Strategizing Against Biased and Probabilistic Adversaries

VED SRIRAMAN
ARNAV AGRAWAL

December 14, 2024

1 MOTIVATION

Behavioral economics has shown that people are not perfectly rational. In particular, humans often fall prey to different biases. Examples include the *gambler's fallacy*, where a player believes that an outcome is "overdue" if it has not been observed recently, or *loss aversion*, where a player is more sensitive to losses than equivalent gains.

Recent work by Blum et al. made several key contributions to understanding how to exploit biased opponents in 2-player, zero-sum, symmetric games [2]. Working with games where payoffs are in $\{-1, 0, 1\}$ and each action is beaten by and beats another distinct action (which they term "permissive games"), they showed that: (1) one can learn to play best responses against deterministically biased opponents without any knowledge of the game matrix, (2) an agent with no information about the game matrix can win nearly every round, and (3) it's possible to infer the specific bias an opponent is playing with through their actions (using a halving algorithm).

A natural extension of this work is to consider opponents who have biased beliefs about how we'll play (i.e., a biased probability distribution) but are otherwise rational. We tackle this setting in two different ways:

1. We consider a stochastic opponent who plays the best response to action x with probability proportional to how likely they think we are to play x
2. We remove the restriction that payoffs must be in $\{-1, 0, 1\}$, allowing for more general game matrices where the opponent may want to hedge against multiple possible actions

To illustrate how non-binary payoffs lead to richer strategic choices, consider the following example in rock-paper-scissors:

Example 1.1. *Consider a modified rock-paper-scissors game where the opponent's payoff matrix is:*

	Rock	*Paper*	*Scissors*
Rock	0	−2	1
Paper	2	0	−1
Scissors	−1	1	0

Suppose the (biased) opponent believes we'll play rock with a $5/8$ *probability and scissors with a* $3/8$ *probability. Then, in a traditional Rock-Paper-Scissors game, they would choose to play Rock (getting an expected payoff of* $3/8$*). However, with the payoffs above, the opponent would instead choose to play Paper (getting an expected payoff of* $7/8$*). So, the best response to a distribution in such as game would differ from the best response to the most likely action in the distribution.*

2 PROBLEM SETUP

We study repeated two-player, zero-sum symmetric games where a player can observe their opponent's actions but does not have access to the payoffs or prior knowledge of the game matrix. This framework serves as a complement to the bandit setting, where the payoffs are observable but the opponent's actions are hidden.

2.1 Common Framework

In both settings we study:

1. Let $A = \{1, 2, \ldots, n\}$ be a set of n actions.
2. Let $f : A^t \to \Delta(A)$ be a function that maps the player's historical actions to a distribution over actions, where:
    - $f(\emptyset)$ is the uniform distribution over A
    - $f(a, S_t)$ denotes the ath element of the distribution $f(S_t)$
    - $S_t \in A^t$ represents the player's historical actions up to time t
3. Let M be an $n \times n$ matrix satisfying for all actions $i, j \in A$:
    - Skew symmetry: $M_{ij} = -M_{ji}$
    - Zero diagonal: $M_{ii} = 0$
4. At each time step $t + 1$:
    (a) The player selects action $a_{t+1} \in A$
    (b) The opponent forms belief $f(\cdot, S_t)$ about our next action
    (c) The player's payoff is given by $M_{a_{t+1}, a_{t+1}^{\text{opp}}}$

2.2 Key Differences

The two settings differ in the following aspects:

1. **Payoff Structure:**
    - *Randomizing Opponent:* Payoffs restricted to $\{-1, 0, 1\}$
    - *Best Responding Opponent:* Payoffs bounded but not restricted to $\{-1, 0, 1\}$
2. **Opponent Behavior:**
    - *Randomizing Opponent:* Plays the best response to action x with probability $f(x, S_t)$
    - *Best Responding Opponent:* Plays to maximize expected utility against the believed distribution $f(\cdot, S_t)$
3. **Learning Objective:**
    - *Randomizing Opponent:* Learn best responses through frequency analysis
    - *Best Responding Opponent:* Learn the game matrix through boundary points

3 OPPONENT RANDOMIZES AT EVERY STEP

3.1 Assumptions for Randomizing Opponent

For this setting, we make two key assumptions about the bias function f:

1. We can construct a sequence of actions S_t such that $f(S_t)$ outputs a *peaked distribution* where one action x has probability $p^* > p$, while all other actions have equal probability p. That is, for any

action $x \in A$, we can find an S_t where:

$$f(x, S_t) = p^* \text{ and } f(a, S_t) = p \text{ for all } a \neq x$$

where $p^* > p$ and $p = \frac{1-p^*}{n-1}$ to ensure the probabilities sum to 1. Note that when payoffs are restricted to $\{-1, 0, 1\}$, the best response to such a peaked distribution is the same as the best response to the most likely action in the distribution.

2. The bias function f is either periodic or can be repeatedly manipulated to produce the above peaked distribution multiple times, allowing us to gather sufficient samples of the opponent's responses.

These assumptions ensure that we can reliably create situations where the opponent believes one action is more likely than others, enabling us to learn their best responses through repeated observations.

3.2 Analysis for Randomizing Opponent

Following from our problem setup in Section 2, we focus on the case where the opponent plays the best response to the player's action x with probability $f(x, S_t)$. That is, $a_{t+1}^{\text{opp}} = \text{BR}(x)$ with probability $f(x, S_t)$.

