As a result, the distributions of these two paritions are very similar, and we observe that simply releasing the public partition as the private dataset ($\epsilon = 0$) achieves very low max error. Consequently, we conduct additional experiments by sampling the public dataset 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. In Figure 5, we observe that even at $\Delta = 0.3$ where our public dataset is comprised of almost entirely males, MWEM+PUB outperforms both benchmarks.

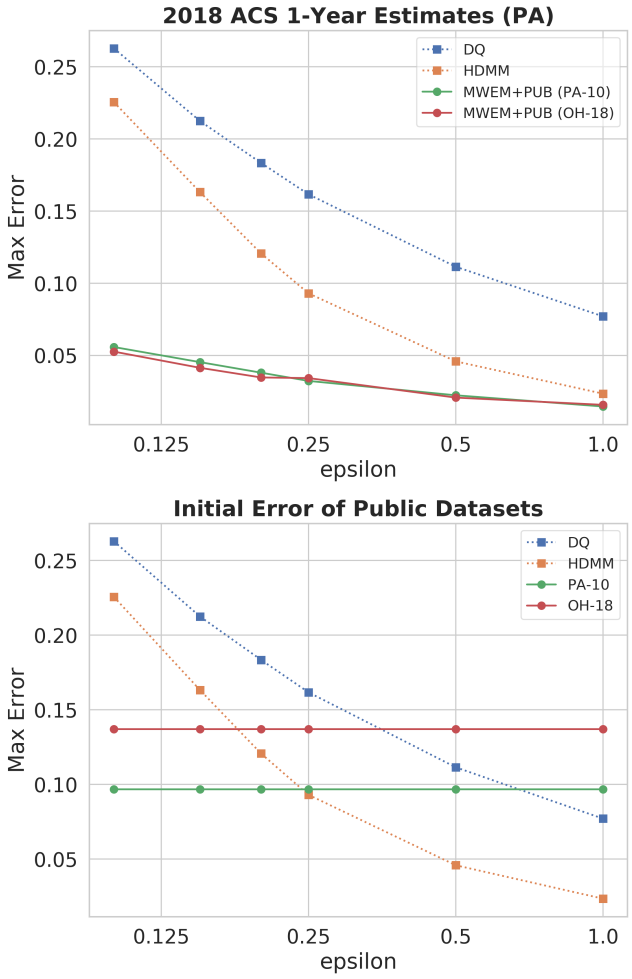


Figure 2: Max error on 3-way marginals across privacy budgets ϵ where $\delta = \frac{1}{N^2}$ and the workload size is 4096. **Top:** We compare MWEM+PUB to the benchmark algorithms. **Bottom:** We evaluate the max error when using the public datasets to answer queries directly ($\epsilon = 0$).

5.3 Ablation studies

To understand how MWEM+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-small, 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 MWEM+PUB on ACS-small is similar to experiments using the full ACS data, with MWEM+PUB outperforming all three benchmarks. We note however that for 5-way marginal

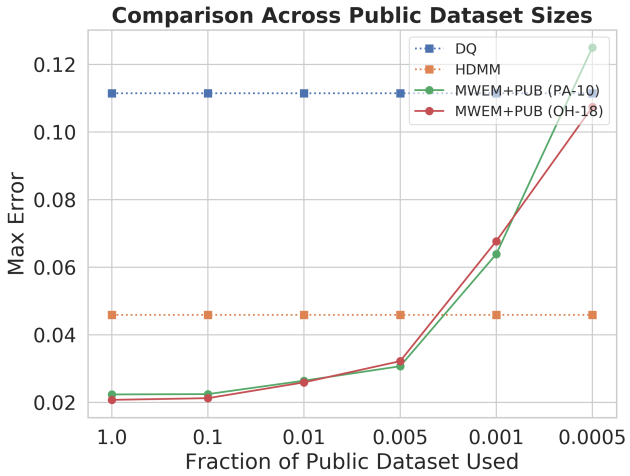


Figure 3: Max error on 3-way marginals while varying the fraction of the public dataset used, where $\epsilon = 0.5$, $\delta = \frac{1}{N^2}$, and workload size is 4096. Results for additional values of ϵ can found in Figure 10.

