Privacy in Statistics and Machine Learning — Spring 2021
Lecture 17 & 18: Equilibrium in Zero-Sum Games

Adam Smith and Jonathan Ullman

As promised in Lecture 15, we're going to see how many algorithms from query release fit into a larger framework based on computing equilibrium in two-player zero-sum games. To start we'll focus on the mathematical background before we get back to query-release.

1 Two-Player Zero-Sum Games and the Minmax Theorem

You've all likely played a two-player zero-sum game before: *Rock, Paper, Scissors.* In case you aren't familiar, two players each get to choose an action from the set { rock, paper, scissors } and the game is decided according to the rules rock-beats-scissors, scissors-beats-paper, paper-beats-rock, and the game is a tie if the players each use the same action. What makes this game *zero-sum* is that one player's loss is the other player's gain, and vice versa. Either the two players tie. Exactly one player wins and the other loses.

In this lecture we'll study the key concept of *equilibrium* in two-player zero-sum games, and then discuss applications to query release. Mathematically, we can model a two-player, zero-sum game using the following notation:

- There are two *players*, which we will name Rowena and Colin.
- Rowena has a set of *actions* $\mathcal{R}$ and Colin has a set of actions C.
- There is a *payoff matrix* $M \in \mathbb{R}^{|\mathcal{R}|\times|C|}$ where $M_{i,j}$ represents the *reward* that Rowena gets from Colin if she plays action $i \in \mathcal{R}$ and Colin plays action $j \in C$. Thus, $-M_{i,j}$ is the reward for Colin given the same pair of actions.

What makes the game zero-sum is that Rowena "wins" $M_{i,j}$ and Colin "wins" $-M_{i,j}$, so the amount the two players win is always $M_{i,j} - M_{i,j} = 0$. Thus, the two players' goals are directly opposed, Rowena wants to maximize her reward and Colin wants to minimize her reward.

In this model, we can represent Rock, Paper, Scissors as a game where $\mathcal{R} = C = \{1, 2, 3\}$ with $\{1, 2, 3\}$ representing rock, paper, and scissors, respectively. The payoff matrix is

$$\begin{bmatrix} 0 & -1 & +1 \\ +1 & 0 & -1 \\ -1 & +1 & 0 \end{bmatrix} \tag{1}$$

where we assume that Rowena gets 1 from Colin if she wins the game and vice versa, and both players get 0 in the event of a tie.

Even if you're intuitively familiar with rock, paper, scissors, we'll see the tools needed to understand strategic behavior in games where the solution is less obvious, such as the simple game described by a payoff matrix like this one.

$$\begin{bmatrix} +2 & -1 \\ -2 & +3 \end{bmatrix} \tag{2}$$

1.1 Who Goes First?

If you've played Rock, Paper, Scissors recently, you probably remember that both players typically play *simultaneously*, and the game gets pretty uninteresting if one player has to pick an action first. In general, suppose I require that Rowena goes first and let's think about how she would choose her action. Whatever action $i \in \mathcal{R}$ she takes, Colin will play the action $j \in C$ that maximizes his rewards, or equivalently that minimizes Rowena's reward. Thus, if Rowena plays i, then Colin will choose the action $\arg\min_j M_{i,j}$ and Rowena will get $\min_j M_{i,j}$. This is known as a *best response* for Colin. Thus, Rowena should play i to maximize the amount she will get, knowing how Colin will respond to her action. In particular, she should play $\arg\max_i \min_j M_{i,j}$ and her reward will be $\max_i \min_j M_{i,j}$. For Rock, Paper, Scissors we all know that whatever Rowena plays, Colin has a way to win the game, so Rowena will always lose if she has to go first. In other words,

$$\max_i \min_j M_{i,j} = -1 \tag{3}$$

Now, what if Colin goes first? By a completely symmetric argument, we know that whatever action j that Colin plays, Rowena will chose the action i to maximize her reward, thus Colin should play to minimize Rowena's reward knowing how she will respond. Thus, Rowena's reward when Colin plays first is going to be $\min_j \max_i M_{i,j}$. For Rock, Paper, Scissors, if Colin goes first, then Rowena will always win, or, in other words,

$$\min_j \max_i M_{i,j} = 1 \tag{4}$$

One simple fact is that for any two-player zero-sum game, you would prefer to be the player who chooses their action second rather than the player who chooses their action first. In our notation, this comes out as the following inequality.

