PRIVACY PRESERVING MACHINE LEARNING

LECTURE 5: DIFFERENTIALLY PRIVATE STOCHASTIC GRADIENT DESCENT

Aurélien Bellet (Inria)
Master 2 Data Science, University of Lille
REMINDER: EMPIRICAL RISK MINIMIZATION (ERM)

- $D = \{(x_i, y_i)\}_{i=1}^n$: training points drawn i.i.d. from distribution $\mu$ over $\mathcal{X} \times \mathcal{Y}$
- Models $h_\theta : \mathcal{X} \rightarrow \mathcal{Y}$ parameterized by $\theta \in \Theta \subseteq \mathbb{R}^p$
- $L(\theta; x, y)$: loss of model $h_\theta$ on data point $(x, y)$
- $\hat{R}(\theta; D) = \frac{1}{n} \sum_{i=1}^n L(\theta; x_i, y_i)$: empirical risk of model $h_\theta$
- $\psi(\theta)$: regularizer on model parameters (e.g., $\ell_2$ norm)

**Empirical Risk Minimization (ERM)**

$$\hat{\theta} \in \arg \min_{\theta \in \Theta} [F(\theta; D) := \hat{R}(\theta; D) + \lambda \psi(\theta)]$$

where $\lambda \geq 0$ is a trade-off hyperparameter.
REMINDER: USEFUL PROPERTIES

- We typically work with loss functions that are **differentiable in $\theta$**: for $(x, y) \in \mathcal{X} \times \mathcal{Y}$, we denote the gradient of $L$ at $\theta$ by $\nabla L(\theta; x, y) \in \mathbb{R}^p$.

- We also like the loss function, its gradient and/or the regularizer to be **Lipschitz**.

**Definition (Lipschitz function)**

Let $l > 0$. A function $f$ is $l$-Lipschitz with respect to some norm $\| \cdot \|$ if for all $\theta, \theta' \in \Theta$:

$$|f(\theta) - f(\theta')| \leq l \| \theta - \theta' \|.$$

If $f$ is differentiable and $\| \cdot \| = \| \cdot \|_2$, the above property is equivalent to:

$$\| \nabla f(\theta) \|_2 \leq l, \quad \forall \theta \in \Theta.$$
It is also useful when the loss and/or regularizer are convex or strongly convex.

**Definition (Strongly convex function)**

Let $s \geq 0$. A differentiable function $f$ is $s$-strongly convex if for all $\theta, \theta' \in \Theta$:

$$f(\theta') \geq f(\theta) + \nabla f(\theta) ^\top (\theta - \theta') + \frac{s}{2} \| \theta - \theta' \|^2_2,$$

or equivalently:

$$\left( \nabla f(\theta) - \nabla f(\theta') \right) ^\top (\theta - \theta') \geq s \| \theta - \theta' \|^2_2,$$

For $s = 0$, we simply say that $f$ is convex.
Algorithm: DP-ERM via output perturbation $A_{DP-ERM}(D, L, \psi, \lambda, \epsilon, \delta)$

1. Compute ERM solution $\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^p} F(\theta)$
2. For $j = 1, \ldots, p$: draw $Y_j \sim \mathcal{N}(0, \sigma^2)$ independently for each $j$, where $\sigma = \frac{2\sqrt{2\ln(1.25/\delta)}}{n\lambda\epsilon}$
3. Output $\hat{\theta} + Y$, where $Y = (Y_1, \ldots, Y_p) \in \mathbb{R}^p$

Theorem (DP guarantees for DP-ERM via output perturbation)

Let $\epsilon, \delta > 0$ and $\Theta = \mathbb{R}^p$. For $\psi$ differentiable and 1-strongly convex, and $L(\cdot; x, y)$ convex, differentiable and 1-Lipschitz, $A_{DP-ERM}(\cdot, L, \psi, \epsilon, \delta)$ is $(\epsilon, \delta)$-DP.
1. Differentially Private SGD

2. Summary of DP-ERM results
DIFFERENTIALLY PRIVATE SGD
1. It requires **restrictive assumptions** on the loss function and regularizer

2. The sensitivity is likely to be **pessimistic** as it treats ERM as a black box
Another approach is to design differentially private ERM solvers.

Such a solver (optimization algorithm) must interact with the data only through DP mechanisms.

