Homework #1 Solutions
CS 6501: Learning and Game Theory (Fall'19)

Minbiao Han   Jing Ma   Haifeng Xu

General Instructions The assignment is meant to be challenging. Feel free to discuss with fellow students, however please write up your solutions independently (e.g., start writing solutions after a few hours of any discussion) and acknowledge everyone you discussed the homework with on your writeup. The course materials are all on the course website: http://www.haifeng-xu.com/cs6501fa19. You may refer to any materials covered in our class. However, any attempt to consult outside sources, on the Internet or otherwise, for solutions to any of these homework problems is *not* allowed.

Whenever a question asks you to "show" or "prove" a claim, please provide a formal mathematical proof. These problems have been labeled based on their difficulties. `Short` problems are intended to take you 5-15 minutes each and `medium` problems are intended to take 15-30 minutes each. `Long` problems may take anywhere between 30 minutes to several hours depending on whether inspiration strikes.

Finally, please write your solutions in latex — hand written solutions will not be accepted. Hope you enjoy the homework!

Problem 1

Consider a Linear Program (LP) in the following standard form where $c \in \mathbb{R}^n, A \in \mathbb{R}^{m\times n}, b \in \mathbb{R}^m$.

$$\begin{aligned} \text{maximize} \quad & c^T \cdot x \\ \text{subject to} \quad & Ax \le b \\ & x \ge 0 \end{aligned} \tag{1}$$

Prove the following facts about the LP.

1. (`Short,3 points`) The feasible region of the LP is a convex set.

   Solution. For any point x_1, x_2 in the feasible region of above LP, $\forall p \in [0,1]$, let $x_3 = p \cdot x_1 + (1-p)\cdot x_2$, we have:
   $Ax_3 = A(p\cdot x_1 + (1-p)\cdot x_2) = pAx_1 + (1-p)Ax_2 \le pb + (1-p)b = b$,
   $x_3 = p\cdot x_1 + (1-p)\cdot x_2 \ge 0\cdot p + (1-p)\cdot 0 = 0$.
   Thus, $\forall p \in [0,1]$, $p\cdot x_1 + (1-p)\cdot x_2$ is in the feasible region. So it is a convex set. ■

2. (`Short,3 points`) At any vertex of the feasible region, n linearly independent constraints are satisfied with equality (a.k.a. *tight*). Note that there are $n+m$ linear constraints in LP (1) because $x \ge 0$ account for n linear constraints.

Solution. Let x^* be any vertex and denote all the tight constraints at x^* as $\alpha_l \cdot x^* = \beta_l$ for l in some set L (these constraints have the form of either $a_i x^* = b_i$ or $x_i^* = 0$). So all the other constraints *strictly* hold at x^*.

Prove by contradiction. Suppose the conclusion is not true, that is, we can find at most $n-1$ independent tight constraints at x^*. Then the subspace spanned by vectors in $\{\alpha_l : l \in L\}$ has dimension at most $n-1$. Therefore, we can find a non-zero direction $d \in \mathbb{R}^n$ that is orthogonal to α_l, i.e., $d \cdot \alpha_l = 0$, for all $l \in L$. We claim that $x^* \pm \epsilon d$ is feasible for any sufficiently small ϵ. This is because $\alpha_l(x^* \pm \epsilon d) = \alpha_l x^* + \epsilon \alpha_l \cdot d = \beta_l$ remains tight for all $l \in L$. Since ϵ is sufficiently small, all the un-tight constraints remain un-tight at $x^* \pm \epsilon d$. However, this is a contradiction to the fact that x^* is a vertex. ■

3. (`Short, 3 points`) The set of optimal solutions of the LP is a convex set.

   Solution. For any point x_1, x_2 in the set of optimal solutions of the LP, $\forall p \in [0,1]$, $x_3 = p \cdot x_1 + (1-p) \cdot x_2$, we have:
   x_3 is also feasible (proven by 1);
   Let the optimal objective value be $opt = c^T x_1$, then $c^T \cdot x_3 = c^T \cdot (p \cdot x_1 + (1-p) \cdot x_2) = opt$;
   Thus, $\forall p \in [0,1]$, $p \cdot x_1 + (1-p) \cdot x_2$ is also in the set of optimal solutions, so it is a convex set. ■

4. (`Short, 3 points`) We learned in class that the dual of LP (1) is the following LP (2)

$$\begin{array}{ll} \text{minimize} & b^T \cdot y \\ \text{subject to} & A^T y \geq c \\ & y \geq 0 \end{array} \tag{2}$$

   Prove that the dual of LP (2) is the original LP (1).