Definition 3.1. The *degree of bias*, denoted $b(n)$, of a function $f : A^t \longrightarrow \Delta(A)$ is the ratio of p^* to p in the peaked distribution, where p^* is the probability of the most likely action and $p = \frac{1-p^*}{n-1}$ is the probability assigned to all other actions. That is,

$$b(n) = \frac{p^*}{p} = \frac{p^*(n-1)}{1-p^*}$$

Theorem 3.2. *For any action x, if we play according to a sequence S_t such that $f(x, S_t) = p^*$ and $f(a, S_t) = p$ for all $a \neq x$, then as soon as the difference D between the number of occurrences of the most frequent and second most frequent actions in the opponent's play history reaches*

$$\frac{\log(n-2) - \log\left(\frac{\epsilon}{1-\epsilon}\right)}{\log\left(b(n)\right)}$$

we can conclude with probability at least $1-\epsilon$ that the most frequently played action by the opponent is the best response to x.

Proof. Let $\mathcal{D}_t = \{a_\tau^{\text{opp}} : f(x, S_\tau) = p^* \text{ and } f(a, S_\tau) = p \text{ for all } a \neq x\}$ be the subset of observed opponent actions where we played according to sequences that induced peaked distributions. Let H_i be the hypothesis that action i is the best response to Rock. Let i^* be the action we have observed the most in $\mathcal{D}_t$. Then, we want to show that $\mathbb{P}(H_{i^*} \mid \mathcal{D}_t) \geq 1-\epsilon$ when $D \geq \frac{\log(n-2)-\log\left(\frac{\epsilon}{1-\epsilon}\right)}{\log(b(n))}$. Note that by Bayes' rule and the law of total probability,

$$\mathbb{P}(H_{i^*} \mid \mathcal{D}_t) = \frac{\mathbb{P}(\mathcal{D}_t \mid H_{i^*}) \cdot \mathbb{P}(H_{i^*})}{\mathbb{P}(\mathcal{D}_t)} = \frac{\mathbb{P}(\mathcal{D}_t \mid H_{i^*}) \cdot \mathbb{P}(H_{i^*})}{\sum_{j \in A} \mathbb{P}(\mathcal{D}_t \mid H_j) \cdot \mathbb{P}(H_j)}$$

Since we have no prior knowledge, $\mathbb{P}(H_i)$ is the same for all i except for action x, since we know that an action cannot be the best response to itself. So, $\mathbb{P}(H_x) = 0$.Therefore,

$$\begin{aligned}\mathbb{P}(H_{i^*} \mid \mathcal{D}_t) &= \frac{\mathbb{P}(\mathcal{D}_t \mid H_{i^*})}{\sum_{j \in A \setminus \{x\}} \mathbb{P}(\mathcal{D}_t \mid H_j)} \\ &= \frac{\mathbb{P}(\mathcal{D}_t \mid H_{i^*})}{\mathbb{P}(\mathcal{D}_t \mid H_{i^*}) + \sum_{j \in A \setminus \{i^*, x\}} \mathbb{P}(\mathcal{D}_t \mid H_j)} \\ &= \frac{1}{1 + \sum_{j \in A \setminus \{i^*, x\}} \frac{\mathbb{P}(\mathcal{D}_t \mid H_j)}{\mathbb{P}(\mathcal{D}_t \mid H_{i^*})}}\end{aligned}$$

Now, note that for any action i, $\mathbb{P}(\mathcal{D}_t \mid H_i) = \sum_{p^i \in P^i} \prod_{\tau=1}^{t} p^i[a_\tau^{\text{opp}}]$, where P^i is the set of distributions that the opponent can play with under the hypothesis H_i. Since we know that our dataset contains best responses to the peaked distribution at action x, we know (by our problem setup) that $p^i[i] = p^*$ and $p^i[j] = p$ for all $j \neq i$ for all $p^i \in P^i$. [1]

Let k_i be the number of times action i is observed in $\mathcal{D}_t$. Then, we have $\mathbb{P}(\mathcal{D}_t \mid H_i) = \prod_{\tau=1}^{t} p^i[a_\tau^{\text{opp}}] = (p^*)^{k_i}(p)^{t-k_i}$. Let action j^* be the second most frequently observed action in $\mathcal{D}_t$. Then, we have $\mathbb{P}(\mathcal{D}_t \mid H_{j^*}) \geq \mathbb{P}(\mathcal{D}_t \mid H_j)$ for all $j \in A \setminus \{i^*, x\}$. As a result, $\frac{\mathbb{P}(\mathcal{D}_t|H_{j^*})}{\mathbb{P}(\mathcal{D}_t|H_{i^*})} \geq \frac{\mathbb{P}(\mathcal{D}_t|H_j)}{\mathbb{P}(\mathcal{D}_t|H_{i^*})}$ for all $j \in A \setminus \{i^*, x\}$. There are $n-2$ such actions, summing up this inequality over all $j \in A \setminus \{i^*, x\}$ gives us

$$\sum_{j \in A \setminus \{i^*, x\}} \frac{\mathbb{P}(\mathcal{D}_t \mid H_j)}{\mathbb{P}(\mathcal{D}_t \mid H_{i^*})} \leq (n-2) \cdot \frac{\mathbb{P}(\mathcal{D}_t \mid H_{j^*})}{\mathbb{P}(\mathcal{D}_t \mid H_{i^*})} = (n-2) \cdot \frac{(p^*)^{k_{j^*}}(p)^{t-k_{j^*}}}{(p^*)^{k_{i^*}}(p)^{t-k_{i^*}}}$$

$$= (n-2) \cdot \left(\frac{p^*}{p}\right)^{k_{j^*}-k_{i^*}} = (n-2) \cdot b(n)^{-D}$$

