Multiplicative Weights Algorithms

CompSci 590.03
Instructor: Ashwin Machanavajjhala
This Class

• A Simple Multiplicative Weights Algorithm

• Multiplicative Weights for privately publishing synthetic data
Multiple Experts Problem

Will it rain today?

Yes  Yes  Yes  No

What is the best prediction based on these experts?
Multiple Experts Problem

• Suppose we know the best expert (who makes the least error), then we can just return that expert says.
  – This is the best we can hope for.

• We don’t know who the best expert is.
  – But we can learn ... we know whether it rained or not at the end of the day.

Qn: Is there an algorithm that learns over time who the best expert is, and has an accuracy that close to the best expert?
Weighted Majority Algorithm

[Littlestone&Warmuth ‘94]

Algorithm:

\[ Is \frac{\sum_i w_i y_i}{\sum_i w_i} > \frac{1}{2} ? \]
Weighted Majority Algorithm

1-\(\epsilon\) 1-\(\epsilon\) 1-\(\epsilon\)
\(\times\) \(\times\) \(\times\)

“Experts”

Yes Yes Yes No

Algorithm

\[
\text{Is } \frac{\sum_i w_i y_i}{\sum_i w_i} > \frac{1}{2} ?
\]

Yes!

Truth

No
Multiplicative Weights Algorithm

- Maintain weights (or probability distribution) over experts.

Answering/Prediction:
- Answer using weighted majority, OR
- Randomly pick an expert based on current probability distribution. Use random experts answer.

Update:
- Observe truth.
- Decrease weight (or probability) assigned to the experts who are wrong.
Error Analysis

Theorem:

After $t$ steps,

- let $m(t,j)$ be the number of errors made by expert $j$
- let $m(t)$ be the number of errors made by algorithm
- let $n$ be the number of experts,

$$\forall j, \quad m(t) \leq \frac{2 \ln n}{\varepsilon} + 2(1 + \varepsilon)m(t,j)$$
Error Analysis: Proof

• Let $\varphi(t) = \Sigma w_i$. Then, $\varphi(1) = n$.

• When the algorithm makes a mistake,
  $\varphi(t+1) \leq \varphi(t) \left(\frac{1}{2} + \frac{1}{2}(1-\varepsilon)\right) = \varphi(t)(1-\varepsilon/2)$

• When the algorithm is correct,
  $\varphi(t+1) \leq \varphi(t)$

• Therefore,
  $\varphi(t) \leq n(1-\varepsilon/2)^{m(t)}$
Error Analysis: Proof

- $\varphi(t) \leq n(1-\varepsilon/2)^{m(t)}$

- Also, $W_j(t) = (1-\varepsilon)^{m(t,j)}$

- $\varphi(t) \geq W_j(t) \Rightarrow n(1-\varepsilon/2)^{m(t)} \geq (1-\varepsilon)^{m(t,j)}$

- Hence, $m(t) \geq 2/\varepsilon \ln n + 2(1+\varepsilon)m(t,j)$
Multiplicative Weights

This algorithm technique has been used to solve a number of problems:

- Packing and covering Linear programs (Plotkin-Shmoys-Tardos)
- Log n approximation for many NP-hard problems (set cover ...)
- Boosting
- Zero sum games
- Network congestion
- Semidefinite programs

[Arora, Hazan, Kale ‘05]
This Class

• A Simple Multiplicative Weights Algorithm

• Multiplicative Weights for privately publishing synthetic data
Workload-aware Synthetic Data Generation

Input:

Q, a workload of (expected/typical) linear queries of the form $\Sigma_{x} q(x)$, and each $q(x)$ is in the range $[-1,1]$

D, a database instance

T, number of iterations

$\varepsilon$, differential privacy parameter

Output:

A, a synthetically generated dataset such that for all $q$ in Q, $q(A)$ is close to $q(D)$
Multiplicative Weights Algorithm

- Let $n$ be the number of records in $D$, and $N$ be the number of values in the domain.

Initialization

- Let $A_0$ be a weight function that assigns $n/N$ weight to each value in the domain.
Multiplicative Weights

• Let $n$ be the number of records in $D$, and $N$ be the number of values in the domain.
• Let $A_0$ be a weight function that assigns $n/N$ weight to each value in the domain.