   Solution. Linear program (2) can be written as the standard form LP:

$$\begin{array}{ll} \text{maximize} & (-b)^T \cdot y \\ \text{subject to} & (-A)^T x \leq -c \\ & y \geq 0 \end{array} \tag{3}$$

   The dual of the LP (3) is thus

$$\begin{array}{ll} \text{minimize} & (-c)^T \cdot x \\ \text{subject to} & (-A)x \geq -b \\ & x \geq 0 \end{array} \tag{4}$$

   which is equivalent to the original LP (1). Therefore, the dual of the dual is the original LP. ■

Problem 2

Prove the following projection lemma and separating hyperplane theorem.

1. (**Projection Lemma**, `Medium, 5 points`) Let $Z \subset \mathbb{R}^n$ be a nonempty closed convex set and $y \notin Z$ be any point in $\mathbb{R}^n$. Prove that there exists $z^* \in Z$ that has the minimum l_2 distance from y among all $z \in Z$. Moreover, $\forall z \in Z$ we have $(y - z^*)^T \cdot (z - z^*) \leq 0$. (hint: use Weierstrass' Theorem).

Solution. Choose any $z_0 \in Z$ and let $Z_0 = \{z \in Z : ||z - y||_2 \leq ||z_0 - y||\}$, thus Z_0 is a compact set (note, Z itself may not be compact since it may be unbounded). We have $\min_{z \in Z} ||z - y||_2 = \min_{z \in Z_0} ||z - y||_2$ and by Weierstrass Theorem, $\min_{z \in Z_0} ||z - y||_2$ always achieves a minimum point $z^* \in Z_0 \subseteq Z$, so $z^* \in Z$ also minimizes $\min_{z \in Z} ||z - y||_2$ as desired.

Let $z \in Z$ and $\epsilon \in [0, 1]$, then by definition we have for any $\epsilon \in (0, 1)$,

$$
\begin{aligned}
& \|y - [(1-\epsilon) \cdot z^* + \epsilon \cdot z]\|_2 \geq \|y - z^*\|_2 \\
\Rightarrow\quad & 2\epsilon(y - z^*)^T(z^* - z) + \epsilon^2(z^* - z)^T(z^* - z) \geq 0 \\
\Rightarrow\quad & (y - z^*)^T(z^* - z) + \frac{\epsilon}{2}(z^* - z)^T(z^* - z) \geq 0 \\
\Rightarrow\quad & (y - z^*)^T(z^* - z) \geq -\frac{\epsilon}{2}(z^* - z)^T(z^* - z)
\end{aligned}
$$

Since this holds for any $\epsilon \in (0, 1)$, let $\epsilon \to 0$, then we have $(y - z^*)^T(z^* - z) \geq 0$ or equivalently $(y - z^*)(z - z^*) \leq 0$, as desired.

■

2. (**Separating Hyperplane Theorem**, `Short, 3 points`) Let $Z \subset \mathbb{R}^n$ be a nonempty closed convex set and let $y \notin Z$ be any point in $\mathbb{R}^n$. Prove that there exists a hyperplane $\alpha^T \cdot x = \beta$ that strictly separates y from Z. That is, $\alpha^T \cdot z \geq \beta$ for any $z \in Z$ but $\alpha^T \cdot y < \beta$.

   Solution. Let $z^* \in Z$ minimizes the l_2 distance from y among all $z \in Z$, and $\alpha = z^* - y$, $\beta = \frac{1}{2}\alpha \cdot (z^* + y)$.

   Since $y \notin Z$, we know $\alpha \neq 0$ and thus $0 < \alpha^T \cdot \alpha = \alpha^T \cdot z^* - \alpha^T \cdot y$. Therefore,

$$
\begin{aligned}
& \alpha^T \cdot z^* > \alpha^T \cdot y \\
\Rightarrow\quad & \alpha^T(z^* + y) > 2\alpha^T \cdot y \\
\Rightarrow\quad & \beta > \alpha^T \cdot y
\end{aligned}
$$

   On the other hand, based on the projection lemma, we have $0 \geq (y - z^*)(z - z^*) = -\alpha(z - z^*)$, which implies $\alpha z \geq \alpha z^*$. Since $\alpha z^* > \alpha y$ as shown above, we have $\alpha z \geq \alpha z^* > \frac{1}{2}(\alpha z^* + \alpha y) = \beta$ as desired. ■

Problem 3: Linear Programming for Machine Learning

In this question, you will learn to formulate some machine learning problems as linear programs. Let us assume that there are n data points $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$ where $\mathbf{x}_i \in \mathbb{R}^m$ is interpreted as an m-dimensional feature vector and $y_i \in \mathbb{R}$ is the corresponding label.

