Lecture 16 Data-Adaptive DP in Machine Learning

Yu-Xiang Wang

COMPUTER SCIENCE
UC SANTA BARBARA
Computing. ReInvented.

Logistics

- Last lecture with new materials.
- We may have short lecture next Monday if I don't finish everything today.
- Remaining lectures will be for
  - Project consultation
  - Homework discussion
  - Anything on your mind
- I will be in this lecture hall. All are welcome.

Recap: data-dependent DP algorithms

- Smooth sensitivity
- Distance-to-Instability
- Propose-Test-Release
- Privately Releasing Local-Sensitivity

Recap: distance-to-instability

- Distance to instability
  - d(x) = d(x; {x" | f(x") ≠ f(neighbor of x") })
    = d(x; {x" | f(x") ≠ f(x) }) **– 1**
- The Dist2Instability mechanism:

- Proof: Observe that decision is post-processing of Laplace mechanism.

  Case A: If f(x) = f(x') => | d(x) – d(x')| ≤ 1

  Case B: If f(x) ≠ f(x') => d(x) = d(x') = 0

Recap: Propose-Test-Release

- Propose a bound on LS
- Privately test it by adding noise.
  - $d(x, \beta) = d(x, \{x'' | LS(x'') > \beta\})$
  - Output $\perp$ if $d(x, \beta) + Lap\left(\frac{1}{\epsilon}\right) < \frac{\log\frac{1}{\delta}}{\epsilon}$
  - Else output $f(x) + Lap\left(\frac{1}{\epsilon}\right)$
- Proof idea similar to "Distance-to-instability"
  - Case A: $LS(x) > \beta$ => $d(x, \beta) = 0$ Test fails with low probability.
  - Case B: $LS(x) \leq \beta$ => Composition to two Laplace Mechanisms

Recap: Privately releasing local sensitivity

Lemma: Let $\tilde{\Delta}_f(D)$ satisfies ε-DP and

$$\mathbb{P}\left[\Delta_f(D) \geq \tilde{\Delta}_f(D)\right] \leq \delta$$

Then $f(D) + \mathrm{Lap}(\tilde{\Delta}_f(D)/\epsilon)$ satisfies (2ε,δ)-DP.

This is computationally efficient if we can release the local sensitivity efficiently.

Example: Output perturbation of DP-GLM with Lipschitz, smooth and convex losses.

See a more general statement and proof in
Appendix G.6 of this paper: https://sites.cs.ucsb.edu/~yuxiangw/docs/spectral_privatelda.pdf

Summary: Data-dependent DP algorithms so far

	Applicability	Computationally efficiency
Smooth sensitivity	Numerical queries (does not scale to high-dimension)	Efficient when SS or other smooth upper bound of LS is efficient
Dist2Instability	Arbitrary queries But need LS = 0 in neighborhood of x.	Efficient when dist2instability function is efficiently computable.
PTR	Numerical queries. Need a good guess of a stable LS upper bound	Efficient when dist2largeLS function is efficiently computable.
Privately Bounding LS	Numerical queries.	Efficient when LS can be bounded and privately released efficiently.

This lecture

- Beyond local sensitivity
  - Per-instance differential privacy
  - pDP to DP conversion

- Examples of data-dependent algorithms in differentially private machine learning

- Open problems / good research directions in DP

Example: Data-Dependent Differentially Private ERM

- Convex, Lipschitz and Smooth losses
- Local sensitivity

Lemma 17 (Stability of smooth learning problems, Lemma 14 of (Wang, 2017)). *Assume ℓ and r be differentiable and their gradients be absolute continuous. Let $\hat{\theta}$ be a stationary point of $\sum_i \ell(\theta, z_i) + r(\theta)$, $\hat{\theta}'$ be a stationary point $\sum_i \ell(\theta, z_i) + \ell(\theta, z) + r(\theta)$ and in addition, let $\eta_t = t\hat{\theta} + (1-t)\hat{\theta}'$ denotes the interpolation of $\hat{\theta}$ and $\hat{\theta}'$. Then the following identity holds:*

$$\hat{\theta} - \hat{\theta}' = \left[\int_0^1 \left(\sum_i \nabla^2 \ell(\eta_t, z_i) + \nabla^2 \ell(\eta_t, z) + \nabla^2 r(\eta_t)\right) dt\right]^{-1} \nabla \ell(\hat{\theta}, z)$$
$$= -\left[\int_0^1 \left(\sum_i \nabla^2 \ell(\eta_t, z_i) + \nabla^2 r(\eta_t)\right) dt\right]^{-1} \nabla \ell(\hat{\theta}', z).$$

- Output perturbation

What if we the mechanism is not just adding noise?

