5 Privacy in Machine Learning

Note takers: Ahmadipour, El Ahmad, Jose, Lachi, Lalanne, Oukfir, Nesterenko, Ogier, Siviero, Valla

Instructor: Rachel Cummings (Columbia University)

5.1 Introduction

Privacy considerations arise as soon data is collected on individuals, on group on individuals, on moral personas, ... More specifically, we look at the setup where one processes data D through a mechanism $\mathcal{M}$ which can be anything from data publication, basic statistics computation, decision rule learning, complex machine learning tasks, ..., and wants the result $\mathcal{M}(D)$ to be made public. The natural question on a privacy standpoint is whether the mechanism $\mathcal{M}$ can be "reverted" in order to learn sensitive information from D. For instance, if $\mathcal{M}$ is the identity function, the publication of $\mathcal{M}(D)$ leaks full information about D and even though the notion of privacy is not rigorously defined yet, we can intuitively qualify such mechanism as "non-private".

This manuscript is a transcription of Prof. Rachel Cummings' lecture titled *Privacy in Machine Learning* that was given at the 2022 Spring School of Theoretical Computer Science at the CIRM, Marseille, France. Any error in this document may be due to its transcription and cannot be imputed to Prof. Cummings.

The lecture organizes as follows:

5.2 Defining privacy - Lecture 1

Even though the notion of privacy might seem natural at first, it is important to give it a good definition. We will start by trying to answer the question *What is privacy ?*

Attempt 1. *Privacy is about protecting identities.* This definition is natural. Something is private if it doesn't allow identifying you. As a result, it might be natural to consider that an algorithm is private if and only if it doesn't leak personally identifiable information (PPI). In practice, it is the main definition of privacy on a legal point of view. For instance, the French RGPD regulation instances (CNIL) consider that a mechanism is private when it makes a *sufficient effort* in hiding the identities. However, the more we look into it, the less convincing this definition becomes. First because it is extremely subjective, but mainly because it only shifts the problem. Indeed, what could be considered PPI or not? For instance, the last names and first names of people from a database seem to be natural PPI's. But what about their sum? Their encoding on a different alphabet? The application of any function on them? What about the correlation with other information such as the zip code, the income or the number of children? As a result, this definition has shown many failures in the past. For instance, research has shown that the search history of people can fully identify them, even with anything considered PPI's at the time removed (https://www.nytimes.com/2006/08/09/technology/09aol.html). On the other hand, it has also been shown that removing the PPI's can block inference and learning (Dwork et al.) and can only result in noise. As a result, this definition is better than nothing, but it is far from being future-proof, it both isn't really *private* while still partially blocking learning, and it requires a lot of legal effort in order to classify what is identifiable.

Attempt 2. *Privacy is about protecting people's freedoms from harm.* This definition is much stronger than the previous one. However, by the absolute aspect of this promise, it forces $\mathcal{M}(D)$ being independent of D. For instance, if researchers were to find correlations between smoking and lung cancer while not being able to learn if their patient smoked or not (in order to protect them from loosing their insurance), it would be a hard task. This definition of privacy thus has the drawback of completely blocking inference and learning.

Attempt 3. *Privacy is when almost no more information can be obtained with an analysis on the same dataset without a person's data.* This definition of privacy is interesting. Indeed, one can deduce the private information on an individual of D from $\mathcal{M}(D)$ if this individual has a huge impact on the result, i.e. when the result would have been significantly different without its information. As a result, privacy is obtained when $\mathcal{M}(D)$ is relatively invariant up to the addition or removal of any element of D. This definition of privacy will be adopted and rigorously defined through the concept of *differential privacy* in the rest of this lecture because it is the most future-proof and usable (even if it is still not clear for now) definition of privacy that research has come up with up to this day.

5.3 Differential Privacy - Lecture 1

The privacy of the mechanism $\mathcal{M}$ is achieved through randomization of its output. Formally, for $\varepsilon, \delta \geq 0$,

Definition. [DMNS '06] An algorithm $\mathcal{M} : T^n \to R$ is (ε, δ)-differentially private if $\forall$ neighboring $D, D' \in T^n$ and $\forall S \subseteq R$,

$$P[\mathcal{M}(D) \in S] \leq e^{\varepsilon} P\left[\mathcal{M}\left(D'\right) \in S\right] + \delta$$

where the randomness is taken on the coin tosses of $\mathcal{M}$.

Note that this definition bounds the "max amount" that one person's data can change the output of a computation. Furthermore, it is a worst case over all pairs of neighboring datasets. In particular,

- it doesn't matter what everyone else's data are,
- it doesn't matter what data you have,
- it doesn't depend on the future usage of $\mathcal{M}(D)$,
- if your data has huge influence, it will be hard to distinguish from your neighbors.

