What Can We Learn Privately?

Shiva Prasad Kasiviswanathan, Homin K. Lee, Kobbi Nissim,
Sofya Raskhodnikova, Adam Smith
FOCS 2008
Presented by *Yanjie Ze*

August 24, 2022

Section: Introduction

Core Problem

We ask:
what concept classes can be learned privately, namely, by an algorithm whose output does not depend too heavily on any one input or specific training example?

Differential Privacy

Utilize the notion of differential privacy. Advantage:

- provides rigorous guarantees even in the presence of a malicious adversary with access to arbitrary auxiliary information.

Learning Privately

What computational tasks can be performed while maintaining privacy?

Learning Privately

We examine **probabilistically approximately correct (PAC)** learning model from computational learning theory.
Assume:

- entries z_i of the database are random examples generated i.i.d. from the underlying distribution $\mathcal{D}$ and labeled by a target concept c.

Contributions

1. **A Private Version of Occam's Razor.**
2. **An Efficient Private Learner for Parity.**
3. Equivalence of Local ("Randomized Response") and SQ Learning.
4. Separation of Interactive and Noninteractive Local Learning.

Implications

- "Anything" learnable is privately learnable using few samples.
- Learning with noise is different from private learning.
  Our efficient private learner for parity dispels the similarity between learning with noise and private learning.

Section: Preliminaries

Subsection: Differential Privacy

Differential Privacy

Definition 1 (ϵ-differential privacy)

A randomized algorithm $\mathcal{A}$ is ϵ-differentially private if for all neighboring databases z, z', and for all sets $\mathcal{S}$ of outputs,

$$\Pr[\mathcal{A}(\mathrm{z}) \in \mathcal{S}] \leq \exp(\epsilon) \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) \in \mathcal{S}\right].$$

The probability is taken over the random coins of $\mathcal{A}$.

Differential Privacy

Let $\mathrm{Lap}(\lambda)$ denote the Laplace probability distribution with mean 0 , standard deviation $\sqrt{2}\lambda$, and p.d.f. $f(x) = \frac{1}{2\lambda} e^{-|x|/\lambda}$.

Theorem 2 (Laplacian Mechanism)

For a function $f : D^n \to \mathbb{R}$, define its global sensitivity $GS_f = \max_{\mathrm{z},\mathrm{z}'} |f(\mathrm{z}) - f(\mathrm{z}')|$ where the maximum is over all neighboring databases z, z'. Then, an algorithm that on input z returns $f(\mathrm{z}) + \eta$ where $\eta \sim \mathrm{Lap}(GS_f/\epsilon)$ is ϵ-differentially private.

Subsection: Learning Theory

Learning Theory

- A concept is a function that labels *examples* taken from the domain X by the elements of the range Y.
- A concept class C is a set of concepts.

We focus on binary classification problems, in which the label space Y_d is $\{0, 1\}$ or $\{+1, -1\}$; the parameter d thus measures the size of the examples in X_d.
The concept classes are ensembles $\mathcal{C} = \{\mathcal{C}_d\}_{d \in \mathbf{N}}$ where $\mathcal{C}_d$ is the class of concepts from X_d to Y_d.

Learning Theory

- Let $\mathcal{D}$ be a distribution over labeled examples in $X_d \times Y_d$.
- *A learning algorithm* is given access to $\mathcal{D}$ (the method for accessing $\mathcal{D}$ depends on the type of learning algorithm).
- It outputs a hypothesis $h : X_d \rightarrow Y_d$ from a hypothesis class $\mathcal{H} = \{\mathcal{H}_d\}_{d \in \mathbb{N}}$.

Learning Theory

The goal: minimize the misclassification error of h on $\mathcal{D}$, defined as

$$\text{err}(h) = \Pr_{(x,y)\sim\mathcal{D}}[h(x) \neq y].$$

The success of a learning algorithm is quantified by parameters α and β.

- α is the desired error.
- β bounds the probability of failure to output a hypothesis with this error.