1. (**Data Fitting**, `Short, 4 points`) We want to construct a linear predictive model $\mathbf{a} \cdot \mathbf{x}$ to fit the value y. One possible loss function of this fitting is the worst-case error, i.e., $\max_i |\mathbf{a} \cdot \mathbf{x}_i - y_i|$. Show that computing the linear predictive model that minimizes the worst-case error can be formulated as a linear program.

Solution. Let $z = \max_i |\mathbf{a} \cdot \mathbf{x}_i - y_i|$, then the predictive model which minimizes the worst-case error can be formulated as the following (non-linear yet) program:

$$\begin{array}{ll} \text{minimize} & z \\ \text{subject to} & |\mathbf{a} \cdot \mathbf{x}_i - y_i| \le z, \quad \text{for } \forall i \in [m]. \\ & z \ge 0 \end{array}$$

Unfolding the absolute value symbol, we can obtain the following linear program:

$$\begin{array}{ll} \text{minimize} & z \\ & \mathbf{a} \cdot \mathbf{x}_i - y_i \le z \\ & -\mathbf{a} \cdot \mathbf{x}_i + y_i \le z \\ & z \ge 0 \end{array}$$

■

2. (**Linear Classification**, `Short, 4 points`) When $y_i \in \{-1, 1\}$ is a binary label for data point i, this gives rise to a binary classification problem. In linear classification, we seek to find a hyperplane $\mathbf{a} \cdot \mathbf{x} - b = 0$ that strictly separates the data points with label 1 from the points with label -1. That is, $\mathbf{a} \cdot \mathbf{x}_i - b > 0$ if $y_i = 1$ and $\mathbf{a} \cdot \mathbf{x}_i - b < 0$ if $y_i = -1$ (note that the requirement "$<$" or "$>$" is strict). For convenience, we also call such a hyperplane *separating hyperplane*. Show that computing a separating hyperplane or asserting that it does not exist can be formulated as a linear feasibility problem (i.e,, a linear program with an arbitrary objective function).

   (Note that in linear programs, any linear constraint *cannot* have strict inequalities like "$>$" or "$<$"; see the general form of LPs in Lecture 2 slides.)

   Solution. Notice that if such a hyperplane $\mathbf{a} \cdot \mathbf{x} - b = 0$ exists, then there must exist a small enough $\epsilon > 0$ such that $\mathbf{a} \cdot \mathbf{x}_i - b \ge \epsilon$ if $y_i = 1$ and $\mathbf{a} \cdot \mathbf{x}_i - b \le -\epsilon$ if $y_i = -1$. Without loss of generality, we can further assume $\epsilon = 1$ because we divide both sides of the inequalities by $1/\epsilon$. As a result, the following linear feasibility problem computes such a hyperplane

$$\begin{array}{ll} \mathbf{a} \cdot \mathbf{x}_i - b \ge 1, & \text{for } \forall y_i = 1. \\ \mathbf{a} \cdot \mathbf{x}_i - b \le -1, & \text{for } \forall y_i = -1. \end{array} \tag{5}$$

■

Problem 4: Rock-Paper-Scissor

In this problem, you will learn to master the rock-paper-scissor game. Recall that the game has the following payoff structure where each utility (x, y) means the row player receives x and the column player receives y.

1. (`Short, 3 points`) Prove that the above rock-paper-scissor has a *unique* Nash equilibrium, which is that each player picks one of $\{Rock, Paper, Scissor\}$ uniformly at random.

	Rock	Paper	Scissor
Rock	(0, 0)	(-1, 1)	(1, -1)
Paper	(1, -1)	(0, 0)	(-1, 1)
Scissor	(-1, 1)	(1, -1)	(0, 0)

Table 1: Payoffs of the Standard Rock-Paper-Scissor Game

Solution. It is easy to see that uniform randomization is a Nash equilibrium. We prove it is the unique one. First, it is easy to verify that there is no pure Nash Equilibrium for this game. Let $x \in \Delta_3, y \in \Delta_3$ denote a pair of Nash equilibrium strategies for the row and column player, respectively. Since this is a zero-sum game, we know that x must be a maximin strategy, that is

$$\begin{aligned} x &= \operatorname*{argmax}_{x \in \Delta_3} \min\{x \cdot (0,1,-1), x \cdot (-1,0,1), x \cdot (1,-1,0)\} \\ &= \operatorname*{argmax}_{x \in \Delta_3} \min\{x_2 - x_3, x_3 - x_1, x_1 - x_2\} \end{aligned}$$

It is easy to see that if x_1, x_2, x_3 do not equal each other, $\min\{x_2 - x_3, x_3 - x_1, x_1 - x_2\}$ will be a strictly negative number whereas $x_1 = x_2 = x_3 = 1/3$ will make $\min\{x_2 - x_3, x_3 - x_1, x_1 - x_2\} = 0$. Therefore, the argmax problem must solve to the unique solution $x_1 = x_2 = x_3 = 1/3$, which thus is the unique maximin strategy. Similarly, there is also a unique column player minimax strategy, thus the uniform randomization is the unique Nash equilibrium.