Furthermore, differential privacy does not block learning "DP addresses the paradox of learning nothing about an individual while learning useful information about a population. It is a definition, not an algorithm."- The Algorithmic Foundations of Differential Privacy, Dwork and Roth.

5.3.1 The role of the privacy parameters

This definition of privacy relies on two *privacy parameters*, ε and δ. They both impact how private the resulting mechanism is, but they do not play a symmetric role.

The role of ε. If a mechanism is (ε, δ)-DP, it is also (ε', δ)-DP if $\varepsilon' > \varepsilon$. As a result, the smaller ε, the stronger the constraint on privacy. The two following limit behaviors arise:

- $\varepsilon = 0$: Perfect privacy, where the result cannot depend at all on the data. As a result, no learning is possible.
- $\varepsilon = +\infty$: No privacy since the constraint vanishes. Privacy is no longer implied by the definition.

We want to be somewhere in the middle and the "correct "choice of ε is an open question depending on the sensitivity of the data.

The role of δ. Similarly, we can observe that the smaller δ, the stronger the privacy guarantees. δ differs from ε because:

- It gives a small additive slack in the privacy guarantee.

- It allows for a family of output distributions that are not all absolutely continuous with respect to each other. Imagine D, D' are neighboring databases and say $P[M(D) \in S] > 0$ and $P\left[M\left(D'\right) \in S\right] = 0$. Without δ :

$$0 < P[M(D) \in S] \leq e^{\varepsilon} P\left[M\left(D'\right) \in S\right] \leq e^{\varepsilon} \cdot 0 = 0$$

- Even with uniform support, it allows for an easier mechanism design.

In order to tune δ, we can fall back on the following observations and interpretations of this parameter:

- δ may be viewed as the probability under which the output mechanism does not respect the ε-DP guarantee.
- Hence, δ may be viewed as a *relaxation* term.
- If $\delta = 1$ then we're back to no privacy, even for $\varepsilon = 0$:

$$P[M(D) \in S] \leq e^{0} P\left[M\left(D'\right) \in S\right] + 1$$

- We have to take $\delta \ll \frac{1}{n}$. Indeed, when $\varepsilon = 0$ (which should give full privacy when $\delta = 0$), one can easily check that the mechanism that picks a random person from the database and output their data is $(0, 1/n)$-DP.

Remark: One might think that the definition of differential privacy is arbitrary, and it is. However, it is becoming increasingly adopted because this is the best that has been proposed to this date. Indeed, it ensures strong privacy guarantees while allowing for a nice algebra of private mechanisms (as we will see later). As a consequence, it is both conceptually powerful and handy, in a way that wasn't matched by previous definitions (such as k-anonymity).

Remark: The randomization of the output of the mechanism is at the core of this definition. Besides, one can easily check that trying to obtain privacy with a mechanism that is pointwise almost surely constant under $(\varepsilon, 0)$-DP results in a mechanism that is constant on all databases. Hence, one must be willing to pay a pointwise variance in order to obtain privacy.

5.3.2 Algebra of private mechanisms:

Private mechanisms come with three handy properties of post-processing, composition and group privacy that make them usable in practice.

Post-processing DP is immune to post-processing: If $M(D)$ is (ε, δ)-differentially private and f is any function (possibly stochastic), then $f(M(D))$ is (ε, δ)-differentially private. To put it simply, it is impossible to compute a function of the output of the private algorithm and make it "less" private. "No adversary (function f) can break the privacy guarantee "

Composition DP is robust under composition: If $M_1, \ldots, M_k$ are (ε, δ)-differentially private, then: $M(D) \equiv (M_1(D), \ldots, M_k(D))$ is $(k\varepsilon, k\delta)$-differentially private.

If multiple analyses are performed on the same data, as long as each one satisfies DP, all the information released taken together will still satisfy DP (albeit with a degradation in the parameters) This result quantifies the common heuristic: Privacy degrades gracefully as more computations are performed on the same dataset. The linear scaling in both ε and δ can be further improved via advanced composition: If $M_1, \ldots, M_k$ are (ε, δ)-differentially private and adaptively chosen, then: $M(D) \equiv (M_1(D), \ldots, M_k(D))$ is $(\varepsilon', k\delta + \delta')$-differentially private for

$$\varepsilon' = \varepsilon\sqrt{2k \log \frac{1}{\delta'}} + k\varepsilon \left(e^{\varepsilon} - 1\right) = \theta(\sqrt{k}\varepsilon)$$

Composition allows composing *simple* private procedures in order to obtain *complex* private algorithms.

Group Privacy Privacy guarantee depends on the group size: If two datasets D, D' differ in k entries and M is (ε, δ)-differentially private, then for all outputs S :

$$\mathbb{P}[M(D) \in S] \leq e^{k\varepsilon}\mathbb{P}\left[M\left(D'\right) \in S\right] + ke^{\varepsilon(k-1)}\delta.$$

In other words, DP guarantees for *individuals* generalizes to DP guarantees for *communities*.