$$\max_i \min_j M_{i,j} \le \min_j \max_i M_{i,j} \tag{5}$$

Exercise 1.1. Prove (5).

1.2 Randomized Strategies and the Minmax Theorem

As we've seen, the game Rock, Paper, Scissors is pretty boring if require one of the players to go first. What if we add *randomization* to the mix? That is, suppose Rowena still has to act first, but now Rowena doesn't have to pick a specific action i. Instead, she has to pick a *probability distribution* $\mathbf{r}$ over actions, which we can represent as a column vector $\mathbf{r} \in \Delta(\mathcal{R})$. Given this probability distribution over actions, Colin will then get to choose a probability distribution $\mathbf{c}$ over actions, represented as a column vector $\mathbf{c} \in \Delta(C)$. We will sometimes call these distributions *(randomized) strategies.* One these two strategies are chosen, the players sit back and watch while an action i is chosen according to $\mathbf{r}$ and an action j is chosen *independently* according to $\mathbf{c}$ and Rowena wins $M_{i,j}$.

Given a pair of strategies $\mathbf{r}, \mathbf{c}$, the expected payoff to Rowena is

$$\mathop{\mathbb{E}}_{\substack{i\sim\mathbf{r}\\ j\sim\mathbf{c}}}\left(M_{i,j}\right) = \sum_{\substack{i\in\mathcal{R}\\ j\in C}} \mathbf{r}_i \mathbf{c}_j M_{i,j} = \mathbf{r}^\top M\mathbf{c} \tag{6}$$

where we will often use the matrix expression as a compact way of representing the expected payoff for the pair of strategies (but won't need any fancy linear algebra).

Using the same logic as before, if Rowena has to choose her strategy first, then she will choose $\mathbf{r}$ to maximize her expected reward knowing that Colin will play a best-response, and she will get a payoff of

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} \tag{7}$$

and similarly if Colin has to choose her strategy first, then he will choose $\mathbf{c}$ to minimize Rowena's expected reward, knowing that she will play a best-response, and then Rowena will get a payoff of

$$\min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} \tag{8}$$

Let's see what will happen in Rock, Paper, Scissors when Rowena goes first and chooses some distribution $\mathbf{r}$. Remember that $\mathbf{r}_1$ is the probability of playing rock, $\mathbf{r}_2$ is the probability of paper, and $\mathbf{r}_3$ is the probability of scissors. You all probably intuitively see that the best thing for Rowena to do is to play the uniform distribution $\mathbf{r} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. In this case, no matter what strategy Colin plays, Rowena's expected reward is 0. Thus, for this game,

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} \geq 0 \tag{9}$$

It's also not too hard to check that if Rowena plays any strategy other than the uniform distribution, Colin has a response that makes Rowena's expected reward strictly negative. Thus we have

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} = 0 \tag{10}$$

Exercise 1.2. Prove that if Rowena plays any strategy other than $\mathbf{r} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ then Colin has a strategy $\mathbf{c}$ such that $\mathbf{r}^\top M\mathbf{c} < 0$.

What happens if Colin plays first? By symmetry of the game, Colin also must play the uniform distribution $\mathbf{c}$ or else Rowena will receive a strictly positive reward. Thus

$$\min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} = 0 \tag{11}$$

So, at least for Rock, Paper, Scissors, as long as the players get to choose randomized strategies, it doens't matter who has to pick their strategy first! It turns out that this isn't specific to Rock, Paper, Scissors, and is actually true for *any* two-player, zero-sum game. This fact is what's known as the celebrated *minmax theorem*, due to John von Neumann.

