CIS 800/002 The Algorithmic Foundations of Data Privacy October 25, 2011

Lecture 12

Lecturer: Aaron Roth *Scribe: Aaron Roth*

Interactive Query Release Mechanisms

In this lecture we will revisit the query release problem, and show how the iterative construction mechanism can be adapted to work in the interactive setting, in which the data analyst may choose his queries adaptively, and the mechanism must provide the answers to each query as they arrive, before seeing the next query. Recall that the simple Laplace mechanism worked in this setting, but that because it added independent noise to each query, its privacy guarantee degraded linearly with the number of queries asked. We have since seen mechanisms with privacy guarantees that degrade only logarithmically in the number of queries asked, but they have the advantage of knowing the identities of all of the queries that they must answer ahead of time, and use this information to carefully calibrate (and correlate) the noise that they add. A-priori, it is not clear whether in the *interactive* setting, facing an adversary who can adaptively choose his queries from a huge class, whether anything better than the Laplace mechanism should be possible. But indeed, as we will see, we can recover the strongest bounds that we proved in the non-interactive setting here again in the interactive setting.

Recall the sparse vector technique which we say last time, which answers adaptively chosen streams of low-sensitivity queries by comparing them to a threshold. It answers the queries if their (noisy) answers fall above the threshold, and otherwise simply reports $\perp$ for "below threshold". The magic of the analysis was that the privacy degrades only linearly in the number of above threshold queries, independently of the total number of queries asked. It automatically halts after answering c queries for some given parameter c so that its worst-case privacy loss can be bounded. We proved the following guarantees:

Definition 1 (Accuracy) *We will say that Sparse is (α, β)-accurate for a sequence of k queries $Q_1, \ldots, Q_k$, if except with probability at most β, the algorithm does not abort before Q_k, and for all $a_i \in \mathbf{R}$:*

$$|a_i - Q_i(D)| \leq \alpha$$

and if for all $a_i = \perp$:

$$Q_i(D) \leq T + \alpha$$

.

Theorem 2 *For any sequence of k queries $Q_1, \ldots, Q_k$ such that $L(T) \equiv |\{i : Q_i(D) \geq T - \alpha\}| \leq c$, Sparse$(T, C)$ is (α, β) accurate for:*

$$\alpha = 2\sigma\left(\log k + \log\frac{2}{\beta}\right) = \frac{4c(\log k + \log(2/\beta))}{\epsilon n}.$$

Theorem 3 *The sparse vector algorithm is ϵ-differentially private.*

We'll also recall our definition of a distinguisher, which we used as a subroutine in the iterative construction mechanism[1]

Definition 4 *An $(F(\epsilon), \gamma)$ private distinguisher with respect to a class of queries $\mathcal{C}$ is an ϵ-differentially private algorithm $A : \mathbf{N}^{|\mathcal{X}|} \times \mathbf{N}^{|\mathcal{X}|} \to \mathcal{C}$ with the following property. For any pair $D, D' \in \mathbf{N}^{|\mathcal{X}|}$ such that there exists a $Q^* \in \mathcal{C}$ such that $|Q^*(D) - Q^*(D')| \geq 3F(\epsilon)$, $A(D, D') = Q$ such that $|Q(D) - Q(D')| \geq F(\epsilon)$, except with probability at most γ.*

Observe that we can use the sparse vector algorithm as a distinguisher!

[1] Actually, this is a different definition... But it works just as well.

SparseDistinguish(D, D')

Let $c \leftarrow 1$, $\alpha \leftarrow \frac{4c(\log|\mathcal{C}|+\log(2/\gamma))}{\epsilon n}$, $T \leftarrow 2\alpha$.
Initialize Sparse(T, c, ϵ) with a stream of queries Q_i on database D, outputting a stream of answers a_i.
for each query $Q \in \mathcal{C}$ **do**
  Let $Q_i(\cdot) \leftarrow |Q(\cdot) - Q(D)|$ (i.e. for any D', $Q_i(D') = |Q(D') - Q(D)|$).
  if $a_i \neq \perp$ **then**
    Halt and Output Q
  end if
end for

Theorem 5 *SparseDistinguish is an $(F(\epsilon), \gamma)$-private distinguisher for $\mathcal{C}$ with:*

$$F(\epsilon) = \frac{4(\log|\mathcal{C}| + \log(2/\gamma))}{\epsilon n}$$

Proof This follows directly from the utility and privacy theorems for the sparse vector algorithm. The mechanism accesses the data only though the sparse vector algorithm, and so it is ϵ-differentially private. We set $T = 2\alpha = 2F(\epsilon)$ By the utility theorem, Sparse is (α, τ) accurate, and so except with probability at most τ, if there exists a query Q with

$$|Q(D') - Q(D)| \geq T + \alpha = 3\alpha = 3F(\epsilon)$$

then we must have $a_i \neq \perp$. But again by the utility theorem, in this case we have: $|Q(D') - Q(D)| \geq T - \alpha = \alpha = F(\epsilon)$ which proves the theorem. ■

Note that this is essentially the same bound we got by using the exponential mechanism as a distinguisher. Of course it is not clear why we have gained: running this distinguisher takes almost as long: we still need to enumerate and evaluate all queries in $\mathcal{C}$.

But note that we have some freedom in how to implement SparseDistinguish: in particular, we can enumerate through the queries in $\mathcal{C}$ in arbitrary, even adversarial order. i.e. even in an order chosen by a data analyst who wants to know the answers to the queries in $\mathcal{C}$ evaluated on D.

