Differential Privacy in the Streaming World

Aleksandar (Sasho) Nikolov
Rutgers University
The Streaming Model

1, 4, 5, 19, 145, 14, 5, 5, 16, 4
+, -, +, -, +, +, -, +, -, +

- Underlying frequency vector $A = A[1], ..., A[n]$
  - start with $A[i] = 0$ for all $i$.

- We observe an online sequence of updates:
  - Increments only (cash register):
    - Update is $i_t \rightarrow A[i_t] := A[i_t] + 1$
  - Fully dynamic (turnstile):
    - Update is $(i_t, \pm 1) \rightarrow A[i_t] := A[i_t] \pm 1$

- Requirements: compute statistics on $A$
  - Online, $O(1)$ passes over the updates
  - Sublinear space, $\text{polylog}(n,m)$
Typical Problems

  - related: $L_p$ norms

- Distinct elements: $F_0 = \#\{i: A[i] \neq 0\}$

- $k$-Heavy Hitters: output all $i$ such that $A[i] \geq F_1/k$

- Median: smallest $i$ such that $A[1] + \ldots + A[i] \geq F_1/2$
  - Generalize to Quantiles

- Different models:
  - Graph problems: a stream of edges, increments or dynamic
    - matchings, connectivity, triangle count
  - Geometric problems: a stream of points
    - various clustering problems
When do we need this?

- The universe size \( n \) is huge.

- Fast arriving stream of updates:
  - IP traffic monitoring
  - Web searches, tweets

- Large unstructured data, external storage:
  - multiple passes make sense

- Streaming algorithms can provide a \textit{first rough approximation}
  - decide whether and when to analyze more
  - fine tune a more expensive solution

- Or they can be the \textit{only feasible solution}
Outline

- Introduction to small space streaming
- Small space & differential privacy
- Privacy under continual observation
- Pan-privacy
A taste: the AMS sketch for $F_2$  

$h:[n] \rightarrow \{\pm 1\}$ is 4-wise independent

$$E[X^2] = F_2 \quad E[X^4]^{1/2} \leq O(F_2)$$
The Median of Averages Trick

Average: reduces variance by $\alpha^2$.

Median: reduces probability of large error to $\delta$.

<table>
<thead>
<tr>
<th>$X_{11}$</th>
<th>$X_{12}$</th>
<th>$X_{13}$</th>
<th>$X_{14}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$X_{21}$</td>
<td>$X_{22}$</td>
<td>$X_{23}$</td>
<td>$X_{24}$</td>
</tr>
<tr>
<td>$X_{31}$</td>
<td>$X_{32}$</td>
<td>$X_{33}$</td>
<td>$X_{34}$</td>
</tr>
<tr>
<td>$X_{41}$</td>
<td>$X_{42}$</td>
<td>$X_{43}$</td>
<td>$X_{44}$</td>
</tr>
<tr>
<td>$X_{51}$</td>
<td>$X_{52}$</td>
<td>$X_{53}$</td>
<td>$X_{54}$</td>
</tr>
</tbody>
</table>

Average: $rac{1}{\alpha^2}$

Median: $X$
Outline

- Introduction to small space streaming
- Small space & differential privacy
- Privacy under continual observation
- Pan-privacy
Defining Privacy for Streams

- We will use *differential privacy*.
- The database is represented by a stream
  - online stream of transactions
  - offline large unstructured database
- Need to define *neighboring inputs*:
  - **Event level privacy**: differ in a single update
    1, 4, 5, 19, 145, 14, 5, 5, 16, 4
    1, 1, 5, 19, 145, 14, 5, 5, 16, 4
  - **User level privacy**: replace some updates to $i$ with updates to $j$
    1, 4, 5, 19, 145, 14, 5, 5, 16, 4
    1, 4, 3, 19, 145, 14, 3, 5, 16, 4
- We also allow the changed updates to be placed somewhere else
Streaming & DP?

