Learning and Privacy

- To do machine learning, we need data.
- What if the data contains sensitive information?
- Even if the (person running the) learning algo can be trusted, perhaps the output of the algorithm reveals sensitive info.
- E.g., using search logs of friends to recommend query completions:

  Why are __
  Why are my feet so itchy?
Learning and Privacy

• To do machine learning, we need data.

• What if the data contains sensitive information?

• Even if the (person running the) learning algo can be trusted, perhaps the output of the algorithm reveals sensitive info.

• E.g., SVM or perceptron on medical data:
  - Suppose feature $j$ is has-green-hair and the learned $w$ has $w_j \neq 0$.
  - If there is only one person in town with green hair, you know they were in the study.

Learning and Privacy

• To do machine learning, we need data.

• What if the data contains sensitive information?

• Even if the (person running the) learning algo can be trusted, perhaps the output of the algorithm reveals sensitive info.

• An approach to address these problems:

  Differential Privacy
A preliminary story

• A classic result from theoretical crypto:
  - Say you want to figure out the average numeric grade of people in the room, without revealing anything about your own grade other than what is inherent in the answer.

• Turns out you can actually do this. In fact, any function at all. “secure multiparty computation”.
  - It’s really cool. Want to try?

• Anyone have to go to the bathroom?
  - What happens if we do it again?

Differential privacy “lets you go to the bathroom in peace”
Differential Privacy

High level idea:

• What we want is a protocol that has a probability distribution over outputs:

such that if person \( i \) changed their input from \( x_i \) to any other allowed \( x_i' \), the relative probabilities of any output do not change by much.

• This would effectively allow that person to pretend their input was any other value they wanted.

Bayes rule:

\[
\frac{\Pr(x_i|output)}{\Pr(x_i'|output)} = \frac{\Pr(output|x_i)}{\Pr(output|x_i')} \cdot \frac{\Pr(x_i)}{\Pr(x_i')} \cdot e^{\frac{-\epsilon^2}{2}}
\]

(Posterior \( \approx \) Prior)

Differential Privacy: Definition

It’s a property of a protocol \( A \) which you run on some dataset \( X \) producing some output \( A(X) \).

• \( A \) is \( \epsilon \)-differentially private if for any two neighbor datasets \( S, S' \) (differ in just one element \( x_i \rightarrow x_i' \)),

for all outcomes \( v \),

\[
e^{-\epsilon} \leq \frac{\Pr(A(S)=v)}{\Pr(A(S')=v)} \leq e^{\epsilon}
\]

\( \approx 1-\epsilon \)

probability over randomness in \( A \)

\( \approx 1+\epsilon \)
**Differential Privacy: Definition**

It's a property of a protocol $A$ which you run on some dataset $X$ producing some output $A(X)$.

- $A$ is $\epsilon$-differentially private if for any two neighbor datasets $S$, $S'$ (differ in just one element $x_i \rightarrow x_i'$),

View as model of **plausible deniability**

(pretend after the fact that my input was really $x_i'$)

for all outcomes $v$,

$$e^{-\epsilon} \leq \frac{\Pr(A(S)=v)}{\Pr(A(S')=v)} \leq e^{\epsilon}$$

\[\approx 1-\epsilon\] probability over randomness in $A$

\[\approx 1+\epsilon\]

**Differential Privacy: Methods**

It's a property of a protocol $A$ which you run on some dataset $X$ producing some output $A(X)$.

- Can we achieve it?
  - Sure, just have $A(X)$ always output 0.
  - This is perfectly private, but also completely useless.
  - Can we achieve it while still providing useful information?
Laplace Mechanism

Say have \( n \) inputs in range \([0,b]\). Want to release average while preserving privacy.

- Changing one input can affect average by \( \leq \frac{b}{n} \).
- Idea: take answer and add noise from Laplace distrib \( p(x) \propto e^{-\frac{|x|\epsilon n}{b}} \)
- Amount of noise added will be \( \approx \pm \frac{b}{(n\epsilon)} \).
- To get an overall error of \( \pm \gamma \), you need a sample size \( n = \frac{b}{\gamma\epsilon} \).
- Get a utility/privacy/database-size tradeoff.
- If want to estimate mean of a distribution up to \( \pm \gamma \) and the database is an iid sample, then for \( \gamma < \epsilon \) you can get privacy "for free".
Laplace mechanism more generally

- E.g., $f =$ standard deviation of income
- E.g., $f =$ result of some fancy computation.

Global Sensitivity of $f$:

$$GS_f = \max_{\text{neighbors } X, X'} |f(X) - f(X')|$$

- Just add noise $\text{Lap}(GS_f / \epsilon)$.

What can we do with this?

- Interface to ask questions
- Run learning algorithms by breaking down interaction into series of queries with noisy answers.
- **But, each answer leaks some privacy:**
  - If $k$ questions and want total privacy loss of $\epsilon$, better answer each with $\epsilon/k$. 
Can run SQ algorithms

• Anything learnable via Statistical Queries is learnable differentially privately using Laplace mechanism.

