CIS 800/002 The Algorithmic Foundations of Data Privacy October 13, 2011

Lecture 9

Lecturer: Aaron Roth *Scribe: Aaron Roth*

Database Update Algorithms: Multiplicative Weights

We'll recall (again) some definitions from last time:

Definition 1 (Database Update Sequence) *Let $D \in \mathbf{N}^{|\mathcal{X}|}$ be any database and let $\{(D^t, Q_t, v_t)\}_{t=1,\ldots,L} \in (\mathcal{D} \times \mathcal{C} \times \mathbf{R})^L$ be a sequence of tuples. We say the sequence is an $(U, D, \mathcal{C}, \alpha, L)$-*database update sequence *if it satisfies the following properties:*

1. $D^1 = U(\emptyset, \cdot, \cdot)$,
2. *for every* $t = 1, 2, \ldots, L$, $|Q_t(D) - Q_t(D^t)| \geq \alpha$,
3. *for every* $t = 1, 2, \ldots, L$, $|Q_t(D) - v_t| < \alpha$,
4. *and for every* $t = 1, 2, \ldots, L-1$, $D^{t+1} = U(D^t, Q_t, v_t)$.

Definition 2 (Database Update Algorithm (DUA)) *Let $U : \mathcal{D} \times \mathcal{C} \times \mathbf{R} \to \mathcal{D}$ be an update rule and let $B : \mathbf{R} \to \mathbf{R}$ be a function. We say U is a $B(\alpha)$-*DUA for query class $\mathcal{C}$ *if for every database $D \in \mathbf{N}^{|\mathcal{X}|}$, every $(U, D, \mathcal{C}, \alpha, L)$-database update sequence satisfies $L \leq B(\alpha)$.*

Using the exponential mechanism as a distinguisher, we proved the following utility theorem about the IC mechanism:

Theorem 3 *Given a $B(\alpha)$-DUA, the Iterative Construction mechanism is (α, β) accurate and ϵ-differentially private for:*

$$\alpha \geq \frac{8B(\alpha/2)}{n\epsilon}\left(\log \frac{\mathcal{C}}{\gamma}\right)$$

and (ϵ, δ)-differentially private for:

$$\alpha \geq \frac{16\sqrt{B(\alpha/2)\log(1/\delta)}}{n\epsilon}\left(\log \frac{\mathcal{C}}{\gamma}\right)$$

so long as $\gamma \leq \beta/(2B(\alpha/2))$.

We plugged in the Median Mechanism, based on the existence of small nets, to get:
Plugging this in to the IC mechanism, we get:

Theorem 4 *Instantiated with the median mechanim, the Iterative Construction mechanism is (α, β) accurate and ϵ-differentially private for:*

$$\alpha \geq \frac{32 \log |\mathcal{X}| \log |\mathcal{C}|}{n\epsilon\alpha^2}\left(\log \frac{\mathcal{C}}{\gamma}\right)$$

$$\alpha = \tilde{O}\left(\frac{\log |\mathcal{X}| \log^2 |\mathcal{C}| \log(1/\beta)}{\epsilon n}\right)^{1/3}$$

and (ϵ, δ)-differentially private for:

$$\alpha \geq \frac{32\sqrt{\log |\mathcal{X}| \log |\mathcal{C}| \log(1/\delta)}}{n\epsilon\alpha}\left(\log \frac{\mathcal{C}}{\gamma}\right)$$

$$\alpha \geq \tilde{O}\left(\frac{(\log |\mathcal{X}| \log^3 |\mathcal{C}| \log(1/\delta) \log^2(1/\beta))^{1/4}}{\sqrt{\epsilon n}}\right)$$

We now give a more sophisticated database update algorithm for linear queries. It will work by maintaining a distribution $\hat{D}^t$ over the data universe X. A linear query is a natural generalization of a counting query, which we considered earlier.

Although this new mechanism will only apply to linear queries (The median mechanism worked for generic classes of queries), it will have significantly improved running time, and (slightly) improved accuracy.

Definition 5 *A linear query is a vector $Q \in [0,1]^{|X|}$, evaluated as $Q(D) = \frac{1}{n}\langle Q, D\rangle$. Equivalently, we can view Q as a function $Q : X \to [0,1]$, and evaluate:*

$$Q(D) = \frac{\sum_{x_i \in D} Q(x_i)}{n} = \frac{\sum_{i=1}^{|X|} Q(x_i) \cdot D[i]}{n}$$

Algorithm 1 The Multiplicative Weights (MW) Algorithm. It is instantiated with a parameter $\eta \leq 1$.