Error measures other than misclassification are considered in supervised learning $\left(e.g., L_2^2\right)$. We study only misclassification error here, since for binary labels it is equivalent to the other common error measures.

PAC Learning

One assumption: the examples are labeled consistently with some target concept c from a class $\mathcal{C}$: namely, $c \in \mathcal{C}_d$ and $y = c(x)$ for all (x, y) in the support of $\mathcal{D}$. In the PAC setting, $\text{err}(h) = \Pr_{x \sim \mathcal{X}}[h(x) \neq c(x)]$.

PAC Learning

Definition 3 (PAC Learning)

A concept class $\mathcal{C}$ over X is PAC learnable using hypothesis class $\mathcal{H}$ if there exist an algorithm $\mathcal{A}$ and a polynomial poly $(\cdot, \cdot, \cdot)$ such that for all $d \in \mathbb{N}$, all concepts $c \in \mathcal{C}_d$, all distributions $\mathcal{X}$ on X_d, and all $\alpha, \beta \in (0, 1/2)$, given inputs α, β and $\mathrm{z} = (z_1, \cdots, z_n)$, where $n = \text{poly}\,(d, 1/\alpha, \log(1/\beta))$, $z_i = (x_i, c\,(x_i))$ and x_i are drawn i.i.d. from $\mathcal{X}$ for $i \in [n]$, algorithm $\mathcal{A}$ outputs a hypothesis $h \in \mathcal{H}$ satisfying

$$\Pr[\text{err}(h) \leq \alpha] \geq 1 - \beta. \tag{1}$$

The probability is taken over the random choice of the examples z and the coin tosses of $\mathcal{A}$.

PAC Learning

Class $\mathcal{C}$ is (inefficiently) PAC learnable if there exists some hypothesis class $\mathcal{H}$ and a PAC learner $\mathcal{A}$ such that $\mathcal{A}$ PAC learns $\mathcal{C}$ using $\mathcal{H}$.
Class $\mathcal{C}$ is efficiently PAC learnable if $\mathcal{A}$ runs it time polynomial in $d, 1/\alpha$, and $\log(1/\beta)$.

Agnostic Learning

Definition 4 (Agnostic Learning)

(Efficiently) agnostically learnable is defined identically to (efficiently) *PAC* learnable with two exceptions: (i) the data are drawn from an arbitrary distribution $\mathcal{D}$ on $X_d \times Y_d$; (ii) instead of Equation (1) the output of $\mathcal{A}$ has to satisfy:

$$\Pr[\mathrm{err}(h) \leq OPT + \alpha] \geq 1 - \beta,$$

where $OPT = \min_{f \in \mathcal{C}_d} \{\mathrm{err}(f)\}$. As before, the probability is taken over the random choice of z, and the coin tosses of $\mathcal{A}$.

Definitions 3 and 4 capture distribution-free learning, in that they do not assume a particular form for the distributions $\mathcal{X}$ or $\mathcal{D}$.

Section: Private PAC and Agnostic Learning

Definition

We define private PAC learners as algorithms that satisfy definitions of both differential privacy and PAC learning.
Difference:

- Learning must succeed on average over a set of examples drawn i.i.d. from $\mathcal{D}$ (often under the additional promise that $\mathcal{D}$ is consistent with a concept from a target class).
- Differential privacy, in contrast, must hold in the worst case, with no assumptions on consistency.

Definition

Definition 5 (Private PAC Learning)

Let d, α, β be as in Definition 4 and $\epsilon > 0$. Concept class $\mathcal{C}$ is (inefficiently) privately PAC learnable using hypothesis class $\mathcal{H}$ if there exists an algorithm $\mathcal{A}$ that takes inputs $\epsilon, \alpha, \beta, \mathrm{z}$, where n, the number of labeled examples in z, is polynomial in $1/\epsilon, d, 1/\alpha, \log(1/\beta)$, and satisfies

Privacy For all $\epsilon > 0$, algorithm $\mathcal{A}(\epsilon, \cdot, \cdot, \cdot)$ is ϵ-differentially private (Definition 1);

Utility Algorithm $\mathcal{A}$ PAC learns $\mathcal{C}$ using $\mathcal{H}$ (Definition 3). $\mathcal{C}$ is *efficiently privately PAC learnable* if $\mathcal{A}$ runs in time polynomial in $d, 1/\epsilon, 1/\alpha$, and $\log(1/\beta)$.