- Large unstructured database of transactions

- Estimate how many distinct users initiated transactions?
  - i.e. $F_0$ estimation

- Can we satisfy both the streaming and privacy constraints?
  - $F_0$ has sensitivity 1 (under user privacy)
  - Computing $F_0$ exactly takes $\Omega(n)$ space
  - Classic sketches from streaming may have large sensitivity
Oblivious Sketch

- Flajolet and Martin [FM 85] show a sketch $f(S)$
  - $O(\log n)$ bits of storage
  - $F_0/2 \leq f(S) \leq 2F_0$ with constant probability

- **Obliviousness:** distribution of $f(S)$ is *entirely* determined by $F_0$
  - similar to functional privacy [Feigenbaum Ishai Malkin Nissim Strauss Wright 01]

- Why it helps:
  - Pick noise $\eta$ from discretized $\text{Lap}(1/\varepsilon)$
  - Create new stream $S'$ to feed to $f$:
    - If $\eta < 0$, ignore first $\eta$ distinct elements
    - If $\eta > 0$, insert elements $n+1, \ldots, n+\eta$

- Distribution of $f(S')$ is a function of $\max\{F_0 + \eta, 0\}$: $\varepsilon$-DP (user)

- **Error:** $F_0/2 - O(1/\varepsilon) \leq f(S) \leq 2F_0 + O(1/\varepsilon)$

- **Space:** $O(1/\varepsilon + \log n)$
  - can make $\log n$ w.h.p. by first inserting $O(1/\varepsilon)$ elements
Open Problems

- When can a streaming estimate of a low-sensitivity function be computed privately, in small space?
  - does privacy & small space ever require more error than either?

- Can we go beyond low-sensitivity, and local sensitivity?
  - $F_2$ has high sensitivity and high local sensitivity
  - Lipschitz extensions [Kasiviswanathan Nissim Raskhodnikova Smith 13] relevant?

- What can we say about graph problems, clustering problems?
  - Private coresets [Feldman Fiat Kaplan Nissim 09]
Outline

- Introduction to small space streaming
- Small space & differential privacy
- Privacy under continual observation
- Pan-privacy
Continual Observation

- In an online stream, often need to *track* the value of a statistic.
  - number of reported instances of a viral infection
  - sales over time
  - number of likes on Facebook

- **Privacy under continual observation** [Dwork Naor Pitassi Rothblum 10]:
  - At each time step the algorithm outputs the value of the statistic
  - The *entire sequence* of outputs is $\varepsilon$-DP (usually event level)

- **Results:**
  - A single counter (number of 1’s in a bit stream) [DNPR10]
  - Time-decayed counters [Bolot Fawaz Muthukrishnan Nikolov Taft 13]
  - Online learning [DNPR10] [Jain Kothari Thakurta 12] [Smith Thakurka 13]
  - Generic transformation for monotone algorithms [DNPR10]
Binary Tree Technique [DPNR10], [Chan Shi Song 10]

Sensitivity of tree: $\log m$

Add $\text{Lap}(\log m/ \varepsilon)$ to each node
Binary Tree Technique

Each prefix: sum of \( \log m \) nodes

\( \rightarrow \) polylog error per query
Open Problems

- What is the optimal error possible for the counter problem?

- Privacy under continual observation for statistics that are not easily decomposable?

- User level?

- Expect privacy under continual observation to be ever more relevant
  - We usually want to track our statistics over time
  - Work on it!
Outline

- Introduction to small space streaming
- Small space & differential privacy
- Privacy under continual observation
- Pan-privacy
Pan Privacy

- Differential privacy guarantees that the results of our computation are private.

- What if data is requests by subpoena, leaked after a security breach, an unauthorized employee looks at it?

- Can we guarantee that intermediate states are also private?
  - Makes sense for online data: not stored.

- **Pan-privacy** [Dwork Naor Pitassi Rothblum Yekhanin 10]:
  - For each $t$: the state of the algorithm after processing the $t$-th update and the final output are jointly $\varepsilon$-DP.
  - Can be event level or user level.