In iteration $j$ in $\{1,2,...,T\}$,

• Pick query $q$ from $Q$ with maximum error
  – $\text{Error} = q(D) - q(A_{i-1})$
Multiplicative Weights

- Let \( n \) be the number of records in \( D \), and \( N \) be the number of values in the domain.
- Let \( A_0 \) be a weight function that assigns \( n/N \) weight to each value in the domain.

In iteration \( j \) in \( \{1, 2, \ldots, T\} \),
- Pick query \( q \) from \( Q \) with maximum error
- Compute \( m = q(D) \)
Multiplicative Weights

- Let $n$ be the number of records in $D$, and $N$ be the number of values in the domain.
- Let $A_0$ be a weight function that assigns $n/N$ weight to each value in the domain.

In iteration $j$ in $\{1, 2, \ldots, T\}$,

- Pick query $q$ from $Q$ with maximum error
- Compute $m = q(D)$
- Update Weights
  - $A_i(x) \propto A_{i-1}(x) \cdot \exp(q(x) \cdot (m - q(A_{i-1}))/2n)$

Output: $\text{average}_i(A_i)$
Update rule

$$A_i(x) \propto A_{i-1}(x) \cdot \exp\left( q(x) \cdot \frac{m - q(A_{i-1})}{2n} \right)$$

If $q(D) - q(A) > 0$,
then increase the weight of records with $q(x) > 0$, and
decrease the weight of records with $q(x) < 0$

If $q(D) - q(A) < 0$,
then decrease the weight of records with $q(x) > 0$, and
increase the weight of records with $q(x) < 0$
Error Analysis

Theorem:
For any database $D$, and any set of linear queries $Q$, MWEM outputs an $A$ such that:

$$\max_{q \in Q} |q(A) - q(D)| \leq 2n \sqrt{\frac{\log |\text{dom}|}{T}}$$
Error Analysis: Proof

\[ \max_{q \in Q} |q(A) - q(D)| = \max_{q \in Q} \left| q \left( \text{avg} A_i \right) - q(D) \right| \]
\[ \leq \text{avg} \max_{i \in Q} |q(A_i) - q(D)| \leq \text{avg} \max_{i \in Q} \text{maxerr}_i \]

Consider the potential function: \( \varphi(i) = \sum_{x \in \text{dom}} \frac{D(x)}{n} \log \left( \frac{D(x)}{A_i(x)} \right) \)

\[ \varphi(i) \geq 0, \text{ and } \varphi(i) \leq \log |\text{dom}| \]
Error Analysis: Proof

Claim: \( \forall i, \varphi(i - 1) - \varphi(i) \geq \left( \frac{q(D) - q(A_i)}{2n} \right)^2 \)

\[
\text{avg maxerr}_i = \frac{1}{T} \sum_{i=1}^{T} |q(D) - q(A_i)|
\]

\[
\leq \frac{1}{T} \sum_{i=1}^{T} \left( 2n \sqrt{\varphi(i - 1) - \varphi(i)} \right)
\]

\[
\leq 2n \sqrt{\frac{1}{T} \sum_{i=1}^{T} (\varphi(i - 1) - \varphi(i))}
\]

\[
= 2n \sqrt{\frac{\varphi(0) - \varphi(T)}{T} = 2n \sqrt{\frac{\log|\text{dom}|}{T}}}
\]
Synthetic Data Generation with Privacy

Input:
- $Q$, a workload of (expected/typical) linear queries of the form $\Sigma_x q(x)$, and each $q(x)$ is in the range $[-1,1]$
- $D$, a database instance
- $T$, number of iterations
- $\epsilon$, differential privacy parameter

Output:
- $A$, a synthetically generated dataset such that for all $q$ in $Q$, $q(A)$ is close to $q(D)$
MWEM

Let \( n \) be the number of records in \( D \), and \( N \) be the number of values in the domain.

Initialization

- Let \( A_0 \) be a weight function that assigns \( n/N \) weight to each value in the domain.
Let \( n \) be the number of records in \( D \), and \( N \) be the number of values in the domain.

Let \( A_0 \) be a weight function that assigns \( n/N \) weight to each value in the domain.

In iteration \( j \) in \( \{1, 2, \ldots, T\} \),

- Pick query \( q \) from \( Q \) with max error using Exponential Mechanism
  - Parameter: \( \varepsilon/2T \)
  - Score function: \( |q(A_{i-1}) - q(D)| \)

More likely to pick those queries for which the answer on the synthetic data is very different from the answer on the true data.
Let \( n \) be the number of records in \( D \), and \( N \) be the number of values in the domain. Let \( A_0 \) be a weight function that assigns \( n/N \) weight to each value in the domain.