- Example: Revisiting linear regression
  - Posterior sampling mechanism:

$$p(\theta|X, \mathbf{y}) \propto e^{-\frac{\gamma}{2}\left(\|\mathbf{y}-X\theta\|^2+\lambda\|\theta\|^2\right)}.$$

- The distribution depends jointly on the data and on the hyperparameters of the mechanisms

General idea: Working with privacy loss random variables

- The output space can be arbitrary, but the space of the privacy loss RV is 1-D.

- We can
  1. Work out the privacy loss random variables
  2. Figuring out what part of it depends on the data
  3. Release an upper bound of these data-dependent quantities differentially privately.
  4. Calibrate noise to privacy budget according to this upper bound.

Detour: Per-instance Differential Privacy

Definition 2.2 (Per-instance Differential Privacy)**.** For a fixed data set Z and a fixed data point z. We say a randomized algorithm $\mathcal{A}$ satisfy (ϵ, δ)-per-instance-DP for (Z, z) if, for all measurable set $\mathcal{S} \subset \Theta$, it holds that

$$P_{\theta \sim \mathcal{A}(Z)}(\theta \in \mathcal{S}) \leq e^{\epsilon} P_{\theta \sim \mathcal{A}([Z,z])}(\theta \in \mathcal{S}) + \delta,$$
$$P_{\theta \sim \mathcal{A}([Z,z])}(\theta \in \mathcal{S}) \leq e^{\epsilon} P_{\theta \sim \mathcal{A}(Z)}(\theta \in \mathcal{S}) + \delta.$$

- Remarks:
  - Defining DP for each pair of neighboring datasets.
  - Measure the privacy loss for each individual z given a fixed dataset Z (or [Z,z])
  - Can be viewed as taking ε as a function
- Properties:
  - Composition / Post-processing and many other properties.
  - DP can be obtained by maximizing over Z,z

W. (2019). Per-instance differential privacy. *Journal of Privacy and Confidentiality*, *9*(1)
https://journalprivacyconfidentiality.org/index.php/jpc/article/download/662/675

Visualizing pDP vs DP upper bound output perturbation in linear regression

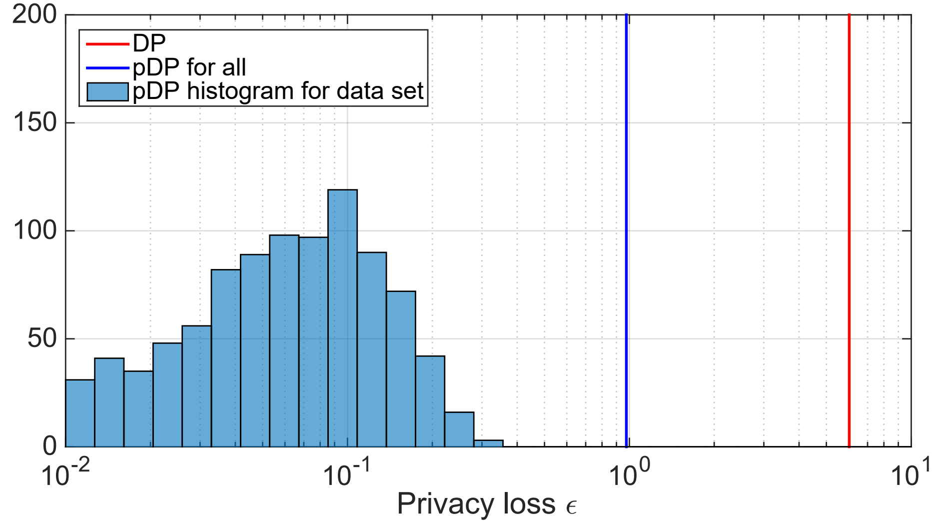

Figure 1: Illustration of the privacy loss ϵ of an output perturbation algorithm under DP, pDP for all, as well as the distribution of pDP's privacy loss for data points in the data set. The data set is generated by a linear Gaussian model, where the design matrix is normalized such that each row has Euclidean norm 1 and y is also clipped at $[-1, 1]$. The output perturbation algorithm releases $\hat{\theta} \sim \mathcal{N}((X^T X + I)^{-1} X\mathbf{y}, \sigma^2 I)$ with $\sigma = 4$. Our choice of $\delta = 10^{-6}$.

