A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis

Roger A. Hallman
Roger.Hallman.TH@dartmouth.edu

Fall Quarter 2018

1 Introduction

We live in an ever-more interconnected society where Internet-connected devices are embedded into objects which we interact with on a daily bases, and consequently transmit data about us and our actions. This generated data often takes the form of statistical information and is stored in statistical databases. One need not dig too deeply in the current news to find documentation of sensitive statistical information being compromised, indeed data subject identification by statistical inference is possible even when statistical data holders take reasonable steps to anonymize the data in their possession. Statistical analysis of sensitive information can yield important benefits. For example, this analysis may be used to inform public policy decisions or to help private businesses generate revenue by improved targeting of advertisements. However, this information can be misused.

A protection against deanonymization by statistical inference, known as "Differential Privacy" (DP), was presented by Dwork [4] in 2006, upon which a strong body of privacy research has been built. At a high level, DP (the fundamentals of which are provided in Section 2) adds noise to query answers to protect data subject privacy while also providing accurate answers. Among the central questions of DP research is this tradeoff between utility and privacy and the number of queries which can be optimally run. Early results suggested that sequences of differentially private queries that were smaller in number than the individual data subjects in the database could be answered with considerable accuracy. However, it was also shown that there were families of queries for which "too many" queries could lead to privacy violations. Nonetheless, these query families could be run in a privacy preserving manner if accuracy was sacrificed.

In general, there are two classes of mechanisms which attempt to optimize the number of queries answered while preserving privacy and accuracy: Non-interactive and interactive mechanisms. Non-interactive Mechanisms [2, 6] are characterized by a set of queries that is specified in advance. [2] demonstrated a privacy-preserving, non-interactive mechanism for a predefined set $\mathcal{C}$ of counting queries where error scaled logarithmically with the number of queries being answered. [6] presented a non-interactive mechanism that ran in polynomial time in N and k, with error scaling roughly as $\left(\frac{1}{\sqrt{n}}\right) \cdot k^{o(1)}$. Both of these non-interactive approaches output a synthetic data base, (i.e., a database of entries from the parent database where for each query, the fraction of data subjects who satisfy the answer are within the error bound). Interactive mechanisms, on the other hand, relaxed the requirement that query sets be predefined. [8] presented a method of answering interactive counting queries where error scaled $\left(\frac{1}{\sqrt[3]{n}}\right) \cdot \text{polylog}(k)$, however this mechanism ran in super-polynomial time. Some important questions about interactive mechanisms include:

1. Is there a polynomial time interactive mechanism with non-trivial error on all databases?
2. If such an algorithm exists, could its error scale to a sampling error of $\left(\frac{1}{\sqrt{n}}\right)$ and grow only logarithmically with k queries?

3. Do there exist interactive mechanisms that can match (or nearly so) the hardness results in [6]?

4. Are there relaxations that will permit such an interactive mechanism to run in sub-linear or poly-logarithmic time?

In this paper, we review a paper from Hardt and Rothblum entitled "A Multiplicative Weights Mechanism for Privacy-Preserving Data Analysis" [7], which presented an algorithm that they call a "Private Multiplicative Weights" (PMW) mechanism. The PMW answers affirmatively to questions 1-3 and makes progress on question 4. Moreover, the PMW runs in linear time and provides worst-case accuracy guarantees.

The rest of this paper is organized as follows: Section 2 makes up for a significant shortcoming of [7] by providing the reader with an introduction to DP. Section 3 presents the PMW mechanism and theorems for its privacy and accuracy performance. [7] has (at the time of this writing) nearly 300 citations on Google Scholar–it is clearly highly regarded within the DP research community–and Section 4 presents an overview of more recent papers that have built off of the present work.

2 Background Information

We provide background information on Differential Privacy which is a 21^{st} Century development with which readers may be unfamiliar. This background information is provided in part because the paper that this report reviews does not do an adequate job of providing background information, and assumes that the reader has at least a familiarity with the fundamentals of the topic.