5.3.3 Neighboring databases

Note that for now, we did not properly define the notion of neighboring databases. Usually, we say that two databases are neighbors iff their content differs on at most one person's data. This informal definition can take multiple forms depending on the structure of the database.

- If the databases $\boldsymbol{x}$ and $\boldsymbol{y}$ are order-sensitive and of fixed size n, we usually say that x and y are neighbors when $\|\boldsymbol{x} - \boldsymbol{y}\|_0 \leq 1$. Databases are then compared according to their order sensitive Hamming distance.
- If the databases $\boldsymbol{x}$ and $\boldsymbol{y}$ are order-insensitive and of fixed size n, we usually say that x and y are neighbors when $\inf_\sigma \|\boldsymbol{x} - \sigma(\boldsymbol{y})\|_0 \leq 1$ where σ is any permutation that permutes the entries of $\boldsymbol{y}$. Not that if those databases are built on a countable set, this definition is equivalent to $\|h(\boldsymbol{x}) - h(\boldsymbol{y})\|_1 \leq 2$ where the function h transforms a database into its histogram (i.e. the vector counting the occurrences of the elements). Databases are then compared according to their order insensitive Hamming distance.
- If the databases $\boldsymbol{x}$ and $\boldsymbol{y}$ are order-insensitive and of possibly arbitrary sizes $n_{\boldsymbol{x}}$ and $n_{\boldsymbol{y}}$, we usually say that x and y are neighbors when $\|h_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{x}) - h_{\boldsymbol{x},\boldsymbol{y}}(\boldsymbol{y})\|_1 \leq 1$ where $h_{\boldsymbol{x},\boldsymbol{y}}$ refers to the histogram function that builds on the supports of $\boldsymbol{x}$ and $\boldsymbol{y}$ (which is countable). Databases are then compared according to their size insensitive Hamming distance.

Independently of the definition, we write $\boldsymbol{x} \sim \boldsymbol{y}$ when $\boldsymbol{x}$ and $\boldsymbol{y}$ are neighbors. All those definitions are not equivalent, but it is often clear which one to use depending on the setup.

Most of the results do not depend on the definition of neighboring databases, but when they do, it will be specified.

5.4 Private Mechanism Design - Lecture 1

This section presents simple building blocks for designing private mechanisms.

5.4.1 Laplace Mechanism

Given a unction f defined on a set of databases and valued in a real vector space, how can one mimic the behavior of f with a private mechanism? The Laplace mechanism gives a simple answer to this question by adding Laplace noise to the expected result scaled to the *sensitivity* of f.

Definition. The sensitivity of a function f is defined as

$$\Delta f = \max_{\boldsymbol{x} \sim \boldsymbol{y}} \|f(\boldsymbol{x}) - f(\boldsymbol{y})\|_1 \ .$$

Examples.

- If f counts the number of people with blue eyes, $\Delta f = 1$.
- If f is a histogram function built on a finite quantization of the data space, $\Delta f = 1$ with size-insensitive neighboring definition and $\Delta f = 2$ otherwise.
- If f is an averaging function, $\Delta f = \infty$ generally. however, if the data points live in set of l_1 diameter D, $\Delta f = D/n$ with the size-sensitive neighboring definitions and $\Delta f = D$ with the size-insensitive neighboring definition.

Laplace Mechanism - Definition The Laplace mechanism for f with privacy parameter ε is defined as

$$\mathcal{M}_L(\boldsymbol{x}, f, \varepsilon) = f(\boldsymbol{x}) + [\mathrm{Lap}(0, \Delta f/\varepsilon)]$$

where $[\mathrm{Lap}(0, \Delta f/\varepsilon)]$ refers to a vector (of size the output dimension) of i.i.d. random variables with centered Laplace distributions of standard derivation $\Delta f/\varepsilon$.

The structure of the noise allows for *pure* differential privacy (i.e. $\delta = 0$).

Theorem 5.1: Laplace Mechanism - Privacy

$\mathcal{M}_L(\cdot, f, \varepsilon)$ is $(\varepsilon, 0)$-differentially private.

