CIS 800/002 The Algorithmic Foundations of Data Privacy November 17, 2011

Lecture 19

Lecturer: Aaron Roth *Scribe: Aaron Roth*

Streaming Algorithms: Continual Release

In this lecture we consider a different problem in the context of private streaming algorithms. Consider a stream $\sigma \in X^T$ where $X \in \{0,1\}$: i.e. each element σ_i of the stream is a bit. A natural problem is to privately count the number of 1s in the stream, and to maintain a running count: i.e. to be able to output a count at every time step of the prefix of the stream seen so far.

Definition 1 *A continual observation mechanism on a stream $\sigma \in X^T$ is a randomized mapping $M(\sigma) : \mathcal{N} \to \mathbf{R}$ such that $M(\sigma)(t)$ is independent of σ_i for all $i > t$.*

We are interested in estimating the running count of a stream $\sigma \in \{0,1\}^T$. We write:

$$c_\sigma(t) = \sum_{i=1}^{t} \sigma_i$$

and define accuracy with respect to a query c_σ:

Definition 2 *A continual observation mechanism $M(\sigma)$ is $(\alpha(t), \beta)$ accurate for a query $c_\sigma : \mathcal{N} \to \mathbf{R}$ if except with probability at most β we have for all $t \in [T]$:*

$$|c_\sigma(t) - M(\sigma)(t)| \leq \alpha(t)$$

Note that we will be interested here in event-level privacy: privacy with respect to changing just a single element of the stream. Since in this case $X = \{0,1\}$, it would not make sense to try and report a count privately if we considered two streams which differed in an arbitrary number of 1s to be neighboring...

Let us warm up by considering a couple of simple mechanisms that we might try to solve this problem. To analyze these, we'll use a simple analogue of the Chernoff bound for sums of Laplace random variables:

Theorem 3 *Suppose $Y_1, \ldots, Y_k \sim Lap(1/\epsilon)$. Let $Y = \sum_{i=1}^{k} Y_i$. Then:*

$$\Pr[|Y| \geq t] \leq \exp\left(\frac{-t^2\epsilon^2}{6k}\right)$$

In particular:

$$\Pr\left[|Y| \geq \frac{\sqrt{6k \log \frac{1}{\beta}}}{\epsilon}\right] \leq \beta$$

Algorithm1(ϵ)
  Let $\epsilon' \leftarrow \epsilon/T$
  Let $c_0 \leftarrow 0$
  for $t = 1$ to T **do**
    Let $c_t \leftarrow c_{t-1} + \sigma_t$, $\nu_t \leftarrow \text{Lap}(1/\epsilon')$
    Output $c_t + \nu_t$
  end for

Ok, this algorithm sucks. We are adding just enough noise to guarantee ϵ-differential privacy: each entry σ_i appears in $\leq T$ counts $c_1, \ldots, c_T$, and so we add noise $\text{Lap}(T/\epsilon)$ to each count (σ_1 actually

Algorithm2(ϵ)
  Let $c_0 \leftarrow 0$
  for $t = 1$ to T **do**
    Let $\nu_t \leftarrow \text{Lap}(1/\epsilon)$, $\hat{\sigma}_t \leftarrow \sigma_t + \nu_t$, $c_t \leftarrow c_{t-1} + \hat{\sigma}_t$,
    Output c_t
  end for

appears in all T counts, so we can't add less noise). The count c_t is never larger than T, but we are adding noise $\text{Lap}(T/\epsilon)$ at every step! We don't get non-trivial error.