2.1 Differential Privacy

Tore Dalenius [3] expressed the fundamental statement of data privacy in statistical databases: *Anything that can be learned about a respondent from the statistical database should be learnable without access to the database.* This was a longstanding challenge until Dwork's seminal 2006 work on DP [4], which provides security against probabilistic data subject identification. In this section, we take a deeper dive into DP which ensures that individual data subjects assume no additional risk by adding their information to a statistical database [5].

2.2 Formalizing Differential Privacy

Unless otherwise stated, we quote definitions, remarks, and theorems in the subsection from [4, 5]. Unless otherwise specified, the term "database" refers to a statistical database and a database D is a set of rows. Given two databases, D_1 and D_2 differ in at most one element if one is a proper subset of the other and the larger database contains exactly one additional row. A randomized function $\mathcal{K}$ is the algorithm applied by the database curator when releasing information.

Definition 2.1. ε-differential privacy. A randomized function $\mathcal{K}$ gives ε-differential privacy if for all data in sets D_1 and D_2 differing on at most one element, and $S \subseteq \text{Range}(\mathcal{K})$,

$$\Pr[\mathcal{K}(D_1) \in S] \leq \exp(\varepsilon) \times \Pr[\mathcal{K}(D_2) \in S]$$

The probability is taken over the coin tosses of $\mathcal{K}$.

We hope that as long as $\mathcal{K}$ satisfies this definition, $\mathcal{K}$ would guarantee that no outputs would become significantly more or less likely in the event that a data subject's data was removed from D. (There should be a similar guarantee if a new data subject's data is added to D.) While DP is not an absolute guarantee of privacy, it is a strong guarantee because it is a statistical property about the behavior of $\mathcal{K}$ and is independent auxiliary available information or available computational power. Moreover, such a mechanism $\mathcal{K}$ provides a guarantee of privacy even when an adversary possesses knowledge of every other row of D. Dwork also claims that Definition 2.1 extends to data subject groups within D (as well as the case when a single data subject contributes to multiple rows of D).

Given a "small" $c > 1$ data subjects, Definition 2.1 is modified by bounding the probability dilation by $\exp(c\varepsilon)$.

2.2.1 But how to find $\mathcal{K}$?!?

Let f be a query function on D and $a = f(D)$ be the answer of the query function. $\mathcal{K}$ adds appropriately chosen random noise to a. Dwork provides two illustrative examples:

1. Simple – "Count the number of rows in D satisfying a given predicate."
2. Complex – "Compute the median value for each column; if the Column 1 median exceeds the Column 2 median, then output a histogram of the numbers of points in the set S of orthants, else provide a histogram of the numbers in different set T of orthants.

The simple query outputs a vector of values, while the complex query is an adaptively chosen sequence of two vector-valued queries and depends on the true answer of the first query. Moreover, the complex query is also a function of the noise added by $\mathcal{K}$.

2.2.2 Getting the noise right

ε-differential privacy is achieved by the addition of random noise, the magnitude of which is chosen as a function of the largest change a single participant could have on the query function, the *sensitivity* of f.

Definition 2.2. $L1$-sensitivity. Given a distribution on databases, $\mathcal{D}$, and for $f : \mathcal{D} \to \mathbb{R}^d$, the $L1$-sensitivity of f is

$$\Delta f = \max_{D_1, D_2} ||f(D_1) - f(D_2)||_1$$

for all D_1, D_2 differing by at most one element.

$L1$-sensitivity is a property of the function and is independent of any particular database. Moreover, many database queries (for instance, simple counts such as "how many rows have property P") will have small Δf's, usually a $\Delta f \leq 1$. DP achieves its best results, succeeds at providing accurate query results while minimizing the risk of data subject identification, when Δf is small.

