Exploitation and Safety in General Sum Games

Abstract

We describe a method for an agent playing a general-sum normal form game to balance the rewards of exploiting a prediction of opponent behavior with the risks of being exploited by a self-interested opponent while guaranteeing a worst-case safety margin. Our algorithm, Restricted Stackelberg Response with Safety, calculates a probability distribution over the agent’s moves that balances those concerns. The probability distribution is generated by finding a modification of the Stackelberg response in a modified game which reflects the assumption that the opponent will behave according to the prediction with some probability, but may best-respond. We prove that the tradeoff provided by a Restricted Stackelberg Response between taking advantage of a prediction and avoiding exploitation can be computed. We show experimental results using Restricted Stackelberg Response with Safety in a general-sum game.

Introduction

In a multi-agent environment an agent needs to predict its opponent’s behavior to estimate the likely outcomes of its own actions. Best responding to a prediction can leave an agent arbitrarily vulnerable, but ignoring the prediction to guard against exploitation leaves the agent unable to exploit a predictable opponent.

We introduce Restricted Stackelberg Response with Safety (RSRS), a novel method of choosing a mixed strategy given a general sum game and prediction of the opponent’s action. RSRS uses a weight parameter which represents the probability of the prediction being correct and uses it to adjust the trade-off between exploiting the prediction and avoiding being exploited.

We assume that the agent has some method of predicting the opponent’s behavior. In this paper we use a fictitious play model, which assumes the opponent is playing a fixed mixed strategy drawn from a Dirichlet prior.

Given a prediction of opponent behavior, there are three approaches an agent can take to choose a response: it can assume the prediction is correct and best-respond, it can prepare for the worst possible outcome, and it can be paranoid and assume that the opponent will predict its behavior and act accordingly. Note that the second two options are identical in zero-sum games but may be different in a general-sum game. An agent cannot simultaneously address all options, and addressing exclusively a single option can result in arbitrarily poor performance. However, as we will show, it is possible to find a mixed strategy which balances those options.

The main contribution of the paper is to propose Restricted Stackelberg Response with Safety (RSRS) as a method for choosing a mixed strategy in general sum games. We describe how to calculate RSRS, prove a property of the tradeoff it introduces, and empirically demonstrate its effectiveness in a repeated general sum game.

Related Work

A lot of work has been done to develop opponent models to predict the outcome of actions in multi-agent environments. The simplest opponent model is a stationary distribution over actions. This is the model underlying fictitious play (Fudenberg & Levine 1998), where agents assume a stationary opponent, calculate the most likely distribution for such an opponent, and best-respond to it.

For the game of poker, a Bayesian opponent model can be modified by combining it with a pre-computed equilibrium (Ganzfried & Sandholm 2011). Best responding to the new model is less vulnerable to exploitation. This approach is valuable, but not useful for every environment because it is partially dependent on hidden information in the game of poker to avoid exploitation by a best-responding opponent.

Modeling an opponent is an essential cognitive process for humans engaged in adversarial situations. Mental models (Johnson-Laird 1983) can have different levels, going from level 0 (my strategy), to level 1 (the opponent strategy), level 2 (what the opponent think I will do), and so on. A cognitive hierarchy can be used (Camerer, Ho, & Chong 2004) to explain human behavior in a wide variety of games. Machine learning approaches typically are limited to level 0 (what shall I do) and 1 (what will my opponent do). Iocaine Powder (Egnor 2000) used a recursive model to successfully play in a Rock-Paper-Scissors tournament. Recursive models have also been applied to the Lemonade Stand Game (Wunder et al. 2011). We explore the effect of using RSRS with recursive models.
AWESOME (Conitzer & Sandholm 2007) best-responds to a stationary opponent, but arrives at a Nash Equilibrium in self-play. AWESOME detects the opponent’s play, detects stationarity, exploiting it if possible, retreating to equilibrium play when it detects a non-stationary opponent.

The algorithm in (Powers, Shoham, & Vu 2007) achieves a best response against some classes of opponents, (stationary opponents and opponents who converge to stationary), achieves a pareto-optimal Nash equilibrium in self-play, and is guaranteed to achieve its security value, regardless of the opponent. The algorithm uses payoff triggers – adopting its minimax strategy when its average payoff falls below its security value. This and AWESOME could be augmented with RSRs, but at the cost of potentially losing the performance guarantees.

In (Wang et al. 2011) a prediction of opponent behavior is made based on the assumption of stationarity. When responding, the assumption made is that the opponent will best-respond, but from within a limited mixed strategy set – i.e. the opponent cannot deviate too far from prior play. This is an interesting alternative to our model in which the opponent can best respond with no limits on its strategy set, but is occasionally forced to play according to the prediction. It is not clear from the paper if the strategies discovered matched a restricted stackelberg response with safety.