The idea is to perturb only the quantities accessed by a particular solver.
For simplicity, let us assume that $\psi(\theta) = 0$ (no regularization)

Denote by $\Pi_\Theta(\theta) = \arg\min_{\theta' \in \Theta} \|\theta - \theta'\|_2$ the projection operator onto $\Theta$

**Algorithm: Non-private (projected) SGD**

- Initialize parameters to $\theta^{(0)} \in \Theta$
- For $t = 0, \ldots, T - 1$:
  - Pick $i_t \in \{1, \ldots, n\}$ uniformly at random
  - $\theta^{(t+1)} \leftarrow \Pi_\Theta(\theta^{(t)} - \gamma_t \nabla L(\theta^{(t)}; x_{i_t}, y_{i_t}))$
- Return $\theta^{(T)}$

SGD is a natural candidate solver: simple, flexible, scalable, heavily used in ML

How to design a DP version of SGD?
• We have already seen ingredients to do this in previous lectures

• Assume that $L(\cdot; x, y)$ is $l$-Lipschitz with respect to the $\ell_2$ norm for any $(x, y) \in \mathcal{X} \times \mathcal{Y}$

• Then we know that for all $x, y, \theta$ we have $\|\nabla L(\theta; x, y)\| \leq l$

• Therefore, at any step $t$ of SGD, the $\ell_2$ sensitivity of $\nabla L(\theta^{(t-1)}; x_i, y_i)$ is bounded by $2l$ and we can use the Gaussian mechanism

• It feels like we can do better...
Theorem (Amplification by subsampling [Balle et al., 2018])

Let $\mathcal{X}$ be a data domain and $S : \mathcal{X}^n \rightarrow \mathcal{X}^m$ be a procedure such that $S(D)$ returns a random subset of $m$ records sampled uniformly without replacement from $D$. Let $\mathcal{A}$ be an $(\varepsilon, \delta)$-DP algorithm. Then $\mathcal{A} \circ S$ satisfies $(\varepsilon', \frac{m}{n} \delta)$-DP with $\varepsilon' = \ln \left(1 + \frac{m}{n} (e^\varepsilon - 1)\right)$.

- The amplification effect is due to the secrecy of the samples
- For simplicity of exposition, we will use the following approximation: when $\varepsilon \leq 1$, $\ln \left(1 + \frac{m}{n} (e^\varepsilon - 1)\right) \leq 2 \frac{m}{n} \varepsilon$ (but in practice the tight version above should be used!)
- The proof and results with other sampling schemes can be found in [Balle et al., 2018]
### Algorithm: Differentially Private SGD $\mathcal{A}_{DP-SGD}(D, L, \varepsilon, \delta)$

- Initialize parameters to $\theta^{(0)} \in \Theta$ (must be independent of $D$)
- For $t = 0, \ldots, T - 1$:
  - Pick $i_t \in \{1, \ldots, n\}$ uniformly at random
  - $\eta^{(t)} \leftarrow (\eta_{1}^{(t)}, \ldots, \eta_{p}^{(t)}) \in \mathbb{R}^{p}$ where each $\eta_{j}^{(t)} \sim \mathcal{N}(0, \sigma^{2})$ with $\sigma = \frac{16l\sqrt{\ln(2/\delta)\ln(2.5T/\delta n)}}{\varepsilon n}$
  - $\theta^{(t+1)} \leftarrow \Pi_{\Theta}\left(\theta^{(t)} - \gamma_{t}(\nabla L(\theta^{(t)}; x_{i_t}, y_{i_t}) + \eta^{(t)})\right)$
- Return $\theta^{(T)}$

- More data (larger $n$) $\rightarrow$ less noise added to each gradient
- More iterations (larger $T$) $\rightarrow$ more noise added to each gradient

### Theorem (DP guarantees for DP-SGD)

Let $\varepsilon \leq 1, \delta > 0$. Let the loss function $L(\cdot; x, y)$ be $l$-Lipschitz w.r.t. the $\ell_2$ norm for all $x, y \in \mathcal{X} \times \mathcal{Y}$. Then $\mathcal{A}_{DP-SGD}(\cdot, L, \varepsilon, \delta)$ is $(\varepsilon, \delta)$-DP.
Proof.

- Recall that for a query with $\ell_2$ sensitivity $\Delta$, achieving $(\varepsilon', \delta')$ with the Gaussian mechanism requires to add noise with standard deviation $\sigma' = \frac{\sqrt{2 \ln(1.25/\delta')} \Delta}{\varepsilon'}$

- So with $\Delta = 2l$, $\sigma = \frac{16l \sqrt{T \ln(2/\delta) \ln(2.5T/\delta n)}}{n \varepsilon}$, each noisy gradient is $\left(\frac{n \varepsilon}{4 \sqrt{2T \ln(2/\delta)}}, \frac{\delta n}{2T}\right)$-DP

- Now, taking into account the randomness in the choice of $i_t$ using privacy amplification by subsampling, each noisy gradient is in fact $\left(\frac{\varepsilon}{2 \sqrt{2T \ln(2/\delta)}}, \frac{\delta}{2T}\right)$-DP