Definition

Definition 6 (Private Agnostic Learning)

(Efficient) private agnostic learning is defined analogously to (efficient) private PAC learning with Definition 4 replacing Definition 3 in the utility condition.

Difficulty

- Evaluating the quality of a particular hypothesis is easy: one can privately compute the fraction of the data it classifies correctly.
- The difficulty of constructing private learners lies in finding a good hypothesis in what is typically an exponentially large space.

Subsection: A Generic Private Agnostic Learner

A Generic Private Agnostic Learner

In this section, we present a private analogue of a basic consistent learning result, often called the cardinality version of Occam's razor.

This classical result shows that a PAC learner can weed out all bad hypotheses given a number of labeled examples that is logarithmic in the size of the hypothesis class.

Our generic private learner is based on the exponential mechanism of McSherry and Talwar.

A Generic Private Agnostic Learner

Let $q : D^n \times \mathcal{H}_d \to \mathbb{R}$ take a database z and a candidate hypothesis h, and assign it a score $q(\mathrm{z}, h) = -\mid \{i : x_i$ is misclassified by h, i.e., $y_i \neq h(x_i)\} \mid$.

- the score is minus the number of points in z misclassified by h.

The classic Occam's razor argument assumes a learner that selects a hypothesis with maximum score (minimum empirical error). *Instead*, our private learner $\mathcal{A}_q^\epsilon$ is defined to sample a random hypothesis with probability dependent on its score:

$\mathcal{A}_q^\epsilon(\mathrm{z})$**: Output hypothesis** $h \in \mathcal{H}_d$ **with probability proportional to** $\exp\left(\frac{\epsilon q(\mathrm{z},h)}{2}\right)$**.**

- Since the score ranges from $-n$ to 0 , hypotheses with low empirical error are exponentially more likely to be selected than ones with high error.

A Generic Private Agnostic Learner

Algorithm $\mathcal{A}_q^\epsilon$ fits the framework of McSherry and Talwar, and so is ϵ-differentially private.

Lemma 7

The algorithm $\mathcal{A}_q^\epsilon$ is ϵ-differentially private.

This follows from the fact that changing one entry z_i in the database z can change the score by at most 1.

A Generic Private Agnostic Learner

Theorem 8 (Generic Private Learner)

For all $d \in \mathbb{N}$, any concept class $\mathcal{C}_d$ whose cardinality is at most $\exp(\text{poly}(d))$ *is privately agnostically learnable using $\mathcal{H}_d = \mathcal{C}_d$. More precisely, the learner uses $n = O\left(\left(\ln |\mathcal{H}_d| + \ln \frac{1}{\beta}\right) \cdot \max\left\{\frac{1}{\epsilon\alpha}, \frac{1}{\alpha^2}\right\}\right)$ labeled examples from $\mathcal{D}$, where ϵ, α, and β are parameters of the private learner. (The learner might not be efficient.)*

A Generic Private Agnostic Learner (Proof)

Proof.

Let $\mathcal{A}_q^{\epsilon}$ be as defined above. The privacy condition in Definition 1 is satisfied by Lemma 7. We now show that the utility condition is also satisfied.

Consider the event $E = \left\{\mathcal{A}_q^{\epsilon}(\mathrm{z}) = h \text{ with } \mathrm{err}(h) > \alpha + OPT\right\}$. We want to prove that $\Pr[E] \leq \beta$.