Proof. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two neighboring databases. $\mathcal{M}_L(\boldsymbol{x}, f, \varepsilon)$ and $\mathcal{M}_L(\boldsymbol{y}, f, \varepsilon)$ have distributions that are absolutely continuous with respect to Lebesgue measure that are strictly positive almost everywhere. We may compare the ratio of these densities.

$$\begin{aligned}
\frac{\mathbb{P}[\mathcal{M}_L(\boldsymbol{x}, f, \varepsilon) = z]}{\mathbb{P}[\mathcal{M}_L(\boldsymbol{y}, f, \varepsilon) = z]} &= \frac{\mathbb{P}[[\text{Lap}(0, \Delta f/\varepsilon)] = z - f(\boldsymbol{x})]}{\mathbb{P}[[\text{Lap}(0, \Delta f/\varepsilon)] = z - f(\boldsymbol{y})]} \\
&= \frac{\Pi_i \mathbb{P}[\text{Lap}(0, \Delta f/\varepsilon) = z_i - f(\boldsymbol{x})_i]}{\Pi_i \mathbb{P}[\text{Lap}(0, \Delta f/\varepsilon) = z_i - f(\boldsymbol{y})_i]} \\
&= \frac{\Pi_i \frac{\varepsilon}{2\Delta f} e^{-\frac{\varepsilon |f(\boldsymbol{x})_i - z_i|}{\Delta f}}}{\Pi_i \frac{\varepsilon}{2\Delta f} e^{-\frac{\varepsilon |f(\boldsymbol{y})_i - z_i|}{\Delta f}}} = \Pi_i e^{-\frac{\varepsilon(|f(\boldsymbol{x})_i - z_i| - |f(\boldsymbol{y})_i - z_i|)}{\Delta f}} \\
&\leq \Pi_i e^{-\frac{\varepsilon |f(\boldsymbol{x})_i - f(\boldsymbol{y})_i|}{\Delta f}} = e^{-\frac{\varepsilon \sum_i |f(\boldsymbol{x})_i - f(\boldsymbol{y})_i|}{\Delta f}} \\
&= e^{-\frac{\varepsilon \|f(\boldsymbol{x}) - f(\boldsymbol{y})\|_1}{\Delta f}} \leq e^{\varepsilon} \, .
\end{aligned}$$

So for any Borel set S,

$$\begin{aligned}
\mathbb{P}[\mathcal{M}_L(\boldsymbol{x}, f, \varepsilon) \in S] &= \int \mathbb{P}[\mathcal{M}_L(\boldsymbol{x}, f, \varepsilon) = z] dz \\
&\leq e^{\varepsilon} \int \mathbb{P}[\mathcal{M}_L(\boldsymbol{y}, f, \varepsilon) = z] dz = e^{\varepsilon} \mathbb{P}[\mathcal{M}_L(\boldsymbol{y}, f, \varepsilon) \in S] \, ,
\end{aligned}$$

which concludes the proof. □

Furthermore, the tail bounds of the Laplace distribution give the following utility guarantee:

Theorem 5.2: Laplace Mechanism - Accuracy

$$\mathbb{P}\left[\|f(x) - y\|_1 \leq \log\left(\frac{d}{\beta}\right) \cdot \left(\frac{\Delta f}{\varepsilon}\right)\right] \geq 1 - \beta$$

where d is the output dimension.

This is our first example of a privacy-utility tradeoff. With the Laplace mechanism, the higher the privacy guarantees are, the more degraded the utility is. Also, we can notice that the higher the sensitivity, the lower the utility.

5.4.2 Exponential Mechanism

The Laplace mechanism works great when the output space is a real vector space and when the utility of the output can be measured with the l_1 norm. But what if the output space has a different structure (ex texts) or what if the utility does not depend directly on the l_1 norm? The exponential mechanism solves this problem by allowing mechanism design with an arbitrary utility function.

The exponential mechanism has to assign a numeric score to each possible output

Assign a specific numeric score to each possible output.

Quality of outcome measured by score function: $q : \mathbf{N}^{|X|} x \mathcal{R} \rightarrow \mathbf{R}$ where q(x,r) is a measure of how good outcome r would be on database x

Choice of q should depend on application

Reasonable quality score?

Smooth degradation of outputs.

Score function sensitivity

Definition:

The sensitivity of a score function $q : \mathbf{N}^{|X|} x \mathcal{R} \rightarrow \mathbf{R}$
$\Delta q = \max r \in \mathcal{R} \max x, y$ neighbors

Exponential Mechanism [MT07]

Definition: Given a quality score q:

Essentially we do a "biased sampling" with an exponential bias.

Example:

Most common eye color? X = brown, blue, green $x \in \mathbf{N}^{|X|}$ database of eye colors $\mathcal{R} = X$ $q(x, r)$ = # people in database x with eye color r Δq = 1 as each person can have at most one eye color

Theorem 5.3: MT'07

The exponential Mechanism M is ε differentially private

$$\frac{\mathbb{P}[\mathcal{M}_E(x, q, \varepsilon) = r]}{\mathbb{P}[\mathcal{M}_E(y, q, \varepsilon) = r]} \leq e^{\varepsilon}$$

Proof.

$$\frac{\mathbb{P}[\mathcal{M}_E(x, q, \varepsilon) = r]}{\mathbb{P}[\mathcal{M}_E(y, q, \varepsilon) = r]} = (definitionofexponentialmech) = (lawofexponents, sameasproofinLaplacemech) = ...firs$$

The first term is similar to what we saw in the Laplace Mechanism, so suing the same techniques we can show that:

This means we can swap x and y at the above cost. So, for the second term,

□

Accuracy:

$$\mathbb{P}[q(x,r) - \max r^{'} \in \mathcal{R}q(x,r^{'} \leq \frac{2\Delta q \cdot ln(|\mathcal{R}|/\beta)}{\varepsilon}] \leq \beta$$

High probability to pick an outcome that is close to the best possible outcome.