Theorem 1.3 (Minmax Theorem [vN28]). *For any two-player zero-sum game with payoff matrix M,*

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} = \min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} = \mathrm{val}(M) \tag{12}$$

where the quantity $\mathrm{val}(M)$ *is called the* value *of the game.*

In particular, a pair of strategies $(\mathbf{r}, \mathbf{c})$ such that $\mathbf{r}^\top M\mathbf{c} = \mathrm{val}(M)$ are called an *equilibrium* of the game in the sense that neither player can improve their payoff by changing their strategy. A strategy $\mathbf{r}$ such that $\min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} = \mathrm{val}(M)$ is sometimes called a *minmax strategy* and, unsurprisingly, a strategy $\mathbf{c}$ such that $\max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c}$ is sometimes called a *maxmin strategy.*

The relatively easy part of the minmax theorem is showing that Rowena does *at least* as well if she chooses her strategy second as she does if she chooses it first.

Exercise 1.4. Prove the "easy" direction of the minmax theorem,

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} \le \min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} \tag{13}$$

In the next lecture we'll prove the much harder part of the minimax theorem, which says that Rowena can do just as well if she goes first. While there are lots of ways to prove this theorem, we will show a cool proof that deduces the minimax theorem as a consequence of the existence of no-regret online learning algorithms! In addition to being very simple (now that we've done all the hard work of proving that no-regret learning is possible), this proof will also give us some nice, computationally efficient algorithms for computing val(M) and finding equilibrium strategies.

Exercise 1.5. Suppose players take turns playing a best response to one another's strategy (which is sometimes called *best-response dynamics*). Rowena chooses some action $\mathbf{r}_1$, then Colin best-responds with $\mathbf{c}_1 = \arg\min_{\mathbf{c}} r_1^\top Mc$, then Rowena best-responds with $\mathbf{r}_2 = \arg\max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c}_1$, and so on. Specifically,

$$\mathbf{c}_t = \arg\min_{\mathbf{c}} \mathbf{r}_t^\top M\mathbf{c} \text{ and } \mathbf{r}_t = \arg\max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c}_{t-1}$$

What will happen in this process as $t \longrightarrow \infty$? Will $\mathbf{r}_t$ and $\mathbf{c}_t$ converge to an equilibrium of the game?

1.3 Proof of the Minmax Theorem

We're now going to prove the "hard" direction of Theorem 1.3, which informally says that Rowena can always do just as well if she has to choose her action first than she could if she chose her action second. Or, in our notation

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} \ge \min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} \tag{14}$$

Our proof is going to rely on the existence of no-regret learning algorithms in a cute way. Let's start by defining two sequences of distributions on actions $\mathbf{r}_1, \dots, \mathbf{r}_T \in \Delta(\mathcal{R})$ and $\mathbf{c}_1, \dots, \mathbf{c}_T \in \Delta(C)$, one for Rowena and one for Colin. Analogous to how we defined no-regret learning, we'll say that Rowena's sequences of actions has *regret* α to Colin's sequence if Rowena could not improve her reward by more than α per round by using some fixed alternative action $\mathbf{r}$ and likewise for Colin.

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t^\top M\mathbf{c}_t \ge \max_{\mathbf{r}} \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}^\top M\mathbf{c}_t - \alpha \tag{15}$$

and

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t^\top M\mathbf{c}_t \le \min_{\mathbf{c}} \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t^\top M\mathbf{c} + \alpha \tag{16}$$

Note that having low regret is a property of the *pair* of sequences, although just for linguistic convenience we will sometimes simply describe one of the sequences as having low regret.

We'll ignore for now how we can obtain sequences that have no regret, but, perhaps unsurprisingly, we can obtain them using no-regret algorithms.