- DP-SGD is an adaptive composition of $T$ DP mechanisms, so by advanced composition (using the simple corollary in lecture 3) we obtain that it is $(\varepsilon, \delta)$-DP
Theorem (Utility guarantees for DP-SGD [Bassily et al., 2014])

Let $\Theta$ be a convex domain of diameter bounded by $R$, and let the loss function $L$ be convex and $l$-Lipschitz over $\Theta$. For $T = n^2$ and $\gamma_t = O(R/\sqrt{t})$, DP-SGD guarantees:

$$\mathbb{E}[F(\theta^{(T)})] - \min_{\theta \in \Theta} F(\theta) \leq O\left(\frac{lR\sqrt{p \ln(1/\delta)} \ln^{3/2}(n/\delta)}{n\epsilon}\right).$$

If the objective $F$ is also $s$-strongly convex, then for $T = n^2$ and $\gamma_t = 1/st$ we have:

$$\mathbb{E}[F(\theta^{(T)})] - \min_{\theta \in \Theta} F(\theta) \leq O\left(\frac{l^2p \ln(1/\delta) \ln^2(n/\delta)}{s\epsilon^2 n^2}\right).$$

- The utility gap with respect to the non-private model reduces with $n$
- Privacy induces a larger cost for high-dimensional models
- We see notable differences between the convex and strongly convex cases
We will rely on a very general lemma giving convergence rates for SGD algorithms.

**Lemma ([Shamir and Zhang, 2013])**

Let $F$ be a convex function over a convex domain $\Theta$ with diameter bounded by $R$. Consider any SGD algorithm $\theta^{(t+1)} \leftarrow \Pi_\Theta(\theta^{(t)} - \gamma_t g_t)$ where $g_t$ satisfies $\mathbb{E}[g_t] = \nabla F(\theta^{(t)})$ and $\mathbb{E}[\|g_t\|^2] \leq G^2$. By setting $\gamma_t = \frac{R}{G \sqrt{t}}$, we have

$$\mathbb{E}[F(\theta^{(T)}) - \min_{\theta \in \Theta} F(\theta)] \leq 2RG \left( \frac{2 + \log T}{\sqrt{T}} \right).$$

If $F$ is also $s$-strongly convex, then setting $\gamma_t = \frac{1}{s t}$ gives

$$\mathbb{E}[F(\theta^{(T)}) - \min_{\theta \in \Theta} F(\theta)] \leq \frac{17G^2(1 + \log T)}{sT}.$$
Proof of the theorem.

- Denote by $g_t = \nabla L(\theta^{(t)}; x_i, y_i) + \eta^{(t)}$ the noisy gradient at step $t$

- Let us examine $E[g_t]$ and $E[\|g_t\|^2]$

- We have $E[g_t] = \frac{1}{n} \sum_{i=1}^{n} \nabla L(\theta^{(t)}; x_i, y_i) + E[\eta^{(t)}] = \nabla F(\theta^{(t)}; D)$, hence $g_t$ is an unbiased estimate of the gradient of the objective function at $\theta^{(t)}$

- Furthermore, since $\nabla L(\theta^{(t)}; x_i, y_i)$ and $\eta^{(t)}$ are independent and $L$ is $l$-Lipschitz:

\[
E[\|g_t\|^2] = E[\|\nabla L(\theta^{(t)}; x_i, y_i)\|^2] + E[\|\eta^{(t)}\|^2] \\
\leq l^2 + \rho \frac{256l^2 T \ln(2/\delta) \ln(2.5T/\delta n)}{\varepsilon^2 n^2}
\]
Proof of the theorem.

• It remains to plug our results in the previous lemma and to set $T$ appropriately.

• For the convex case, we get:

$$\mathbb{E}[F(\theta^{(T)}) - \min_{\theta \in \Theta} F(\theta)] \leq O\left( \frac{lR \ln T}{\sqrt{T}} + \frac{lR \sqrt{pT \ln(T) \ln(1/\delta) \ln(T/\delta n)}}{n \varepsilon \sqrt{T}} \right)$$

• For the $s$-strongly case, we get:

$$\mathbb{E}[F(\theta^{(T)}) - \min_{\theta \in \Theta} F(\theta)] \leq O\left( \frac{l^2 \ln T}{sT} + \frac{l^2 pT \ln(T) \ln(1/\delta) \ln(T/\delta n)}}{\varepsilon^2 n^2 sT} \right)$$

• In both cases, choosing $T = n^2$ balances the two terms ("optimization error" and "privacy error") and gives the result.
• In practice one should apply the tighter versions of amplification by subsampling and advanced composition to obtain better performance

