Pan-Private Streaming Algorithms

Cynthia Dwork, Moni Naor, Toni Pitassi,
Guy Rothblum, Sergey Yekhanin

Privacy-Preserving Data Analysis

Statistical analysis of sensitive data comes with huge social benefit, e.g. medical studies

Concern: data is sensitive, what about privacy? "Better privacy, better data"

Goal: get both **utility** and **privacy**

Differential Privacy [DwMcNiSm06]

Guarantee: outcome of the analysis is nearly identical (in a strong sense) whether any user is in or out of the dataset

Participation does not increase risk of privacy violation

ε-differentially private algorithm *A*

For **all** DBs D and **all** events T

$$e^{-\varepsilon} \leq \frac{\Pr_A[A(D+Me)\in T]}{\Pr_A[A(D-Me)\in T]} \leq e^{\varepsilon} \approx 1+\varepsilon$$

“Modern” Differential Privacy Literature

In most works trusted curator collects sensitive data, publishes privacy-preserving analysis.

Including:

- Counting/Histogram Algorithms
- Arbitrary Functions
- Statistical Estimators
- Learning Algorithms
- Approximation Algorithms
- Geometric Algorithms and Core-Sets

And more…

“Post-Modern” Private Data Analysis

How can we support a well-intentioned curator?

Even well-intentioned curators subject to **mission creep** (“think of the children”), subpoena, security breach…

- Pro baseball anonymous drug tests
- Facebook policies to protect users from application developers

Goal: curator **accumulates** statistical information, but **never stores sensitive data** about individuals

Suggests streaming; Insufficient!

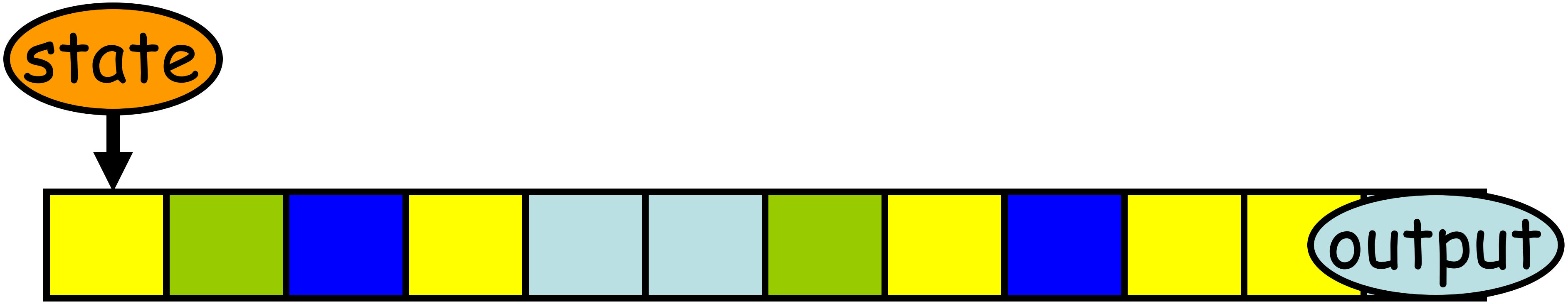
streaming permits storing info about selected individuals

each individual may have multiple elements in the stream

Pan-privacy: algorithm private **inside and out**

Pan-Privacy Model

Data is a **stream** of items, each item belongs to a user algorithm sees each item and updates internal state, generates output at end of stream (**single-pass**)



Pan-Privacy: for every two **adjacent streams**, at any **single point in time**, **internal state** (and final output) are differentially private.

Pan-Privacy Model

Adjacent streams: User-level privacy

Two streams are adjacent if they differ only in one user's (***potentially numerous***) data items

Number of observations: Attacker's view

Output + **One intrusion** or **multiple intrusions**

Streaming: want **small space**

Pan-Privacy: for every two **adjacent streams**, at any **single point in time**, **internal state** (and final output) are differentially private.