In iteration \( j \) in \( \{1, 2, \ldots, T\} \),

1. Pick query \( q \) from \( Q \) with max error using Exponential Mechanism.
2. Compute \( m = q(D) \) using Laplace Mechanism
   - Parameter: \( \varepsilon/2T \)
   - \( m = q(D) + \text{Lap}(2T/\varepsilon) \)
Let $n$ be the number of records in $D$, and $N$ be the number of values in the domain.

Let $A_0$ be a weight function that assigns $n/N$ weight to each value in the domain.

In iteration $j$ in $\{1, 2, \ldots, T\}$,

- Pick query $q$ from $Q$ with max error using Exponential Mechanism
- Compute $m = q(D)$ using Laplace Mechanism
- Update Weights
  \[ A_i(x) \propto A_{i-1}(x) \cdot \exp\left( q(x) \cdot \frac{(m - q(A_{i-1}))}{2n} \right) \]

Output: $\text{average}_i(A_i)$
Update rule

\[ A_i(x) \propto A_{i-1}(x) \cdot \exp(q(x) \cdot (m - q(A_{i-1}))/2n) \]

If noisy \( q(D) - q(A) > 0 \),
then increase the weight of records with \( q(x) > 0 \), and
decrease the weight of records with \( q(x) < 0 \)

If noisy \( q(D) - q(A) < 0 \),
then decrease the weight of records with \( q(x) > 0 \), and
increase the weight of records with \( q(x) < 0 \)
Error Analysis

Theorem:
For any database $D$, and any set of linear queries $Q$, with probability at least $1 - 2T/|Q|$, MWEM outputs an $A$ such that:

$$\max_{q \in Q} |q(A) - q(D)| \leq 2n \sqrt{\frac{\log |\text{dom}|}{T}} + \frac{10T \log |Q|}{\epsilon}$$
Error Analysis: Proof

$$\max_{q \in Q} |q(A) - q(D)| = \max_{q \in Q} \left| q \left( \frac{1}{i} \sum_{i} A_i \right) - q(D) \right| \leq \max_{i} \max_{q \in Q} |q(A_i) - q(D)| \leq \max_{i} \text{maxerr}_i$$

1. But exponential mechanism picks $q_i$, which might not have the maximum error!

$$P(|q_i(A_i) - q_i(D)| < \text{maxerr}_i - r) < |Q| \cdot e^{-\frac{\varepsilon r}{4T}}$$

When $r = \frac{8T \log |Q|}{\varepsilon}$, we get w.p. $1 - \frac{1}{|Q|}$

$$\text{maxerr}_i \leq |q_i(A_i) - q_i(D)| + \frac{8T \log |Q|}{\varepsilon}$$
Error Analysis: Proof

\[
\max_{q \in Q} |q(A) - q(D)| = \max_{q \in Q} \left| q\left( \text{avg}A_i \right) - q(D) \right|
\leq \text{avg} \max_{i, q \in Q} |q(A_i) - q(D)| \leq \text{avg maxerr}_i
\]

1. In each iteration with probability at least \(1 - 1/|Q|\), error in the query picked by exponential mechanism is smaller than max error by at most

\[
8T \frac{\log |Q|}{\varepsilon}
\]

2. **We add noise to \(m = q(D)\).** But with probability at least \(1 - 1/|Q|\) in each iteration, the noise added by Laplace is at most

\[
2T \frac{\log |Q|}{\varepsilon}
\]
Error Analysis

Theorem:
For any database $D$, and any set of linear queries $Q$, with probability at least $1 - \frac{2T}{|Q|}$, MWEM outputs an $A$ such that:

\[
\max_{q \in Q} |q(A) - q(D)| \leq 2n \sqrt{\frac{\log|\text{dom}|}{T}} + \frac{10T \log |Q|}{\varepsilon}
\]
Optimizations

• Output $A_T$ rather than the average
• In update step, use queries picked in all previous rounds for which $(m-q(A))$ is large.
• Can improve the solution by initializing $A_0$ with noisy counts.
Next Class

• Implementations of Differential Privacy
  – How to write programs with differential privacy
  – Security issues due to incorrect implementation
  – How to convert any program to satisfy differential privacy
References

Hardt & Rothblum, “A multiplicative weights mechanism for privacy-preserving data analysis”, FOCS ’10