Fact 1.6. *For any two-player zero-sum game, and every T, there exists a pair of sequences of distributions over actions $\mathbf{r}_1, \dots, \mathbf{r}_T$ and $\mathbf{c}_1, \dots, \mathbf{c}_T$ that have regret at most α_T to one another. Moreover, we can make α_T arbitrarily close to 0 by taking T large enough.*

Given these sequences, we will construct a single strategy for each player by taking the *average* strategy used by each player in the sequence. Specifically,

$$\hat{\mathbf{r}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t \text{ and } \hat{\mathbf{c}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{c}_t \tag{17}$$

Now that we have these strategies, we can show the following.

$$\begin{aligned}
\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} &\geq \min_{\mathbf{c}} \hat{\mathbf{r}}^\top M\mathbf{c} && (18)\\
&= \min_{\mathbf{c}} \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t^\top M\mathbf{c} && (19)\\
&\geq \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t^\top M\mathbf{c}_t - \alpha && \text{(By (16))}\\
&\geq \max_{\mathbf{r}} \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}^\top M\mathbf{c}_t - 2\alpha && \text{(By (15))}\\
&= \max_{\mathbf{r}} \mathbf{r}^\top M\hat{\mathbf{c}} - 2\alpha && (20)\\
&\geq \min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} - 2\alpha && (21)
\end{aligned}$$

Therefore we have statement we want, except for this small error of 2α. However, since we can take the regret bound α to be arbitrarily small by taking T arbitrarily large, we get the desired statement! That is,

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} \geq \min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} \tag{22}$$

This completes the proof of the "hard" direction.

Obtaining (approximate) equilibrium strategies. One important observation is that the proof actually gives us a bit more than just the statement that there exists an equilibrium, specifically it says that if $\mathbf{r}_1, \ldots, \mathbf{r}_T$ and $\mathbf{c}_1, \ldots, \mathbf{c}_T$ each have regret α to one another, then $(\hat{\mathbf{r}}, \hat{\mathbf{c}})$ are *2α-approximate equilibrium strategies*, meaning that

$$\min_{\mathbf{c}} \hat{\mathbf{r}}^\top M\mathbf{c} \geq \text{val}(M) - 2\alpha \text{ and } \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} \leq \text{val}(M) + 2\alpha \tag{23}$$

Thus finding, no-regret strategies actually gives us a way to compute (approximate) equilibrium of two-player zero-sum games.

Exercise 1.7. Prove the statement above that $(\hat{\mathbf{r}}, \hat{\mathbf{c}})$ are 2α-approximate minmax strategies.

1.4 Finding No-Regret Strategies

Now that we know the importance of no-regret strategies, we can look at methods for computing them. There are two particularly useful ones, which can be mixed and matched (i.e. Rowena can use one approach and Colin can use another). We'll consider both of these from Rowena's perspective, but the analogous method works for Colin, by symmetry.

1. **Best response.** Suppose that when Colin plays a strategy $\mathbf{c}_t$, Rowena plays a best-response strategy $\mathbf{r}_t = \arg\max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c}_t$. (Note that, without loss of generality, $\mathbf{r}_t$ can be a deterministic strategy.) Then we can show

$$\frac{1}{T}\sum_{t=1}^{T} \mathbf{r}_t^\top M\mathbf{c}_t = \frac{1}{T}\sum_{t=1}^{T} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c}_t \geq \max_{\mathbf{r}} \frac{1}{T}\sum_{t=1}^{T} \mathbf{r}^\top M\mathbf{c}_t \tag{24}$$

More generally, Rowena can play an *α-approximate best-response*, which is any strategy $\mathbf{r}_t$ such that $\mathbf{r}_t^\top M\mathbf{c}_t \geq \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} - \alpha$. Playing α-approximate best-responses leads to a strategy with regret at most α.