MW(D^t, Q_t, v_t):
  if $D^t = \emptyset$ **then**
    Output: $D^1 \in \mathbf{N}^{|\mathcal{X}|} : D^0_i = \frac{1}{|\mathcal{X}|}$ for all $x_i \in \mathcal{X}$.
  end if
  if $v_t < Q_t(D^t)$ **then**
    Let $r_t = Q_t$
  else
    Let $r_t = 1 - Q_t$ (i.e. for all i, $r_t(i) = 1 - Q_t(i)$)
  end if
  Update: For all $i \in [|\mathcal{X}|]$ Let

$$\hat{D}^{t+1}_i = \exp(-\eta r_t(x_i)) \cdot D^t_i$$

$$D^{t+1}_i = \frac{\hat{D}^{t+1}_i}{\sum_{j=1}^{|\mathcal{X}|} \hat{D}^{t+1}_j}$$

  Output D^{t+1}.

Lets think about what the MW algorithm is trying to do. Recall that the median mechanism attempted to maintain a "distribution" over databases consistent with queries seen so far. The MW mechanism, on the other hand, is maintaining an explicit probability distribution over the *data universe*. This will turn out to be sufficient for answering *linear* queries, and as a result, the algorithm will be more efficient.

Why a probability distribution? It turns out that for linear queries, we can think of databases as equivalent to distributions over the data universe. Recall that for a database $D \in \mathbf{N}^{|\mathcal{X}|}$, and a linear query $Q \in [0,1]^{|\mathcal{X}|}$, we defined $Q(D) = \frac{1}{n}\langle Q, D\rangle$, where $n = ||D||_1$. Suppose we consider a normalized version of our database, $\hat{D} \in \mathbf{R}^{|X|}$, where $\hat{D}[i] = D[i]/n$. Note that we have $\sum_{i=1}^{|X|} \hat{D}[i] = 1$: i.e. $\hat{D}$ is a probability distribution over X. We also have

$$Q(\hat{D}) = \langle Q, \hat{D}\rangle = \frac{1}{n}\langle Q, D\rangle = Q(D)$$

i.e. normalizing D to be a probability distribution does not change the value of any linear query. We may therefore without loss of generality reason about D as if it is a probability distribution. The MW algorithm seeks to learn the probability distribution D, as it is reflected in the answers to a set of linear queries.

The strategy to analyze the MW algorithm will be to keep track of a potential function Ψ measuring the similarity between the hypothesis database D^t at time t, and the true database D. We will show:

1. The potential function does not start out too large.

2. The potential function decreases by a significant amount at each update round.

3. The potential function is always non-negative.

Together, these 3 facts will force us to conclude that there cannot be too many update rounds!

Let us now begin the analysis:

Theorem 6 *Letting parameter $\eta = \alpha/2$, the Multiplicative Weights algorithm is a $B(\alpha)$-database update algorithm for $B(\alpha) = \frac{4\log|\mathcal{X}|}{\alpha^2}$ for every class of linear queries $\mathcal{C}$.*

Proof We must show that any sequence $\{(D^t, Q_t, v_t)\}_{t=1,\ldots,L}$ with the property that $|Q^t(\mathbf{D}^t) - Q^t(D)| > \alpha$ and $|v_t - Q^T(D)| < \alpha$ cannot have $L > \frac{4\log|\mathcal{X}|}{\alpha^2}$.

We define our potential function as follows. Recall that we here view the database as a probability distribution – i.e. we assume $||D||_1 = 1$. Of course this does not require actually modifying the real database. The potential function that we use is the relative entropy, or KL divergence, between D and D^t.

$$\Psi_t \stackrel{\text{def}}{=} D(D||D^t) = \sum_{i=1}^{|\mathcal{X}|} D[i]\log\left(\frac{D[i]}{D^t[i]}\right)$$

We begin with a simple fact:

Proposition 7 *For all t: $\Psi_t \geq 0$, and $\Psi_1 \leq \log|\mathcal{X}|$.*

Proof Relative entropy (KL-Divergence) is always a non-negative quantity, by the log-sum inequality. To see that $\Psi_1 \leq \log|\mathcal{X}|$, recall that $D^1[i] = 1/|\mathcal{X}|$ for all i, and so $\Psi_1 = \sum_{i=1}^{|\mathcal{X}|} D[i]\log(|\mathcal{X}|D[i])$. Noting that D is a probability distribution, we see that this quantity is maximized when $D[1] = 1$ and $D[i] = 0$ for all $i > 1$, giving $\Psi_i = \log|\mathcal{X}|$. ■

We will now argue that at each step, the potential function drops by at least $\alpha^2/4$. Because the potential begins at $\log|\mathcal{X}|$, and must always be non-negative, we therefore know that there can be at most $L \leq 4\log|X|/\alpha^2$ steps in the database update sequence. To begin, let us see exactly how much the potential drops at each step:

