Rényi DP & composition results

Recall that differential privacy was intuitively defined by ensuring M_D & $M_{D'}$ are "close", according to HS divergence.

Why not other "distance" measures?

We already know that TV doesn't work.

It turns out that Rényi divergence is a good candidate.

Definition. (Mironov 2017) We say that a randomized mechanism M is (α, ε)-Renyi DP (or (α, ε)-RDP) for $\alpha > 1$ & $\varepsilon \geq 0$ if

$$D_\alpha(M_D \| M_{D'}) \leq \varepsilon$$

the main motivation [which made RDP extremely popular in AI] is the following result:

Theorem. For any $\alpha > 1$, we have

$$D_\alpha(P_X \| Q_X) + \min_x D_\alpha(P_{Y|X=x} \| Q_{Y|X=x}) \leq D_\alpha(P_{XY} \| Q_{XY}) \leq D_\alpha(P_X \| Q_X) + \max_x D_\alpha(P_{Y|X=x} \| Q_{Y|X=x})$$

In particular, if $P_{XY} = P_X \cdot P_Y$ & $Q_{XY} = Q_X \cdot Q_Y$, then:

$$D_\alpha(P_{XY} \| Q_{XY}) = D_\alpha(P_X \| Q_X) + D_\alpha(P_Y \| Q_Y).$$

Proof. See Lemma 2.2 of "Concentrated Differential Privacy" by Bun & Steinke.

this simple result leads to the following simple composition result that resembles basic composition.

Theorem. Let M^1 be (α, ε_1)-RDP & M^2 be (α, ε_2)-RDP. Then their (adaptive) composition is $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.

Proof. Simple application the previous theorem.

According to this result, k-fold (adaptive) composition of any (α, ε)-RDP is $(\alpha, k\varepsilon)$-RDP.

Limitation of RDP. (α, ε)-RDP guarantee doesn't enjoy any operational interpretation, in terms of hypothesis testing.

In fact, it was shown* that D_α with $\alpha > 1$ has nothing to do with binary hypothesis testing performance.

* Read "Hypothesis testing interpretation & RDP" by Balle et al., AISTAT 2020.

How to fix it? RDP privacy guarantee needs to be converted back to approximate DP.

Theorem. If M is (α, ξ)-RDP with $\alpha > 1$, then it is (ε, δ)-DP with $\delta \in (0,1)$ & $\varepsilon = \xi - \frac{1}{\alpha - 1} \log \frac{1}{\delta}$.

Mironov

Moments accountant: was the state-of-the-art technique for studying composition of iterative algorithms used for training AI models (such as SGD). We discuss it further later.

Statistical Learning

Let $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a dataset of features x_i & labels y_i. For instance feature might be images, texts or medical records of individuals & labels are the type of animal in the image (cat vr. dog), or the diagnostic.

The main goal in machine learning is to discover the relationship between features & labels. This allows us to come up with a classifier. Think of classifier as a conditional distribution $P_{Y|X}$.

Any such classifiers are modelled by parameters $\theta \in \mathbb{R}^d$.

To find the "best" classifier that fits a dataset, we must find $\theta \in \mathbb{R}^d$ that describes the relationship between x_i & y_i for all $i \in \{1, 2, \ldots, n\}$.

To mathematically formulate this goal, we take a problem-specific **loss function** l:

$l(\theta, (x, y))$ quantifies the loss associated with representing the relationship between feature x & label y using a classier parameterized by θ.

Example:

1- Linear regression: think of linear regression as a continuous classifier.

$x_i \in \mathbb{R}^d$, $y_i \in \mathbb{R}$, $\theta \in \mathbb{R}^d$

regressor is parameterized by $\theta \in \mathbb{R}^d$ as $\langle \theta, x \rangle$ (inner product)

loss function: $\ell(\theta, (x_i, y_i)) = |y - \langle \theta, x \rangle|^2$

* Goal is to find θ that minimizes loss for each point: $\sum_{i=1}^{n} \ell(\theta, (x_i, y_i))$

2- Logistic regression: Here, label $y \in \{-1, +1\}$ binary classification

classifier is parametrized by $\theta \in \mathbb{R}^d$ as

$$P_{Y|X=x}(+1) = \frac{1}{1 + e^{-\langle \theta, x \rangle}}.$$

the goal is to find θ such that

$\frac{1}{1 + e^{-\langle \theta, x_i \rangle}}$ is maximized when $y_i = +1$

& $1 - \frac{1}{1 + e^{-\langle \theta, x_i \rangle}}$ is maximized when $y_i = -1$

$$1 - \frac{1}{1 + e^{-\langle \theta, x_i \rangle}} = \frac{1}{1 + e^{\langle \theta, x_i \rangle}}$$

for all data points in the dataset.

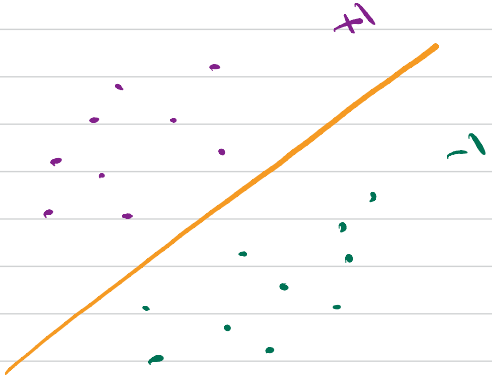
thus, the loss function is defined to be

$$\ell(\theta; (x, y)) = \log\left(1 + e^{-y\langle \theta, x \rangle}\right)$$

3_ Support Vector machine (SVM): Binary label $y_i \in \{-1, +1\}$

$P_{Y|X=x}(+1) = 1$ if $\langle \theta, x \rangle > 0$

« deterministic classifier »



The goal is to find $\theta \in \mathbb{R}^d$ & $b \in \mathbb{R}$

such that

$$y_i \langle \theta, x_i \rangle \geq 0$$

So any reasonable loss increases with

$-y \langle \theta, x \rangle$.

One potential loss: $\ell((\theta, x), y)) = \mathbb{1}_{\{y \langle \theta, x \rangle < 0\}}$

↑ called Zero-one loss

Since this function is not differentiable & not convex, we usually consider this loss:

$$\ell(\theta, (x,y)) = \max(0, 1 - y\langle x, \theta \rangle)$$

↑ called hinge loss



4. Geometric median: Unsupervised learning, so label is not given.

We are interested in median:

$$\ell(\theta, (x,y)) = \|\theta - x\|_2$$

of $\sum_{i=1}^{n} \ell(\theta, (x_i, y_i))$

thus, the minimizer θ^* is the geometric median of all feature.

In all these examples, we are interested in the following optimization problem:

$$\theta^* := \arg\min_{\theta \in \mathbb{R}^d} \sum_{i=1}^{n} \ell(\theta, (x_i, y_i))$$

ERM (Empirical risk minimization)

Revealing θ^* might compromise privacy!

- For instance, if $\ell(\theta, (x, y)) = \|\theta - x\|_2$, then if the number of datapoints is odd, then θ^* is equal

to one of the point.

- In SVM, the optimal θ^* is close to one datapoint

- Model inversion attack