$\mathcal{K}_f$ is query-specific privacy function which computes $f(X)$ adds noise according to a scaled symmetric exponential distribution with variance σ^2 , and described by the density function:

$$\Pr[\mathcal{K}_f(X) = a] \propto \frac{\exp(-||f(X) - a||_1)}{\sigma^2}$$

The privacy function $\mathcal{K}_f$'s distribution has independent, exponentially distributed random variables as coordinates. Thus, $\mathcal{K}_f$ adds symmetric exponential noise to each coordinate of $f(X)$. The exact value of σ^2 is determined by the following theorem (the proof of which is provided in [4]:

Theorem 2.1. For $f : \mathcal{D} \to \mathbb{R}^d$, $\mathcal{K}_f$ gives $\left(\frac{\Delta f}{\sigma}\right)$-differential privacy. ε-differential privacy is achieved by setting $\sigma = \frac{\varepsilon}{\Delta f}$.

Returning to the complex query in 2.2.1, the importance of choosing noise as a function of the sensitivity of the entire query should be clear. Consider the case of histogram queries, in which the domain of data elements is partitioned into k classes and the query's true answer is the k-tuple of the exact number of database points in each class. This is a set of k queries, each of sensitivity 1, Theorem 2.1 guarantees that ε-differential privacy is achieved by using noise distributed according to a symmetric exponential with $\sigma^2 = \frac{k}{\varepsilon}$. However, given D_1, D_2, $||f(D_1) - f(D_2)||_1 = 1$, as only a single cell of the histogram changes, it suffices to apply Theorem 2.1 and select $d = k$ and $\Delta f = 1$, which adds noise with $\sigma^2 = \frac{1}{\varepsilon}$.

Remark. The authors in [7] generate their noise as a Laplacian distribution $Lap(\sigma)$ centered at 0 with scaling σ and a corresponding density $f(x) = \frac{e^{-\frac{|x|}{\sigma}}}{2\sigma}$.

3 Main Conclusions

Now that an exposition on the fundamentals of DP has been provided, we are ready to describe the PMW, as presented in [7], as well as prove its privacy and accuracy claims.

Definition 3.1. Private Multiplicative Weights (PMW) Mechanism. **Parameters:** A subset of the coordinates $V \subseteq U$ with $|V| = M$ (by default $V = U$), intended number of rounds $k \in \mathbb{N}$, privacy parameters $\varepsilon, \delta > 0$ and failure probability $\beta > 0$. Let

$$\sigma = \frac{10 \log\left(\frac{1}{\delta}\right) \sqrt[4]{\log M}}{\sqrt{n} \cdot \varepsilon}$$
$$\eta = \frac{\sqrt[4]{\log M}}{\sqrt{n}}$$
$$T = 4\sigma \cdot \left(\log k + \log\left(\frac{1}{\beta}\right)\right)$$

Input: A database $D \in U^n$ corresponding to a histogram $x \in \mathbb{R}^n$
Algorithm: Set $y_0[i] = x_0[i] = \frac{1}{M}$ for all $i \in V$. In each round $t \leftarrow 1, 2, ..., k$, when receiving a linear query f_t, do the following:

1. Sample $A_t \sim Lap(\sigma)$. Compute the noisy answer $\hat{a}_t \leftarrow \langle f_t, x \rangle + A_t$.
2. Compute the difference $\hat{d}_t \leftarrow \hat{a}_t - \langle f_t, x \rangle$. If $|\hat{d}_t| \leq T$, then set $w_t \leftarrow 0, x_t \leftarrow x_{t-1}$, output $\langle f_t, x \rangle$, and proceed to the next iteration. If $|\hat{d}_t| > T$, then set set $w_t = 1$ and (a) for all $i \in V$, update $y_t[i] \leftarrow x_{t-1}[i] \cdot e^{-\eta \cdot r_t[i]}$, where $r_t[i] = f_t[i]$ if $\hat{d}_t > 0$ and $r_t[i] = 1 - f_t[i]$ otherwise; (b) normalize $x_t[i] \leftarrow \frac{y_t[i]}{\sum_{i \in V} y_t[i]}$; let $m = \sum_{j=1}^{t} w_j$. If $m > n \cdot \sqrt{\log M}$ then abort and output a 'failure' message. Otherwise, output the noisy answer $\hat{a}_t$ and proceed to the next iteration.