Here is another algorithm that sucks less:

This algorithm is a little better. Note that it still preserves ϵ-differential privacy: each entry σ_i appears only once, in $\hat{\sigma}_i$, and we add noise $\text{Lap}(1/\epsilon)$ to this. Moreover, we have for all t:

$$c_t = \sum_{i=1}^{t} \sigma_i + \sum_{i=1}^{t} \nu_i$$

So the error of this mechanism at each step t is simply $E_t = |\sum_{i=1}^{t} \nu_i|$. By our theorem above, except with probability β, we have for all t:

$$|E_t| \leq O\left(\frac{\sqrt{t}\log\frac{t}{\beta}}{\epsilon}\right)$$

This is already non-trivial error! Can we do better? Lets examine the sources of error in the previous two algorithms. Both of these algorithms work by computing partial sums:

Definition 4 *A P-sum is a partial sum of consecutive items. Write:*

$$\Sigma[i,j] = \sum_{t=i}^{j} \sigma_t$$

We can think of both of the algorithms that we have seen as simply releasing a collection of p-sums. Algorithm 1 releases T noisy p-sums $\hat{\Sigma}[1,t]$ for each $t \in [T]$, simply computes $M(\sigma)(t) = \hat{\Sigma}[i,t]$. Algorithm 2 releases T noisy p-sums $\hat{\Sigma}[i,i]$ for $i \in [T]$ and computes $M(\sigma)(t) = \sum_{i=1}^{t} \hat{\Sigma}[i,i]$.

Suppose an algorithm releases a collection of p-sums such that a single element in the stream can appear in at most k of the p-sums. Then the sensitivity of the output is k, and to preserve privacy, each p-sum must be perturbed with noise $\text{Lap}(k/\epsilon)$. Suppose further that each answer $M(\sigma)(t)$ is the sum of ℓ of these noisy p-sums. Then the error term is at most:

$$|E_t| = |M(\sigma)(t) - c_\sigma(t)| = |\sum_{i=1}^{\ell} \text{Lap}(k/\epsilon)| \leq O\left(k\frac{\sqrt{\ell}\log\frac{t}{\beta}}{\epsilon}\right)$$

except with probability β.

Indeed, Algorithm 1 had $k = T$ and $\ell = 1$, and Algorithm 2 had $k = 1$ and $\ell = t$. So, to develop an algorithm with lower error, we can simply try and develop a way of releasing a count by combining partial sums that has a better tradeoff between k and ℓ. This is what the binary mechanism does:

To analyze this algorithm in the p-sum framework, first note that by design, every output is the sum of at most $\ell = \{i : b_i(t) = 1\} \leq \log T$ p-sums. Moreover, each σ_t is a member of at most a single p-sum of length 2^i for each i, and so is a member of at most $k = \log T$ p-sums. Hence, the algorithm preserves differential privacy, and moveover, except with probability β, we have at each time step the error is at most:

$$|E_t| = O\left(k\frac{\sqrt{\ell}\log\frac{t}{\beta}}{\epsilon}\right) = O\left(\frac{\log T\sqrt{\log t}\log\frac{t}{\beta}}{\epsilon}\right)$$

BinaryCount(ϵ, T)

Let $\epsilon' \leftarrow \epsilon / \log T$.
for $t = 1$ to T **do**
  Express t in binary: $t = \sum_{j=1}^{\log T} b_j(t) \cdot 2^j$
  Let $i \leftarrow \min_j : b_j(t) \neq 0$ be the least non-zero bit of t.
  Let $a_i \leftarrow \sum_{j<i} a_j + \sigma_t$ $(a_i = \Sigma[t - 2^i + 1, t])$
  For $j < i$ Let $a_j \leftarrow 0$, $\hat{a}_j \leftarrow 0$.
  Let $\hat{a}_i \leftarrow a_i + \text{Lap}(1/\epsilon')$.
  Output $M(\sigma)(t) = \sum_{i:b_i(t)=1} \hat{a}_i$
end for

Note also that there are never more than $\log T$ sums "active" at any one time, and so the algorithm can be implemented using only $\log^2 T$ space. Modifications are also possible to remove the need to know T ahead of time...

Bibliographic Information The content and presentation of this lecture is from Chan, Shi, and Song, "Private and Continual Release of Statistics", 2010. Dwork, Naor, Pitassi, and Rothblum also obtain these results in the same model, together with matching lower bounds in "Differential Privacy under Continual Observation", 2010.