Differential Privacy and Pan-Private Algorithms

Cynthia Dwork, Microsoft Research
A Dream?

Census, medical, educational, financial data, commuting patterns, web traffic; OTC drug purchases, query logs, social networking, …
Reality: Sanitization Can’t be Too Accurate

- “Overly accurate” answers to “too many” questions leads to blatant non-privacy [Dinur-Nissim ‘03; DMT07, DY08, HT09…] 

- Results may have been over-interpreted
  - Subject of another talk
  - Ask me about this
Success: Interactive Solutions

Multiple Queries, Adaptively Chosen

e.g. $n / \text{polylog}(n)$, noise $o(\sqrt{n})$

Accuracy eventually deteriorates as # queries grows

Has led to intriguing non-interactive “solutions”

Both Achieve Differential Privacy
A Linkage Attack

- Using “innocuous” data in one dataset to identify a record in a different dataset containing both innocuous and sensitive data

- Made Famous by Sweeney
  - Linked HMO data release to voter registration data

- Netflix Prize Training Data
  - Linked to the IMDb  [NS’07]
Many Other Successful Privacy Attacks

- Against Privacy by Aggregation
  - Including implementation by secure multi-party computation, secure function evaluation – it’s not the how, it’s the what
  - Simple differencing
- Against Auditing/Inference Control [KMN05, KPR00]
- Against Tokenized Search Logs [KNPT07]
- Against Randomized IDs
  - Analysis of the AOL dataset [Free for all]
  - Discovering relationships in social networks [BDK07]
- Against K-anonymity [MGK06]
  - Proposed L-diversity
- Against L-Diversity [XT07]
  - Proposed M-Invariance
- …But wait, there’s more! [KS08]
What is Going Wrong?

- Failure to account for “auxiliary information”
  - *In vitro* versus *in vivo*
  - Other datasets, previous version of current dataset, “innocuous” inside information, etc.

- Literally, Ad Hoc
  - The defined set of excluded (“bad”) outcomes fails to capture certain privacy breaches

- Needed:
  - An “Ad Omnia,” achievable, definition that composes automatically and obliviously with (past and future) information, databases
Dalenius / Semantic Security

- An ad omnia guarantee: Dalenius, 1977
  - Anything that can be learned about a respondent from the statistical database can be learned without access to the database
- Popular Intuition: prior and posterior views about an individual shouldn’t change “too much”.
  - Problematic: My (incorrect) prior is that everyone has 2 left feet.
- Provably Unachievable [DN06]

- Shift Focus to Risk Incurred by Joining (or Leaving) DB
  - “Differential Privacy” [DMNS06]
  - Before/After interacting vs Risk when in/not in DB
$\varepsilon$ - Differential Privacy

$K$ gives $\varepsilon$-differential privacy if for all neighboring $D_1$ and $D_2$, and all $C$

$\mu \text{ range}(K): \Pr[K(D_1) 2 C] \leq e^\varepsilon \Pr[K(D_2) 2 C]$

Neutralizes all linkage attacks.
Composes unconditionally and automatically: $\sum_i \varepsilon_i$
(ε, δ) - Differential Privacy

\( K \) gives (ε, δ) - differential privacy if for all neighboring D1 and D2, and all \( C \) \( \mu \) range(\( K \)): \( \Pr[ K(D1) \leq C] \leq e^\varepsilon \Pr[ K(D2) \leq C] + \delta \)

Neutralizes all linkage attacks.
Composes unconditionally and automatically: \( \sum_i \varepsilon_i + \sum_i \delta_i \)
Techniques and Programming Methodologies

SuLQ, Addition of Noise, Exponential Mechanism
Subject of Many Other Talks

- General Techniques (not exhaustive)
  - Privacy-preserving programming interface: SuLQ [DN04,BDMN05, DKMMN06]
    - Available Instantiation/generalization: PINQ [M09]
  - Addition of Laplacian or Gaussian Noise [DMNS06,NRS07,DN04 etc.]
  - Discrete Outputs [MT07]
  - Synthetic Databases [BLR08,DNRRV09]
  - Optimization Function Perturbation [CM09]
  - Geometric Approach [HT09]

- Specific Tasks (not exhaustive)
  - Contingency Tables / OLAP Cubes [BDMT05]
  - PAC Learning [NRS08], Boosting [DRV]
  - Logistic Regression [CM09,MW09]
  - Statistical Estimators [DL09,S09]
  - Halfspace Queries [BDMN05,BLR08]
Eternal Vigilance is the Price of Privacy

- Social Privacy Maxim:
  - Only use the data for the purpose for which they were collected
- Mission Creep
  - “Think of the children!”
- One defense: non-interactive sanitization
  - The data can be destroyed once sanitization is complete

- NEW: “pan-private” streaming algorithms [DNPRY]
  - Never store the data; no tempting target
  - “Pan-Private:” even internal state is differentially private
Pan-Privacy Model

