Local Model
for Differentially Private Data Analysis

Sofya Raskhodnikova
Boston University

Some slides are based on slides by Adam Smith (Boston University)
Typical examples: census, medical studies, what big companies want to publish about our data…

Two conflicting goals

- Protect privacy of individuals
  - Differential privacy [Dwork McSherry Nissim Smith 06]
- Give accurate answers
Two datasets $x, x'$ are **neighbors** if they differ in one person’s data.
Privacy Definition

An algorithm $A$ is $\epsilon$-differentially private if for all pairs of neighbors $x, x'$ and all sets of answers $S$:

$$\Pr[A(x) \in S] \leq e^{\epsilon} \Pr[A(x') \in S]$$
Properties of Differential Privacy

- Composition:
  If algorithms $A_1$ and $A_2$ are $\epsilon$-differentially private then algorithm that outputs $(A_1(x), A_2(x))$ is $2\epsilon$-differentially private

- Meaningful in the presence of arbitrary external information
Basic Privacy Models

Local Noninteractive

Local (Interactive)

Centralized

• Advantages of the local model:
  – private data never leaves person's hands
  – no single point of failure
  – highly distributed

• Disadvantage of the local model:
  – data-thirsty (more data for the same accuracy)
  – Exponentially more data for learning parity
Deployments of the Local Privacy Model


https://github.com/google/rappor

Chrome
Differential Privacy in the Local Model

Privacy Definition

A randomizer $R$ is $\epsilon$-differentially private if for all pairs of values $x_i, x_i'$ and all sets of answers $S$:

$$\Pr[R(x_i) \in S] \leq e^\epsilon \Pr[A(x_i') \in S]$$

• The requirement that the ratio $\frac{\Pr[R(x_i)=a]}{\Pr[A(x_i')=a]}$ be bounded predates differential privacy

[Efvimievski Gehrke Srikant 03]
Randomized Response [Warner 65]

- Canonical example of a local algorithm
- Invented to help get truthful answers on sensitive YES/NO survey questions.
- Each person has data $x_i \in \mathcal{X}$
  - Given $f: \mathcal{X} \rightarrow \{-1,1\}$, analyst needs the average of $f(x_i)$
  - Can deduce, e.g., the proportion of diabetics
- Randomization operator takes $y \in \{-1,1\}$:

$$R(y) = \begin{cases} 
  +y & \text{w. p. } \frac{e^{\epsilon}}{e^{\epsilon}+1} \\
  -y & \text{w. p. } \frac{1}{e^{\epsilon}+1}
\end{cases}$$
Randomized Response

- Randomization operator takes $y \in \{-1, 1\}$:
  \[ R(y) = \begin{cases} 
  +y & \text{w. p. } \frac{e^\epsilon}{e^\epsilon + 1} \\
  -y & \text{w. p. } \frac{1}{e^\epsilon + 1} 
\end{cases} \]

- $E[R(y)] = y \cdot \frac{e^\epsilon}{e^\epsilon + 1} - y \cdot \frac{1}{e^\epsilon + 1} = y \cdot \frac{e^\epsilon - 1}{e^\epsilon + 1}$

- If we rescale by $c_\epsilon = \frac{e^\epsilon + 1}{e^\epsilon - 1}$, then $E[c_\epsilon \cdot R(y)] = y$

- We can estimate the average of $f(x_i)$
  \[ A(x_1, \ldots, x_n) = \frac{1}{n} \sum_{i} c_\epsilon \cdot R(f(x_i)) \]

**Lemma.** $E \left[ \left\| A(x) - \frac{1}{n} \sum_{i} f(x_i) \right\| \right] \leq \frac{c_\epsilon}{\sqrt{n}} \approx \frac{1}{\epsilon \sqrt{n}}$. 
Randomized Response: Generalization

• Can be generalized to estimating the averages of functions of the form $f : \mathcal{X} \to [-1,1]$