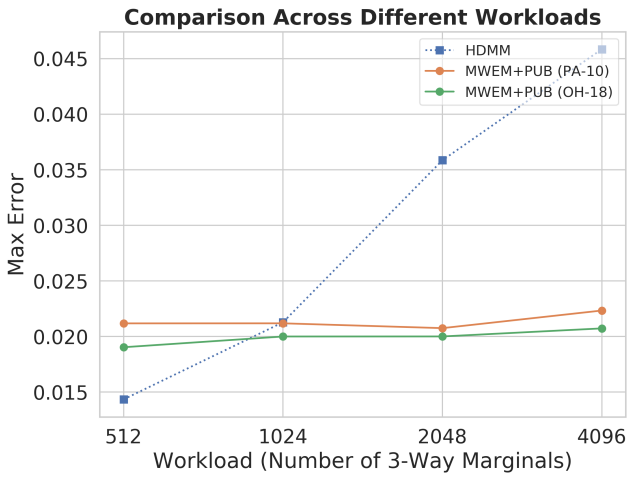


Figure 4: Comparison of MWEM+PUB (using the enitre public dataset) against HDMM on 3-way marginals while varying the workload size. Results displayed are for $\epsilon = 0.5$ and $\delta = \frac{1}{N^2}$. Results for additional values of ϵ can found in Figure 11.

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 when it is feasible to run the algorithm.

Next recall from section 3 that that MWEM+PUB makes two modifications to MWEM: (1) MWEM+PUB maintains a distribution over the public domain rather than the entire data domain, and (2) MWEM+PUB initializes its approximating distribution to the distribution of the public dataset. To understand how each modification impacts the perfor-

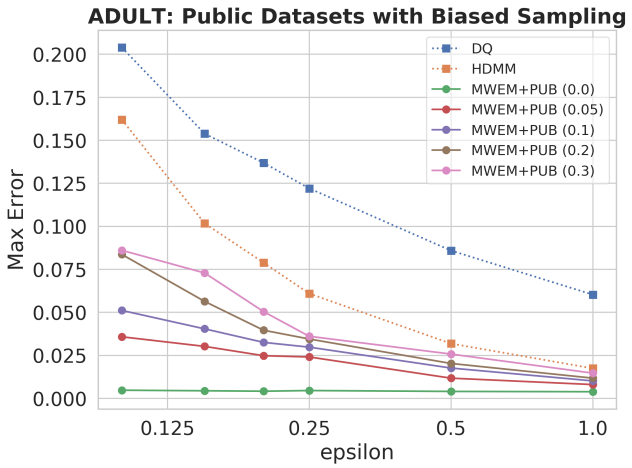


Figure 5: Max error on 3-way marginals across privacy budgets ϵ where $\delta = \frac{1}{N^2}$ and the workload size is 256. When running MWEM+PUB, Each public dataset is created by sampling from the public partition with some bias Δ over the attributes *sex* (labeled as MWEM+PUB (Δ)).

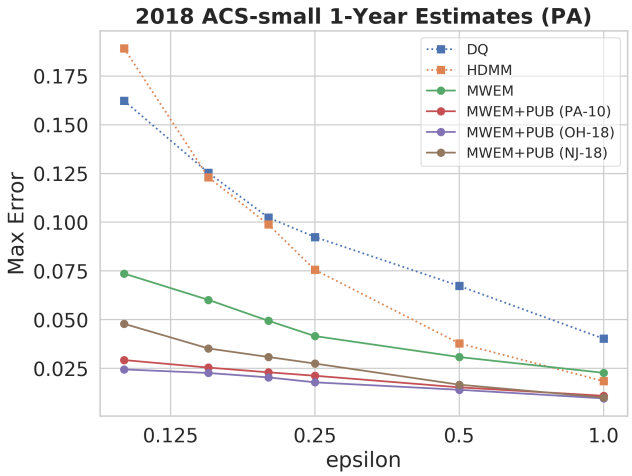


Figure 6: Max error on 5-way marginals across privacy budgets ϵ where $\delta = \frac{1}{N^2}$. We use the maximum workload size (3003) and run MWEM+PUB with the entire public dataset.

mance of our algorithm, we evaluate MWEM+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 MWEM+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 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 roughly 2.5K, which is significantly smaller than that of the