■

2. (`Medium, 5 points`) Consider the situation where the column player is forbidden to play *Scissor* (equivalently, the last column of the above payoff matrix is deleted). What is the Nash equilibrium of this new variant of the game.

   Solution. The game is still zero-sum. Applying the maximin theorem, it can be shown that the following is the unique Nash equilibrium: mixed strategy for row player is $\mathbf{Pr}(Paper) = \frac{2}{3}$ and $\mathbf{Pr}(Scissor) = \frac{1}{3}$; the mixed strategy for the column player is $\mathbf{Pr}(Rock) = \frac{1}{3}$ and $\mathbf{Pr}(Paper) = \frac{2}{3}$. ■

3. (`Short, 3 points`) Consider the situation where the two players are encouraged to collaborate. In particular, if they play the same action, each will receive 0.5. This results in the following game variant. What is the Nash equilibrium of this new game?

	Rock	Paper	Scissor
Rock	(0.5, 0.5)	(-1, 1)	(1, -1)
Paper	(1, -1)	(0.5, 0.5)	(-1, 1)
Scissor	(-1, 1)	(1, -1)	(0.5, 0.5)

Table 2: Payoffs of the Rock-Paper-Scissor Game with Encouraged Collaboration

Solution. It can be verified that playing uniformly at random is still a Nash equilibrium of this game. The following question will show that this is actually the unique Nash equilibrium — in fact, it is the unique correlated equilibrium. ■

4. (`Medium, 5 points`) Imagine that you are an outsider who watches two players playing the above game variant with encouraged collaboration, and you can recommend actions to the two players using a correlated equilibrium. If you want to *maximize* the sum of the two players' expected utilities, which correlated equilibrium should you use? If you want to *minimize* the sum of their expected utilities, which correlated equilibrium should you use?

Solution. Let s_1 denote the row player's action, s_2 denote the column player's action, and S denote the set of pure strategy profiles. The the correlated equilibrium is a distribution over S, denoted by $P(s_1, s_2)$, that satisfies the following linear constraints.

$$\begin{aligned} &\sum_{s_{-i}} P(s_i, s_{-i}) \cdot u_i(s_i, s_{-i}) \geq \sum_{s_{-i}} P(s_i, s_{-i}) \cdot u_i(s_i', s_{-i}), \quad \text{for } i \in \{1,2\}, s_i, s_i'. \\ &P \text{ is a distribution.} \end{aligned} \tag{6}$$

For notational convenience, let

$$\begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ x_3 & y_3 & z_3 \end{bmatrix}$$

denote the probabilities for the following action profiles.

$$\begin{bmatrix} (Rock, Rock) & (Rock, Paper) & (Rock, Scissor) \\ (Paper, Rock) & (Paper, Paper) & (Paper, Scissor) \\ (Scissor, Rock) & (Scissor, Paper) & (Scissor, Scissor) \end{bmatrix}$$

Then the correlated equilibrium that minimizes the sum of player's utilities can be computed by the following linear program.

$$\begin{aligned} \text{minimize} \quad & x_1 + y_2 + z_3 \\ \text{subject to} \quad & 0.5x_1 - y_1 + z_1 \geq x_1 + 0.5y_1 - z_1 \\ & 0.5x_1 - y_1 + z_1 \geq -x_1 + y_1 + 0.5z_1 \\ & x_2 + 0.5y_2 - z_2 \geq 0.5x_2 - y_2 + z_2 \\ & x_2 + 0.5y_2 - z_2 \geq -x_2 + y_2 + 0.5z_2 \\ & -x_3 + y_3 + 0.5z_3 \geq 0.5x_3 - y_3 + z_3 \\ & -x_3 + y_3 + 0.5z_3 \geq x_3 + 0.5y_3 - z_3 \\ & 0.5x_1 - x_2 + x_3 \geq x_1 + 0.5x_2 - x_3 \\ & 0.5x_1 - x_2 + x_3 \geq -x_1 + x_2 + 0.5x_3 \\ & y_1 + 0.5y_2 - y_3 \geq 0.5y_1 - y_2 + y_3 \\ & y_1 + 0.5y_2 - y_3 \geq -y_1 + y_2 + 0.5y_3 \\ & -z_1 + z_2 + 0.5z_3 \geq 0.5z_1 - z_2 + z_3 \\ & -z_1 + z_2 + 0.5z_3 \geq z_1 + 0.5z_2 - z_3 \\ & x_1 + y_1 + z_1 + x_2 + y_2 + z_2 + x_3 + y_3 + z_3 = 1 \\ & x, y, z \geq 0 \end{aligned} \tag{7}$$

