One of the main tasks in learning theory is to predict the behavior of future data based on observations of past data. For example, if you have observed the input-output pairs

input	output
2	4
4	16
1	1
9	81

you may conclude that you are looking at the function $f(t) = t^2$ and can predict the value of any future input of your choice.

Learning methods are usually robust in the sense that their outcome does not depend on the presence of any particular data item. Unlike in other settings where we had to work (hard) to achieve private data release, in learning privacy sometimes comes for free or with just a little bit of extra work.

1 Private and probably approximately correct

We now consider a model where the objective is to learn an unknown Boolean function $f\colon \{0,1\}^d \to \{0,1\}$ coming from some class C. The past data consists of independent random input-output pairs of the form $(r, f(r))$ where r is sampled from some distribution D.

In approximately correct learning, we want to use this data to learn f well enough in order to predict most future values $f(r)$ where r is sampled from D.

For example, suppose C is the set of ANDs over all possible 2^d subsets of the input bits:

$$C = \left\{f\colon f(r) = \wedge_{i\in S} r_i, S \subseteq [d]\right\}$$

and D is the uniform distribution over $\{0,1\}^n$. Suppose $d = 6$ and we observe the data items:

r	$f(r)$
`001011`	`1`
`000000`	`0`
`000010`	`1`

We can conclude that $f(r) = r_5$ as this is the only function in C that is consistent with the data. On the other hand, if we observe

r	$f(r)$
001011	0
010110	0
100010	0
110100	0
000100	0

then there are several functions in C that are consistent with the data, but we can at least rule out the possibilities that $f(r)$ equals any of the functions 1, r_1, r_2, r_3, r_4, r_5, or r_6. If we make the hypothesis $f(r) = r_1 \wedge \cdots \wedge r_6$, our prediction may not always be accurate but we can be sure to be correct at least 3/4 of the time.

The fundamental theorem of probably approximately correct (PAC) learning states that a moderate amount of data is sufficient to produce an approximately correct hypothesis, with high probability over the choice of the data.

Let $x = ((x_1, y_1), \ldots, (x_n, y_n)) \in (\{0,1\}^{d+1})^n$ be a database of possible input-output pairs. We say a hypothesis function $h \in C$ is *consistent* with x if $h(x_i) = y_i$ for every i between 1 and n.

Theorem 1. *For every class C, function $f \in C$, distribution D, and database x consisting of $n = 2(\ln|C|)/\alpha$ i.i.d. samples of the form $(x_i, f(x_i))$, $x_i \sim D$,*

$$\Pr_x\big[\Pr_{r\sim D}[h(r) = f(r)] \geq 1 - \alpha\big] \geq 1 - e^{-\alpha n/2}.$$

where $h \in C$ is any function consistent with x.

Proof. Fix f and let BAD be the subset of C consisting of those h such that $\Pr_{r\sim D}[h(r) \neq f(r)] > \alpha$. Then

$$\begin{aligned}
\Pr_x[\text{there exists } h \in BAD \text{ that is consistent with } x] &\leq \sum_{h\in BAD} \Pr[h \text{ is consistent with } x] \\
&= \sum_{h\in BAD} \Pr[h(x_i) = f(x_i) \text{ for all } i \in [n]] \\
&< \sum_{h\in BAD} (1-\alpha)^n \\
&\leq |C|(1-\alpha)^n \\
&\leq e^{\ln|C| - \alpha n}.
\end{aligned}$$

Choosing $n = 2\ln|C|/\alpha$ gives the desired bound. □

What about differential privacy? A reasonable model for privacy is to view the database as private and the hypothesis as public. To obtain privacy we should randomize the choice of hypothesis. To preserve accuracy, Theorem 1 suggests that we should favor those hypotheses that are consistent with the database. To this end we represent utility by a consistency score

$$u(x, h) = -|\{i\colon h(x_i) \neq y_i\}|.$$

The highest utility score of zero is given to those h that are consistent with x; each inconsistency is penalized by -1. The sensitivity of u is at 1 as changing one row of the database can create at most one additional inconsistency with respect to a fixed hypothesis. It follows that the exponential mechanism, which chooses hypothesis h with probability proportional to $\exp(\varepsilon u(x,h)/2)$, is ε-differentially private.

