Anonymous Communications: One-to-Many

Giulia Fanti
Fall 2019
Based in part on slides by Anupam Datta, Piotr Mardziel
Administrative

• HW4 due Nov. 22 (<2 weeks from now)
  • Please hold off on “Fairness in Classification” problem
  • HW3 grades out on Gradescope/Canvas

• Recitation on Friday (Sruti)
  • Anonymous communication

• If you want feedback on your project, please come to OH!
In-class Quiz

• On Canvas
Last time

• Review of equalized odds vs equal opportunity
  • Revisit geometric interpretation

• Disparate impact
  • Metric for measuring
  • How to prevent it
Today

• Overview of fairness techniques & how they relate to each other

• Wrap up Unit 2

• Start Unit 3 on Anonymous + Privacy-Preserving Communication
Mistake from last time

• Does equalized odds imply group fairness?
• Work it out with your partner

• Equalized Odds
\[ P[\hat{Y} = 1 | A = 0, Y = y] = P[\hat{Y} = 1 | A = 1, Y = y] \]

• Group Fairness
\[ P[\hat{Y} = 1 | A = 0] = P[\hat{Y} = 1 | A = 1] \]
How does this help explain the profit results from last time?

<table>
<thead>
<tr>
<th>Method</th>
<th>Profit (% relative to max profit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max profit</td>
<td>100</td>
</tr>
<tr>
<td>Race blind</td>
<td>99.3</td>
</tr>
<tr>
<td>Equal opportunity</td>
<td>92.8</td>
</tr>
<tr>
<td>Equalized odds</td>
<td>80.2</td>
</tr>
<tr>
<td>Group fairness (demographic parity)</td>
<td>69.8</td>
</tr>
</tbody>
</table>
Disparate impact (relaxed group fairness)

Individual Fairness

Equal Opportunity

Equalized Odds

Group Fairness (strict equality)

Disparate impact (relaxed group fairness)

Fairness: High-Level View

Metrics

Enforcement Algorithms
Fairness: High-Level View

### Metrics
- **Modify Input Data**
  - “Certifying & Removing Disparate Impact”

### Enforcement Algorithms
- **Train Fair Classifier**
  - “Fairness through awareness”
- **Modify Biased Model**
  - “Equality of opportunity in supervised learning”

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Prevents any future training from exhibiting bias
- Can enforce whatever fairness metric you want
- * Allows post-facto modifications to models
  * Requires less data access
- Can destroy data utility
- Requires you to know ahead of time protected features
- Can hurt utility
Unit II: Learning from Big Data
Summary of Concepts

<table>
<thead>
<tr>
<th>Privacy</th>
<th>Fairness</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Risks</strong></td>
<td></td>
</tr>
<tr>
<td>Deanonymization</td>
<td>Bias in algorithms</td>
</tr>
<tr>
<td>Membership inference</td>
<td></td>
</tr>
<tr>
<td>Model inversion</td>
<td></td>
</tr>
<tr>
<td><strong>Metrics</strong></td>
<td>Group fairness</td>
</tr>
<tr>
<td>k-anonymity (and variants)</td>
<td>Individual fairness</td>
</tr>
<tr>
<td>Global (database) differential privacy</td>
<td>Disparate impact</td>
</tr>
<tr>
<td>Local differential privacy</td>
<td>Equalized odds</td>
</tr>
<tr>
<td><strong>Mitigations</strong></td>
<td>Equal opportunity</td>
</tr>
<tr>
<td>Data redaction</td>
<td>Data alterations</td>
</tr>
<tr>
<td>Data clustering</td>
<td>Classifier learning algos</td>
</tr>
<tr>
<td>DP mechanisms</td>
<td>Classifier modification algos</td>
</tr>
<tr>
<td>Federated learning</td>
<td></td>
</tr>
</tbody>
</table>
What should you be able to do?

- Identify privacy and fairness risks in ML/big data pipelines
  - Make a list of "things you should be worried about based on deanonymization approach"

- Propose mechanisms for mitigating those risks
  - E.g., design DP, unbiased learning pipelines
  - Implement such a pipeline (HW3, HW4)

- Evaluate the privacy (or fairness) vs utility cost of these mitigations
Next up:
Privacy-Preserving Communication

Unit III
Overview of the Unit

1. One-to-many communication
2. Point-to-point communication

Many techniques in both spaces rely on the same few algorithmic tools.
• Scenario: Suppose you need to send your passport via email

Now Google and Yahoo have your passport!
What can we do about this?