Define the training error of h as

$$\mathrm{err}_T(h) = \left|\{i \in [n] \mid h(x_i) \neq y_i\}\right| / n = -q(\mathrm{z}, h)/n$$

By Chernoff-Hoeffding bounds (Lemma 9),

$$\Pr\left[\left|\mathrm{err}(h) - \mathrm{err}_T(h)\right| \geq \rho\right] \leq 2\exp\left(-2n\rho^2\right)$$

for all hypotheses $h \in \mathcal{H}_d$. Hence,

$$\Pr\left[\left|\mathrm{err}(h) - \mathrm{err}_T(h)\right| \geq \rho \text{ for some } h \in \mathcal{H}_d\right] \leq 2\left|\mathcal{H}_d\right|\exp\left(-2n\rho^2\right)$$

Chernoff-Hoeffding Bound

Lemma 9 (Real-valued Additive Chernoff-Hoeffding Bound)

Let $X_1, \ldots, X_n$ be i.i.d. random variables with $\mathbb{E}[X_i] = \mu$ and $a \leq X_i \leq b$ for all i. Then for every $\delta > 0$,

$$\Pr\left[\left|\frac{\sum_i X_i}{n} - \mu\right| \geq \delta\right] \leq 2\exp\left(\frac{-2\delta^2 n}{(b-a)^2}\right)$$

A Generic Private Agnostic Learner (Proof)

We now analyze $\mathcal{A}_q^\epsilon(\mathrm{z})$ conditioned on the event that for all $h \in \mathcal{H}_d, |\mathrm{err}(h) - \mathrm{err}_T(h)| < \rho$. For every $h \in \mathcal{H}_d$, the probability that $\mathcal{A}_q^\epsilon(\mathrm{z}) = h$ is

$$\frac{\exp\left(-\frac{\epsilon}{2} \cdot n \cdot \mathrm{err}_T(h)\right)}{\sum_{h' \in \mathcal{H}_d} \exp\left(-\frac{\epsilon}{2} \cdot n \cdot \mathrm{err}_T\left(h'\right)\right)}$$

$$\leq \frac{\exp\left(-\frac{\epsilon}{2} \cdot n \cdot \mathrm{err}_T(h)\right)}{\max_{h' \in \mathcal{H}_d} \exp\left(-\frac{\epsilon}{2} \cdot n \cdot \mathrm{err}_T\left(h'\right)\right)}$$

$$= \exp\left(-\frac{\epsilon}{2} \cdot n \cdot \left(\mathrm{err}_T(h) - \min_{h' \in \mathcal{H}_d} err_T\left(h'\right)\right)\right)$$

$$\leq \exp\left(-\frac{\epsilon}{2} \cdot n \cdot \left(\mathrm{err}_T(h) - (OPT + \rho)\right)\right)$$

A Generic Private Agnostic Learner (Proof)

Hence, the probability that $\mathcal{A}_q^{\epsilon}(\mathrm{z})$ outputs a hypothesis $h \in \mathcal{H}_d$ such that $\mathrm{err}_T(h) \geq OPT + 2\rho$ is at most $|\mathcal{H}_d| \exp(-\epsilon n\rho/2)$.
Now set $\rho = \alpha/3$.
If $\mathrm{err}(h) \geq OPT + \alpha$ then $|\mathrm{err}(h) - \mathrm{err}_T(h)| \geq \alpha/3$ or $\mathrm{err}_T(h) \geq OPT + 2\alpha/3$.
Thus $\Pr[E] \leq |\mathcal{H}_d| \left(2\exp\left(-2n\alpha^2/9\right) + \exp(-\epsilon n\alpha/6)\right) \leq \beta$ where the last inequality holds for

$$n \geq 6\left(\left(\ln|\mathcal{H}_d| + \ln\frac{1}{\beta}\right) \cdot \max\left\{\frac{1}{\epsilon\alpha}, \frac{1}{\alpha^2}\right\}\right)$$

(Recall: $E = \left\{\mathcal{A}_q^{\epsilon}(\mathrm{z}) = h \text{ with } \mathrm{err}(h) > \alpha + OPT\right\}$)

Subsection: Private Learning with VC Dimension Sample Bounds

Private Learning with VC Dimension Sample Bounds

In the non-private case one can also bound the sample size of a PAC learner in terms of the Vapnik-Chervonenkis (VC) dimension of the concept class.