What about the accuracy of this mechanism? By Theorem 6 from Lecture 2, the probability that $u(x,h)$ is smaller than $-\alpha n/2$ — that is, that the hypothesis has more than $\alpha n/2$ inconsistencies with respect to x — is at most $|C|e^{-\varepsilon\alpha n/4}$. Using the same definition of BAD, we reason as in the proof of Theorem 1:

$$\begin{aligned}\Pr_x[\exists h \in BAD \text{ s.t. } u(x,h) \geq -\alpha n/2] &\leq \sum_{h \in BAD} \Pr[u(x,h) \geq -\alpha n/2] \\ &= \sum_{h \in BAD} \Pr[h(x_i) \neq f(x_i) \text{ for at most } \alpha n/2 \text{ rows } i] \\ &< \sum_{h \in BAD} e^{-\alpha^2 n/2} \\ &\leq |C| e^{-\alpha^2 n/2}.\end{aligned}$$

The second inequality is the Chernoff bound (the events $h(x_i) \neq f(x_i)$ are independent and each occurs with probability more than α). By a union bound, we get that

$$\begin{aligned}\Pr_{x,h}\big[\Pr_{r\sim D}[h(r) \neq f(r)] > \alpha\big] &\leq \Pr_{x,h}[u(x,h) < -\alpha n/2] + \Pr_x[\exists h \in BAD \text{ s.t. } u(x,h) \geq -\alpha n/2] \\ &< |C| e^{-\varepsilon\alpha n/4} + |C| e^{-\alpha^2 n/2}\end{aligned}$$

so if $n \geq 8\ln|C|/(\alpha \min\{\alpha, \varepsilon\})$ the mechanism is both accurate and private with high probability.

2 Online learning with experts

You are at the Happy Valley racetrack and want to bet in the next horse race but know nothing about horses. You have access to K experts that predict possibly different winners. Knowing nothing about the experts either, you choose one of them at random and go with their prediction.

Once the race is over and you collect your earnings (if any), it is time to bet on the next horse. Now you have some additional information about the experts so perhaps you can do better than choosing one at random. The ones that predicted well in the first race should be trusted more. On the other hand, putting too much trust in any particular expert may not be such a good idea because his correct prediction may just have been a stroke of good luck.

The multiplicative weights update algorithm strikes a balance between going with the winner and choosing at random. At each time step t, the algorithm chooses an expert $i \in \{1, \ldots, K\}$ and obtains a reward of $r_t(i) \in [0,1]$. The rewards of the different experts may be interdependent and may even depend on the choices that the algorithm made in previous time steps.

Algorithm MW:

Set y to be the uniform distribution over the set of experts $\{1, \ldots, K\}$.
For each time step t from 1 to T:
    Sample expert i_t from the distribution y.
    Observe the rewards $r_t(1), \ldots, r_t(K)$ and collect $r_t(i_t)$.
    For each expert i,
        Multiply $y(i)$ by $e^{\alpha r_t(i)}$ and normalize y to $\sum_{i=1}^{K} y(i) = 1$.

The overall reward of this algorithm depends on the performance of the experts; in general, one can never do better than the most successful expert. The MW algorithm performs almost as well as this expert.

Theorem 2. *For $0 < \alpha \leq 1.79$ and every $i \in [K]$,*

$$\mathrm{E}\Big[\sum_{t=1}^{T} r_t(i_t)\Big] \geq \sum_{t=1}^{T} r_t(i) - \alpha T - \ln K/\alpha.$$

In particular, if we set $\alpha = \sqrt{\ln K/T}$, the expected deviation in reward from that of the best expert is at most $2\sqrt{T \ln K}$, or $2\sqrt{\ln K/T}$ per time step.