We claim that the optimal solution to the above LP is $x_1 = y_1 = z_1 = x_2 = y_2 = z_2 = x_3 = y_3 = z_3 = \frac{1}{9}$, which is a correlated equilibrium (in fact a Nash equilibrium, as proved in the previous problem) and achieves objective value $1/3$. To see this, adding all the constraints, except the last two, in LP 7 gives us $2(x_1 + y_2 + z_3) \geq y_1 + z_1 + x_2 + z_2 + x_3 + y_3$. Adding $x_1 + y_2 + z_3$ on both sides of this inequality and using the fact that $x_1 + y_1 + z_1 + x_2 + y_2 + z_2 + x_3 + y_3 + z_3 = 1$, we obtain

$x_1 + y_2 + z_3 \geq 1/3$. Therefore, the optimal solution to LP (7) is at least $1/3$. So uniform distribution over S is indeed optimal.

Maximizing the sum of player utilities turns out to be trickier. But it turns out that one can use LP solver to verify that the feasible region of LP (7) only has a single point $x_1 = y_1 = z_1 = x_2 = y_2 = z_2 = x_3 = y_3 = z_3 = \frac{1}{9}$. So this correlated equilibrium is unique, and also maximizes sum of player's utilities.

■

Problem 5: Stackelberg Games

In this problem, you will learn another type of games called **Stackelberg games**. A Stackelberg game is a two-player game but with *sequential* player moves. In particular, a normal-form Stackelberg game is described by two matrices $A, B \in \mathbb{R}^{n\times m}$ where A is the payoff matrix of the row player who has action set $[n] = \{1, \cdots, n\}$ and B is the payoff matrix of the column player who has action set $[m] = \{1, \cdots, m\}$. The row player moves first (call *her* the *leader*) and the column player (call *him* the *follower*) moves second and thus can see the row player's strategy and then responds with his best action. Similar to the argument we saw in class, such a best response can without loss of generality be a pure best response. Sometimes there may be multiple best responses. In this case we assume that the follower is a benign player so that he will always pick the one that is the best for the leader, i.e., the follower breaks ties in favor of the leader.

It is not difficult to see that, after seeing the leader's strategy — either pure strategy or mixed strategy — the follower's best response action is easy to compute. That is, simply check the utility of each follower action $j \in [m]$ and then pick the best one. Therefore, research in Stackelberg games mainly focuses on computing the optimal leader strategy, which is also called the leader's Strong Stackelberg Equilibrium (SSE) strategy.

Answer the following questions about Stackelberg games.

1. `(Short, 3 points)` **A warm-up example.** Recall the traffic light game from Lecture 4, as follows. Assume that the row player is the leader and she can only play a pure strategy[1], what is the leader's SSE strategy?

	STOP	GO
STOP	(-3, -2)	(-3, 0)
GO	(0, -2)	(-100, -100)

Table 3: Payoffs of the Traffic Light Game

Solution. We claim that the leader's SSE strategy is GO. First, when leader plays GO, the follower's best response is to $STOP$, in which case the leader receives utility 0. Note that the leader's largest possible utility is 0 as that is the largest number in all his payoffs, so playing GO is the leader's best strategy to commit to. ■

[1]For example, maybe because the follower can observe whatever pure action the leader takes.

2. (`Short, 3 points`) Consider the normal-form Stackelberg game and assume that the leader can only play a pure strategy. Show that there is a $\mathcal{O}(nm)$ time algorithm that computes the leader's pure SSE strategy.

   Solution. For each pure leader strategy $i \in [n]$ to commit to, the leader can calculate her utility by predicting which j the follower plays, which is a best follower response (i.e., it maximizes $B(i, j)$). If there are multiple follower best response, the follower shall pick the one favors the leader. This takes time $O(m)$.

   By enumerating all n leader strategies, the leader can figure out the best pure strategy to commit to. In total the running time is $n \times O(m) = O(mn)$. ■

3. (`Medium, 5 points`) We now consider the case where the leader can play a mixed strategy. To compute the leader's SSE (mixed) strategy, consider the following simpler *SSE with promise* problem. That is, imagine that there is an oracle who promises us that when the leader plays the mixed SSE strategy, the follower's best response action will be j^*. Show that given this credible promise, the leader's SSE strategy can be computed by a linear program.