• If $y \in [-1,1]$, first round it to 1 or -1:

$$\text{Round}(y) = \begin{cases} 
+1 & \text{w. p. } \frac{1+y}{2} \\
-1 & \text{w. p. } \frac{1-y}{2} 
\end{cases}$$

• Define $RR(y) = R(\text{Round}(y))$

• $E[RR(y)] = E[\text{Round}(y)] = \frac{1+y}{2} - \frac{1-y}{2} = y$

• We can estimate the average as before.
Power of Local Models

Local Noninteractive = Nonadaptive SQ

Local = SQ

Containment is strict: there are computational tasks for which noninteractive protocols require \textit{exponentially} larger $n$ than interactive ones.
**Statistical Query (SQ) Algorithms**

- An **SQ algorithm** can perform its computation by accessing the data via an SQ oracle.

  \[ f: \mathcal{X} \rightarrow [0,1] \]

  \[ \mathbb{E}_{y \sim P}[f(y)] \pm \tau \]

- Distribution \( P \) could be the distribution from which the data drawn or the empirical distribution over the data set.

- A **nonadaptive** algorithm specifies all its queries in advance.

- Huge fraction of basic learning/optimization algorithms can be expressed in SQ form [Kearns 93]

---

**Theorem** [Blum Dwork McSherry Nissim 05]

Any SQ algorithm can be simulated by a private algorithm.
(Noninteractive) $\text{Local} = \text{(Nonadaptive) SQ}$

**Theorem**

Any $q$-query (nonadaptive) SQ algorithm with tolerance $\tau$ can be simulated by an $\epsilon$-DP (noninteractive) local algorithm if $n \geq \frac{q \ln q}{\tau^2 \epsilon^2}$.

Local protocol for an SQ query:

- use a different group of $n/q$ people
- for each $i$, compute bit $\text{RR}(f(x_i))$
- average the noisy bits and rescale

- Participants can compute noisy bits on their own
- $\text{RR}$ (applied by each participant) is differentially private
- If all SQ queries are known in advance (non-adaptive), the protocol is non-interactive
(Noninteractive) Local = (Nonadaptive) SQ

**Theorem**

When the data is sampled i.i.d. from an unknown distribution $P$, any (noninteractive) local algorithm can be simulated by a (nonadaptive) SQ algorithm.

**Technique:** Rejection sampling

**Proof idea** [noninteractive case]:

- To simulate a randomizer $R: D \rightarrow W$ on entry $x_i$, need to output each $w \in W$ with probability $p(w) = \Pr_{y \sim P}[R(y) = w]$.

- Let $q(w) = \Pr[R(0) = w]$. (Approximates $p(w)$ up to factor $e^\varepsilon$).

1. Sample $w$ from $q(w)$.
2. Output $w$ with probability $\frac{p(w)}{q(w)e^\varepsilon}$.
3. With the remaining probability, repeat from (1).

- Use SQ queries to estimate $p(w) = \Pr_{y \sim P}[R(z) = w] = \sum_y \Pr_{y \sim P}[y] \cdot \Pr[R(y) = w]$.

**Idea:**

$$p(w) = \Pr_{y \sim P}[R(z) = w] = \sum_y \Pr_{y \sim P}[y] \cdot \Pr[R(y) = w] = \mathbb{E}_{y \sim P}[\Pr[R(y) = w]]$$
Summary

• We characterized the class of problems solvable in local noninteractive and local interactive models
  – with respect to sample size
  – up to polynomial factors

• Many specific tasks are studied (learning, heavy hitters, histograms, optimization problems, clustering,...)
  – Best algorithms often use sketching techniques
  – Information-theoretic techniques were developed for lower bounds
    [Beimel Nissim Omri 08, Chan Shi Song 12, Duchi Jordan Wainwright 13,...]

• For specific tasks, need to optimize
  – Amount of data need for specific accuracy
  – Running time
  – Communication
  – Server memory