- Password protect the file
- Secret sharing (Shamir, 1979)
  - Important idea
  - Generalizations are widely-used
1. Want to transmit: 
   \[ x \in S \]

2. Generate random shares 
   \[ z_1, z_2, z_3, \text{ where } z_i \sim \text{Unif}(S) \]
   s.t. \[ z_1 + z_2 + z_3 = 0 \]

3. Send randomized data over network

\[ \text{Sum} \rightarrow x \]
Properties of secret sharing

• Correctness
  • The destination always receives the desired message
  • Because the noise cancels out

• Information-theoretic secrecy w.r.t. up to $n - 1$ colluding relays
  • I.e., any colluding set of $\leq n - 1$ relays learns no information about $x$
  • Prove this with your partner
What are some weaknesses of this algorithm?

• Requires nodes to
  • Participate reliably
  • Obey protocol

• Assumes a certain topology between the source and destination

We can solve a lot of these problems with coding theory!
What is a (channel) code?

Source

| 1 0 1 0 |

Channel

| 1 0 x 0 |

Dest

Goal: Add **redundancy** to correct for errors!
First attempt: **Repetition coding**

**Problem:** Repetition coding adds a lot of overhead!
Second attempt: **Reed-Solomon Codes**

- Widely used in many applications (e.g., distributed storage, CDs)
- Let $x = (x_1, \ldots, x_k) \in F^k$ be the message
  1. Encode $x$ in the coefficients of a degree $k - 1$ polynomial
     \[ p(a) = \sum_{i=1}^{k} x_i a^{i-1} \]
  2. Evaluate $p(a)$ at $n \geq k$ different points $a_1, \ldots, a_n$ of the field $F$

Q: How many points can be **erased** while still recovering $x$?
A: $n - k$ (because any $k + 1$ points will reconstruct $p(a)$)

**Remark:** RS Codes can also correct up to $\frac{n-k}{2}$ errors!
1. Want to transmit: \( x \in \mathbb{F}^k \)

2. Generate coded polynomial

\[
p(a) = \sum_{i=1}^{k} x_i a^{i-1}
\]

3. Evaluate \( p(a) \) at \( n \) points and transmit over network

Interpolate Polynomial

\( x \)
How can secret sharing help us with our email problem?
Related ideas are used often in security- or privacy-sensitive systems

- Bank safe deposit boxes
  - Require two keys to access

- Threshold cryptography
  - Used to ensure that any k-out-of-n parties can decrypt a secret (but no fewer)

- Next: Dining Cryptographer (DC) networks
Dining Cryptographers

• Make a message public in a perfectly untraceable manner (1988)

The Dining Cryptographers Problem:
Unconditional Sender and Recipient Untraceability

David Chaum
Centre for Mathematics and Computer Science, Kruislaan 413, 1098 SJ Amsterdam, The Netherlands

• Information-theoretic anonymity guarantee
  • This is an unusually strong form of security: defeats adversary who has unlimited computational power

• Impractical, requires huge amount of randomness
  • In group of size N, need N random bits to send 1 bit
Three-Person DC Protocol

Three cryptographers are having dinner. Either NSA is paying for the dinner, or one of them is paying, but wishes to remain anonymous.

Cryptographers = clients
NSA pays/someone pays = 1 bit message
Three-Person DC Protocol

1. Each diner flips a coin and shows it to his left neighbor.
   • Every diner will see two coins: his own and his right neighbor’s

2. Each diner announces whether the two coins are the same.
   • If he is the payer, he lies (says the opposite).

3. Odd number of “same” $\Rightarrow$ NSA is paying;
   • Even number of “same” $\Rightarrow$ one of them is paying
   • But a non-payer cannot tell which of the other two is paying!
Non-Payer’s View: Same Coins

Without knowing the coin toss between the other two, non-payer cannot tell which of them is lying.
Non-Payer’s View: Different Coins

Without knowing the coin toss between the other two, non-payer cannot tell which of them is lying.

“same”

“same”

“same”

“same”

payer

payer

“different”

payer

payer
Superposed Sending

• This idea generalizes to any group of size N
• For each bit of the message, every user generates 1 random bit and sends it to 1 neighbor
  • Every user learns 2 bits (his own and his neighbor’s)
• Each user announces own bit XOR neighbor’s bit
• Sender announces own bit XOR neighbor’s bit XOR message bit
• XOR of all announcements = message bit
  • Every randomly generated bit occurs in this sum twice (and is canceled by XOR), message bit occurs once
DC-Based Anonymity is Impractical

- Requires secure pairwise channels between group members
  - Otherwise, random bits cannot be shared
- Requires massive communication overhead and large amounts of randomness
+ DC-net (a group of dining cryptographers) is robust even if some members collude
  - Guarantees perfect anonymity for the other members