Lecture 9 Differentially Private Machine Learning

Yu-Xiang Wang
Logistic notes

• Submit your HW1 and Project proposal if you haven’t yet

• HW2 is on the course website
  • Except a code template for Q4 which I am still finalizing.
  • You should start working on the theoretical parts
Recap: Last lecture

• Composition of mechanism-specific representations
  • RDP accountant
  • Fourier accountant

• Unified treatment via a dominating privacy loss random variable
  • And its characteristic function

• autodp: How you would represent DP mechanism and compute privacy loss
Recap: Mechanism specific analysis and privacy accounting

<table>
<thead>
<tr>
<th>Functional view</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>$D_{\alpha}(P</td>
<td></td>
<td>Q) \leq \epsilon(\alpha), \forall \alpha \geq 1$</td>
</tr>
<tr>
<td>$\mathbb{E}_q[(\frac{p}{q} - e^\epsilon)^+] \leq \delta(e^\epsilon), \forall \epsilon \geq 0$</td>
<td>Interpretable.</td>
<td>messy composition.</td>
</tr>
<tr>
<td>Trade-off function $f$</td>
<td>Interpretable, CLT</td>
<td>messy composition.</td>
</tr>
<tr>
<td>Probability density of $\log(p/q)$</td>
<td>Natural composition via FFT</td>
<td>Limited applicability.</td>
</tr>
</tbody>
</table>

Table 1: Modern functional views of DP guarantees and their pros and cons.

- Renyi DP is qualitatively different from approximate DP. Composition is quite natural with RDP.

- The composition of privacy-profiles and tradeoff functions are equivalent and somewhat messy.
  - The key to get it to work is to find a dominating pair
  - Using $\phi$-function representation, we get the natural composition of RDP, and the tightness of privacy-profile / tradeoff functions.
Recap: Analytical Fourier accountant

• Composition: simply add up the log of phi functions
• Conversion to approx. DP via Levy’s formula
• Conversion to tradeoff function via duality.

Recap: autodp: automating differential privacy computation (for both laypersons and experts)

• Users describe their randomized algorithm to autodp

• autodp focuses on computing privacy losses

Open source project: https://github.com/yuxiangw/autodp

pip install autodp
This lecture

• Introduction to Differentially Private Machine Learning
  • Problem setup and notations
  • Examples
  • A learning theoretic study of the problem

• Posterior sampling mechanism
  • When the loss functions are bounded
  • A new analysis of the when they are not
(Optional) reading materials


• “Privacy-for-free: Posterior Sampling and Stochastic Gradient Monte Carlo” https://arxiv.org/abs/1502.07645


• Other (not covered in this lecture)
  • “What can we learn privately?” https://arxiv.org/abs/0803.0924
Differentially Private Machine Learning

Data

Learning algorithm

Classifier

f

Supervised Learning
Clustering
Kernel Density Estimation

Cluster centers
Estimated density function

Differentially private learning algorithm

Supervised Learning
Clustering
Kernel Density Estimation

Classifier
Cluster centers
Estimated density function

Feature-label pairs
Unlabeled features
Feature points

Feature-label pairs
Unlabeled features
Feature points
Example: Recommender System

Deep Neural Network based Recommendation Engine
Example: Recommender System

“If your recommendation engine is private, then an adversary can’t infer whether a particular user was present”
A closely related setting: Federated Learning

Illustration extracted from McMahan and Ramage (2017)

MPC ensures the security of distributed computation.
DP eliminates the risk from the output itself.

Additional considerations
- Communication cost
- Size of the model
- Rounds of adaptivity
Two regimes of Federated Learning with DP

*Need flexible tools for algorithm design.
*Need tight privacy accounting to standardize empirical benchmarking of DP methods.

Notations and problem setup

• Data space

• Hypothesis space / model space

• Loss functions

• Learning algorithm
Example 1: Linear / Logistic regression

- Data space
- Hypothesis space / model space
- Loss functions
- Learning algorithm
Example 2: PAC learning / binary classification

- Data space
- Hypothesis space / model space
- Loss functions
- Learning algorithm
Example 3: k-means clustering

• Data space

• Hypothesis space / model space

• Loss functions

• Learning algorithm
Example 4: Recommender System / Matrix factorization

• Data space

• Hypothesis space / model space

• Loss functions

• Learning algorithm
What do we know about machine learning methods theoretically?

• Learnability

• Sample complexity of learning

• Key component: Uniform convergence / VC-theory
The fundamental problems in learning with differential privacy

• Private learnability
  • What problems are learnable with DP

• Sample complexity of private learning
  • Among those that are privately learnable, what is the number of samples needed to learn

• Efficient algorithms?
PAC learning with finite hypothesis space

• Standard statistical learning via Hoeffding’s inequality and a union bound
Generic private optimization algorithm via exponential mechanism?

- Recap: utility of the EM

**Theorem 3.11.** Fixing a database $x$, let $\mathcal{R}_{\text{OPT}} = \{ r \in \mathcal{R} : u(x, r) = \text{OPT}_{u}(x) \}$ denote the set of elements in $\mathcal{R}$ which attain utility score $\text{OPT}_{u}(x)$. Then:

$$\Pr\left[ u(\mathcal{M}_{E}(x, u, \mathcal{R})) \leq \text{OPT}_{u}(x) - \frac{2\Delta u}{\varepsilon} \left( \ln \left( \frac{|\mathcal{R}|}{|\mathcal{R}_{\text{OPT}}|} \right) + t \right) \right] \leq e^{-t}$$

- Let’s apply this to the PAC learning problem with finite hypothesis space
What happens with continuous hypothesis space (satisfying VC-dim is finite)?