Adding 1 to both sides maintains the inequality, and then taking the reciprocal of both sides inverts the inequality, giving us

$$\mathbb{P}(H_{i^*} \mid \mathcal{D}_t) = \frac{1}{1 + \sum_{j \in A \setminus \{i^*, x\}} \frac{\mathbb{P}(\mathcal{D}_t|H_j)}{\mathbb{P}(\mathcal{D}_t|H_{i^*})}} \geq \frac{1}{1 + (n-2) \cdot b(n)^{-D}}$$

where D is the difference between the number of occurrences of the most frequent and second most frequent actions in $\mathcal{D}_t$.

Note that $b(n) \geq 1$, since $p^* \geq p$. So, as D increases, the denominator of the above fraction decreases, and we can be more confident in the hypothesis. Setting $D \geq \frac{\log(n-2) - \log\left(\frac{\epsilon}{1-\epsilon}\right)}{\log(b(n))}$, we get:

$$\begin{aligned}
& D\log(b(n)) \geq \log(n-2) - \log\left(\frac{\epsilon}{1-\epsilon}\right) \\
\implies & -D\log(b(n)) \leq \log\left(\frac{\epsilon}{1-\epsilon}\right) - \log(n-2) \\
\implies & b(n)^{-D} \leq \frac{\epsilon}{(1-\epsilon)\cdot(n-2)} = \frac{1}{(1-\epsilon)\cdot(n-2)} - \frac{1}{n-2} \\
\implies & (n-2)\cdot b(n)^{-D} + 1 \leq \frac{1}{1-\epsilon} \\
\implies & \mathbb{P}(H_{i^*} \mid \mathcal{D}_t) \geq \frac{1}{(n-2)\cdot b(n)^{-D} + 1} \geq 1 - \epsilon
\end{aligned}$$

So, we have shown that $\mathbb{P}(H_{i^*} \mid \mathcal{D}_t) \geq 1 - \epsilon$ when $D \geq \frac{\log(n-2) - \log\left(\frac{\epsilon}{1-\epsilon}\right)}{\log(b(n))}$. □

This result is valid for any biased opponent that plays according to our setup and has f satisfying our assumptions. We can find $b(n)$ for any given f by solving for p^* and achieve different bounds on how fast we can learn the opponent's best response to a given action. The following table summarizes how quickly we can learn the opponent's best response to a given action for different classes of $b(n)$:

[1] If we didn't use the peaked distribution, we would have to consider a distribution corresponding to each possible derangement of the set of actions such that action i was the best response to x. This is because H_i only asserts that i is the best response to x and so must be the most observed action. It doesn't assert anything about the ordering of the rest of the actions. Using the peaked distribution significantly simplifies our analysis here at the cost of some generality.

Class of $b(n)$	**Description**	**Required Differences**
Constant	$b(n) > 1$	$O(\log(n))$
Exponential	$b(n) = c^n$, $c > 1$	$O(\log(n)/n)$
Polynomial	$b(n) = n^k$, $k > 0$	$O(1)$
Approaching 1	$b(n) = \frac{n^k}{n^k - c}$, $k, c > 0$	$O\left(n^k \log(n)\right)$

Where we obtain the last bound using a Taylor expansion and first order approximation.[2] Consider a version of Gambler's Fallacy, where the opponent thinks we'll play an action x with probability proportional to $1 - \text{freq}(x)$. That is, $f(x, S_t) = \frac{1-\text{freq}(x)}{\sum_{a \in A}(1-\text{freq}(a))} = \frac{1-\text{freq}(x)}{n-1}$. This yields $b(n) = \frac{n-1}{n-2}$, which is approaching 1 as n increases. The required differences to learn the best response to any given action are $O(n \log(n))$.

This process significantly simplifies the problem of finding bounds on how quickly we can learn the opponent's best response to a given action. It matches our intuition that we can learn faster against more strongly biased opponents.

4 OPPONENT BEST RESPONDS

4.1 Learning the Payoff Matrix

Unlike the randomizing opponent setting where we learn best responses through frequency analysis, here we attempt to learn the game matrix M directly. Our key insight is that if we can find boundary points where two actions are equally optimal for the opponent, we can construct a linear system that recovers M up to error δ_{final}.

We leverage that the opponent:

- Forms belief $f(\cdot, S_t)$ about our next action based on our history S_t
- Plays a best response to maximize expected utility against this belief

By carefully controlling our action frequencies, we can steer the opponent's beliefs to boundary points where two actions become equally optimal responses. These boundary points provide linear constraints on M that allow us to reconstruct it. Once we have an accurate estimate of M, we can win the game on every round with low error. We provide a detailed motivating example in Section 6.1 of the Appendix.

4.2 Boundary Points

We first show that boundary points exist for every pair of actions, and then present our strategy for finding them. We can define a boundary point as follows:

Definition 4.1 (Boundary Point)**.** For actions $i, j \in \mathcal{A}$, a boundary point $\mathbf{p}^{ij}$ is a probability distribution where:

$$(M\mathbf{p}^{ij})_i = (M\mathbf{p}^{ij})_j \geq (M\mathbf{p}^{ij})_k \text{ for all } k$$

That is, actions i and j are both optimal best responses to $\mathbf{p}^{ij}$.