Definition 10 (VC dimension)

A set $S \subseteq X_d$ is shattered by a concept class $\mathcal{C}_d$ if $\mathcal{C}_d$ restricted to S contains all $2^{|S|}$ possible functions from S to $\{0, 1\}$. The VC dimension of $\mathcal{C}_d$, denoted $VCDIM(\mathcal{C}_d)$, is the cardinality of a largest set S shattered by $\mathcal{C}_d$.

Private Learning with VC dimension Sample Bounds

We can extend Theorem 8 to classes with finite VC dimension, but the resulting sample complexity also depends logarithmically on the size of the domain from which examples are drawn.

Private Learning with VC dimension Sample Bounds

Corollary 11

Every concept class $\mathcal{C}_d$ is privately agnostically learnable using hypothesis class $\mathcal{H}_d = \mathcal{C}_d$ with $n = O\left(\left(\mathrm{VCDIM}\left(\mathcal{C}_d\right) \cdot \ln |X_d| + \ln \frac{1}{\beta}\right) \cdot \max\left\{\frac{1}{\epsilon\alpha}, \frac{1}{\alpha^2}\right\}\right)$ labeled examples from $\mathcal{D}$. Here, ϵ, α, and β are parameters of the private agnostic learner, and $\mathrm{VCDIM}\left(\mathcal{C}_d\right)$ is the VC dimension of $\mathcal{C}_d$. (The learner is not necessarily efficient.)

(Comparison: $n = O\left(\left(\ln |\mathcal{H}_d| + \ln \frac{1}{\beta}\right) \cdot \max\left\{\frac{1}{\epsilon\alpha}, \frac{1}{\alpha^2}\right\}\right)$)

Proof.

Sauer's lemma (see, e.g., [42]) implies that there are $O\left(|X_d|^{VCDIM(\mathcal{C}_d)}\right)$ different labelings of X_d by functions in $\mathcal{C}_d$. We can thus run the generic learner of the previous section with a hypothesis class of size $|\mathcal{H}_d| = O\left(|X_d|^{VCDIM(\mathcal{C}_d)}\right)$. The statement follows directly. $\square$

Private Learning with VC dimension Sample Bounds

Our original proof of the corollary used a result of Blum, Ligget and Roth [14] (which was inspired, in turn, by our generic learning algorithm) on generating synthetic data. The simpler proof above was pointed out to us by an anonymous reviewer.

Section: An Efficient Private Learner for PARITY

Subsection: PARITY Learner

PARITY Learner

Let PARITY be the class of parity functions $c_r : \{0,1\}^d \to \{0,1\}$ indexed by $r \in \{0,1\}^d$, where $c_r(x) = r \odot x$ denotes the inner product modulo 2 .

In this section, we present an efficient private PAC learning algorithm for PARITY.

PARITY

The standard (non-private) PAC learner for PARITY:

- look for the hidden vector r by solving a system of linear equations imposed by examples $(x_i, c_r(x_i))$ that the algorithm sees
- It outputs an arbitrary vector consistent with the examples, i.e., in the solution space of the system of linear equations

We want to design a private algorithm that emulates this behavior. A major difficulty:

- The private learner's behavior must be specified on all databases z, even those which are not consistent with any single parity function.
- The standard PAC learner would simply fail in such a situation (we denote failure by the output $\perp$). In contrast, the probability that a private algorithm fails must be similar for all neighbors z and z'.