• Short answer: No DP algorithms can learn the VC class.

• Example: Learning a threshold function.
Any pure-DP algorithms will fail in learning threshold functions.

(Beimel et al. 2013) (Chaudhuri and Hsu. 2015)
Formalizing the lower bound.

(Beimel et al. 2013) (Chaudhuri and Hsu. 2015)
Is this an issue of pure-DP being too restrictive? Can we learn a VC class under approximate DP?

\[ \omega\left(\frac{1}{n}\right) \quad \delta(n) \quad O\left(\frac{1}{n}\right) \]

- Slower
- Faster

Learnability = Approx. Private Learnability

Learnability \neq Approx. Private Learnability

- Bun et. al. “Differentially private release and learning of threshold functions.” FOCS’15
What is the key problem?

• Statistical learning requires a very strong notion of learning:
  • No assumptions on data distributions
  • Uniformly consistent learning for all distributions

• Differential privacy says that even if the data distributions are disjoint, they still need to be somewhat similar.
  • The construction relies on an exponentially large set of distributions (each converging to a point-mass).
Remedies to this problem

• Assume the loss function is Lipschitz
  • 0-1 loss doesn’t work, but we can use surrogate losses, e.g., logistic loss, hinge losses.

• Assume some regularity conditions on the probability distributions of data
  • e.g., Bounded probability density.

• Then most problems are learnable by an exponential mechanism.

Check point: Learning with differential privacy in theory

• Finite hypothesis class
  • Exponential mechanism gives asymptotically vanishing additional error.
  • Does not benefit from being realizable.

• Continuous hypothesis class (bounded VC-dim)
  • No DP algorithm gives consistent learning
  • A Packing lower bound

• However, there are weak assumptions we can add
  • Lipschitz loss functions
  • data-distributions with bounded density
Connections of the Exponential mechanism to Bayesian learning

• Bayesian philosophy
  • I have a prior belief
  • When I collect data, I update my belief

• Deriving the posterior using iid data:

Connections of the Exponential mechanism to Bayesian learning

- Getting one sample from the posterior distribution is equivalent to exponential mechanism

**Algorithm 1** One-Posterior Sample (OPS) estimator

**input** Data \( X \), log-likelihood function \( \ell(\cdot|\cdot) \) satisfying \( \sup_{x,\theta} \|\ell(x|\theta)\| \leq B \) a prior \( \pi(\cdot) \).

Privacy loss \( \epsilon \).

1. Set \( \rho = \min\{1, \frac{\epsilon}{2B}\} \).
2. Re-define log-likelihood function and the prior \( \ell'(\cdot|\cdot) := \rho \ell(\cdot|\cdot) \) and \( \pi'(\cdot) := (\pi(\cdot))^\rho \).

**output** \( \hat{\theta} \sim P(\theta|X) \propto \exp\left(\sum_{i=1}^{N} \ell'(\theta|x_i)\right) \pi'(\theta) \).

- Some classical results from statistics
  - Asymptotic normality of the Bayesian posterior
  - Bernstein-Von Mises Theorem.

- Utility of the OPS estimator

**Proposition 9.** Under the same assumption as Proposition 8, if we set a different \( \epsilon \) by rescaling the log-likelihood by a factor of \( \frac{\epsilon}{4B} \), then the the One-Posterior sample estimator obeys

\[
\sqrt{n}(\hat{\theta} - \theta_0) \overset{weakly}{\to} \mathcal{N}\left(0, (1 + \frac{4B}{\epsilon})\mathbb{I}^{-1}\right),
\]

in other word, the estimator has an ARE of \( (1 + \frac{4B}{\epsilon}) \).
Pros and cons of OPS

• Pros:
  • Generically applicable
  • Strong learning bounds under weak assumptions
  • No need to change existing learning workflow

• Cons:
  • It makes the distribution more diffused.
  • Can get only one sample (hard to do inference)
  • Requires bounded (clipped) loss functions
  • Computationally inefficient in general
Improved analysis of exponential mechanism with strong convexity

• Assume \( \pi(\theta) = e^{-r(\theta)} \) where \( r(\theta) \) is \( \mu \)-strongly convex, i.e., the prior is strongly log-concave.

• Assume that the loss-function is Lipschitz

\[
| - \log p(x|\theta) + \log p(x|\theta') | \leq L \| \theta - \theta' \|
\]

• Then

\( \hat{\theta} \sim P(\theta|x_1, ..., x_n) \propto e^{(-\tau \sum \log p(x_i|\theta) - r(\theta))} \)

obeys \((\epsilon, \delta) - \text{DP}\) if

\[
\tau = O\left( \frac{\epsilon \sqrt{\mu}}{L \sqrt{\log(1/\delta)}} \right)
\]

Idea of the proof for the improved analysis of EM

• The privacy loss random variable is

• Apply the tail bound lemma

• The strong log-concavity + Lipschitz assumption ensures that \( \hat{\theta} \) satisfies a “Log-Sobolev Inequality”
  • Which ensures a subgaussian-like tail bound for all Lipschitz functions of \( \hat{\theta} \)
  • And a bound on the KL-divergence.

Reiterate the main points

• Bayesian learning
  • Just a scaling of posterior sampling
  • For bounded log-likelihood functions, exponential mechanism is a consistent learner

• Boundedness is not needed if we use a strong prior
  • A tighter analysis of the exponential mechanism
  • Use a prior to ensure that PLRV is concentrated
Next lecture

• Convex empirical risk minimization

• Objective perturbation