CIS 800/002 The Algorithmic Foundations of Data Privacy November 15, 2011

Lecture 18

Lecturer: Aaron Roth *Scribe: Aaron Roth*

Streaming Algorithms: User Level Pan Privacy

Suppose we want to compute some statistic on a gigantic *stream* of data, that we get to see one element at a time. Maybe this stream represents the names of users who type in a specific search query on Google. In this setting, the stream $\sigma = \sigma_1, \ldots, \sigma_k, \ldots$ where each $\sigma_i \in X$ represents the name of a single individual, among a set of $|X|$ possible individuals. Maybe we want to estimate the density of the stream: what fraction of elements from X appear in σ at least once? A couple of issues arise:

1. The stream might be much to large to store in memory, so we want to do this with memory much less than the length of the stream.
2. We want to protect the privacy of any individual in the stream, even if they appear in the stream many times.
3. Maybe we even want to offer privacy guarantees if someone hacks into our servers and gets to observe the internal state of the algorithm...

Lets think about these issues one at a time. The first issue is relatively straightforward, and is a constraint on the algorithm (common in streaming settings) independent of privacy.

The second issue relates to how we define neighboring streams. Recall that differential privacy is defined with respect to a neighbor relation:

Definition 1 *A streaming algorithm $A : X^* \to R$ is ϵ-differentially private if for all pairs of* neighboring *streams $\sigma, \sigma' \in X^*$, and all events $S \subseteq R$:*

$$\Pr[M(\sigma) \in S] \leq \exp(\epsilon) \Pr[M(\sigma') \in S]$$

If we want to protect *user level* privacy (i.e. protecting whether or not a user *appears* in the stream, independently of how many times he appears), then we can define two streams to be neighbors if they differ in any number of occurrences of a single user $x \in X$:

Definition 2 *Write σ^{-x} to denote the stream that results from σ after all instances of x have been removed. Two streams $\sigma, \sigma' \in X^*$ are user-level neighbors if there exists some $x \in X$ such that $\sigma^{-x} = \sigma'^{-x}$.*

The third issue relates to how we define the output of a mechanism $M(\sigma)$. A mechanism has a set of internal states S, a function Update : $S \times X \to S$ which updates its internal state given its last state, and the next element of the stream, and a function Output : $S \to R$. Normally we think of the output of the mechanism $M(\sigma)$ as simply being the result of Output(s) where s is the final state of the mechanism. For a stream prefix $\sigma^{\leq i}$ we can also write Update$(\sigma^{\leq i})$ and Output$(\sigma^{\leq i})$ to denote the mapping to states and outputs given by running update on each element of the stream, and then running output on the final element. We might want *pan-privacy* with respect to a single un-announced intrusion.

Definition 3 *For a mechanism $M : X^* \to R$ with internal states from set S, write $M^i : X \to S \times R$ for the mechanism with output $M^i(\sigma) = (M(\sigma), Update(\sigma^{\leq i}))$ that also outputs the state of the mechanism at time i. M is ϵ-pan-private with respect to a single intrusion of M^i is ϵ-differentially private for all i.*

The idea is that M should be differentially private even if an adversary unexpectedly gets access to the state of the mechanism at any (single) time. We could similarly define pan privacy with respect to multiple intrusions...

We will see how to solve the density estimation problem with a pan-private streaming algorithm guaranteeing user level privacy. Recall our initial example: a stream consisting of the names of users from some universe X, who search on Google/Bing/Yahoo using a particular search term. We wish to estimate the fraction of users who appear in the stream.

Definition 4 *The density of a stream $\sigma \in X^*$ is*

$$d(\sigma) = \frac{1}{|X|}\{x : \exists\ i : \sigma_i = x\}$$

We will give a user-level pan-private algorithm for estimating the density of a stream σ. We will sample bits from two distributions over bits $\{0,1\}$. $\mathcal{D}_0$ is the distribution such that $\Pr[1] = \frac{1}{2}$. $\mathcal{D}_1(\epsilon)$ is the distribution such that $\Pr[1] = \frac{1}{2} + \frac{\epsilon}{4}$.

Density$(\epsilon, \alpha, \beta)$

Let $m = \frac{200 \log 1/\beta}{\epsilon^2 \alpha^2}$.

Sample a set of m representatives M of elements $x \in X$ and construct a table of size m. For each $x \in M$, generate a value $b_x \sim D_0$.

for each i **do**

If $\sigma_i \in M$ then let $b_{\sigma_i} \sim D_1$.

end for

Compute $\theta = \frac{1}{m}\sum_{x \in M} b_x$

Output $\hat{\theta} = \frac{4(\theta - \frac{1}{2})}{\epsilon} + \text{Lap}\left(\frac{1}{\epsilon m}\right)$

The key insights are:

1. The algorithm provides user level privacy because if x has not appeared in the stream then $b_x \sim \mathcal{D}_0$, and otherwise $b_x \sim \mathcal{D}_1$, no matter how many times x has appeared.

2. The algorithm provides pan-privacy because D_0 and D_1 are two distributions that are "differentially private" and no other information other than a sample from these distributions is stored about each x. $\hat{\theta}$ is then re-randomized with Laplace noise to provide privacy even if the adversary has seen the state of the mechanism at some previous time step.

Theorem 5 *Density preserves user level 2ϵ-pan privacy.*

Proof Let σ, σ' be x-adjacent. For $x \notin M$, no information about x is stored, and perfect privacy is guaranteed. For $x \in M$:

$$\frac{\Pr[b_x = 1|\sigma]}{\Pr[b_x = 1|\sigma']} \leq \frac{1/2 + \epsilon/4}{1/2} = 1 + \epsilon/2 \leq \exp(\epsilon/2)$$

and so privacy is guaranteed against a single intrusion. Even conditioned on knowing the state of the algorithm, the output is ϵ-differentially private by the guarantees of the Laplace mechanism. ■

Theorem 6 *Except with probability β:*

$$|d(\sigma) - \hat{\theta}| \leq \alpha$$

Proof We have 3 sources of error to control. First, let $d(M)$ denote the density within the subsample m. By the additive Chernoff bound:

$$\Pr[|d(M) - d(\sigma)| \geq \alpha/3] \leq \exp(-(2/9)m\alpha^2) \leq \frac{\beta}{3}$$

Next, note that:

$$E[\theta] = d(M)(\frac{1}{2} + \frac{\epsilon}{4}) + \frac{(1 - d(M))}{2} = \frac{1}{2} + \frac{\epsilon d(M)}{4}$$

Hence $\mathrm{E}[\hat{\theta}] = d(M)$. We have:

$$\begin{aligned}
\Pr[|\hat{\theta} - d(M) \geq 2\alpha/3] &\leq \Pr[|\theta - E[\theta]| \geq \frac{\alpha}{3} \cdot \frac{\epsilon}{4}] + \Pr[\mathrm{Lap}\left(\frac{1}{\epsilon m}\right) \geq \frac{\alpha}{3}] \\
&\leq \exp\left(-\frac{2m\alpha^2\epsilon^2}{144}\right) + \exp\left(-\frac{\epsilon m\alpha}{3}\right) \\
&\leq \frac{2\beta}{3}
\end{aligned}$$

which completes the proof. ■

Bibliographic Information The content of this lecture is from Dwork, Naor, Pitassi, Rothblum, and Yekhanin, "Pan Private Streaming Algorithms", 2010.