What Can We Learn Privately?

Adam Smith
Penn State

Joint work with
Shiva Kasiviswanathan (Penn State)
Homin Lee (Columbia)
Kobbi Nissim (Ben-Gurion)
Sofya Raskhodnikova (Penn State)
Private Learning Algorithms

• **Goal:** machine learning algorithms that protect the privacy of individual examples (people, organizations,...)

• **Desiderata**
  - **Privacy:** Worst-case guarantee (differential privacy)
  - **Learning:** Distributional guarantee (PAC learning)

• **This talk**
  - Feasibility results
  - Open questions
Differential Privacy

$x'$ is a neighbor of $x$ if they differ in one row
Definition: A is indistinguishable if, for all neighbors \(x, x'\), for all subsets \(S\) of transcripts

\[
\Pr[A(x) \in S] \leq (1 + \epsilon)\Pr[A(x') \in S]
\]
Differential Privacy

\[ \Pr[A(x) \in S] \leq (1 + \epsilon) \Pr[A(x') \in S] \]

**Definition:** A is indistinguishable if, for all neighbors \( x, x' \), for all subsets \( S \) of transcripts:

\[ \Pr[A(x) \in S] \leq (1 + \epsilon) \Pr[A(x') \in S] \]
PAC learning

- **Z**: a random variable over domain D.
- **C**: a set of concepts $C = \{ c : D \rightarrow \{0, 1\} \}$

Examples $z_i \sim Z$

Labels $y_i = \ell(x_i)$

$\ell : D \rightarrow \{0, 1\}$
PAC learning

- $Z$ : a random variable over domain $D$.
- $C$ : a set of concepts $C = \{ c : D \rightarrow \{0, 1\} \}$

Examples $z_i \sim Z$
Labels $y_i = \ell(x_i)$
$\ell : D \rightarrow \{0, 1\}$
PAC learning

- \( Z \) : a random variable over domain \( D \).
- \( C \) : a set of concepts \( C = \{ c : D \to \{0, 1\} \} \)

Examples \( z_i \sim Z \)
Labels \( y_i = \ell(x_i) \)
\( \ell : D \to \{0, 1\} \)

Definition: A agnostically PAC-learns \( C \) on \( Z \) if, for all \( \ell \), with high prob. over \( z_1, \ldots, z_n \) i.i.d.: \( \Pr_{z \sim Z} [h(z) = \ell(z)] \leq \text{OPT} - \alpha \)

where \( \text{OPT} = \sup_{c' \in C} \Pr[c'(z) = c(z)] \)

\# examples \( n \)
running time of \( A \) \( \bigg\} \text{poly} \left( \frac{1}{\alpha}, \text{desc-length}(c') \right) \)
Private PAC learning

• Say \( A \) is a **private PAC learner** for \( C \) on \( Z \) if
  - \( A \) is a **PAC learner** for \( C \) on \( Z \) and
  - \( A \) is \( \varepsilon \)-indistinguishable for \( \varepsilon = o(1) \)
Private PAC learning

• Say $A$ is a private PAC learner for $C$ on $Z$ if
  - $A$ is a PAC learner for $C$ on $Z$ and
  - $A$ is $\epsilon$-indistinguishable for $\epsilon = o(1)$

• First attempt: Apply sample-aggregate to non-private learning algorithm

Intuition: Replace $f$ with a less sensitive function $\tilde{f}$.

$\tilde{f}(x) = g(f(sample_1), f(sample_2), \ldots, x_{kt}, \ldots, x_{kt})$

aggregation function

noise calibrated to sensitivity of $\tilde{f}$

output
Private PAC learning

- Say \( A \) is a **private PAC learner** for \( C \) on \( Z \) if
  - \( A \) is a **PAC learner** for \( C \) on \( Z \) and
  - \( A \) is \( \varepsilon \)-indistinguishable for \( \varepsilon = o(1) \)

- **First attempt:** Apply sample-aggregate to non-private learning algorithm

- **Problem:** there may be many good hypotheses. Different samples may produce different-looking hypotheses.
Private PAC learning

• Say $A$ is a private PAC learner for $C$ on $Z$ if
  - $A$ is a PAC learner for $C$ on $Z$ and
  - $A$ is $\varepsilon$-indistinguishable for $\varepsilon = o(1)$
Private PAC learning

• Say \( A \) is a **private PAC learner** for \( C \) on \( Z \) if
  - \( A \) is a **PAC learner** for \( C \) on \( Z \) and
  - \( A \) is \( \epsilon \)-indistinguishable for \( \epsilon = o(1) \)

• **Theorem**: Any PAC learnable concept can be learned privately, using polynomially-many samples but possibly exponential running time.
Private PAC learning

- Say \( A \) is a **private PAC learner** for \( C \) on \( Z \) if
  - \( A \) is a **PAC learner** for \( C \) on \( Z \) and
  - \( A \) is \( \epsilon \)-indistinguishable for \( \epsilon = o(1) \)

- **Theorem**: Any PAC learnable concept can be learned privately, using polynomially-many samples but possibly exponential running time.