[2]For $b(n) = \frac{n^k}{n^k - c}$ where $k, c > 0$, we can write $b(n) = \frac{1}{1-\frac{c}{n^k}}$. Using the Taylor expansion of $\log(1-x)$ with $x = \frac{c}{n^k}$, we get $\log(b(n)) = \frac{c}{n^k} + O(\frac{1}{n^{2k}})$. This yields a required difference of $D = O(n^k \log(n))$, which is significantly larger than the other cases.

4.2.1 Existence of Boundary Points

Theorem 4.2. *For any pair of indices (i, j) with $i < j$, there exists a probability vector $\mathbf{p}^{ij} \in \Delta^{n-1}$ such that $(M\mathbf{p}^{ij})_i = (M\mathbf{p}^{ij})_j$.*

4.2.2 Finding Boundary Points

To find boundary points, we must steer the opponent's empirical distribution through our choice of actions. We can do this by playing actions proportionally to the target distribution. That is, to find each $\mathbf{p}^{ij}$ where $(M\mathbf{p}^{ij})_i = (M\mathbf{p}^{ij})_j$, we:

1. Use binary search to identify target distributions
2. For each target, play actions proportionally to steer the opponent's empirical distribution
3. Observe which action becomes dominant

4.2.3 Algorithms

The binary search algorithm (SteeringBinarySearch) and the steering subroutine (SteerOpponentDistribution) are presented in full detail in Section 6.2 of the Appendix. The binary search algorithm is used to find boundary distributions that make two actions equally optimal. The steering subroutine is used to steer the opponent's empirical distribution to a target distribution in each iteration of the binary search.

4.3 Analysis

Let's distinguish between different error parameters:

- δ_{steer}: accuracy of steering (how close $f(\cdot, S_T)$ is to $\mathbf{p}_{\text{target}}$)
- δ_{binary}: precision of binary search
- δ_{final}: final error between estimated and true $\mathbf{p}^{ij}$

Lemma 4.3 (Steering Accuracy Per Iteration)**.** *Let t be the current round before steering begins. The SteerOpponentDistribution algorithm first requires $(n-1)t$ rounds to reset to uniform distribution, then plays for an additional $N = O(\frac{1}{\delta_{steer}^2}\log(\frac{n}{\delta_{steer}}))$ rounds to achieve $\|f(\cdot, S_{t+N+(n-1)t}) - \mathbf{p}_{target}\| \leq \delta_{steer}$ with probability at least $1 - \delta_{steer}$.*

Lemma 4.4 (Binary Search with Steering Error)**.** *If SteerOpponentDistribution achieves steering error δ_{steer}, then after $O(\log(1/\delta_{binary}))$ iterations, SteeringBinarySearch finds a point $\hat{\mathbf{p}}^{ij}$ such that:*

$$\|\hat{\mathbf{p}}^{ij} - \mathbf{p}^{ij}\| \leq O(\delta_{binary} + \delta_{steer})$$

where p^{ij} is a true boundary point satisfying $(Mp^{ij})_i = (Mp^{ij})_j$.

4.4 Matrix Recovery

To recover M, we:

1. Find approximate boundary points $\hat{\mathbf{p}}^{ij}$ for all $i < j$
2. Each $\hat{\mathbf{p}}^{ij}$ gives us an equation $(M\hat{\mathbf{p}}^{ij})_i = (M\hat{\mathbf{p}}^{ij})_j$
3. These equations form a linear system $A\mathbf{x} = \mathbf{0}$
4. Solving this system recovers the entries of M

Theorem 4.5 (Matrix Recovery with Steering). *Given estimates $\hat{\mathbf{p}}^{ij}$ for all pairs (i, j) with $i < j$, where each $\|\hat{\mathbf{p}}^{ij} - \mathbf{p}^{ij}\| \leq O(\delta_{binary} + \delta_{steer})$, we can recover an estimate $\hat{M}$ of the game matrix M such that:*

$$\|\hat{M} - M\| \leq O(\kappa(\delta_{binary} + \delta_{steer}))$$

where κ is the condition number of the resulting linear system.

4.5 Complete Algorithm and Complexity

Combining all components, we present our complete learning algorithm and analyze its total complexity.

Theorem 4.6 (Complete Learning Algorithm). *There exists an algorithm that learns a zero-sum symmetric game matrix M up to error δ_{final} using $O(n^{n^2})$ total plays. This exponential bound arises from the need to reset the opponent's distribution before each steering attempt.*

4.6 Improved Complexity with Recency Bias

We can significantly improve our complexity bounds by considering an opponent who exhibits recency bias - only considering the last W actions when forming their beliefs.

Definition 4.7 (Recency-Biased Opponent). At time t, the opponent's belief $f(x, S_t)$ is proportional to the frequency of action x in the last W actions of S_t, where W is a fixed constant:

$$f(x, S_t) = \frac{\text{count}(x \text{ in last } W \text{ actions of } S_t)}{W}$$

This modification allows us to reset the opponent's distribution by playing a desired sequence of length W, so that the complexity of each steering attempt becomes independent of t_k.