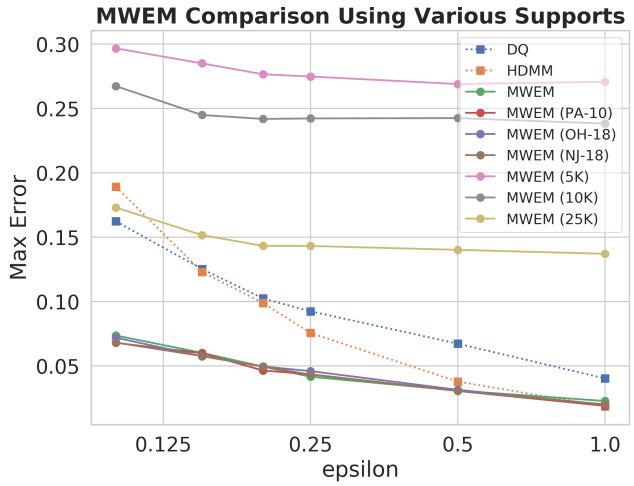


Figure 7: Max error on 5-way marginals across privacy budgets ϵ where $\delta = \frac{1}{N^2}$ and workload size is 3003. We run MWEM while maintaining a distribution over the domains of our public datasets (PA-10, OH-18, NJ-18) rather than the entire universe. In addition, we sample C rows from the data domain and run MWEM with $C \in \{5000, 10000, 25000\}$ (labeled as MWEM (C)).

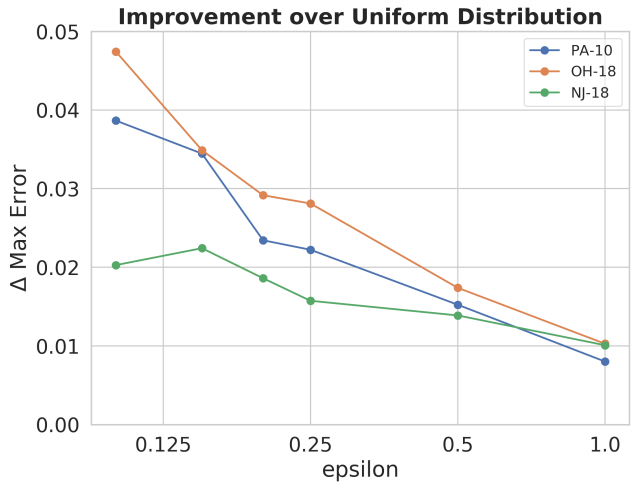


Figure 8: Difference in max error (**higher** is better) of MWEM+PUB vs. MWEM initialized with a uniform distribution over the public data domain. Experiments are run across privacy budgets ϵ where $\delta = \frac{1}{N^2}$. We evaluate over 5-marginal queries at the maximum workload size of 3003.

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 2x, 4x, and 10x the size of our public data domains, MWEM performs very poorly. Consequently, we conclude that for these set of attributes and queries, our public datasets offer sufficient supports with significantly reduced dimensionality.

(2) To evaluate the initialization of the approximating distribution in MWEM+PUB, we compare running MWEM+PUB to running MWEM with A_0 as a uniform distribution over the public data domain. We observe in Figure 8 that using the distribution of the public dataset significantly improves performance for smaller values of epsilon. As our privacy budget ϵ increases, the difference in max errors between MWEM and MWEM+PUB shrinks. However, the improvement of MWEM+PUB at $\epsilon = 1.0$ is nonnegliglble, pushing the max error of MWEM below HDMM (Figure 6).

6 Conclusion and Future Work

In this paper, we introduced MWEM+PUB, an enhancement to MWEM that uses public data to dramatically reduce its running time and memory requirements while still achieving high accuracy. We empirically evaluate our method on the 2018 ACS dataset for Pennsylvania and show that there exist a number of choices for public datasets that allow MWEM+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 MWEM+PUB.

7 Appendix

7.1 Additional Figures

We provide more comprehensive results for Figures 2, 3, and 4 in Figures 9, 10, and 11 respectively. In Figure 9, we observe that the performance of MWEM+PUB does not decrease significantly until we use under 0.5% of the public dataset. In Figure 11, we observe that across all epsilon, the max error of HDMM increases as the number of marginals increase while the max error of MWEM+PUB remains relatively constant.

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* .

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* .