Best possible means highest quality score

Close depedns on high probabiity guarantee.

Exponential Privacy Accuracy trade-off

5.5 DP and online/adaptive statistics

A ε-DP algorithm is more noisy, but does this hurt generalization? Training score is worse, but this can also prevent overfitting.

5.5.1 DP and generalization

Theorem 5.1. *An ε-DP algorithm cannot overfit by more than ε*

We want the learning with DP samples to be (almost) as good as with the underlying distribution (not compared to the ground truth).

Reminder (Group Privacy) If S, S' differ in k elements, then $k\varepsilon$-privacy

DP private learners generalize well Notions of generalization:

- DP generalization: "similar samples should have similar output." DP-guarantee are strong worst case guarantee
- Weaker notion: Robust Generalization "no adversary can use the output to find a hypothesis that overfits"
- Stronger notion: Perfect Generalization "output reveals nothing about the sample". (Does not compare against a sample changed by one, but against the true underlying distribution. Means you are perfectly generalizing from the sample)

Why don't we change DP def to include distribution? eg for some rare databases, provide weaker privacy. However rare events are precisely the ones we are trying to protect. This is not an issue here

5.5.2 DP and adaptive analysis

How to do data analysis in a robust way?

What can go wrong? To learn global truth, the agent sends multiple queries sequentially, adapting new queries depending on the previous answer. [DFHPRR15] This can cause overfitting.

Particularly a risk in fields where scientists share one dataset (*eg* astronomy, historical datasets in economy)

`AboveNoisyThreshold` for multiple threshold queries [DNPR '10]

This is a DP algorithm for detecting which queries in a stream have answer above a given threshold.

input: database X, query stream $\{f_1, \dots\}$ with sensitivity Δ, privacy parameter ε and threshold T.
$\hat{T} := T + \mathrm{Lap}(2\Delta/\varepsilon)$;
for *each query* f_i **do**
  $Z_i \sim \mathrm{Lap}(4\Delta/\varepsilon)$ *(we add noise twice!)*;
  if $f_i(x) + Z_i > \hat{T}$ **then**
    output `Above` and halt
  else
    output `Below`
  end
end

Remarks This Algorithm compares noisy answer against a noisy threshold (fixed in advance). It can be proven $(\varepsilon, 0)$ DP and satisfies a composition privacy for k queries with only $\varepsilon = \log k$, can answer exponentially many queries (by composition theorem)! (vs composition of queries gives k or $\sqrt{k}$). Finally, ANT halts once it finds a single above threshold query, we need another algorithm if we would like to find multiple above threshold queries.

`SparseVector` to do threshold queries and do something with the results above threshold

Combine ANT and Laplace mechanism to release the answers.

Applicable to many problems

`Reusable Holdout`

Randomly partition D in training D_t and holdout D_h

When training a model, we only test generalization when testing on holdout, but this is true if holdout is used only once! (it needs to be considered as a fresh sample). The idea here is to access holdout only through DP algorithm, then no overfitting on holdout.

Input: training set S_t, holdout set S_h, threshold T, tolerance τ, budget B.
$\hat{T} := T + \text{Lap}(4\tau)$
For each query $\varphi : X \to [0,1]$:
if $B < 1$ **then**
  output `Below` and halt
else
  if $|E_{S_h}(\varphi) - E_{S_t}(\varphi)| > \hat{T} + Lap(8\tau)$ **then**
    output $E_{S_h}(\varphi) + \text{Lap}(2\tau)$
    $B = B - 1$
    $T = T + \text{Lap}(4\tau)$
  else
    output $E_{S_t}(\varphi)$
  end
end

Same algo as SV with

- check if answer on holdout is close to answer on training set (*i.e.* above noisy threshold)

  if no release noisy answer on holdout

  otherwise just release answer on training

- As in SV, we have a privacy budget, counting if we can still access holdout

DP and accuracy are quantified (see slides).

Theorem 5.2. *Thresholdout is $B/(\tau n)$-DP. For all adaptively chosen queries $\{\varphi_1, \ldots, \varphi_m\}$, for all i such that a_i isn't "bellow" the threshold, for all $t > 0$:*

$$\Pr[|a_i - \Pr(\varphi_i)| > T + (t+1)\tau] \leq 6\exp(-\tau^2/2) + \exp(-t/8).$$

5.5.3 DP and sequential hypothesis testing

Try to address the replication crisis, how to get meaningful p-values?