Lemma 4.8 (Improved Steering with Recency Bias). *For any target distribution $\mathbf{p}_{target}$ and error parameter δ_{steer}, with a recency-biased opponent using window size $W \geq \frac{1}{\delta_{steer}}$, SteerOpponentDistribution requires $O(W + \frac{1}{\delta_{steer}^2}\log(\frac{n}{\delta_{steer}}))$ rounds to achieve $\|f(\cdot, S_t) - \mathbf{p}_{target}\| \leq \delta_{steer}$ with probability at least $1 - \delta_{steer}$.*

Theorem 4.9 (Improved Complete Learning Algorithm). *With a recency-biased opponent using window size $W = \Theta(\frac{\kappa}{\delta_{final}})$, there exists an algorithm that learns a zero-sum symmetric game matrix M up to error δ_{final} using*

$$O\left(n^3 \cdot \log\left(\frac{\kappa}{\delta_{final}}\right) \cdot \left(\frac{\kappa}{\delta_{final}} + \frac{\kappa^2}{\delta_{final}^2}\log\frac{n\kappa}{\delta_{final}}\right)\right)$$

total plays, which is polynomial in all parameters.

This improved bound shows that learning is tractable when facing a more realistic opponent who exhibits recency bias. The polynomial complexity makes this approach practical for moderately-sized games, especially when the window size W is reasonable.

5 DISCUSSION

Our work presents two frameworks for learning against probabilistic opponents in repeated zero-sum games: one for opponents who play according to predicted likelihoods, and another for opponents who maximize expected utility. While these frameworks provide theoretical foundations for understanding probabilistic opponent behavior, several important directions remain for future work.

5.1 Extensions to Non-Stationary Behavior

A natural extension would be to consider opponents exhibiting non-stationary behavior. Factors such as learning, adaptation, and context-specific considerations lead to dynamic decision-making processes. For instance, in iterated games like the Prisoner's Dilemma, players may start with a particular strategy but adjust their choices based on prior outcomes and the perceived behavior of others. This adaptability reflects the absence of rigid strategic patterns. This motivates several interesting directions:

- Opponents who play different biased strategies according to a time-varying distribution
- Decay models where opponents become increasingly resistant to manipulation over time
- Mixed models combining multiple types of behavioral biases with varying degrees of influence
- Extending the frameworks to handle partial observability or noisy feedback

These extensions would better reflect real-world scenarios where opponents adapt and learn from interactions, making them progressively harder to exploit.

5.2 Limitations and Potential Improvements

Our current frameworks have several limitations that suggest areas for improvement:

1. In our first framework, we could generalize the bias measure to use arbitrary probability vectors $\mathbf{p} = [p_1, \ldots, p_n]$, with the ratio between the highest probability p^* and second-highest probability serving as the degree of bias. This would provide a more flexible framework for modeling different types of behavior, and would allow us to steer the opponent's behavior more effectively.
2. For our second framework involving utility-maximizing opponents, we found that certain behavioral biases (like gambler's fallacy) were difficult to incorporate through the $f(x, S_t)$ function. This was primarily because such biases are not easily "steerable" in the way our algorithm requires. Developing alternative approaches that can accommodate a broader range of behavioral biases remains an open challenge.

5.3 Minimum actions to steer frequency-based biases can be formulated as an integer program

When looking at opponents whose bias was dependent on the frequency of our actions, we found that finding the minimum number of actions needed to reach a target distribution could be formulated as an integer programming problem. For instance, suppose we are starting with an empty action history and want to reach some distribution p^{target}. Consider $f(S_t)$ such that $f(x, S_t) \propto n_x(S_t)$, where $n_x(S_t)$ is the number of times action x was played in the history S_t. Suppose we want each value in the distribution to be within some δ of p^{target}. Setting each n_x to be a (integer) decision variable, we obtain the following integer (linear) program:

$$\begin{aligned} & \min \sum_{k=1}^{n} n_k \\ \text{s.t.} \quad & n_x - (p_x^{\text{target}} + \delta)\left(\sum_{k=1}^{n} n_k\right) \leq 0, \quad \forall x \\ & (p_x^{\text{target}} - \delta)\left(\sum_{k=1}^{n} n_k\right) - n_x \leq 0, \quad \forall x \end{aligned}$$

Techniques like branch and bound can be used to solve this problem, but the worst-case runtime would be exponential in the number of actions. This indicates that optimal steering of the bias function, especially when it is dependent on the entire action history is a particularly difficult problem.

REFERENCES

[1] D. Bindel. Perturbation theory. https://www.cs.cornell.edu/~bindel/class/cs6210-f12/notes/lec10.pdf, 2012. Lecture notes for CS 6210: Matrix Computations, Cornell University, Fall 2012.

[2] A. Blum and M. Dutz. Winning without observing payoffs: Exploiting behavioral biases to win nearly every round. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024.

6 APPENDIX

6.1 Motivating Examples from Section 4.2

To learn this matrix against a probabilistic opponent maximizing expected utility, we can use the concept of "boundary points" in the probability simplex:

Example 6.1. *In rock-paper-scissors, consider the state of the game at time t. Let $\mathbf{p} = (p_1, p_2, p_3)$ represent the opponent's belief, which is his probability distribution over our actions (with $p_1 + p_2 + p_3 = 1$ and $p_i \geq 0$ for all i). The opponent's expected utility is given by the matrix-vector product $M\mathbf{p} = \mathbf{u}$:*

$$\underbrace{\begin{pmatrix} 0 & M_{12} & M_{13} \\ M_{12} & 0 & M_{23} \\ -M_{13} & M_{23} & 0 \end{pmatrix}}_{M} \cdot \underbrace{\begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix}}_{\mathbf{p}} = \underbrace{\begin{pmatrix} M_{12}p_2 + M_{13}p_3 \\ -M_{12}p_1 + M_{23}p_3 \\ -M_{13}p_1 + -M_{23}p_2 \end{pmatrix}}_{\mathbf{u}}$$