2. **No-regret learning.** Suppose that Rowena uses a no-regret learning algorithm, such as multiplicative weights, to select strategies $\mathbf{r}_1, \ldots, \mathbf{r}_T$. That is, we generate a sequence of *losses* $\ell_t \in \mathbb{R}^{|\mathcal{R}|}$ giving a loss for each action Rowena can take by setting $\ell_t = -M\mathbf{c}_t$. If Rowena uses an algorithm that guarantees regret at most α to any sequence of losses, then Rowena's resulting sequence of strategies $\mathbf{r}_1, \ldots, \mathbf{r}_T$ will have regret at most α to Colin's sequence $\mathbf{c}_1, \ldots, \mathbf{c}_T$.[1]

Exercise 1.8. Prove that Rowena can achieve regret α to any sequence of strategies by Colin by either playing an α-approximate best-response or by using an online learning algorithm with regret at most α.

2 Applications to Query Release

The reason zero-sum games are relevant for query release is because we can write the following *query-release game.* Suppose we have a dataset $\mathbf{x} \in \mathcal{U}^n$ with $|\mathcal{U}| = m$ and a set of linear queries $f_1, \ldots, f_k$ defined by predicates $\varphi_1, \ldots, \varphi_k : \mathcal{U} \to [0, 1]$. We'll assume that the set of queries is closed under complement so that for every query φ_j, the query $1 - \varphi_j$ is also in the set. We can define the following two-player zero-sum game:

1. Rowena will be the *query player* and her action set is $\mathcal{R} = [k]$.

2. Colin will be the *data player* and his action set is $\mathcal{U}$.

3. For a pair of actions (i, z) for $i \in [k]$ and $z \in \mathcal{U}$, Rowena's payoff is

$$M_{i,z} = f_i(\mathbf{x}) - \varphi_i(z) \tag{25}$$

Let's make a couple of observations about the game. First, the sets of actions for each player depend only on the set of queries, and not the data. However, the payoff matrix depends on the dataset $\mathbf{x}$. In particular, if we let $\mathbf{x}, \mathbf{x}'$ be neighboring datasets, and M, M' be the corresponding payoff matrices, then for every $i \in [k]$ and $z \in \mathcal{U}$,

$$|M_{i,z} - M'_{i,z}| = |f_i(\mathbf{x}) - f_i(\mathbf{x}')| \leq 1/n.$$

Second, we can analyze the equilibria of this game. Notice that, since Colin's actions are data-domain elements in $\mathcal{U}$, thus a randomized strategy $\mathbf{c} \in \Delta(\mathcal{U})$ for Colin is a distribution over data-domain elements. That is, Colin's randomized strategies are just histograms corresponding to datasets!

[1] Note that the no-regret algorithms that we've seen assume that the loss vectors are in $[0, 1]^{|\mathcal{R}|}$, so we need that the payoff matrix M has entries bounded in $[0, 1]$ to apply those algorithms directly. However, by shifting-and-scaling the losses, we can get no-regret strategies for any game with payoffs bounded in $[-a, a]$ with a dependence on a in the regret bound.

Therefore, Colin always has a randomized strategy that ensures Rowena's payoff is at most 0, simply play $\mathbf{c} = \mathbf{x}$. If Colin does so, then no matter what Rowena plays,

$$\mathbf{r}^\top M\mathbf{x} = \mathop{\mathbb{E}}_{i\sim\mathbf{r}} (f_i(\mathbf{x}) - f_i(\mathbf{x})) = 0 \tag{26}$$

Thus, the value of the game is at most 0 and it's not too hard to see that it is, in fact, exactly 0, because the queries are closed under complement. That is,

$$\max_{\mathbf{r}} \min_{\mathbf{c}} \mathbf{r}^\top M\mathbf{c} = \min_{\mathbf{c}} \max_{\mathbf{r}} \mathbf{r}^\top M\mathbf{c} = 0 \tag{27}$$