Intuition

Intuitively, the reason PARITY can be learned privately is:

- When a new example (corresponding to a new linear constraint) is added, the space of consistent hypotheses shrinks by *at most* a factor of 2.
- This holds unless the new constraint is *inconsistent* with previous constraints. In the latter case, the size of the space of consistent hypotheses goes to 0.
- Thus, the solution space changes *drastically* on neighboring inputs only when the algorithm fails (outputs $\perp$).
- The fact that algorithm outputs $\perp$ on a database z and a valid (non $\perp$) hypothesis on a neighboring database z' might lead to privacy violations. To avoid this, our algorithm always outputs $\perp$ with probability at least $1/2$ on any input (Step 1).

PARITY Learner

A PRIVATE LEARNER FOR PARITY, $\mathcal{A}(z, \epsilon)$

1. With probability $1/2$, output $\perp$ and terminate.
2. Construct a set S by picking each element of $[n]$ independently with probability $p = \epsilon/4$.
3. Use Gaussian elimination to solve the system of equations imposed by examples, indexed by S: namely, $\{x_i \odot r = c_r(x_i) \ : \ i \in S\}$. Let V_S denote the resulting affine subspace.
4. Pick $r^* \in V_S$ uniformly at random and output c_{r^*}; if $V_S = \emptyset$, output $\perp$.

PARITY Learner

The proof of $\mathcal{A}$'s utility follows by considering all the possible situations in which the algorithm fails to satisfy the error bound, and by bounding the probabilities with which these situations occur.

Lemma 12 (Utility of $\mathcal{A}$)

Let $\mathcal{X}$ be a distribution over $X = \{0,1\}^d$. Let $\mathrm{z} = (z_1, \ldots, z_n)$, where for all $i \in [n]$, the entry $z_i = (x_i, c(x_i))$ with x_i drawn i.i.d. from $\mathcal{X}$ and $c \in$ PARITY. If $n \geq \frac{8}{\epsilon\alpha}(d \ln 2 + \ln 4)$ then

$$\Pr[\mathcal{A}(\mathrm{z}, \epsilon) = h \text{ with error}(h) \leq \alpha] \geq \frac{1}{4}.$$

PARITY Learner

Proof.
By standard arguments in learning theory [1], $|S| \geq \frac{1}{\alpha}\left(d \ln 2 + \ln \frac{1}{\beta}\right)$ labeled examples are sufficient for learning PARITY with error α and failure probability β.
Since $\mathcal{A}$ adds each element of $[n]$ to S independently with probability $p = \epsilon/4$, the expected size of S is $pn = \epsilon n/4$.
By the Chernoff bound (Theorem 13), $|S| \geq \epsilon n/8$ with probability at least $1 - e^{-\epsilon n/16}$.
We set $\beta = \frac{1}{4}$ and pick n such that $\epsilon n/8 \geq \frac{1}{\alpha}(d \ln 2 + \ln 4)$

[1] Kearns, Michael J., and Umesh Vazirani. An introduction to computational learning theory. MIT press, 1994.

Multiplicative Chernoff Bounds

Theorem 13 (Multiplicative Chernoff Bounds)

Let $X_1, \ldots, X_n$ *be i.i.d. Bernoulli random variables with* $\Pr[X_i = 1] = \mu$. *Then for every* $\phi \in (0, 1]$,

$$\Pr\left[\frac{\sum_i X_i}{n} \geq (1+\phi)\mu\right] \leq \exp\left(-\frac{\phi^2 \mu n}{3}\right)$$

and

$$\Pr\left[\frac{\sum_i X_i}{n} \leq (1-\phi)\mu\right] \leq \exp\left(-\frac{\phi^2 \mu n}{2}\right)$$

PARITY Learner

Proof.

We now bound the overall success probability.

$\mathcal{A}(\mathrm{z}, \epsilon) = h$ with $\mathrm{err}(h) \leq \alpha$ unless one of the following bad events happens:

1. $\mathcal{A}$ terminates in Step 1,
2. $\mathcal{A}$ proceeds to Step 2, but does not get enough examples: $|S| < \frac{1}{\alpha}(d \ln 2 + \ln 4)\big)$,
3. $\mathcal{A}$ gets enough examples, but outputs a hypothesis with error greater than α.