In more plain language, in each round t, a series of linear queries f_t are presented over U and x_t denotes a fractional histogram (with $\text{domain}(x_t) = V \subseteq U$, and $|V| \ll |U|$) computed in round t. a_t is the true answer to query t, and $\hat{a}_t$ denotes this same answer with noise added to it. d_t is the difference between a_t and the answer given by x_{t-1}, $\hat{d}_t$ is the difference between $\hat{a}_t$ and the answer given by x_{t-1}. If $|\hat{d}_t| \lessapprox \frac{1}{\sqrt{n}}$ then t is a "lazy" round and $w_t = 0$, otherwise t is an "update" round and $w_t = 1$. A lazy round, when $|\hat{d}_t|$ is small, outputs $f_t(x_{t-1})$ and sets $x_t \leftarrow x_{t-1}$. An update round occurs when x_t needs to be improved (i.e.,$|\hat{d}_t|$ is not small) using the PMW mechanism and bringing $\hat{x}_t$ "closer" to an accurate answer on f_t. However, the number of update rounds must be bounded to prevent privacy violations: if the number of update rounds in a series of queries grows to be $\gtrapprox n$, then the mechanism fails and terminates.

3.1 Proving PMW Utility

We present the two main theorems of [7], which support their claims of the PMW mechanism's utility guarantees. Theorems are presented first, along with explanations and remarks, while proofs will come in subsections 3.1.1 and 3.1.2. These proofs rely on a series of lemmas which we will present to sketch the proofs. The authors also provide an analysis of privacy guarantees for the PMW mechanism, however we omit this analysis as it is not germane at present.

Theorem 3.1. Utility of the PMW. Let U be a data universe of size N. For any $k, \varepsilon, \delta, \beta > 0$, the PMW mechanism is an (ε, δ)-differentially private interactive mechanism. For any database of size n, the mechanism is (α, β, k)-accurate for adaptive counting queries over U, where

$$\alpha = \mathcal{O}\left(\frac{\log\left(\frac{1}{\delta}\right) \sqrt[4]{N} \cdot \left(\log k + \log\left(\frac{1}{\beta}\right)\right)}{\varepsilon \sqrt{n}}\right).$$

The running time in answering each query is $N \cdot \text{poly}(n) \cdot \text{polylog}\left(\frac{1}{\beta}, \frac{1}{\varepsilon}, \frac{1}{\delta}\right)$.

PMW error is a function of N and k and grows at $\approx \left(\frac{1}{\sqrt{n}}\right) \cdot \log k$. Moreover, the PMW mechanism can be used to generate synthetic databases with similar error and running times in a non-interactive setting. In fact, in this environment, the PMW can achieve sub-linear or polylogarithmic running times.

Definition 3.2. Smooth and Pseudo-Smooth Databases. A historgram $x \in \mathbb{R}^U$ such that $\sum_{u \in U} x_u = 1$ and for all $u \in U$, $x_u \geq 0$ is ξ-smooth if, for all $u \in U$, we have $x_u \leq \xi$. That is, a histogram or distribution y over U is ξ-smooth if, for every $u \in U$, the probability of u by y is at most ξ. A histogram $x \in \mathbb{R}^U$ such that $\sum_{u \in U} x_u = 1$ and for all $u \in U$, $x_u \geq 0$ is (ξ, ϕ)-pseudo-smooth with respect to a set $\mathcal{C}$ if there exists a ξ-smooth histogram $x^\star$ such that $|\langle f, x\rangle - \langle f, x^\star\rangle| \leq \phi$.