   Use one or two sentences to briefly explain how we can still compute the leader's SSE strategy efficiently even without the oracle's promise, by solving m linear programs.

   Solution. Denote the leader's mixed strategy by $x (\in \Delta_n)$. Since the oracle tells us that the optimal follower response is j^*, $x (\in \Delta_n)$ must then indeed induce j^* as the optimal follower response and is the optimal mixed strategy that does so. Thus, x can be computed by the following LP:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i\in[n]} x(i) \cdot A(i, j^*) \\
\text{subject to} \quad & \sum_{i\in[n]} x(i) \cdot B(i, j^*) \geq \sum_{i\in[n]} x_1(i) \cdot B(i, j'), \quad && \text{for } j' \in [m]. \\
& \sum_{i\in[n]} x(i) = 1 \\
& x(i) \in [0, 1], && \text{for } i \in [n].
\end{aligned} \tag{8}$$

   Even without the oracle's promise, the leader's SSE strategy can still be computed by trying out all m possible j^*'s and pick the one that results in the highest leader utility. ■

4. (`Medium, 5 points`) Prove that the leader's utility by playing the SSE mixed strategy (and the follower will best respond) is at least her utility in any Nash equilibrium of the game when players move simultaneously.

   Solution. Without loss of generality, let player A be the leader. Let (x^*, y^*) be any Nash equilibrium of the game, and $u^* = (x^*)^T A y^*$ be the leader's utility in this Nash equilibrium. We prove that committing to x^* will lead to leader utility $u^{lead} \geq u^*$. This is due to the assumption of breaking ties in favor of the leader. In particular, since in Nash equilibrium, player B plays action j with positive probability only when j is a best response. Therefore, u^* is the average leader utility over randomness of all follower best response actions to x^* where as u^{lead} is the maximum leader utility among all follower best response actions. Thus, $u^{lead} \geq u^*$. In SSE, since the leader will pick the best mixed strategy to commit to, so her expected utility would only even be larger than u^*, concluding the proof. ■

Problem 6: Boosting (`Long, 10 points`)

A fundamental concept in learning theory is *boosting*, intuitively means that classifiers that perform only slightly better than random guess can be turned into a classifier that is never wrong. In this question, you will prove a basic version of this celebrated result using the minimax theorem for zero-sum games.

Let $\mathcal{X} = \{x_1, \cdots, x_n\}$ be any feature space and $\mathcal{H} = \{h : X \to \{-1, 1\}\}$ be a set of classifiers over $\mathcal{X}$ (a.k.a., hypothesis class). For example, $\mathcal{H}$ could be the set of all linear classifiers. However, for simplicity, in this question we will assume that $\mathcal{H} = \{h_1, \cdots, h_m\}$ is finite. Let $g : \mathcal{X} \to \{-1, 1\}$ be the ground truth, i.e., the true label of x_j is $g(x_j)$.

The *weak learnability assumption* on $\mathcal{H}$ says that $\mathcal{H}$ is good in the following sense: there exists $\epsilon > 0$ such that for any distribution $p(\in \Delta_n)$ over $\mathcal{X}$, there exists a classifier h_i such that h_i is correct with probability at least $\frac{1}{2} + \epsilon$ for point x drawn from p, or more formally,

$$\sum_{j=1}^{n} p_j \cdot \mathbb{I}[h_i(x_j) = g(x_j)] \geq \frac{1}{2} + \epsilon,$$

where $\mathbb{I}[h_i(x_j) = g(x_j)]$ is the indicator function. That is, $\mathbb{I}[h_i(x_j) = g(x_j)]$ equals 1 if $h_i(x_j) = g(x_j)$ and equals 0 otherwise.

It turns out that weak learnability implies something much stronger — we can combine classifiers in $\mathcal{H}$ to construct a classifier that is always correct (a.k.a., *strong learnability*), formally stated as follows.

If $\mathcal{H}$ satisfies the weak learnability assumption, then there always exists a distribution $q(\in \Delta_m)$ over $\mathcal{H}$ such that the following weighted classifier:

$$h_q(x) = \begin{cases} 1 & if \quad \sum_{i=1}^{n} q_i h_i(x) \geq 0 \\ -1 & otherwise \end{cases}$$

is always correct, that is, $h_q(x) = g(x)$ for any $x \in \mathcal{X}$.

Prove the above statement.