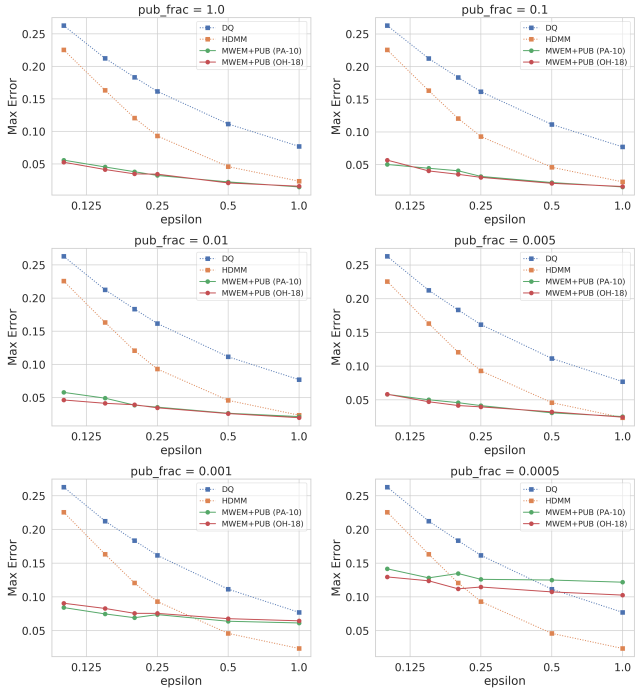


Figure 9: Max error on 3-way marginals across privacy budgets ϵ where $\delta = \frac{1}{N^2}$ and workload size is 4096. We vary the fraction of the public dataset used in each plot shown.

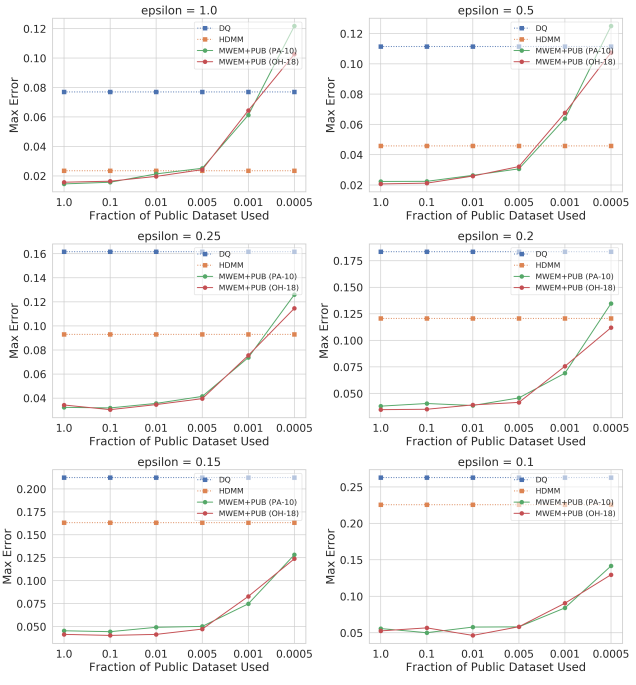


Figure 10: Comparison MWEM+PUB to the benchmark algorithms on 3-way marginals with workload size as 4096 while varying the fraction of the public dataset used. We fix $\delta = \frac{1}{N^2}$ and vary ϵ for each plot shown.

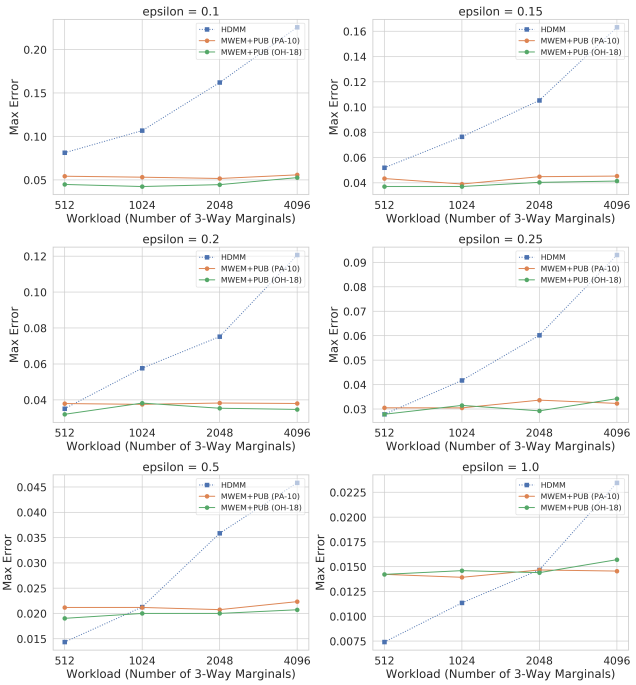


Figure 11: Comparison MWEM+PUB (using the entire public dataset) against to HDMM on 3-way marginals while varying the workload size. We fix $\delta = \frac{1}{N^2}$ and vary the privacy budget ϵ in each plot shown.

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.

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.

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* .