For classification problems: objective perturbation on logistic regression. The (ex post) pDP says the following

$$\epsilon(\hat{\theta}^P, D, D_{\pm z}) \leq \left| -\log\left(1 \pm f''(\cdot)\mu(x)\right) + \frac{1}{2\sigma^2}||\nabla \ell(\hat{\theta}^P; z)||_2^2 \pm \frac{1}{\sigma^2}\nabla J(\hat{\theta}^P; D)^T \nabla \ell(\hat{\theta}^P; z)\right|,$$

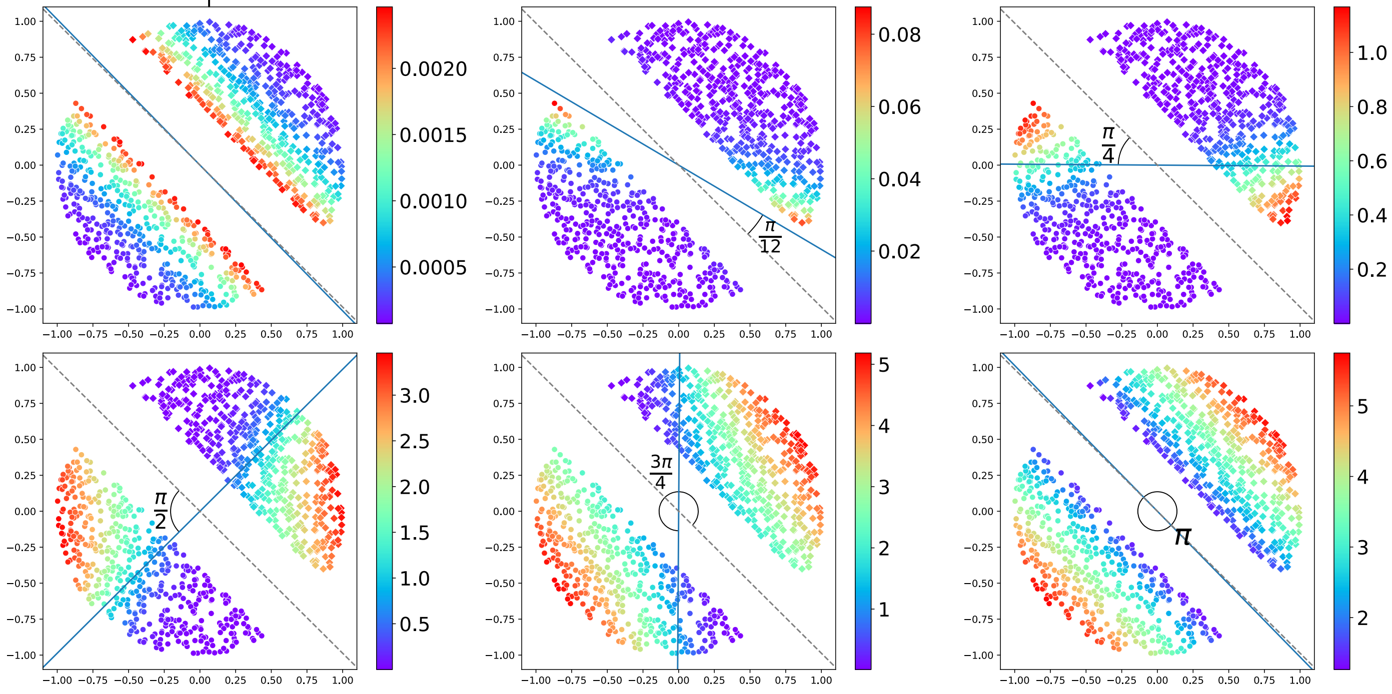

Figure 1: Visualization of *ex-post* pDP losses for logistic regression ($n = 1000, d = 2$).