As usual, observe $x_1, \cdot, x_n$ and have null hypothesis H_0 and interesting alternative hypothesis H_1. p-value is likelihood of seeing the sample assuming the null (reject H_0 if p is small)

Usual threshold is $p < 0.05$, small but still means that there is 5% chance of this sample occurring under the null. In particular, when testing 20 hypotheses, we can expect around one false discovery.

Controlled by False Discovery Rate (FDR). FDR is a measure capturing rate of false rejection of H_0. We want a post processing to control in an offline (sequence of p-values is known in advance), or online manner.

Offline FDR control Just select the k smallest p-values.

Online FDR control framed as an investment problem (because lots of tools and framework for budget, reward) "online alpha-investing rule" then "generalized alpha-investing rule"

Level based on recent discovery (LORD) and SAFFRON add statefulness to estimate current proportion of true nulls.

SAFFRON

Keep a candidate set, estimate fraction of true null from the size of this set. When new value arrives, estimate value of investing in the hypothesis, and current wealth. Gives alpha-investing value α_t.

PAPRIKA

input: p-values $\{p_1, \dots\}$, multiplicative sensitivity parameter η, target FDR level, initial wealth, privacy parameters (ε, δ), expected number of rejection c

$\hat{Z} \sim \text{Lap}(2\eta c/\varepsilon)$;
count $\leftarrow 0$;
for *each p-value* p_t **do**
  $\hat{Z}_t \sim \text{Lap}(4\eta c/\varepsilon)$;
  or candidacy $C_t \leftarrow 1(\log(p_t) < \text{Threshold}_t)$;
  Compute alpha investing rule α_t;
  if $C_t = 1$, count $< c$, $\log(p_t) + Z_t < \log(a_t) + \hat{Z}$ **then**
    output $R_t = 1$;
    count + +;
    resample $\hat{Z} \sim \text{Lap}(2\eta c/\varepsilon)$
  else
    output $R_t = 0$ (fail to reject)
  end
end

Remarks. combine SAFFRON investment estimation with SV

instead of comparing p and α, compare noisy versions of them

looks a lot like ANT but

Multiplicative sensitivity: $\eta(f) := \min\left\{\max \frac{f(x)}{f(y)}, \max\{f(x), f(y)\}\right\}$ (either small ratio, or both are very small anyway)

- keep a candidacy set. In ANT noisy in a symmetric way (fails both way as often), here we want false rejection to be rarer.
- here sensitivity is multiplicative and looking at $\log p$, because sensitivity of p-value can be very high.

Theorem 5.4: PAPRIKA is DP and accurate

PAPRIKA is (ε, δ)-DP **and controls FDR** to below an explicit threshold

Note that for this algorithm $\delta > 0$ (but tiny).

No theoretical guarantee with respect to the power of the method. In experiments, good power requires rather large ε values.

5.5.4 DP and Changepoint Detection

Goal: detect distribution of timeseries changes at $t^\star$

Assume we have offline DP method (reasonable)

How to do it online? DP detect that test statistic is above threshold in the sliding window, and run offline algo on this window

5.6 Online Optimization

summary:

- Private algo for maintaining partial sum
- Private Follow The Approximate Leader

Incoming stream, and we want to adapt the decision based on what was seen before

5.6.1 First idea

Given stream of bits $b_1, \ldots, b_\tau$. At each time t output $\sum_{\tau=1}^{t} b_\tau$

Bad idea 1 At each time t, output $\sum_{\tau=1}^{t} b_\tau + \mathrm{Lap}(1/\varepsilon)$

Then by composition, $\varepsilon = \sqrt{T}\varepsilon' \log 1/\delta$

Accuracy loss $O(1/\varepsilon')$

Good accuracy *or* good privacy. Fix ε and choose $\varepsilon' = \varepsilon/\sqrt{T}$, then we have accuracy loss $O(\sqrt{T}/\varepsilon)$. However this can easily become large

Bad idea 2 Add noise to each $b_i : \hat{b}_i = b_i + \text{Lap}(1/\varepsilon)$

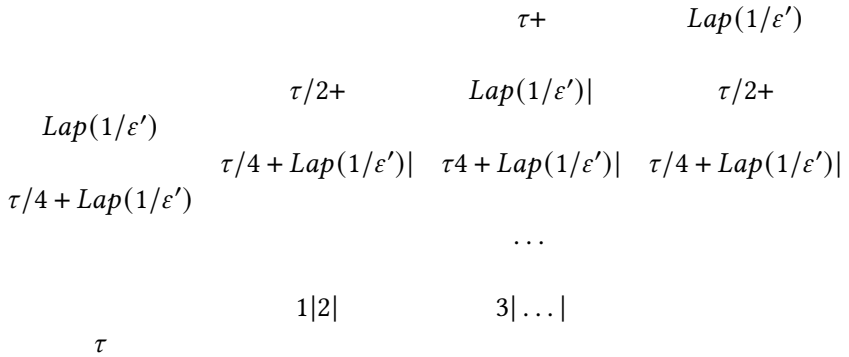
output $\sum_{i=1}^{\tau} \hat{b}_i = \sum_{i=1}^{\tau} b_i + \tau\text{Lap}(1/\varepsilon)$

1. Big noise infrequently 2. Small noise too often

We need a data structure to fix this

Better idea (but really a lie)

Break down into blocks (like a balanced binary tree). Then per sample, add Laplace noise for each block.