Given that the value of the game is 0, we know that any (α-approximate) equilibrium strategy for Colin is a distribution $\hat{\mathbf{x}} \in \Delta(\mathcal{U})$ such that

$$\max_i f_i(\mathbf{x}) - f_i(\hat{\mathbf{x}}) \leq \max_{\mathbf{r}} \mathop{\mathbb{E}}_{i\sim\mathbf{r}} (f_i(\mathbf{x}) - f_i(\hat{\mathbf{x}})) = \max_{\mathbf{r}} \mathbf{r}^\top M\hat{\mathbf{x}} \leq \alpha \tag{28}$$

Again, because the queries are closed under complement, we also have the same bound on $\max_i f_i(\hat{\mathbf{x}}) - f_i(\mathbf{x})$. Therefore we have the following key fact

Fact 2.1. *For any dataset* $\mathbf{x}$ *and any set of linear queries, if* $\hat{\mathbf{x}}$ *is an* α*-approximate equilibrium strategy for Colin, then,* $\hat{\mathbf{x}}$ *represents a dataset such that*

$$\max_i |f_i(\mathbf{x}) - f_i(\hat{\mathbf{x}})| \leq \alpha \tag{29}$$

2.1 Algorithms for Query Release via Equilibrium Computation

To summarize the last section: *equilibrium strategies for Colin in the query-release game correspond to accurate synthetic datasets!* Using this fact, and our observations about how to find equilibrium strategies in the query-release game, we can find lots of interesting algorithms that represent many of the state-of-the-art approaches to private query release, including MWEM and other algorithms we haven't studied yet. In each of these cases, we have to use special steps and observations to make sure that the game is solved in a way that is private (with respect to the dataset $\mathbf{x}$ defining the payoff matrix).

2.1.1 MWEM as an Equilibrium Computation

Recall that MWEM has the following structure (some details omitted):

1. Initialize the strategy of the data player (Colin) to be the uniform distribution, $\mathbf{c}_1 = (\frac{1}{m}, \ldots, \frac{1}{m})$.
2. For $t = 1, \ldots, T$:

   (a) The query player (Rowena) uses the exponential mechanism to choose some

   $$i_t \approx \arg\max_i f_i(\mathbf{x}) - f_i(\mathbf{c}_t)$$

   (b) The data player (Colin) uses a multiplicative-weights update and sets a new strategy

   $$\mathbf{c}_{t+1,z} = C \cdot \mathbf{c}_{t,z} \cdot (1-\eta)^{-\varphi_{i_t}(z)}$$

   where C is some constant to ensure that $\mathbf{c}_{t+1}$ is a probability distribution.

At a high level, we can see that Colin is playing a *no-regret strategy* and Rowena is playing *approximate best response*. Thus, if we run this algorithm for large T (with a suitable choice of η) then it should converge to an approximate equilibrium. Meaning that the average of Colin's strategies, $\hat{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{c}_t$ will be an accurate synthetic databases.

In order to ensure privacy, we need to make sure that the actions of Rowena are made private, which we can achieve using the exponential mechanism. The actions of Colin, once we release the actions of Rowena, don't actually depend on the sensitive data, so we don't have to do anything in addition to make them private.

There is, however, one more small trick we slipped in here. Notice that we make the updates using the loss vector $\ell_t = \varphi_{i_t}(z)$. However, to get a no-regret strategy for Colin, we should be using the loss vector $\ell_t \varphi_{i_t}(z) - f_{i_t}(\mathbf{x})$. However, these loss vectors depend on the dataset, and thus we'd need to use some additional steps to preserve privacy that would compromise accuracy. However, notice that using the "proper" losses would scale $\mathbf{c}_{t+1,z}$ by a factor of $(1-\eta)^{f_{i_t}(\mathbf{x})}$, which is independent of z. Thus, adding this extra term to the loss vector would simply be cancelled out by the renormalization to make $\mathbf{c}_{t+1}$ a distribution! So we may as well not include this factor in the updates to avoid costing privacy.