[Hint: The classification problem can be viewed as a zero-sum game played between a *classifier designer* whose pure strategy is to pick a classifier from $\mathcal{H}$ and an *adversary* whose pure strategy is to pick a data point from $\mathcal{X}$. Think about how to define the payoff matrix of this game and what weak learnability means in the zero-sum game context.]

Solution. We view the classification problem as a zero-sum game, where a classifier designer's pure action is to pick a classifier from $h_i\mathcal{H}$ and an adversary's pure action is to pick a data point $x_j \in \mathcal{X}$, with the payoff matrix u for the designer defined as the follows (utility for the adversary is the opposite):

$$u(i, j) = \begin{cases} 1 & if \quad h_i(x_j) = g(x_j) \\ -1 & otherwise \end{cases}$$

Note that for any i, p, we have

$$u(i, p) = \sum_{j=1}^{n} p_j u(i, j) = \sum_{j:\mathbb{I}(h_i(x_j)=g(x_j))} p_j - \sum_{j:\mathbb{I}(h_i(x_j)\neq g(x_j))} p_j \geq \frac{1}{2} + \epsilon - (\frac{1}{2} - \epsilon) = 2\epsilon.$$

The weak learnability interpreted in the game constructed above means

$$\min_{p \in \Delta_n} \max_{i} u(i, p) \geq 2\epsilon$$

where $u(i,p) = \sum_{j=1}^{n} u(i,j)p_j$. By the Maximin Theorem, we know that

$$\max_{q\in\Delta_m} \min_j u(q,j) = \min_{p\in\Delta_n} \max_i u(i,p) \geq 2\epsilon.$$

Therefore, there exists a q such that $u(q,j) > 0$ for any j. Now if $g(x_j) = 1$, then $h_i(x_j) = u(i,j)$ and $\sum_{i=1}^{n} q_i h_i(x_j) = \sum_{i=1}^{n} q_i u(i,j) > 0$. If $g(x_j) = -1$, then $h_i(x_j) = -u(i,j)$ and $\sum_{i=1}^{n} q_i h_i(x_j) = -\sum_{i=1}^{n} q_i u(i,j) < 0$. Therefore, using such a q as the weight for the classifier defined in the claim will result in correct classification for any x_j. ■

Problem 7: Optimal Strategic Attack to ML Algorithms (`Long, 10 points`)

Strategic or adversarial attacks to machine learning algorithms has been a hot research topic recently. In this question, you will device an *optimal strategic attack* to the machine learning algorithm discussed in Lecture 1 of our class. In particular, we studied the problem of selling a product (with unlimited supply) to N sequentially arriving buyers. All the buyers have the same value $v \in [0,1]$ for the product but v is unknown to the seller. In order to maximize the seller's revenue, we described an online learning algorithm for selling the product that achieves regret $(2\log\log N + 1)$.

In this question, we concern a slight variant of the above problem. That is, the seller sells the product (with unlimited supply) to a *single buyer* who repeatedly shows up for N rounds. The buyer's value $v \in [0,1]$ is unknown to the seller. We assume that the seller still uses exactly the same algorithm as we described in class (you may need to review lecture 1 slides if necessary). Naturally, knowing that the seller is learning his value, the buyer will be strategic about his response at each round. For example, when offered a price p_n at round n, the buyer may intentionally respond with "Reject" even though $v > p_n$ because this will trick the seller to offer lower prices in next rounds. On the other hand, a "Reject" response also leads to buyer utility 0 whereas an "Accept" could have given him a utility of $v - p_n (> 0)$ at round n. Therefore, the strategic buyer who looks to *maximize his total utility* would need to balance between using "Reject" to induce lower prices and using "Accept" to collect positive utilities.

More formally, denote the seller's price at round n by $p_n \in [0,1]$ and the buyer's response by $s_n \in \{0,1\}$ for $n = 1, \cdots, N$, where $s_n = 1$ means the buyer responds with "Accept" at round n and $s_n = 0$ means a buyer response of "Reject". The total utility of the buyer is thus $\sum_{n=1}^{N} s_n(v - p_n)$. Assume that the seller is committed to run the algorithm as described in class (the one achieving $(2\log\log N + 1)$ regret for N repeated buyers), what is the optimal buyer response strategy $s = (s_1, \cdots, s_N)$? Please describe a $o(N)$ time algorithm to compute the optimal buyer response strategy (i.e., your algorithm should be faster than being linear in N), and show that your algorithm correctly computes the optimal buyer strategy.

What if the seller runs the standard binary search algorithm? How to compute the optimal buyer response strategy? Will the buyer gain less or more utility in this case? Prove your answers.