For any pair of actions $i, j \in \mathcal{A}$, we can find probability vectors $\mathbf{p}^{ij}$ where $\mathbf{u}[i] = \mathbf{u}[j]$, indicating points where the opponent is indifferent between actions i and j. This gives us three conditions:

- *For $(1,2)$: Find $\mathbf{p}^{12}$ such that $\mathbf{u}[1] = \mathbf{u}[2] \implies M_{12}p_2 + M_{13}p_3 = -M_{12}p_1 + M_{23}p_3$*
- *For $(1,3)$: Find $\mathbf{p}^{13}$ such that $\mathbf{u}[1] = \mathbf{u}[3] \implies M_{12}p_2 + M_{13}p_3 = -M_{13}p_1 - M_{23}p_2$*
- *For $(2,3)$: Find $\mathbf{p}^{23}$ such that $\mathbf{u}[2] = \mathbf{u}[3] \implies -M_{12}p_1 + M_{23}p_3 = -M_{13}p_1 - M_{23}p_2$*

Each of these conditions is a linear equation in terms of M_{12}, M_{13}, M_{23} once $\mathbf{p}^{ij}$ is known. By choosing appropriate $\mathbf{p}^{ij}$, one can solve these equations to express p_1, p_2, p_3 (the components of these boundary vectors) in terms of M_{12}, M_{13}, M_{23}. In turn, given estimates $\hat{\mathbf{p}}^{ij}$, one can solve the linear system to estimate $\hat{M}_{12}, \hat{M}_{13}, \hat{M}_{23}$.

The key takeaway is that if we can determine such boundary probability distributions $\mathbf{p}^{ij}$ for each pair of actions, we get a system of linear equations that can be solved for the unknown payoff parameters. Note that, in estimating these boundary points, we might not know $\mathbf{p}^{ij}$ exactly; we only have $\hat{\mathbf{p}}^{ij}$ within some error δ. Because our final equations are linear, if $\mathbf{p}^{ij}$ changes by a small amount δ, then the estimated $\hat{M}$ also changes proportionally to δ. This linear dependency ensures that small errors in estimating these boundary points only lead to small (proportionate) errors in the final M. We prove this in the next section. Thus, we have learned the game matrix M in this case, and can use it to win the game on every round with low error.

We can extend this type of argument can be extended to an n-action game, with payoff matrix M being an $n \times n$ skew symmetric matrix, which we do in Section 3.2.

6.2 Algorithms from Section 4.2

Algorithm 1 SteeringBinarySearch(i, j, δ) - finds boundary points

```
1:  Initialize starting distributions:
2:  $\mathbf{p}_a \leftarrow e_{i'}$                                   ▷ should make $i$ dominant
3:  $\mathbf{p}_b \leftarrow e_{j'}$                                   ▷ should make $j$ dominant
4:  $\lambda_{\text{low}} \leftarrow 0$, $\lambda_{\text{high}} \leftarrow 1$
5:  while $\lambda_{\text{high}} - \lambda_{\text{low}} > \delta$ do
6:      $\lambda_{\text{mid}} \leftarrow (\lambda_{\text{low}} + \lambda_{\text{high}})/2$
7:      $\mathbf{p}_{\text{target}} \leftarrow (1 - \lambda_{\text{mid}})\mathbf{p}_a + \lambda_{\text{mid}}\mathbf{p}_b$
8:      SteerOpponentDistribution($\mathbf{p}_{\text{target}}$, $\delta_{\text{steer}}$, $t$)
9:      if opponent plays $i$ then
10:         $\lambda_{\text{low}} \leftarrow \lambda_{\text{mid}}$
11:     else
12:         $\lambda_{\text{high}} \leftarrow \lambda_{\text{mid}}$
13:     end if
14: end while
15: return $\mathbf{p}(\lambda_{\text{mid}})$
```

Algorithm 2 SteerOpponentDistribution(p_{target}, δ_{steer}, t) - steers the opponent's distribution

```
1:  Let $f_{\text{curr}} = f(\cdot, S_t)$ be current frequencies
2:  $f_{\max} = \max_{x \in \mathcal{A}} f_{\text{curr}}(x)$
3:  for $x \in \mathcal{A}$ where $f_{\text{curr}}(x) < f_{\max}$ do          ▷ reset to uniform distribution
4:      Play action $x$ repeatedly for $(f_{\max} - f_{\text{curr}}(x))t$ rounds
5:  end for
6:  $N \leftarrow \left\lceil \frac{1}{\delta_{\text{steer}}^2} \log(\frac{2n}{\delta_{\text{steer}}}) \right\rceil$          ▷ additional rounds needed to steer to target
7:  for $i = 1$ to $N$ do
8:      Play action $x$ with probability $\mathbf{p}_{\text{target}}(x)$
9:  end for
10: Observe opponent's action at round $t + N + (n-1)t$
11: return $f(\cdot, S_{t+N+(n-1)t})$                       ▷ distribution after reset and steering
```

6.3 Proofs from Section 4.2

6.3.1 Proof of Theorem 4.2

Proof. Consider the continuous function $f_{ij}(\mathbf{p}) = (M\mathbf{p})_i - (M\mathbf{p})_j$. We construct two points:

- Let $e_{i'}$ be the unit vector with 1 in position i'
- Define $\mathbf{p}_a = e_{i'}$ such that $i = \text{argmax}_k (M\mathbf{p}_a)_k$
- Define $\mathbf{p}_b = e_{j'}$ such that $j = \text{argmax}_k (M\mathbf{p}_b)_k$

Thus, we have $f_{ij}(\mathbf{p}_a) > 0$ and $f_{ij}(\mathbf{p}_b) < 0$. By the intermediate value theorem, there exists $\mathbf{p}^{ij}$ on the line segment between $\mathbf{p}_a$ and $\mathbf{p}_b$ where $f_{ij}(\mathbf{p}^{ij}) = 0$. □

6.3.2 Proof of Lemma 4.3

Proof. For each action x, our empirical play frequency during the N steering rounds will concentrate around $\mathbf{p}_{\text{target}}(x)$ by Hoeffding's inequality:

$$P(|\frac{\text{count}(x, \text{new plays})}{N} - \mathbf{p}_{\text{target}}(x)| > \delta_{\text{steer}}) \leq 2e^{-2N\delta_{\text{steer}}^2}$$

Taking a union bound over all n actions and setting $N = O(\frac{1}{\delta_{\text{steer}}^2}\log(\frac{n}{\delta_{\text{steer}}}))$ gives us the desired concentration with probability at least $1 - \delta_{\text{steer}}$. Since the opponent's bias function $f(\cdot, S_{t+N+(n-1)t})$ is based on these empirical frequencies, we achieve:

$$\|f(\cdot, S_{t+N+(n-1)t}) - \mathbf{p}_{\text{target}}\| \leq \delta_{\text{steer}}$$

□

6.3.3 Proof of Lemma 4.4

Proof. In each binary search iteration:

1. We aim to query with distribution $\mathbf{p}_{\text{target}}$
2. Actually achieve distribution $f(\cdot, S_T)$ where $\|f(\cdot, S_T) - \mathbf{p}_{\text{target}}\| \leq \delta_{\text{steer}}$
3. This affects the opponent's best response, potentially making them choose differently than they would with exact $\mathbf{p}_{\text{target}}$

The binary search converges to a point $\hat{\mathbf{p}}^{ij}$ where:

$$|(M\hat{\mathbf{p}}^{ij})_i - (M\hat{\mathbf{p}}^{ij})_j| \leq O(\delta_{\text{binary}} + \delta_{\text{steer}})$$

By the continuity of the map $\mathbf{p} \mapsto (M\mathbf{p})_i - (M\mathbf{p})_j$, this implies our bound on $\|\hat{\mathbf{p}}^{ij} - \mathbf{p}^{ij}\|$. □

6.3.4 Proof of Theorem 4.5

Proof. For each pair (i, j) with $i < j$, we have the condition $(M\mathbf{p}^{ij})_i = (M\mathbf{p}^{ij})_j$. Let's expand this:

$$\sum_{\ell=1}^{n} M[i, \ell]p_\ell^{ij} = \sum_{\ell=1}^{n} M[j, \ell]p_\ell^{ij}$$

Rearranging:

$$\sum_{\ell=1}^{n}(M[i, \ell] - M[j, \ell])p_\ell^{ij} = 0$$

Due to skew-symmetry, M is fully determined by its upper triangular entries. Let $\mathbf{x} \in \mathbb{R}^{n(n-1)/2}$ be the vector containing these entries, ordered as:

$$\mathbf{x} = (M[1, 2], M[1, 3], ..., M[1, n], M[2, 3], ..., M[2, n], ..., M[n - 1, n])^T$$

For each pair (i, j), we can rewrite our equation as $\mathbf{a}_{ij}^T\mathbf{x} = 0$, where $\mathbf{a}_{ij} \in \mathbb{R}^{n(n-1)/2}$ is constructed as follows:

- Each component of $\mathbf{a}_{ij}$ corresponds to an upper triangular entry $M[k, \ell]$ $(k < \ell)$

- For entry $M[k,\ell]$, the corresponding component in $\mathbf{a}_{ij}$ is:

$$\begin{cases} p^{ij}_\ell & \text{if } k = i \\ -p^{ij}_\ell & \text{if } k = j \\ -p^{ij}_k & \text{if } \ell = i \\ p^{ij}_k & \text{if } \ell = j \\ 0 & \text{otherwise} \end{cases}$$

Stacking these equations for all pairs (i,j) gives us $A\mathbf{x} = \mathbf{0}$, where each row of A is $\mathbf{a}_{ij}^T$. To make the system well-defined, we normalize by setting $M[1,2] = 1$, giving us:

$$\begin{cases} A\mathbf{x} = \mathbf{0} \\ e_1^T\mathbf{x} = 1 \end{cases}$$

When using estimated points $\hat{\mathbf{p}}^{ij}$, we get perturbed vectors $\hat{\mathbf{a}}_{ij}$. Since each component of $\mathbf{a}_{ij}$ is either 0, p^{ij}_k, or $-p^{ij}_k$:

$$\|\hat{\mathbf{a}}_{ij} - \mathbf{a}_{ij}\| \le O(\|\hat{\mathbf{p}}^{ij} - \mathbf{p}^{ij}\|) \le O(\delta_{\text{binary}} + \delta_{\text{steer}})$$

Therefore:

$$\|\hat{A} - A\| \le O(\delta_{\text{binary}} + \delta_{\text{steer}})$$

When using estimated points $\hat{\mathbf{p}}^{ij}$, we get a perturbed system:

$$\begin{cases} \hat{A}\hat{\mathbf{x}} = \mathbf{0} \\ e_1^T\hat{\mathbf{x}} = 1 \end{cases}$$