Goals

- Any sum uses only $O(\log(T))$ noise terms
- Any noise term is used only $O(\log(T))$ times

NB: instead of bits b_i, we can think of vectors z_i with $\|z_i\| \leq \Delta$. Then use noise $\text{Lap}(\Delta/\varepsilon')$. And replace with $\sum^t z_i$.

Tree Based Aggregation Protocol (TBAP) [Chanet et al. 2010, Dwork et al. 2010]

input : $z_1, \ldots, z_\tau \in \mathbb{R}^d$, Δ l_2-bound on all z_t, ε
output: Sequence of noisy partial sums $v_1, \ldots, v_\tau \in \mathbb{R}^d$
Initialize binary tree A of size $2^{\lfloor \log_2 T \rfloor} - 1$ with leaves $z_1, \ldots, z_\tau$;
for $t = 1, \ldots, T$ **do**
  Accept z_t from data stream;
  Let $P = \{z_t \to \cdots \to \text{root}\}$ be a path from z_t to the root. ;
  Tree update.;
  Let Λ be the first node in P that is a left-child in A. *We only add noise up to a point (the first left-child) then stop* ;
  Let $P_\Lambda = \{z_t \to \cdots \to \Lambda\}$;
  for *all nodes* α *in path* p **do**
    $\alpha \leftarrow \alpha + z_t$;
    if $\alpha \in P_\Lambda$ **then** $\alpha \leftarrow \alpha + \gamma$ where $\gamma \in \mathbb{R}$ sampled $\propto \exp \frac{-\|\gamma\|_2 \varepsilon}{\Delta \lfloor \log_2 T \rfloor}$;
  end
end
Output partial sums;
Initialize $v_t \in \mathbb{R}^d$ to be 0 ;
Let b be a ($\lfloor \log_2 T \rfloor + 1$)-bit binary representation of t. ;
for $i = 1, \ldots, \lfloor \log_2 T \rfloor + 1$ **do**
  if $b_i = 1$ **then**
    always add something*;*
    if *i-th node in P (denoted* $P(i)$*) is a left child* **then**
      $v_t \leftarrow v_t + P(i)$
    else
      $v_t \leftarrow v_t + \texttt{left-sibling}(P(i))$
    end
  end
end
return v_t

NB: Laplace Mechanism $\mathbb{P}x \propto \exp \frac{-\|x\|\varepsilon}{\Delta}$, here $\mathbb{P}\gamma \propto \exp \frac{-\|\gamma\|\varepsilon}{\Delta floor \log_2 T}$

Private follow the Approximate Leader

input : sequence of cost functions $f_1, \ldots, f_\tau$, H, L, C, ε
Initialize $\hat{w}_i \in C$ arbitrarily, output $\hat{w}_i$ **for** $t = 1, \ldots, T$ **do**
  Pass $\nabla f_t(\hat{w}_i), L, \varepsilon$ into TBAP and recieve current partial sum $\hat{v}_t$
  $\hat{w}_{t+1} = arg\min_{w \in C} < \hat{v}_t, w > + \frac{H}{2} \sum_{\tau=1}^{t} \|w - \hat{w}_\tau\|_2^2$
  Output $\hat{w}_{t+1}$
end

5.7 Private Deep Learning

The only thing to to is to adapt a DP gradient descent.

DP-SGD [Abadi et al 2016] (Deep learning with differential privacy, In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security).

input : Dataset $X = (x_1, \dots, x_n)$, loss function $\mathcal{L}_\theta$, learning rate η_t, batch size L, noise multiplier σ, gradient norm bound c

Initialize θ_0 randomly;

(Sample a batch) e.g. Poisson random subsample L_t with pre-example prob L/n ;

for *each* $x_i \in L_t$ **do**

compute $g_t(x_i) = \nabla_{\theta t} \mathcal{L}_{\theta t}(x_i)$;

$\bar{g}_t(x_i) = \frac{g_t(x_i)}{\max\{1, \|g_t(x_i)\|_2 / c\}}$;

$g_t = \frac{1}{|L_t|} \sum_{x_i \in L_t} g_t(x_i) + N(0, \sigma^2 C^2 I)$;

$\theta_{t+1} = \theta_t + \eta_t g_t$

end

output: θ_T and compute overall (ε, δ)-DP bound via privacy accounting.