Hint: think about the following question — if the buyer will reject the offer for k rounds for some $k \leq N$, which k of the N rounds should he choose to reject the offer so that it maximizes his utility?

Solution. We assume $v \geq 1/N$ and N is large enough in this problem

Setting 1: Optimal Seller Algorithm.

We first argue that if the the buyer will reject the offer for k rounds, the optimal buyer strategy is to reject exactly in the first k rounds. Prove by contradiction. Assume, for the purpose of contradiction, that the buyer did not reject all the first k rounds. Then let t_0 be the first round during which the buyer accepts and let t_1 be the last round during which the buyer rejects. Under our assumption, we have $t_1 > k > t_0$.

We claim that if the buyer switches his decision between round t_0 and t_1 — i.e., reject at t_0 and accept at t_1 instead — his utility will strictly increase. This is because if he rejects at t_0, all his prices after round t_0 will be non-increasing and will be strictly decreasing during round t_0 to t_1 during which he accepts at least once (i.e., at t_1). Therefore, his utility strictly increases, contradicting the optimality assumption. Therefore, the optimal buyer strategy must reject in the offer consecutively during the first k rounds.

To figure out the optimal buyer strategy, we only need to compute the optimal number of rounds, denoted as k^*, to reject offers. Note that after $\overline{k} = \lceil \log_2 \log_2 N \rceil$ rounds of rejections, the price will become a constant $2^{-2^{\overline{k}}}$ which is s. This is because after k rounds of rejections, the price is 2^{-2^k}. So when $2^{-2^k} \leq 1/N$ — equivalently, $2^{2^k} \geq N$ or $k \geq \lceil \log_2 \log_2 N \rceil$ — the price will become $1/N$.

We claim k^* must equal $\overline{k}$. Clearly, any $k > \overline{k}$ is *sub-optimal* because when $k > \overline{k}$, the price will remain the same as $1/N$. So more rejections will not decrease the price any more. Since $v > 1/N$, so accepting the offer is always a better strategy for the buyer after $\overline{k}$ rejections.

We now argue any $k \leq \overline{k} - 1$ is also *sub-optimal*. Note that we have $2^{-2^{\overline{k}-1}} > 1/N$ by our choice of $\overline{k}$. The intuitive reason is that if the buyer rejects k times with $k < \overline{k}$, the seller price will be greater than $1/N$ and thus each round the seller's algorithm will increase the price by $\Delta = 2^{-2^k} > 1/N$. This at the end may even make the price higher than 1, but by definition of the buyer strategy, the buyer still has to accept the offer. This weird situation is because k is *too small* and thus cannot be optimal. More concretely, we have

$$
\begin{aligned}
u(k) &= \sum_{i=1}^{N-k} (v - i \cdot 2^{-(2^k)}) \\
&= v(N-k) - \frac{(N-k)(N-k+1)}{2} 2^{-2^k} \\
&\leq v(N-k) - \frac{(N-k)(N-k+1)}{2} \cdot \frac{1}{N} \\
&\leq vN - vk - \frac{N}{2} + (2k-1) + \frac{(k+1)^2}{2N}
\end{aligned}
$$

whereas

$$
u(\overline{k}) = \sum_{i=1}^{N-\overline{k}} (v - 1/N) = (v - 1/N)(N - \overline{k}) = vN - 1 - \overline{k}v + \overline{k}/N
$$

It is easy to see that for large N, we always have $u(k) < u(\overline{k})$ due to the $N/2$ term in $u(k)$ whereas $u(\overline{k})$ only has a deduction term $\overline{k}v = O(\log \log N)$.

As a result, the optimal buyer strategy is to reject $\overline{k}$ rounds to make price become less than $1/N$ and then always accept.

Setting 2: Binary Search Algorithm.

For binary search, similar argument shows that the buyer should still reject during the first k rounds for some carefully chosen k. Binary search updates the price slightly differently, and there is not a simple characterization of the optimal k^*. However, we know that $k^* \leq \overline{k} = \lceil \log_2 N \rceil - 1$, i.e., the round at which the price just decreases below $1/N$ after $\overline{k}$ consecutive rejections. This is because the buyer should not reject more than $\overline{k}$ rounds as the price will not decrease after $\overline{k}$ rejections.

For any $k \leq \overline{k}$, the price at round $t = k+1, \cdots, \overline{k}$ is $2^{-k} - 2^{-t}$ and the price at round $t = \overline{k}+1, \cdots, N$ is always $2^{-k} - 2^{-\overline{k}-1}$. We can thus exactly calculate the $u(k)$ function and then compare them to pick the best k. This takes $O(\log N)$ time. ■