- The first bad event occurs with probability $1/2$.
- If the lower bound on the database size n ($n \geq \frac{8}{\epsilon\alpha}(d \ln 2 + \ln 4)$) is satisfied then the second bad event occurs with probability at most $e^{-\epsilon n/16}/2 \leq 1/8$. The last inequality follows from the bound on n and the fact that $\alpha \leq 1/2$.
- Finally, by our choice of parameters, the last bad event occurs with probability at most $\beta/2 = 1/8$.

The claimed bound on the success probability follows.

PARITY Learner

Lemma 14 (Privacy of $\mathcal{A}$)

Algorithm $\mathcal{A}$ is ϵ-differentially private.

As mentioned above, the key observation in the following proof is that including of any single point in the sample set S increases the probability of a hypothesis being output by at most 2.

PARITY Learner

Proof.

To show that $\mathcal{A}$ is ϵ-differentially private, it suffices to prove that any output of $\mathcal{A}$, either a valid hypothesis or $\perp$, appears with roughly the same probability on neighboring databases z and z'. In the remainder of the proof we fix ϵ, and write $\mathcal{A}(\mathrm{z})$ as shorthand for $\mathcal{A}(\mathrm{z}, \epsilon)$.

We have to show that

1. $\Pr[\mathcal{A}(\mathrm{z}) = h] \leq e^{\epsilon} \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h\right]$ for all neighbors $\mathrm{z}, \mathrm{z}' \in D^n$ and all hypotheses $h \in$ PARITY;
2. $\Pr[\mathcal{A}(\mathrm{z}) = \perp] \leq e^{\epsilon} \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = \perp\right]$ for all neighbors $\mathrm{z}, \mathrm{z}' \in D^n$.

Proof.

We prove the correctness of the first equation first.

Let z and z' be neighboring databases, and let i denote the entry on which they differ. Recall that $\mathcal{A}$ adds i to S with probability p. Since z and z' differ only in the i^{th} entry,

$\Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S] = \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h \mid i \notin S\right]$.

Proof.

Note that if $\Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h \mid i \notin S\right] = 0$, then also $\Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S] = 0$, and hence $\Pr[\mathcal{A}(\mathrm{z}) = h] = 0$ because adding a constraint does not add new vectors to the space of solutions. Otherwise, $\Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h \mid i \notin S\right] > 0$. In this case, we rewrite the probability on z as follows:

$$\Pr[\mathcal{A}(\mathrm{z}) = h] = p \cdot \Pr[\mathcal{A}(\mathrm{z}) = h \mid i \in S] + (1-p) \cdot \Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S],$$

and apply the same transformation to the probability on z'. Then

$$\begin{aligned}
\frac{\Pr[\mathcal{A}(\mathrm{z}) = h]}{\Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h\right]} &= \frac{p \cdot \Pr[\mathcal{A}(\mathrm{z}) = h \mid i \in S] + (1-p) \cdot \Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S]}{p \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h \mid i \in S\right] + (1-p) \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h \mid i \notin S\right]} \\
&\leq \frac{p \cdot \Pr[\mathcal{A}(\mathrm{z}) = h \mid i \in S] + (1-p) \cdot \Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S]}{p \cdot 0 + (1-p) \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h \mid i \notin S\right]} \\
&= \frac{p}{1-p} \cdot \frac{\Pr[\mathcal{A}(\mathrm{z}) = h \mid i \in S]}{\Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S]} + 1
\end{aligned}$$

PARITY Learner

Proof.

We need the following claim:

Claim 1

$\frac{\Pr[\mathcal{A}(\mathrm{z})=h|i\in S]}{\Pr[\mathcal{A}(\mathrm{z})=h|i\notin S]} \leq 2$, *for all* $\mathrm{z} \in D^n$ *and all hypotheses* $h \in \mathrm{PARITY}$*.*

We plug it into the previous equation to get

$$\frac{\Pr[\mathcal{A}(\mathrm{z}) = h]}{\Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = h\right]} \leq \frac{2p}{1-p} + 1 \leq \epsilon + 1 \leq e^{\epsilon}.$$

The first inequality holds since $p = \epsilon/4$ and $\epsilon \leq 1/2$. This establishes the first condition.