The question is the correct size of the noise to add. Difficult a priori because we don't know the variations of the gradient (possibly unbounded). We can clip the gradient to always lie in some range.

Note that there are no ε or δ in the algorithm. Could use composition, but here training for thousands of rounds so even a square-root bound is too large. We have special composition rules for learning with gaussian noise.

Definition 8 (Renyi DP [Mir 17]). *A mechanism M is (α, ε)-RDP if for all neighbours x, x'*

$$RDP(\alpha) := D_\alpha(M(x) \| M(x')) \leq \varepsilon$$

where $D_\alpha(P\|Q) = \frac{1}{\alpha - 1} \log \left(\mathbb{E}_{x \sim x} \left[\left(\frac{P(x)}{Q(x)} \right)^\alpha \right] \right)$

Privacy accounting of DP-SGD via RDP

1. compute subsample RDP parameters for one step $RDP_{t=1}(\alpha)$
2. RDP composition:

   Proposition 3. *If M_1, M_2 respectively are $(\alpha, \varepsilon_1), (\alpha, \varepsilon_2)$-RDP for $\alpha \geq 1$, then the composition is $(\alpha, \varepsilon_1 + \varepsilon_2)$-RDP.*
3. convert to (ε, δ)-DP

   Proposition 4. *If M is (α, ε)-RDP $\forall \alpha \geq 1$, then M is $\left(\varepsilon(\alpha) + \frac{\log(1/\delta)}{\alpha - 1}, \delta\right)$-DP $\forall \delta > 0$.*

   (ε depends on *alpha*).

5.8 Misc.

Who adds noise? Two models:

- *Trusted Curator model:* requires trusted party collects and sees data, add less noise (more accurate)
- *Local Model:* add noise locally (doesn't require trust), more error because can't coordinate noise

Example of local model To give people deniability on a yes/no question, the agent flips two coins and answers truthfully, but if they get two tails they flip their answer. Then still possible to have population level statistics.

DP Synthetic data generation Given a database D, find another database D' that has the same statistical properties as D.

- *Challenges:* datasets are often high dimensional and are required to be correct on many queries:
  - how to measure "accuracy" of a synthetic dataset? (there exist good measures of distance but superpolynomial in the size of the dataset)
  - Computational efficiency of data generation.
- *(Partial) solutions:*

Explaining DP How to communicate to public/policymakers/engineers?

(Partial) solutions: measuring users' privacy expectations from different DP description; finding methods for explaining privacy parameters $\longrightarrow$ **Ongoing work**.

Bibliography

Shipra Agrawal and Nikhil R Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the fifteenth ACM conference on Economics and computation*, pages 989–1006, 2014.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. *Advances in Neural Information Processing Systems*, 30, 2017.

Shipra Agrawal, Vashist Avadhanula, Vineet Goyal, and Assaf Zeevi. Mnl-bandit: A dynamic learning approach to assortment selection. *Operations Research*, 67(5):1453–1485, 2019.

Peter Auer, Thomas Jaksch, and Ronald Ortner. Near-optimal regret bounds for reinforcement learning. *Advances in neural information processing systems*, 21, 2008.

Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o(1/n). *CoRR*, abs/1306.2119, 2013. URL http://arxiv.org/abs/1306.2119.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216. IEEE, 2013.

Peter L Bartlett and Ambuj Tewari. Regal: A regularization based algorithm for reinforcement learning in weakly communicating mdps. *arXiv preprint arXiv:1205.2661*, 2012.

David K. Hammond, Pierre Vandergheynst, and Rémi Gribonval. Wavelets on graphs via spectral graph theory. *Applied and Computational Harmonic Analysis*, 30(2):129–150, 2011. ISSN 1063-5203. doi: https://doi.org/10.1016/j.acha.2010.04.005. URL https://www.sciencedirect.com/science/article/pii/S1063520310000552.

Tor Lattimore and Csaba Szepesvári. *Bandit algorithms*. Cambridge University Press, 2020.

Edoardo Di Napoli, Eric Polizzi, and Yousef Saad. Efficient estimation of eigenvalue counts in an interval. *Numerical Linear Algebra with Applications*, 23(4):674–692, March 2016. doi: 10.1002/nla.2048. URL https://doi.org/10.1002/nla.2048.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss minimization. *arXiv preprint arXiv:1209.1873*, 2012.

David I. Shuman, Mohammad Javad Faraji, and Pierre Vandergheynst. A multiscale pyramid transform for graph signals. *IEEE Trans. Signal Process.*, 64(8):2119–2134, 2016. doi: 10.1109/TSP.2015.2512529. URL https://doi.org/10.1109/TSP.2015.2512529.