Proof. We extend the proof of Theorem 3 in Lecture 3. Let x be any distribution over the set of experts and y_t be the state of distribution y before time step t. Using the same calculation as before, we have

$$\mathrm{Div}(x\|y_t) - \mathrm{Div}(x\|y_{t+1}) = \alpha \,\mathrm{E}_{i\sim x}[r_t(i)] - \ln \mathrm{E}_{i\sim y_t}[e^{\alpha r_t(i)}].$$

Since for $\alpha \leq 1.79$ and any $[0,1]$ bounded random variable X,

$$\ln \mathrm{E}[e^{\alpha X}] \leq \mathrm{E}[e^{\alpha X}] - 1 \leq \mathrm{E}[\alpha X + \alpha^2] = \alpha \,\mathrm{E}[X] + \alpha^2$$

we get that

$$\mathrm{Div}(x\|y_t) - \mathrm{Div}(x\|y_{t+1}) \geq \alpha \,\mathrm{E}_{i\sim x}[r_t(i)] - \alpha \,\mathrm{E}_{i\sim y_t}[r_t(i)] - \alpha^2.$$

We now scale by $1/\alpha$, sum over t, and take linearity of expectation to get

$$\frac{1}{\alpha}\Big(\mathrm{Div}(x\|y_0) - \mathrm{Div}(x\|y_T)\Big) \geq \mathrm{E}_{i\sim x}\Big[\sum_{t=1}^{T} r_t(i_t)\Big] - \mathrm{E}\Big[\sum_{t=1}^{T} r_t(i_t)\Big] - \alpha T.$$

Since y_0 is uniform, $\mathrm{Div}(x\|y_0) \leq \ln K$ and so

$$\mathrm{E}\Big[\sum_{t=1}^{T} r_t(i_t)\Big] \leq \mathrm{E}_{i\sim x}\Big[\sum_{t=1}^{T} r_t(i_t)\Big] - \alpha T - \frac{\ln K}{\alpha}.$$

In particular, if we choose x to be the distribution that assigns probability 1 to expert i we obtain the theorem. □

Let us now view the reward matrix $r_t(i)$ as a private database whose t-th row consists of the vector $(r_t(1), \ldots, r_t(K))$ and the choice of experts $(i_1, \ldots, i_T)$ made by MW as public. If we track how $y(t)$ changes over time, we conclude that the outcome i_t at time step t is chosen with probability

$$\frac{1}{Z_t} \prod_{t'=1}^{t} \exp(\alpha r_t(i)) = \frac{1}{Z_t} \exp\Big(\alpha \sum_{t'=1}^{t} r_t(i)\Big)$$

where Z_t is a normalization constant that does not depend on i. This is an instantiation of the exponential mechanism with utility $U(u,i) = \sum_{t'=1}^{t} r_t(i)$. Since U is 1-Lipschitz, i_t is 2α-differentially private.

To analyze the privacy of MW we view it as a product mechanism that produces T independent outputs, each of which is 2α-differentially private. We can therefore conclude that MW is $2\alpha T$-differentially private. If we want to achieve, say, 1-differential privacy we should therefore set $\alpha = 1/2T$, which makes the bound in Theorem 2 useless.

We also know by Theorem 4 from Lecture 4 that MW is $(12T\alpha^2, e^{-T\alpha})$-differentially private. If we set $\varepsilon = 12T\alpha^2$ we derive the following consequence:

Theorem 3. *For $\varepsilon \leq 1$, mechanism MW is $(\varepsilon, e^{-\Omega(\sqrt{T\varepsilon})})$-differentially private with respect to the database $(r_1, \ldots, r_T)$ and achieves expected reward at least*

$$\mathrm{E}\Big[\sum_{t=1}^{T} r_t(i_t)\Big] \geq \max_{i\in[K]} \sum_{t=1}^{T} r_t(i) - O\big(\ln K \cdot \sqrt{T}/\sqrt{\varepsilon}\big).$$

References

These notes are based on Chapter 11 of the survey *The Algorithmic Foundations of Differential Privacy* by Cynthia Dwork and Aaron Roth.

Theorem 2 is a special case of Theorem 2.4 from the survey *The Multiplicative Weights Update Method: A Meta-Algorithm and Applications* by Sanjeev Arora, Elad Hazan, and Satyen Kale.