Redberg and W. (2021) Privately publishable Per-instance Privacy: https://arxiv.org/abs/2111.02281

Per-instance differential privacy of Posterior Sampling for linear regression?

$$\epsilon(Z,z) \leq \frac{1}{2}\left|-\log(1+\mu) + \frac{\gamma\mu}{(1+\mu)}(y - x^T\hat{\theta})^2\right| + \frac{\mu}{2}\log(2/\delta) + \sqrt{\gamma\mu\log(2/\delta)}|y - x^T\hat{\theta}| \tag{4.3}$$

$$= \frac{1}{2}\left|-\log(1-\mu') - \frac{\gamma\mu'}{1-\mu'}(y - x^T\hat{\theta}')^2\right| + \frac{\mu'}{2}\log(2/\delta) + \sqrt{\gamma\mu'\log(2/\delta)}|y - x^T\hat{\theta}'|. \tag{4.4}$$

- Where

Let $\hat{\theta}$ *and* $\hat{\theta}'$ *be the ridge regression estimate with data set* $X \times \mathbf{y}$ *and* $[X, x] \times [\mathbf{y}, y]$ *and defined the out of sample leverage score* $\mu := x^T(X^TX + \lambda I)^{-1}x = x^T H^{-1} x$ *and in-sample leverage score* $\mu' := x^T[(X')^T X' + \lambda I]^{-1}x = x^T(H')^{-1}x$.

W. (2018) "Revisiting Differentially Private Linear Regression": https://arxiv.org/abs/1803.02596

Maximizing it so we have a bound that covers all individuals

Remark 11. *Let* $L := \|\mathcal{X}\|(\|\mathcal{X}\|\|\theta^*_\lambda\| + \|\mathcal{Y}\|)$, *The* OPS *algorithm for ridge regression with parameter* (λ, γ) *obeys* (ϵ, δ)*-pDP for each data set* (X, y) *and all target* (x, y) *with*

$$\epsilon = \sqrt{\frac{\gamma L^2 \log(2/\delta)}{\lambda + \lambda_{\min}}} + \frac{\gamma L^2}{2(\lambda + \lambda_{\min} + \|\mathcal{X}\|^2)} + \frac{(1 + \log(2/\delta))\|\mathcal{X}\|^2}{2(\lambda + \lambda_{\min})}.$$

- How to make it dataset-independent?

- It depends on just two quantities of interest.

How do we privately release the two quantities?

- The smallest eigenvalue has bounded global sensitivity

- The norm of the the Ridge regression estimate?

$$\Big| \|\hat{\theta}\| - \|\hat{\theta}'\| \Big| \leq \|\hat{\theta} - \hat{\theta}'\| = |y - x^T\hat{\theta}| \sqrt{x^T([X, x]^T[X, x] + \lambda I)^{-2} x}$$

Generalized Propose-Test-Release: Privately releasing per-instance DP bounds

- Your mechanism has parameter ϕ (e.g., noise-level, regularization), the data-dependent quantities $\psi(D, \phi)$.

- Generalizing PTR:
  1. Propose some parameter ϕ, work out the pDP $\epsilon_{\phi}(D, z)$
  2. Privately test if $\max_{z} \epsilon_{\phi}(D, z)$ is smaller than budget ϵ
  3. If so, run this mechanism with parameter ϕ
  4. Otherwise, return $\perp$

- Questions to ask when using this:
  - What if we do not know what parameter ϕ to choose?
  - How to run the private test?

The general recipe: “pDP to DP conversion” that allows calibrating ϕ to privacy budgets

- Your mechanism has parameter ϕ (e.g., noise-level, regularization), the data-dependent quantities $\psi(D, \phi)$.

- pDP function $\epsilon_\phi(D, z)$ depends the data

- We can often write $\max_z \epsilon_\phi(D, z)$ is also data dependent, but we can release a high-probability data-dependent upper bound $\tilde{\epsilon}_\phi(D) \geq \max_z \epsilon_\phi(D, z)$ differentially privately.