- **Proof**: Use McSherry-Talwar exponential sampling
  - “Score” \( q(x, h) = - \#(\text{misclassified examples}) \)
  - Roughly need \( n \geq \text{desc-length}(c') \times \max\left(\frac{1}{\alpha \epsilon}, \frac{1}{\alpha^2}\right) \)
Private PAC learning

- Say $A$ is a **private PAC learner** for $C$ on $Z$ if
  - $A$ is a **PAC learner** for $C$ on $Z$ and
  - $A$ is $\varepsilon$-indistinguishable for $\varepsilon = o(1)$

**Theorem**: Any PAC learnable concept can be learned privately, using polynomially-many samples but possibly exponential running time.

**Proof**: Use McSherry-Talwar exponential sampling

- “Score” $q(x, h) = - \#$(misclassified examples)
- Roughly need $n \geq \text{desc-length}(c') \times \max\left(\frac{1}{\alpha \varepsilon}, \frac{1}{\alpha^2}\right)$

**Questions**:
- Can we get a VC-dimension bound?
- Can we preserve polynomial running time?
What is learnable privately & efficiently?

- **Parity-like Problems**

  - Domain $D = \mathbb{Z}_p^n$
  - Concepts $c(z) = \begin{cases} 
  0 & \text{if } z \odot v = 0 \pmod{p} \\
  1 & \text{if } z \odot v \neq 0 \pmod{p}
\end{cases}$

  - Need to assume that labels are consistent with some concept
    - (Without assumption, this becomes parity with noise)
What is learnable privately & efficiently?

- **Parity-like Problems**
  - Domain $D = \mathbb{Z}_p^n$
  - Concepts $c(z) = \begin{cases} 0 & \text{if } z \odot v = 0 \mod p \\ 1 & \text{if } z \odot v \neq 0 \mod p \end{cases}$
  - Need to assume that labels are consistent with some concept
    - (Without assumption, this becomes parity with noise)

- **Statistical Query algorithms**
  - Statistical Query: ask question of distribution $Z$
  - **Query**: predicate $g : D \times \{0, 1\} \rightarrow \{0, 1\}$
    - **Answer** $\approx \Pr_z[g(z, c(z)) = 0]$  
  - Many common learning algorithms are SQ algorithms
What is learnable privately & efficiently?

• Parity-like Problems
  - Domain $D = \mathbb{Z}_p^n$
  - Concepts $c(z) = \begin{cases} 0 & \text{if } z \circ v = 0 \mod p \\ 1 & \text{if } z \circ v \neq 0 \mod p \end{cases}$
  - Need to assume that labels are consistent with some concept
    - (Without assumption, this becomes parity with noise)

• Statistical Query algorithms
  - Statistical Query: ask question of distribution $Z$
  - Query: predicate $g : D \times \{0, 1\} \rightarrow \{0, 1\}$
  - Answer $\approx \Pr_z [g(z, c(z)) = 0]$
  - Many common learning algorithms are SQ algorithms
What can be learned privately?

\[ \text{PAC}^* = \text{PAC learnable with poly. samples but arbitrary computation} \]

\[ \text{PAC}^* = \text{Private-PAC}^* \]

\[ \text{P} \text{ARITY with noise} \]

\[ \text{P} \text{ARITY} \]

\[ \text{SQ} = \text{Local-p-PAC} \]
Statistical Query Learning

- **Statistical Query**: ask question of distribution $Z$
  - **Query**: predicate $g : D \times \{0, 1\} \rightarrow \{0, 1\}$
  - **Answer**: $\approx Pr_z[g(z, c(z)) = 0]$

- If $n$ is large, then use sum query on data + noise [BDMN]

- Alternative: “local”, decentralized protocol
  - For each $i$, compute bit $b_i = \begin{cases} g(x_i) & \text{w.p. } \frac{1}{2} + \epsilon \\ 1 - g(x_i) & \text{w.p. } \frac{1}{2} - \epsilon \end{cases}$
  - Sum of bits allows approximation to answer

- Local protocols studied extensively in data mining lit.

- **Theorem**: Local-private-PAC = SQ.
What can be learned privately?

\[ \text{PAC}^* = \text{PAC learnable with poly. samples but arbitrary computation} \]

- \[ \text{PAC}^* = \text{Private-PAC}^* \]
- \[ \text{P} \text{ARITY with noise} \]
- \[ \text{SQ} = \text{Local-p-PAC} \]

- \[ \text{P} \text{ARITY} \]
Privacy has other interesting connections to learning

D.P. algorithms are useful as sub-algorithms, to break dependencies

- “Follow the perturbed leader” algorithm for online decision
  \[\text{Kalai-Vempala}\]
- Fixing an issue in \[\text{Vempala-Wang 02}\] for learning Gaussian mixtures

Privacy investigation lead to separations between “adaptive” and “non-adaptive” SQ algorithms.

- Corresponds to interaction in private mechanisms

Good “sensitivity” properties of error lead to good generalization error
Thank you