Adaptive dynamic learning (Burkov & Chaib-draa 2007) can take advantage of simple learning agents by learning how they respond to sequences of play. This shows that an opponent doesn’t need to know an agent’s exact model to exploit it. An exploiting agent capable of taking advantage of a bounded memory opponent is presented in (de Cote & Jennings 2010). Instead of playing a move with the highest expected payoff, it calculates a sequence of actions with the highest expected payoff. This highlights the difficulty of detecting an exploiting opponent, as the opponent may not be playing an immediate best response.

The approaches most similar to our work are Safe Policy Selection (SPS) (?) and Restricted Nash Response (RNR) (?). Safe Policy Selection uses $\epsilon$-safe strategies in a repeated game. This allows the algorithm to exploit a prediction of opponent behavior while guaranteeing that the algorithm will achieve its safety value. Restricted Nash Response, given a prediction in a zero-sum game, constructs a modified game in which the opponent is forced to follow the prediction with probability $w$. By playing a Nash equilibrium of the modified game an agent can exploit the prediction while avoiding the risk that it may be exploited in turn. The value of $w$ determines how those issues are balanced, and they prove that any such equilibrium is equivalent to an $\epsilon$-safe strategy in a zero-sum game. Our proposed strategy RRS works for general sum games.

**Restricted Stackelberg Response with Safety**

RSRS is a method of calculating a compromise strategy for a specific game and prediction given parameters describing the confidence in the prediction and the level of risk aversion to use. RSR is calculated by creating a modified game which reflects the assumption that the opponent will follow the prediction with probability $w$. The strategy which maximizes the agents payoff in the modified game, assuming the opponent best-responds, is computed subject to the condition that the worst case outcome is not more than $\epsilon$ worse than the outcome the agent can guarantee for itself.

The maximum loss parameter ($\epsilon$) dominates the outcome of the algorithm. If $\epsilon = 0$ the algorithm will produce the minimax strategy for the game, while a value of $\epsilon$ equal to the minimax value of the game minus the minimum outcome for the agent will have no effect on the agents strategy. If $\epsilon$ is set high enough, a value of $w = 1$ will produce a best response to the prediction, while $w = 0$ will produce a Stackelberg leader strategy for the game. Intermediate values will produce a compromise between these results.

To illustrate the strategies computed by RSRS we will show a set of example strategies for a general sum version of Rock/Paper/Scissors in which players are bribed to play Rock and punished for playing Paper 1.

<table>
<thead>
<tr>
<th></th>
<th>Rock</th>
<th>Paper</th>
<th>Scissors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rock</td>
<td>1/3</td>
<td>1/3</td>
<td>-2/3</td>
</tr>
<tr>
<td>Paper</td>
<td>-1/3</td>
<td>-1/3</td>
<td>-4/3</td>
</tr>
<tr>
<td>Scissors</td>
<td>1/3</td>
<td>1/3</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: A general sum version of Rock/Paper/Scissors in which the winning player receives a payoff of 1 from the losing player, any player playing Rock receives a bonus of 1/3, and any player playing Paper receives a penalty of 1/3.

Table 2 shows how various strategies computed by RSRS perform in the general sum Rock/Paper/Scissors game, where the opponent’s move is predicted to be Rock. The first three strategies show the three main approaches we consider, while the next three show some available compromises. These strategies form an incomplete Pareto frontier – no option is strictly better than any other option. Compromises like this will always be possible unless the game has a dominants strategy, in which case no compromise is necessary.

**How to compute RSRS**

In this section we formally state the set of definitions and parameters used to calculate RSRS. We construct a Stackelberg leader game which reflects the parameters, and calculate the optimal action using a modification of the technique outlined in (Conitzer & Sandholm 2006).

We will use the following definitions:

- a normal-form game $G = (A_1 = a_1^1, \ldots, a_1^n, A_2 = a_2^1, \ldots, a_2^m, U_1 : A_1 \times A_2 \to \mathbb{R}, U_2 : A_1 \times A_2 \to \mathbb{R})$ consists of a set $(A_1, A_2)$ of actions for each player and a utility function $(U_1, U_2)$ for each player;
- a zero-sum game $G^Z = (A_1, A_2, U_1, -U_1)$ describes a game in which player 2 is attempting to minimize player 1’s payoff.
- the expected payoff to player 1 in $G^Z$, $u_1^Z$
- a prediction $\bar{s}_2 \in \Delta A_2$ is a probability distribution over actions which describes player 2’s behavior.
Table 2: Expected outcomes of various strategies calculated using RSRS in the general sum Rock/Paper/Scissors against three possible opponent strategies.