- Then we can calibrate the parameter ϕ according to the upper bound.

Checkpoint: two new recipes that generalizes PTR

- No restrictions on randomized algorithms.
- Release data-dependent quantities in the privacy loss RV.
- Privately test or release the data-dependent privacy loss accordingly.

(Based on an ongoing work.)

Remainder of the lecture

- Two representative methods in data-adaptive differentially private learning

  - NoisySGD and adaptive clipping

  - PATE and model-agnostic private learning

Noisy SGD with Adaptive Clipping

- NoisySGD
- Idea: As we train the models, most data points would've been classified correctly and the gradients are small. So we can use more aggressive clipping.
- Why not make it 90% percentile of the gradient norm?

Galen, Thakkar, McMahan, Ramaswamy etc.: "Differentially Private Learning with Adaptive Clipping"
https://arxiv.org/abs/1905.03871

Noisy SGD with Adaptive Clipping

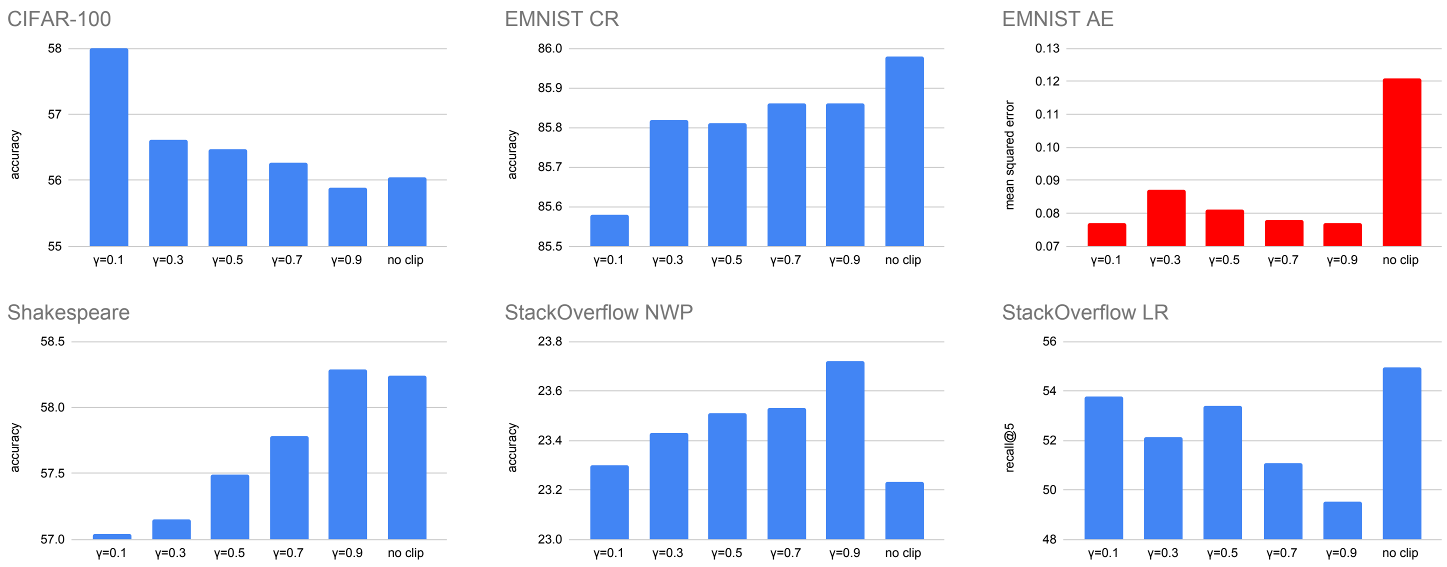

Figure 3: **Impact of clipping without noise.** Performance of the unclipped baseline compared to five settings of γ, from $\gamma = 0.1$ (aggressive clipping) to $\gamma = 0.9$ (mild clipping). The values shown are the evaluation metrics on the validation set averaged over the last 100 rounds. Note that the y-axes have been compressed to show small differences, and that for EMNIST-AE lower values are better.