Let $\Delta A = \hat{A} - A$ denote the perturbation in our matrix (to distinguish from our algorithm's error parameters). By standard perturbation theory for linear systems [1]:

$$\frac{\|\hat{\mathbf{x}} - \mathbf{x}\|}{\|\mathbf{x}\|} \le \kappa(A)\frac{\|\Delta A\|}{\|A\|}$$

where $\kappa(A) = \|A\|\|A^{-1}\|$ is the condition number of A. Since each entry of A is either 0 or a component of some $\mathbf{p}^{ij}$ (or its negative), and each $\mathbf{p}^{ij}$ is a probability vector:

- $\|A\| \le O(1)$ (since entries are bounded by 1)
- $\|\Delta A\| \le O(\delta_{\text{binary}} + \delta_{\text{steer}})$ (since each entry of ΔA differs by at most our steering and binary search errors)

Moreover, since M has bounded entries (being a payoff matrix), $\|\mathbf{x}\|$ is also bounded by a constant. Therefore:

$$\|\hat{\mathbf{x}} - \mathbf{x}\| \le O(\kappa(A)(\delta_{\text{binary}} + \delta_{\text{steer}}))$$

Since the entries of M are exactly the components of $\mathbf{x}$ (and their negatives), we conclude:

$$\|\hat{M} - M\| \le O(\kappa(A)(\delta_{\text{binary}} + \delta_{\text{steer}}))$$

□

6.3.5 Proof of Theorem 4.6

Proof. The algorithm proceeds as follows:

1. For each pair (i, j) with $i < j$:
    - Set $\delta_{\text{binary}} = \delta_{\text{steer}} = O(\delta_{\text{final}}/\kappa)$
    - Find edge vectors $\mathbf{p}_a$ and $\mathbf{p}_b$ that make i and j dominant respectively by using steering
    - Run SteeringBinarySearch$(i, j, \delta_{\text{binary}})$ to find $\hat{\mathbf{p}}^{ij}$
    - For the k-th steering attempt (including edge vector search):
        - Need $O(n-1)t_k$ rounds to reset to uniform, where t_k is rounds played so far
        - Then $O(\frac{\kappa^2}{\delta_{\text{final}}^2}\log(\frac{n\kappa}{\delta_{\text{final}}}))$ additional rounds for steering
        - t_k grows according to the recurrence $T(n)$ analyzed above
    - Binary search needs $O(\log(\kappa/\delta_{\text{final}}))$ iterations
2. Construct and solve the linear system:
    - Form matrix $\hat{A}$ using all $\hat{\mathbf{p}}^{ij}$
    - Solve $\hat{A}\hat{\mathbf{x}} = \mathbf{0}$ with $e_1^T\hat{\mathbf{x}} = 1$
    - Construct $\hat{M}$ from $\hat{\mathbf{x}}$
3. By the Matrix Recovery theorem, $\|\hat{M} - M\| \leq \delta_{\text{final}}$

While each steering attempt requires only $O(\frac{\kappa^2}{\delta_{\text{final}}^2}\log(\frac{n\kappa}{\delta_{\text{final}}}))$ additional rounds, the need to reset the distribution before each attempt leads to exponential growth through the recurrence $T(n)$. This gives us a total complexity of $O(n^{n^2})$ plays, which is superexponential in n. □

6.3.6 Proof of Lemma 4.8

Proof. The window size requirement $W \geq \frac{1}{\delta_{\text{steer}}}$ ensures that we can represent frequencies with sufficient granularity. Consider trying to achieve a target probability p for some action. With a window of size W, the closest we can get is $\lfloor pW \rfloor/W$ or $\lceil pW \rceil/W$, with maximum error $\frac{1}{W}$. Setting $\frac{1}{W} \leq \delta_{\text{steer}}$ gives us $W \geq \frac{1}{\delta_{\text{steer}}}$. □

6.3.7 Proof of Theorem 4.9

Proof. As before, we set $\delta_{\text{steer}} = \delta_{\text{binary}} = O(\delta_{\text{final}}/\kappa)$. This means we need:

$$W \geq \frac{1}{\delta_{\text{steer}}} = O\left(\frac{\kappa}{\delta_{\text{final}}}\right)$$

For each pair (i, j) with $i < j$ (of which there are $O(n^2)$), we must:

1. Find edge vectors $\mathbf{p}_a$ and $\mathbf{p}_b$:
    - Requires trying up to $2n$ different unit vectors
    - Each attempt costs $O(W + \frac{\kappa^2}{\delta_{\text{final}}^2}\log(\frac{n\kappa}{\delta_{\text{final}}}))$ rounds
2. Perform binary search:
    - Requires $O(\log(\kappa/\delta_{\text{final}}))$ iterations
    - Each iteration costs $O(W + \frac{\kappa^2}{\delta_{\text{final}}^2}\log(\frac{n\kappa}{\delta_{\text{final}}}))$ rounds

This gives us total complexity:

$$O(n^2 \cdot (2n + \log(\kappa/\delta_{\text{final}})) \cdot (W + \frac{\kappa^2}{\delta_{\text{final}}^2}\log(\frac{n\kappa}{\delta_{\text{final}}})))$$

Substituting $W = O(\kappa/\delta_{\text{final}})$ yields our final bound. The cubic dependence on n comes from the need to find edge vectors for each pair of actions. $\square$