PARITY Learner

Proof.

The proof of the second condition is similar:

$$\begin{aligned}
\frac{\Pr[\mathcal{A}(\mathrm{z}) = \perp]}{\Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = \perp\right]} &= \frac{p \cdot \Pr[\mathcal{A}(\mathrm{z}) = \perp \mid i \in S] + (1-p) \cdot \Pr[\mathcal{A}(\mathrm{z}) = \perp \mid i \notin S]}{p \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = \perp \mid i \in S\right] + (1-p) \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = \perp \mid i \notin S\right]} \\
&\leq \frac{p \cdot 1 + (1-p) \cdot \Pr[\mathcal{A}(\mathrm{z}) = \perp \mid i \notin S]}{p \cdot 0 + (1-p) \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = \perp \mid i \notin S\right]} \\
&= \frac{p}{(1-p) \cdot \Pr\left[\mathcal{A}\left(\mathrm{z}'\right) = \perp \mid i \notin S\right]} + 1 \\
&\leq \frac{2p}{1-p} + 1 \leq \epsilon + 1 \leq e^{\epsilon}
\end{aligned}$$

In the last line, the first inequality follows from the fact that on any input, $\mathcal{A}$ outputs $\perp$ with probability at least $1/2$. This completes the proof of the lemma.

Thank You.

Proof of Claim 1

We now prove Claim 1.

Proof of Claim 1.

The left hand side

$$\frac{\Pr[\mathcal{A}(\mathrm{z}) = h \mid i \in S]}{\Pr[\mathcal{A}(\mathrm{z}) = h \mid i \notin S]} =$$

$$\frac{\sum_{T \subseteq [n] \setminus \{i\}} \Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T \cup \{i\}] \cdot \Pr[\mathcal{A} \text{ selects } T \text{ from } [n] \setminus \{i\}]}{\sum_{T \subseteq [n] \setminus \{i\}} \Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T] \cdot \Pr[\mathcal{A} \text{ selects } T \text{ from } [n] \setminus \{i\}]}.$$

To prove the claim, it is enough to show that $\frac{\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T \cup \{i\}]}{\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T]} \leq 2$ for each $T \subseteq [n] \setminus \{i\}$. Recall that V_S is the space of solutions to the system of linear equations $\{\langle x_i, r \rangle = c_r(x_i) : i \in S\}$. Recall also that $\mathcal{A}$ picks $r^* \in V_S$ uniformly at random and outputs $h = c_{r^*}$. Therefore,

$$\Pr[\mathcal{A}(\mathrm{z}) = c_{r^*} \mid S] = \begin{cases} 1/|V_S| & \text{if } r^* \in V_S, \\ 0 & \text{otherwise.} \end{cases}$$

Proof.

If $\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T] = 0$ then $\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T \cup \{i\}] = 0$ because a new constraint does not add new vectors to the space of solutions. If $\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T \cup \{i\}] = 0$, the required inequality holds. If neither of the two probabilities is 0 ,

$$\frac{\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T \cup \{i\}]}{\Pr[\mathcal{A}(\mathrm{z}) = h \mid S = T]} = \frac{1/\left|V_{T\cup\{i\}}\right|}{1/\left|V_T\right|} = \frac{\left|V_T\right|}{\left|V_{T\cup\{i\}}\right|} \le 2.$$

The last inequality holds because in $\mathbb{Z}_2$ (the finite field with 2 elements where arithmetic is performed modulo 2), adding a consistent linear constraint either reduces the space of solutions by a factor of 2 (if the constraint is linearly independent from V_T) or does not change the solutions space (if it is linearly dependent on the previous constraints). The constraint indexed by i has to be consistent with constraints indexed by T, since both probabilities are not 0. $\square$