Galen, Thakkar, McMahan, Ramaswamy etc.: "Differentially Private Learning with Adaptive Clipping"
https://arxiv.org/abs/1905.03871

PATE with SVT and large margin

The *PATE* Framework:

1. Randomly partition the private dataset into K splits.
2. Train one "teacher" classifier on each split.
3. Apply the K "teacher" classifiers on public data and *privately release* their majority votes as pseudo-labels.
4. Output the "student" classifier trained on the pseudo-labeled public data.

- Standard Gaussian mechanism release
- Alternative: SVT + Dist2Instability
  - Use add noise to a threshold.
  - If the margin > noisy-threshold,
    - release the exact value of the argmax
    - and continue
  - Otherwise
    - release nothing, update the threshold noise.

Bassily, Thakkar, Thakurta. Model-Agnostic Private Learning via Stability: https://arxiv.org/abs/1803.05101

Alternative way of adapting to large margins in PATE

- Just use Gaussian mechanism
- But work out a data-dependent DP losses

Theorem 6 (informal). *Let $\mathcal{M}$ be a randomized algorithm with (μ_1, ε_1)-RDP and (μ_2, ε_2)-RDP guarantees and suppose that given a dataset D, there exists* a likely *outcome i^* such that $\mathbf{Pr}\left[\mathcal{M}(D) \neq i^*\right] \leq \tilde{q}$. Then the* data-dependent *Rényi differential privacy for $\mathcal{M}$ of order $\lambda \leq \mu_1, \mu_2$ at D is bounded by a function of $\tilde{q}$, μ_1, ε_1, μ_2, ε_2, which approaches 0 as $\tilde{q} \to 0$.*

- Amplification by Large Margin of the voting scores.

Proposition 7. *For any $i^* \in [m]$, we have $\mathbf{Pr}\left[\mathcal{M}_\sigma(D) \neq i^*\right] \leq \frac{1}{2} \sum_{i \neq i^*} \operatorname{erfc}\left(\frac{n_{i^*} - n_i}{2\sigma}\right)$, where* erfc *is the complementary error function.*

Papernot et al. (2018) “Scalable Private Learning with PATE”
https://arxiv.org/abs/1802.08908

Adapting to "large margin" without using data-adaptive DP algorithm

- Select data points according to **active learning** rules
  - Disagreement-based Active Learning [See this excellent ICML tutorial: https://icml.cc/media/icml-2019/Slides/4341.pdf]
- Uses naïve Gaussian mechanisms based queries

Dataset	Method	# Queries	ϵ	$\epsilon_{\texttt{ex post}}$	Accuracy
real-sim	PSQ-NP	1,447	$+\infty$	$+\infty$	0.8234 ± 0.0014
	ASQ-NP	434	$+\infty$	$+\infty$	$\mathbf{0.8289} \pm 0.0008$
	PSQ	1,447	0.5	0.5	0.6355 ± 0.0065
	ASQ	434	0.5	0.5	$\mathbf{0.7389} \pm 0.0014$
	PSQ	1,447	1.0	1.0	0.7550 ± 0.0058
	ASQ	434	1.0	1.0	$\mathbf{0.8040} \pm 0.0009$
	PSQ	1,447	2.0	2.0	0.8025 ± 0.0037
	ASQ	434	2.0	2.0	$\mathbf{0.8231} \pm 0.0009$

Liu et al. (2021) "Revisiting Model-Agnostic Private Learning"
https://arxiv.org/abs/2011.03186

Expanding list of papers on data-dependent DP for learning

- Clustering: [k-means, k-medians, ...]
- Linear regression: [AdaOPS/AdaSSP]
- Statistical estimation: [mean, covariance]
- Statistical inference: [Hypothesis testing, OLS]
- Boosting: [Adapting to margin]
- Topic models: [Spectral LDA]

- Many more…

Good research directions

- Stronger, more practical, more adaptive DP algorithms:
  - Mechanism specific analysis (RDP, CDP, Privacy Profiles) of data-adaptive algorithms
  - Per-instance DP of more algorithms.

- The use of DP in novel context
  - e.g. Adaptive Data Analysis / preventing implicit overfitting
  - For fairness, for truthfulness in mechanism design
  - As a general smoothing trick that induces stability
  - …

- Practical implementation / empirical evaluation of DP
  - Not necessary new methodology. Just off-the-shelf tools are already sufficient for solving many problems!