Remark. The most straightforward method for obtaining a pseudo-smooth database is simply to sample from a smooth histogram.

Theorem 3.2. Utility of the Smooth PMW. Let U be a data universe of size N. For any $\varepsilon, \delta, \beta, \xi, \phi > 0$, the PMW mechanism or an (ε, δ)-differentially private interactive mechanism. For any sequence $\mathcal{C}$ of k interactive counting queries that are fixed in advance (i.e., non-adaptively), and for any database of size n that is (ξ, ϕ)-pseudo-smooth with respect to $\mathcal{C}$, the mecahnism is (α, β, k)-non-adaptively accurate with resect to $\mathcal{C}$, where

$$\alpha = \tilde{\mathcal{O}}\left(\phi + \frac{\log\left(\frac{1}{\delta}\right) \sqrt[4]{\xi N} \cdot \left(\log k + \log\left(\frac{1}{\beta}\right)\right)}{\varepsilon\sqrt{n}}\right).$$

The running time in answering each query is $\xi N \cdot \text{poly}(n) \cdot \text{polylog}\left(\frac{1}{\beta}, \frac{1}{\varepsilon}, \frac{1}{\delta}, \frac{1}{\xi}, \frac{1}{\phi}\right)$.

3.1.1 Proving PMW Performance and Accuracy

Definition 3.3. Potential Function. A target histogram in V is denoted $x^\star \in \mathbb{R}^N$ and need not be equal to x, nor known to the PMW algorithm. The Potential Function is defined as

$$\Phi_t = \text{RE}(x^\star || x_t) = \sum_{i \in V} x^\star[i] \log\left(\frac{x^\star[i]}{x_t[i]}\right).$$

Lemma 3.1. In each update round t, $\Phi_{t-1} - \Phi_t \geq \eta\langle r_t, x_{t-1} - x^\star\rangle - \eta^2$.

Remark. Lemma 3.1 quantifies the potential drop drop in terms of the penalty vector r_t and the parameter η using a multiplicative weights argument.

Definition 3.4. Query Error. $\text{err}(x^\star, f_t) = |\langle f_t, x^\star\rangle - \langle f_t, x\rangle|$. If $x^\star = x$, then $\text{err}(x^\star, f_t) = 0$.

Lemma 3.2. In each round t where $|\hat{d}_t| \geq T$ and $|A_t| \leq \frac{T}{2}$ we have $\langle r_t, x^\star - x_{t-1}\rangle \geq |\langle f_t, x\rangle - \langle f_t, x_{t-1}\rangle| - \text{err}(x^\star, f_t)$.

Remark. Lemma 3.2 connects the inner product $\langle r_t, x^\star - x_{t-1}\rangle$ with the error of x_{t-1} on f_t, measured with respect to the true histogram x. Combining Lemma 3.1 and Lemma 3.2, we arrive at Lemma 3.3, which is necessary to build Lemma 3.4.

Lemma 3.3. In each round t where $|\hat{d}_t| \geq T$ and $A_t \leq \frac{T}{2}$, we have $\Phi_{t-1} - \Phi_t \geq \eta\left(\frac{T}{2} - \text{err}(x^\star, f_t)\right) - \eta^2$

Lemma 3.4. Utility for $V = U$. When the PMW mechanism is run with $V = U$, it is an (α, β, k)-accurate interactive mechanism, where

$$\alpha = \mathcal{O}\left(\frac{\log\left(\frac{1}{\delta}\right) \sqrt[4]{N} \cdot \left(\log k + \log\left(\frac{1}{\beta}\right)\right)}{\varepsilon\sqrt{n}}\right).$$

Remark. Putting the preceding lemmas together completes a proof for Theorem 3.1.

3.1.2 Proving Smooth PMW Performance and Accuracy

We first consider utility in the more general case where $V \subseteq U$.