Example: Stream Density or # Distinct Elements

Universe X of users, estimate what fraction of users in X appear in data stream

Ideas that don't work:

- **Naïve:** keep list of users that have appeared (bad for privacy and large space)
- **Streaming literature:** hash each user, keep track of minimal hash value (bad for privacy)
- **Streaming literature:** keep random sub-sample of users that have appeared (bad for privacy)

Pan-Private Stream Density Estimator

Inspired by randomized response [Warner65]

For each user $x \in X$: store a single bit b_x drawn from D_0 or D_1

- Initially: for every $x \in X$, b_x is $(0,1)$ w.p. $(\frac{1}{2},\frac{1}{2})$ - dist. D_0
- When encountering x in data stream, update b_x to be $(0,1)$ w.p. $(\frac{1}{2}-\varepsilon,\frac{1}{2}+\varepsilon)$ - dist. D_1

Pan-Privacy:

If user x **never** appeared: entry drawn from D_0,

If user x appeared (**any # of times**): entry drawn from D_1

D_0 and D_1 are ε-differentially private

Accuracy: ε known, can reconstruct stream density

Additive error

Density Estimator Parameters

Storage still large: reduce using **sub-sampling**

Additive accuracy: improve using hashing

Theorem [density estimation streaming algorithm]

ε pan-privacy, multiplicative error α

Space is **poly(1/α,1/ε)**

Multiple Intrusions

If intrusions are **announced**, can handle multiple intrusions

accuracy degrades exponentially in # of intrusions

Can we do better?

Theorem [Continual intrusion lower bounds]

If there are either:

1. **Two unannounced** intrusions (for **finite-state** algorithms)
2. **Continual** intrusions (for **any** algorithm)

then additive accuracy cannot be better than $\Omega(n)$

What other statistics have pan-private algorithms?

Pan-private streaming algorithms for:

- Stream density / number of distinct elements
- **t**-cropped mean: mean, over users, of min(**t**,#appearances)
- Fraction of users appearing **k** times exactly
- Fraction of heavy-hitters, users appearing at least **k** times

Incidence Counting

Universe **X** of users. Given **k**, estimate what fraction of users in **X** appear **exactly** **k** times in data stream

Difficulty: can't track individual's # of appearances

Idea: keep track of ***noisy # of appearances***

However… can't accurately track whether individual appeared **0**,**k** or **100k** times.

Different approach: follows "count-min" [CM05] idea from streaming literature

Incidence Counting a la “Count-Min”

Use: pan-private algorithm that gets input:

1. hash function **h: Z→M** (for small range **M**)
2. target **val**

outputs fraction of users with **h(#appearances) = val**

Given this, estimate **k**-incidence as fraction of users with

h(# appearances) = h(k)

Concern: Might we over-estimate? (hash collisions)

Accuracy: If **h** has low collision prob, then with some probability collisions are few and estimate is accurate.

Repeat to amplify (output minimal estimate)

Putting it together

Hash by choosing small random prime p

$h(z) = z \pmod{p}$

Pan-private **modular incidence counter**:

Gets p and val, estimates fraction of users with

$\# \text{ appearances} = val \pmod{p}$

space is $poly(p)$, but small p suffices

Theorem [k-incidence counting streaming algorithm]

ε pan-privacy, multiplicative error α,

upper bound N on number of appearances.

Space is $poly(1/\alpha, 1/\varepsilon, \log N)$

Pan-Private Modular Incidence Counter

For every user x, keep counter $c_x \in \{0,\ldots,p-1\}$

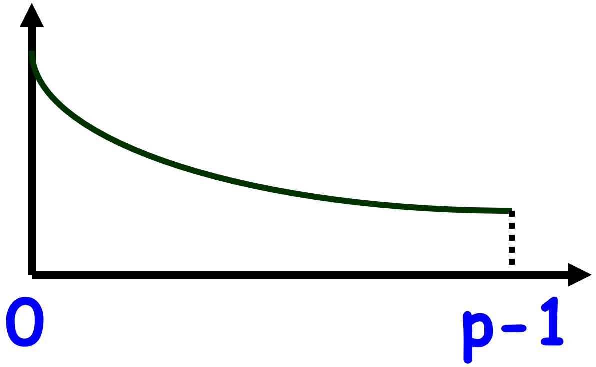
Increase counter (**mod p**) every time user appears