2.1.2 The Dual Query Algorithm

Now that we have the tools to fit MWEM into a general framework of privately computing equilibria of the query-release game, we can try other approaches for computing equilibrium in this game, many of which have nice properties. One such approach is called *DualQuery* [GAH+14]. Roughly speaking, the algorithm swaps the approach for Rowena and Colin—Rowena will now play a no-regret strategy based on multiplicative weights while Colin will play best-responses. At a high level the algorithm has the following form:

1. Initialize the strategy of the query player (Rowena) to be the uniform distribution, $\mathbf{r}_1 = (\frac{1}{k}, \ldots, \frac{1}{k})$.

2. For $t = 1, \ldots, T$:

    (a) The query player (Rowena) takes *samples* $i_{t,1}, \ldots, i_{t,S}$, for some parameter S, from the distribution $\mathbf{r}_t$. Let $\tilde{\mathbf{r}}_t$ be the uniform distribution over these samples.

    (b) The data player (Colin) returns the deterministic action

    $$x_t = \arg\min_{z} \mathop{\mathbb{E}}_{i \sim \tilde{\mathbf{r}}_t} (-\varphi_i(z))$$

    (c) The query player (Rowena) uses a multiplicative weights update based on x_t to obtain a new strategy $\mathbf{r}_{t+1}$.

At a high-level, Colin is playing best responses, whereas Rowena is using a no-regret sequence, so at the end of the game, $\hat{\mathbf{x}} = (x_1, \ldots, x_T)$, which represents the average of Colin's actions $x_1, \ldots, x_T$, should be a synthetic dataset with error α. But there are a number of complications. First, Colin is not quite playing a best response, again because the term $f_i(\mathbf{x})$ is missing. However, since $f_i(\mathbf{x})$ doesn't depend on the choice of z, whether or not it is included doesn't affect Colin's optimization problem, so we can simply not include it to avoid having Colin's response depend on private data.

Second, Colin is not quite playing a best-response to Rowena's strategies $\mathbf{r}_t$, but to an approximation $\tilde{\mathbf{r}}_t$. However, if we choose S large enough, then $\mathbf{r}_t$ and $\tilde{\mathbf{r}}_t$ will be close in a strong sense, so whatever Colin chooses will be close to a maximizer for Rowena's true distribution $\mathbf{r}_t$, and thus Colin is playing approximate best responses to Rowena's distributions.

Last, we need to do something to ensure privacy. It's actually not obvious as the algorithm is described that we've done *anything* to make it private. However, the sampling step is the key. What's not obvious without carefully studying the distribution $\mathbf{r}_t$ obtained by multiplicative weights updates is that it has a nice form where changing the dataset from $\mathbf{x}$ to a neighboring $\mathbf{x}'$ does not change the distribution $\mathbf{r}_t$ much at any point in the algorithm. As a result, we can argue that, for an appropriate choice of S, the samples, and thus the approximate distribution $\tilde{\mathbf{r}}_t$ is actually differentially private!

The key advantage of using this algorithm is that Colin's optimization problem

$$\arg\min_{z} \mathop{\mathbb{E}}_{i \sim \tilde{\mathbf{r}}_t} (-\varphi_i(z))$$

is *independent of the private data*. Thus, even when we have to choose z from an extremely large set, we can use various heuristic solvers (e.g. SAT-solvers, integer programming solvers like CPLEX, or optimizers based on gradient descent) in an "off-the-shelf" way without concern for privacy. The heuristic solvers may or may not actually produce best responses, but the algorithm will be private.

Additional Reading and Watching

- …

References

[GAH+14] Marco Gaboardi, Emilio Jesús Gallego Arias, Justin Hsu, Aaron Roth, and Zhiwei Steven Wu. Dual query: Practical private query release for high dimensional data. In *ICML*, 2014.

[vN28] John von Neumann. Zur theorie der gesellschaftsspiele. *Mathematische annalen*, 100(1):295–320, 1928.