• Statistical query model:

\[ q(x,l) \]

\[ \Pr_B[q(x,f(x))=1] \pm \gamma. \]

• Many algorithms can be re-written to interface via such statistical estimates.

Can run SQ algorithms

• Anything learnable via Statistical Queries is learnable differentially privately using Laplace mechanism.

• Statistical query model:

\[ q(x,l) \]

\[ \Pr_B[q(x,f(x))=1] \pm \gamma. \]

- Really tailor-made for DP.
- In fact, for a single query, Laplace mechanism adds noise \(1/(en)\). Less than \(1/n^{1/2}\) due to sampling.
- Privacy “for free” unless q’s from space of low VC-dim...
Privately learnable = SQ-learnable?

- [KLNRS08]: Actually, anything learnable is learnable in principle with DP.
  - Exponential mechanism for general classes.
    - Assign score to each $f \in C$, exponentially decaying in its suboptimality.
    - Choose from this distrib over $C$.
  - Efficient algorithm for $C = \{\text{parity functions}\}$.
    - Interesting since not known to be efficiently learnable with noise, and provably not SQ-learnable.
  - SQ-learnable = learnable with local privacy, where no centralized database at all.

Local Sensitivity

- Consider $f = \text{median income}$
  - On some databases, $f$ could be *very* sensitive. E.g., 3 people at salary=0, 3 people at salary=b, and you.
  - But on many databases, it’s not.
  - If $f$ is not very sensitive on the actual input $X$, does that mean we don’t need to add much noise?

$$LS_f(X) = \max_{\text{nbrs } X'} |f(X) - f(X')|$$
Local Sensitivity

- Consider $f = \text{median income}$
  - If $f$ is not very sensitive on the actual input $X$, does that mean we don't need to add much noise?
- Be careful: what if sensitivity itself is sensitive?

Smooth Sensitivity

- [NRS07] prove can instead use (roughly) the following smooth bound instead:
  \[
  \max_y \left[ \mathcal{LS}_f(Y) e^{-\epsilon d(X,Y)} \right]
  \]
Smooth Sensitivity

- In principle, could apply sensitivity idea to any learning algorithm (say) that you’d like to run on your data.
- But might be hard to figure out

Objective perturbation [CMS08]

- Idea: add noise to the objective function used by the learning algorithm.
- Natural for algorithms like SVMs that have regularization term.
- [CMS] show how to do this, if use a smooth loss function. Also show nice experimental results.
So far: learning as goal, privacy as constraint

Now: learning as tool for achieving stronger privacy

Answering more questions

"Add iid noise" approach can only answer a limited number of questions before it has to shut down.

- **Fundamental limit**: \( \# \text{questions} \leq |S|^2 \) to preserve this kind of privacy?
- **Output “sanitized database”**: people can examine as they wish?
Idea: back to SQ’s from class of small VC dim

- Fix a class $Q$ of statistical (i.e., counting/n) queries you care about (e.g., all $2^d$ marginals).
- VC-dimension bounds: whp a random subsample of size $O(\text{VCdim}(Q)/\alpha^2)$, will approximate all $q \in Q$ up to $\pm \alpha$.
- If $n \gg \text{VCdim}(Q)/(\epsilon \alpha^2)$, this offers at least $(0, \epsilon)$ privacy. Maybe can invert?

With probability $1 - \epsilon$, nothing is revealed about you, with prob $\epsilon$, everything is revealed about you. We want: with prob 1, very little is revealed about you.

[BLR08] building on [KLNRS08]: Use this with the “exponential mechanism”: Explicit distrib over sets of size $m = O(\text{VCdim}(Q)/\alpha^2)$

$\Pr(S') \propto e^{-O(\epsilon n \text{ penalty}(S'))}$

Penalty($S'$) = max gap $S, S'(Q)$

- Solve for $n$ s.t. bad $S'$ (penalty $\alpha$) have prob $\ll 1/2^m$.
- $-\epsilon n \alpha \ll -md = \left(\frac{\text{VCdim}(Q)}{\alpha^2}\right) d$
Idea: back to SQ’s from class of small VC dim

\[ \Pr(S') \propto e^{-O(\epsilon n \text{ penalty}(S'))} \]

- Solve for \( n \) s.t. bad \( S' \) (penalty) have probability \( \ll 1/2^{md} \).
- Get \( n = O(d \text{ VCdim}(Q)/(\epsilon \alpha^3)) \) sufficient to whp output good sanitized db.

[BLR08] building on [KLNRS08]: Use this with the “exponential mechanism”: Explicit distrib over sets of size \( m=O(\text{VCdim}(Q)/\alpha^2) \)

\[ \text{Penalty}(S') = \max_{s,s'} \text{gap}_{s,s'}(Q) \]