If initially **0** – **no privacy**, but perfect accuracy

If initially random – perfect privacy, but **no accuracy**

Initialize using a distribution **slightly biased towards 0**

$$\Pr[c_x = i] \approx e^{-\varepsilon \cdot i/(p-1)}$$



Privacy: user's #appearances has only small effect on distribution of c_x

Modular Incidence Counter: Accuracy

For $j \in \{0, \dots, p-1\}$

o_j is # users with **observed** “noisy” count j

t_j is true # users that **truly** appear j times $(\text{mod } p)$

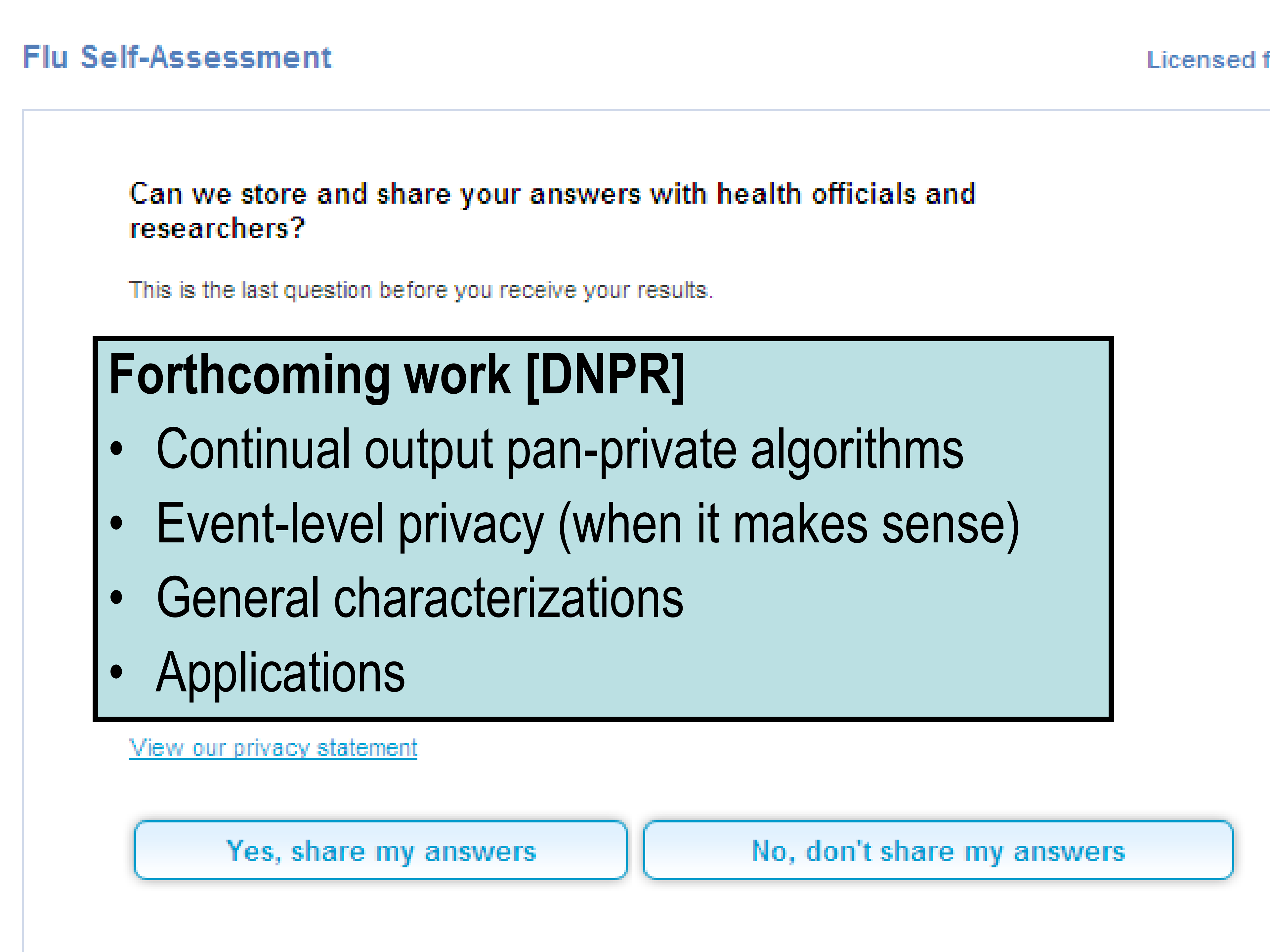
$$o_j \approx \sum_{k=0}^{p-1} t_{j-k \ (\text{mod } p)} \cdot e^{-\varepsilon \cdot k/(p-1)}$$

Using **observed** o_j’s,

get p (approx.) equations in p variables (the t_k’s)

solve using linear programming

argue that solution is close to true counts



Flu Self-Assessment

Licensed f

Can we store and share your answers with health officials and researchers?

This is the last question before you receive your results.

Forthcoming work [DNPR]

- Continual output pan-private algorithms
- Event-level privacy (when it makes sense)
- General characterizations
- Applications

View our privacy statement