- Strategy: keep private statistics on top of sketches.
Warm-up: $F_0$ [DNPRY10]

- Solution: randomized response
- Two distributions: $D_0$ and $D_1$ on {-1, 1}
  - $D_0$ is 1 w.p. 1/2;
  - $D_1$ is 1 w.p. $(1 + \varepsilon)/2$
- Store a big table $X[1], \ldots, X[n]$
  - Initialize all $X[i]$ from $D_0$
- When update $i_t$ arrives, pick $X[i_t]$ from $D_1$
- Can compute $O(n^{1/2}/\varepsilon)$ additive approximation
  - $X = (X[1] + \ldots + X[n])/\varepsilon$
  - $E[X] = F_0$ and $E[X^2] = n/\varepsilon^2$
Cropped $F_1$ [Mir Muthukrishnan Nikolov Wright 11]

- Cropped moments:
  - $F_k(\tau) = |\min\{A[1], \tau\}|^k + |\min\{A[2], \tau\}|^k + \ldots + |\min\{A[n], \tau\}|^k$
  - We’ll be interested in $F_1(\tau)$

- Can pan-privately compute $X$ s.t.
  $F_1(\tau)/2 - O(\tau n^{1/2}/\varepsilon) \leq X \leq F_1(\tau) + O(\tau n^{1/2}/\varepsilon)$

- Idea: keep each $A[i] \mod \tau$, with initial noise
  - What if $A[i] = \tau + 1$?
  - Multiply each $A[i]$ by a random $c_i$ uniform in $[1, 2]$
  - Small $A[i]$ ($\leq \tau/2$) get distorted by at most factor $2$
  - For large $A[i]$, $c_i A[i] \mod \tau$ is large on average

- Range is $\tau$, so noise $O(\tau/\varepsilon)$ per modular counter suffices
Recall, the $k$-Heavy Hitters ($k$-HH) are $i$ s.t. $A[i] \geq F_1/k$ at most $k$ of them.

Approximate the number of $k$-HH

- notation: $H_k$
- a measure of how skewed the data is

Will get pan-private estimator $X$ s.t.:

$$H_k/2 - O(k^{1/2}) \leq X \leq H_k \log k + O(k^{1/2})$$
Say we want to compute an estimate $X$ in $[H_k, H_{ck}]$

Consider:

$$\frac{(F_1(F_1/k) - F_1(F_1/ck))(F_1/k - F_1/ck)}{F_1(F_1/k)}$$

$k$-Heavy Hitters contribute 1

$ck$-Heavy Hitters contribute between 0 and 1

Anything else contributes 0

Error of $O(F_1 n^{1/2}/k \epsilon)$ for $F_1(F_1/k)$ is too much!

Sketch to reduce the universe size $n$
Idea: Use a (CM-type) Sketch

- Hash \([n]\) into \([O(k)]\) (with a pairwise-independent hash)

- Compute the number of heavy buckets (weight \(\geq F_1/k\))
  - at least \(H_k/2\) (balls and bins)
  - no bucket containing items of weight \(\leq F_1/(k \cdot \log k)\) is heavy

- Essentially keeping private statistics on a CM sketch
Lower bounds and Open Problems

- The $O(n^{1/2})$ additive error for $F_0$ is optimal
  - also $O(k^{1/2})$ for $H_k$, by reduction

- Idea: combine streaming-style LBs with reconstruction attacks [MMNW11]
  - stop the algorithm at some time step and grab the private state
  - different continuations of the stream: answer many counting queries from the same state
  - invoke [Dinur Nissim 03] type attacks

- Lower bounds against many passes via connections to randomness extraction [McGregor Mironov Pitassi Reingold Talwar Vadhan 10]

- Do all problems of low streaming complexity admit accurate pan-private algorithm
  - intuitively: less state $\rightarrow$ easier to make private
Summary

- Private analysis of massive online data presents new challenges
  - small space
  - continuous monitoring

- Data is not stored: can ask for algorithms private inside and out

- Tools from small-space streaming algorithms can be useful
  - but we need to view them from a new angle