- Data is a stream of items, e.g., search queries
  - Data of different users interleaved arbitrarily
  - Curator sees each item and updates internal state

**User-Level Pan-Privacy:**
For every possible behavior of user in stream, at any single point in time, internal state and output are differentially private
Example: Stream Density or # Distinct Elements

- Universe of users; estimate what fraction of users in the universe actually appear in the data stream
- Lovely streaming techniques don’t preserve privacy
- Flavor of solution:
  
  Keep a bit $b_x$ for each user $x$
  Initially set bit $b_x$ according to probability distribution $D_0$
  
  $D_0$: $\Pr[b_x = 0] = \Pr[b_x = 1] = \frac{1}{2}$

  Each time $x$ appears, set bit $b_x$ according to distribution $D_1$
  
  $D_1$: $\Pr[b_x = 0] = \frac{1}{2} + 2^{-2}$; $\Pr[b_x = 1] = \frac{1}{2} - 2^{-2}$
Many Fascinating Questions

- Pan-Private Algorithms for Several Tasks
  - Stream density / number of distinct elements
  - \( t \)-cropped mean: mean, over users, of \( \min(t, \# \text{ appearances}) \)
  - Fraction of items appearing \( k \) times exactly
  - Fraction of heavy-hitters, items appearing at least \( k \) times

- Continual Outputs?
  - Yes, for stream density. What else?

- Continual Intrusions?
  - Only really lousy density estimators. Other problems?

- Multiple Unannounced Intrusions? Two?
Backup Slides
Multiple Intrusions?

<table>
<thead>
<tr>
<th>Intrusions</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pan-privacy – one intrusion</td>
<td>One output at end</td>
</tr>
<tr>
<td>Continual intrusions</td>
<td>Continual outputs, state is not completely public</td>
</tr>
</tbody>
</table>

**Theorem** [Continual Intrusion Lower Bound]
For any density estimator, if there are continual intrusions, then additive accuracy cannot be better than $\Omega(n)$

**In Progress**
Multiple *announced* intrusions (accuracy degrades fast)
Multiple *unannounced* intrusions (even 2)
Sublinear Queries (SuLQ) Programming

Query =
\[ h: \text{row} \rightarrow [0,1] \]

Database D

Exact Answer: \( \sum_r h(D_r) \)
Response: Exact+noise
Addition of Laplacian Noise \([\text{DMNS06}]\)

\[ \Delta_1 = \max_{D_1, D_2 \text{ neighbors}} ||f(D_1) - f(D_2)||_1 \]

Theorem: To achieve \(\varepsilon\)-differential privacy, use scaled symmetric noise \([\text{Lap}(R)]^d\) with \(R = \Delta_1/\varepsilon\).
Addition of Gaussian Noise \[ \text{[DMNS06,DKMMN06]} \]
Laplacian vs Gaussian Noise

\[ \Delta_1 = \max_{D1,D2 \text{ neighbors}} \|f(D1) - f(D2)\|_1 \]

Theorem: To achieve \( \varepsilon \)-differential privacy, use scaled symmetric noise \([\text{Lap}(R)]^d\) with \( R = \Delta_1 / \varepsilon \).

\[ \Delta_2 = \max_{D1,D2 \text{ neighbors}} \|f(D1) - f(D2)\|_2 \]

Theorem: To achieve \((\varepsilon, \pm)\)-differential privacy, use noise \( \mathcal{N}(0, R^2)^d \) with \( R = 2[\log(1/\delta)]^{1/2} \Delta_2 / \varepsilon \).
Exponential Mechanism [MT07]

- Given database \( D \), produce output object \( y \)
  - \( y \) can be anything: strings, trees, text, a number between 1 and 10
  - Utility function \( u(D,y) \) measures goodness of \( y \)

- Output \( y \) with (normalized) probability exponential in utility of \( y \)
Techniques and Programming Methodologies

- Sublinear Queries (SuLQ) Programming
- Addition of Laplacian Noise to Outcome: f(D)+Noise
- Addition of Gaussian Noise to Outcome: f(D)+Noise
- Exponential Mechanism

- Use in Combination, eg, Noisy Gradient Descent [MW09]
  - SuLQ computation, using Laplacian noise, of a “noisy gradient”
  - Step size chosen using the exponential mechanism
  - (+ use probabilistic inference to wring out additional accuracy)
  - (+ programmed in PINQ [M07])
Techniques and Programming Methodologies

- Sublinear Queries (SuLQ) Programming
- Addition of Laplacian Noise to Outcome: \( f(D) + \text{Noise} \)
- Addition of Gaussian Noise to Outcome: \( f(D) + \text{Noise} \)
- Exponential Mechanism

- Subsample and Aggregate, Smoothed Sensitivity, Propose-Test-Release, data structures for multiple queries, differentially private synthetic databases, …
- Differential Privacy Branches Out
  - Utility maximization, private approximations, multiparty protocols, auctions, …