• Using moments accountant [Abadi et al., 2016] or Rényi DP [Wang et al., 2019], one can further save a factor \(O(\sqrt{\ln T/\delta})\) in the composition and get better constants

• There are some straightforward extensions of DP-SGD:
  • Mini-batch version: same analysis applies with minor modifications
  • Regularization: can be readily incorporated into the algorithm
  • Non-differentiable loss: if \(L\) is only sub-differentiable (e.g., hinge loss, ReLU), one can use a subgradient instead of the gradient
  • Non-Lipschitz loss: if \(L\) is not Lipschitz (or the constant is hard to bound as in deep neural nets), one can use gradient clipping before adding the noise, see [Abadi et al., 2016]

• It is also possible to improve the \(O(n^2)\) gradient complexity, e.g., down to \(O(n \log n)\) using variance reduction techniques [Wang et al., 2017]
SUMMARY OF DP-ERM RESULTS
DP-ERM: SOME RESULTS FOR THE STRONGLY CONVEX CASE

- Assume convex 1-Lipschitz loss with 1-Lipschitz gradient, 1-strongly convex objective
- Tight lower bound for \((\varepsilon, \delta)\)-DP: \(\Omega(\min\{1, \frac{p}{n^2\varepsilon^2}\})\)
- Upper bounds (ignoring multiplicative dependence on \(\log(1/\delta)\)):

<table>
<thead>
<tr>
<th>Paper</th>
<th>Technique</th>
<th>Excess risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Chaudhuri et al., 2011]</td>
<td>Black box output perturbation</td>
<td>(O\left(\frac{p}{n^2\varepsilon^2}\right))</td>
</tr>
<tr>
<td>[Chaudhuri et al., 2011]</td>
<td>Objective perturbation</td>
<td>(O\left(\frac{p}{n^2\varepsilon^2}\right))</td>
</tr>
<tr>
<td>[Bassily et al., 2014]</td>
<td>Gradient perturbation (this lecture)</td>
<td>(O\left(\frac{p \ln^2(n)}{n^2\varepsilon^2}\right))</td>
</tr>
<tr>
<td>[Wang et al., 2017]</td>
<td>Gradient perturbation with MA + VR</td>
<td>(O\left(\frac{p \ln(n)}{n^2\varepsilon^2}\right))</td>
</tr>
</tbody>
</table>

(MA: Moments Accountant, VR: Variance Reduction)
DP-ERM: SOME RESULTS FOR THE CONVEX CASE

- Assume convex 1-Lipschitz loss with 1-Lipschitz gradient

- **Tight lower bound** for $(\varepsilon, \delta)$-DP: $\Omega(\min\{1, \frac{\sqrt{p}}{n\varepsilon}\})$

- **Upper bounds** (ignoring multiplicative dependence on $\log(1/\delta)$):

<table>
<thead>
<tr>
<th>Paper</th>
<th>Technique</th>
<th>Excess risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Chaudhuri et al., 2011]</td>
<td>Objective perturbation</td>
<td>$O\left(\frac{\sqrt{p}}{n\varepsilon}\right)$</td>
</tr>
<tr>
<td>[Bassily et al., 2014]</td>
<td>Gradient perturbation (this lecture)</td>
<td>$O\left(\frac{\sqrt{p} \ln^{3/2}(n)}{n\varepsilon}\right)$</td>
</tr>
<tr>
<td>[Wang et al., 2017]</td>
<td>Gradient perturbation with MA + VR</td>
<td>$O\left(\frac{\sqrt{p}}{n\varepsilon}\right)$</td>
</tr>
<tr>
<td>[Feldman et al., 2018]</td>
<td>Gradient perturbation with amp. by iteration</td>
<td>$O\left(\frac{\sqrt{p}}{n\varepsilon^2}\right)$</td>
</tr>
</tbody>
</table>

- More results can be found in [Bassily et al., 2014, Wang et al., 2017]

- For problems with more structure, other gradient perturbation algorithms and lower bounds exist, see e.g. [Talwar et al., 2015, Mangold et al., 2021]
Deep learning with differential privacy.
In CCS.

Privacy amplification by subsampling: tight analyses via couplings and divergences.
In NeurIPS.

In FOCS.

Differentially Private Empirical Risk Minimization.

Privacy Amplification by Iteration.
In FOCS.

Differentially Private Coordinate Descent for Composite Empirical Risk Minimization.
Stochastic Gradient Descent for Non-smooth Optimization: Convergence Results and Optimal Averaging Schemes.
In ICML.

Nearly Optimal Private LASSO.
In NIPS.

In NIPS.

Subsampled Renyi Differential Privacy and Analytical Moments Accountant.
In AISTATS.