In this case, we are in a win-win situation. Recall – why did we want a distinguisher in the first place? It was to find a distinguishing query so that we could run a database update algorithm for another step. Now, we can trick the data analyst into doing the work of the distinguisher for us. Suppose we have a database update algorithm that has produced a current hypothesis D'. If the data analyst has asked a query Q that fails to distinguish between D' and D, this is a good case! We can simply answer the analysts question with $Q(D')$, which we know must be a good approximation to $Q(D)$, for otherwise the query would have distinguished the two databases! Moreover, answering with $Q(D')$ does not incur any further privacy cost, and is quickly evaluatable. On the other hand, if the data analyst has asked a query that does distinguish between D and D', we can run our database update algorithm another step to generate our next hypothesis, all without ever needing to make a call to a 'real' distinguisher. In fact, by offloading the task of distinguishing onto the data analyst, the running time per query of our algorithm will now depend exclusively on the running time of the database update algorithm.

Theorem 6 *OnlineIC is ϵ-differentially private.*

Proof This follows because it accesses the data only through Sparse(T, c, ϵ). ■

Theorem 7 *Together with a $B(\alpha)$-Database Update Algorithm, OnlineIC is $(3\alpha, \beta)$-useful for α such that:*

$$\alpha = \frac{4B(\alpha)(\log(2|\mathcal{C}|) + \log(2/\beta))}{\epsilon n}$$

Algorithm 1 The Online Iterative Construction (IC) mechanism takes as input a private database D, a privacy parameter ϵ, accuracy parameters α and β, a $B(\alpha)$- database update algorithm U for a class of queries $\mathcal{C}$, and a stream of queries $\{Q_i\}$ that may be chosen adaptively from $\mathcal{C}$. It outputs a stream of answers $\{a_i\}$.

OnlineIC$(D, \{Q_i\}, U, \epsilon, \alpha, \beta)$

Let $c \leftarrow B(\alpha)$, $T \leftarrow \frac{8c(\log(2|\mathcal{C}|)+\log(2/\beta))}{\epsilon n}$
Initialize Sparse(T, c, ϵ) with a stream of queries $\{Q'_i\}$, outputting a stream of answers a'_i.
Let $t \leftarrow 0$, $D^t \leftarrow U(\emptyset, \cdot, \cdot)$.
for each query Q_i **do**
  Let $Q'_{2i-1}(\cdot) = Q_i(\cdot) - Q_i(D^t)$.
  Let $Q'_{2i}(\cdot) = Q_i(D^t) - Q_i(\cdot)$
  if $a'_{2i-1} = \perp$ and $a'_{2i} = \perp$ **then**
    Let $a_i = Q_i(D^t)$
  else
    if $a'_{2i-1} \in \mathbf{R}$ **then**
      Let $a_i = Q_i(D^t) + a'_{2i-1}$
    else
      Let $a_i = Q_i(D^t) - a'_{2i}$
    end if
    Let $D^{t+1} = U(D^t, Q_i, a_i)$
    Let $t \leftarrow t + 1$.
  end if
end for

Proof First let us assume that the sparse vector algorithm does not halt prematurely. In this case, by the utility theorem, except with probability at most β, we have for all i such that $a_i = Q_i(D^t)$: $|Q_i(D) - Q_i(D^t)| \leq T + \alpha = 3\alpha$, as we wanted. Additionally, for all i such that $a_i = a'_{2i-1}$ or $a_i = a'_{2i}$, we have $|Q_i(D) - a'_i| \leq \alpha$.

Note that we also have for all i such that $a_i = a'_{2i-1}$ or $a_i = a'_{2i}$: $|Q_i(D) - Q_i(D')| \geq T - \alpha = \alpha$. Therefore, (D^t, Q_i, a_i) forms a valid step in a database update sequence. By the properties of a $B(\alpha)$-database update algorithm, there can be at most $B(\alpha) = c$ such update steps, and so the Sparse vector algorithm does not halt prematurely. ■

Recalling that the multiplicative weights algorithm is a $B(\alpha)$-Database update algorithm for linear queries where $B(\alpha) = 4\log|\mathcal{X}|/\alpha^2$ and plugging this in to the above theorem, we get:

Theorem 8 *The Online Iterative Construction mechanism together with the Multiplicative Weights DUA is ϵ-differentially private and (α, β)-useful for:*

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

Note that we could have repeated this analysis instead using our utility theorem for the (ϵ, δ)-private version of the sparse vector algorithm. This would have resulted in the bound:

Theorem 9 *The Online Iterative Construction mechanism together with the Multiplicative Weights DUA is (ϵ, δ)-differentially private and (α, β)-useful for:*

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

Note that these are the same bounds that we derived for the non-interactive version of the interactive construction mechanism, and that we now no longer need a distinguisher! Indeed, the running time per query for queries that do not require running a database update step is trivial, and the running time for all other queries depends only on the running time of the database update algorithm. So coming up with efficient database update algorithms for a class of queries $\mathcal{C}$ automatically yields efficient private query release mechanisms for $\mathcal{C}$ in the interactive setting!

Bibliographic Information The first interactive mechanism taking this form was given by Roth and Roughgarden in "Interactive Privacy via the Median Mechanism", 2010. Hardt and Rothblum gave a mechanism in this framework based on multiplicative weights with both improved accuracy and (greatly) improved running time in "A Multiplicative Weights Mechanism for Privacy Preserving Data Analysis", 2010. The framework was generalized into its current form in Gupta, Roth, and Ullman, "Iterative Constructions and Private Data Release", 2011.