Lemma 8

$$\Psi_t - \Psi_{t+1} \geq \eta\left(r_t(D^t) - r_t(D)\right) - \eta^2$$

Proof

$$\begin{aligned}
\Psi_t - \Psi_{t+1} &= \sum_{i=1}^{|\mathcal{X}|} D[i]\log\left(\frac{D[i]}{D_i^t}\right) - \sum_{i=1}^{|\mathcal{X}|} D[i]\log\left(\frac{D[i]}{D_i^{t+1}}\right) \\
&= \sum_{i=1}^{|\mathcal{X}|} D[i]\log\left(\frac{D_i^{t+1}}{D_i^t}\right) \\
&= \sum_{i=1}^{|\mathcal{X}|} D[i]\log\left(\frac{D_i^t\exp(-\eta r_t(x_i))}{D_i^t}\right) - \log\left(\sum_{i=1}^{|\mathcal{X}|}\exp(-\eta r_t(x_i))D_i^t\right) \\
&= -\sum_{i=1}^{|\mathcal{X}|} D[i]\eta r_t(x_i) - \log\left(\sum_{i=1}^{|\mathcal{X}|}\exp(-\eta r_t(x_i))D_i^t\right)
\end{aligned}$$

$$
\begin{aligned}
&= -\eta r_t(D) - \log\left(\sum_{i=1}^{|\mathcal{X}|} \exp(-\eta r_t(x_i)) D_i^t\right) \\
&\geq -\eta r_t(D) - \log\left(\sum_{i=1}^{|\mathcal{X}|} D_i^t (1 + \eta^2 - \eta r_t(x_i))\right) \\
&= -\eta r_t(D) - \log\left(1 + \eta^2 - \eta r_t(D^t)\right) \\
&\geq \eta\left(r_t(D^t) - r_t(D)\right) - \eta^2
\end{aligned}
$$

The first inequality follows from the fact that:

$$\exp(-\eta r_t(x_i)) \leq 1 - \eta r_t(x_i) + \eta^2 r_t(x_i)^2 \leq 1 - \eta r_t(x_i) + \eta^2$$

The second inequality follows from the fact that $\log(1+y) \leq y$ for $y > -1$. ■

The rest of the proof now follows easily. By the conditions of a database update algorithm, $|v_t - Q_t(D)| < \alpha$. Hence, because for each t: $|Q_t(D) - Q_t(D^t)| \geq \alpha$, we also have that $Q_t(D) > Q_t(D^t)$ if and only if $v_t > Q_t(D^t)$. In particular, $r_t = Q_t$ if $Q_t(D^t) - Q_t(D) \geq \alpha$, and $r_t = 1 - Q_t$ if $Q_t(D) - Q_t(D^t) \geq \alpha$. Therefore, by Lemma 8 and the fact that $\eta = \alpha/2$:

$$\Psi_t - \Psi_{t+1} \geq \frac{\alpha}{2}\left(r_t(D^t) - r_t(D)\right) - \frac{\alpha^2}{4} \geq \frac{\alpha}{2}(\alpha) - \frac{\alpha^2}{4} = \frac{\alpha^2}{4}$$

Finally we know:

$$0 \leq \Psi_L \leq \Psi_0 - L \cdot \frac{\alpha^2}{4} \leq \log|\mathcal{X}| - L\frac{\alpha^2}{4}$$

Solving, we find: $L \leq \frac{4\log|\mathcal{X}|}{\alpha^2}$ This completes the proof. ■

Finally, we can see what bounds we get by plugging in the multiplicative weights DUA into the IC algorithm:

Theorem 9 *Combining the multiplicative weights DUA and the exponential mechanism distinguisher, the IC algorithm is (α, β)-accurate and ϵ-differentially private for:*

$$\alpha = \tilde{O}\left(\left(\frac{\log|X| \log\frac{|C|}{\beta}}{n\epsilon}\right)^{1/3}\right)$$

and (ϵ, δ)-differentially private for:

$$\alpha = \tilde{O}\left(\frac{(\log|X| \log 1/\delta)^{1/4} (\log\frac{|C|}{\beta})^{1/2}}{\sqrt{\epsilon n}}\right)$$

Lets conclude by appreciating the magic that just happened. Unlike the median mechanism or the net mechanism, the multiplicative weights mechanism did not start with any baked in information about the class of queries it was going to answer (such as in the form of a net). In fact, the existence of the multiplicative weights mechanism gives another, unrelated proof that linear queries have small nets! Recall that we already proved via sampling arguments that any set of linear queries C has a net of size $|X|^{\log|C|/\alpha^2}$, by arguing that for every database, there is another database of size only $\log|C|/\alpha^2$ that agrees with it (up to $\pm\alpha$) on any set of linear queries C.

What has the multiplicative weights mechanism shown? It has shown that for any set of C linear queries, we can represent all of the answers (up to $\pm\alpha$) by a sequence of queries from $|C|$ forming a

database update sequence of length $4 \log |X|/\alpha^2$. How many such sequences of queries are there from $|C|$? Exactly $|C|^{4 \log |X|/\alpha^2}$. But this is exactly equal to $|X|^{4 \log |C|/\alpha^2}$ – that is, the MW mechanism proves the existence of the same size net for linear queries! This net is "dual" to the one we already demonstrated: rather than being a collection of databases, it is a collection of query sequences! Yet the net is the same size.

Bibliographic Information The Multiplicative Weights Mechanism was given by Hardt and Rothblum, in "A Multiplicative Weights Mechanism for Privacy Preserving Data Analysis", 2010.