Lemma 3.5. Let $(f_1, ..., f_k)$ be a sequence of k linear queries. Take $\gamma = \inf_{x^\star} \text{err}(x^\star, f_t)$ where $x^\star$ ranges over all histograms supported on V. When the PMW mechanism is run on V with the query sequence above, and with threshold parameter $T` = T + \gamma$, it is an (α, β, k)-non-adaptively accurate interactive mechanism where

$$\alpha = \mathcal{O}\left(\gamma + \frac{\log\left(\frac{1}{\delta}\right)\sqrt[4]{N} \cdot \left(\log k + \log\left(\frac{1}{\beta}\right)\right)}{\varepsilon\sqrt{n}}\right).$$

Combining Lemma 3.5 with Lemma 3.6 provides the proof for Theorem 3.2.

Lemma 3.6. Let U be a data universe and $\mathcal{C}$ be a collection of linear queries over U. Let x be (ξ, ϕ)-pseudo-smooth with respect to $\mathcal{C}$. Take $\alpha, \beta > 0$, and sample uniformly at random (with replacement) $V \subseteq U$ so that $M = |V| = f \max\left\{\xi N \cdot \frac{\left(\log\left(\frac{1}{\beta}\right) + \log|\mathcal{C}|\right)}{\alpha^2}, \log\left(\frac{1}{\beta}\right)\right\}$. Then, with all but β probability over the choice of V, there exists a histogram $x^\star$ with support only over V such that

$$\forall f \in \mathcal{C} : |f(x) - f(x^\star)| \leq \phi + \alpha.$$

4 The PMW in More Recent DP Research

A brief search of Google Scholar shows that [7] has more than 200 citations and continues to influence ongoing research in the field of Differential Privacy. We look at several more recent papers: [9, 1]

[9] explicitly builds upon the PMW mechanism, extending it to the case of convex minimiztion and showing the capability to give accurate and differentially private solutions to exponentially many convex minimization problems on a sensitive dataset. Unfortunately, the algorithm described in this paper runs in exponential time over the dimension of data and there is probably no polynomial time that takes as input a set of k arbitrary differentiable convex loss functions and outputs answers with even $\frac{1}{100}$ accuracy per query.

[1] demonstrates that the Johnson-Lindenstrauss transform can be efficiently applied as a non-interactive mechanism to DP. Specifically, they apply their mechanism to *cut queries*, which treats a database as a graph and asks how many edges cross a specific cut of the graph. The authors note that the PMW mechanism (modified for cut queries), in contrast to their work, always answers correctly with no multiplicative error and can support k adaptively chosen queries.

References

[1] Jeremiah Blocki, Avrim Blum, Anupam Datta, and Or Sheffet. The johnson-lindenstrauss transform itself preserves differential privacy. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 410–419. IEEE, 2012.

[2] Avrim Blum, Katrina Ligett, and Aaron Roth. A learning theory approach to noninteractive database privacy. *Journal of the ACM (JACM)*, 60(2):12, 2013.

[3] Tore Dalenius. Towards a methodology for statistical disclosure control. *statistik Tidskrift*, 15(429-444):2–1, 1977.

[4] Cynthia Dwork. Differential privacy. In *Proceedings of the 33rd International Conference on Automata, Languages and Programming - Volume Part II*, ICALP'06, pages 1–12, Berlin, Heidelberg, 2006. Springer-Verlag.

[5] Cynthia Dwork. Differential privacy: A survey of results. In *International Conference on Theory and Applications of Models of Computation*, pages 1–19. Springer, 2008.

[6] Cynthia Dwork, Guy N Rothblum, and Salil Vadhan. Boosting and differential privacy. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 51–60. IEEE, 2010.

[7] Moritz Hardt and Guy N Rothblum. A multiplicative weights mechanism for privacy-preserving data analysis. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 61–70. IEEE, 2010.

[8] Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774. ACM, 2010.

[9] Jonathan Ullman. Private multiplicative weights beyond linear queries. In *Proceedings of the 34th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems*, pages 303–312. ACM, 2015.