<table>
<thead>
<tr>
<th>Goal</th>
<th>w</th>
<th>e</th>
<th>Strategy</th>
<th>vs. Predicted</th>
<th>vs. Best Response</th>
<th>Worst Outcome</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best Response</td>
<td>1</td>
<td>1.5</td>
<td>(0,1,0)</td>
<td>.667</td>
<td>-1.333</td>
<td>-1.333</td>
</tr>
<tr>
<td>Stackelberg Equilibrium</td>
<td>0</td>
<td>1.5</td>
<td>(444,444,111)</td>
<td>.333</td>
<td>.333</td>
<td>-.333</td>
</tr>
<tr>
<td>Minimax</td>
<td>.5</td>
<td>0</td>
<td>(333,333,333)</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Compromise Stackelberg</td>
<td>.5</td>
<td>1.5</td>
<td>(444,555,0)</td>
<td>.518</td>
<td>-.1484</td>
<td>-.481</td>
</tr>
<tr>
<td>Compromise Minimax</td>
<td>1</td>
<td>2</td>
<td>(333,483,183)</td>
<td>.250</td>
<td>-.2</td>
<td>-.2</td>
</tr>
<tr>
<td>General Compromise</td>
<td>.5</td>
<td>2</td>
<td>(365,444,191)</td>
<td>.226</td>
<td>.226</td>
<td>-.2</td>
</tr>
</tbody>
</table>

- A weight $w \in [0, 1]$ expresses the degree of confidence attached to that prediction.
- A safety margin $e \in [0, \infty]$ expresses the maximum payoff the agent is willing to risk compared to $u_1^i$.

**Definition:** RSRS is the mixed strategy for player 1 which maximizes its expected payoff given the assumption that with probability $w$ the opponent will play according to the prediction and with probability $1 - w$ it will best-respond, subject to the constraint that its expected payoff when played against any opponent action is at least $u_1^i - e$.

To calculate the restricted Stackelberg response with safety, we construct a new payoff function $U'_1$, using $\tilde{s}_2$ and $w$, which reflects the assumptions made:

$$U'_1(a_1, a_2) = w \cdot E[U_1(a_1, \tilde{s}_2)] + (1 - w)U_1(a_1, a_2)$$

where $E[U_1(a_1, \tilde{s}_2)]$ is the expected value of $U_1(a_1, a_2)$ when $a_2$ is drawn from $\tilde{s}_2$. The restricted Stackelberg response with safety for player 1 is the probability distribution $s_1 \in \Delta A_1$ which maximizes the expected value of $U'_1$ under the assumption that player 2 will best respond to the mixed strategy chosen by player 1, subject to the constraint that $E[U'_1(s_1, a_2)] \geq u_1^i - e$ for all $a_2 \in A_2$.

To calculate the optimal strategy for player 1 we use a modification of the technique outlined in (Conitzer & Sandholm 2006). For each opponent strategy $a_2' \in A_2$ we find (if possible) a mixed strategy $s_1 \in \Delta A_1$ which maximizes the agent’s payoff in the modified game when played against $a_2'$, for which $a_2'$ is a best-response for player 2, subject to the constraint that $E[U'_1(s_1, a_2)] \geq u_1^i - e$. $s_1(a_1)$ is the probability assigned to $a_1$ by $s_1$.

For each $a_2'$ maximize over $s_1 \in \Delta A_1$:

$$s_1 = \text{argmax}_{s_1 \in \Delta A_1} \sum_{a_1 \in A_1} s_1(a_1) \cdot U'_1(a_1, a_2')$$

subject to

$$\forall a_2 \in A_2, \sum_{a_1 \in A_1} s_1(a_1) \cdot U_2(a_1, a_2') \geq \sum_{a_1 \in A_1} s_1(a_1) \cdot U_2(a_1, a_2)$$

$$\forall a_2 \in A_2, \sum_{a_1 \in A_1} s_1(a_1) \cdot U'_1(a_1, a_2) \geq u_1^i - e$$

$$\sum_{a_1 \in A_1} s_1(a_1) = 1$$

Solving a set of equations for each opponent action will give us at least 1 and up to $n$ mixed strategies for player 1. The mixed strategy $s_1$ with the highest expected value for $U'_1$ against the opponent’s best response is the Restricted Stackelberg Response with Safety.

Note that the way $e$ dominates $w$ is an effect of how we’ve chosen to define those parameters. It is instructive to consider an alternative in which $w$ dominates $e$. All that is necessary is to alter the constraint involving $e$ to:

$$\forall a_2 \in A_2, \sum_{a_1 \in A_1} s_1(a_1) \cdot U'_1(a_1, a_2) \geq u_1^i - e$$

This guarantees that the loss will be less than $e$ in the modified game instead of the original game. We’ve chosen not to calculate RSRS this way because we feel that $e$ is more useful when it does not depend on $w$.

**Stackelberg Equilibrium** Note that once we have created a modified game to reflect the effect of the $w$ parameter we use a Stackelberg equilibrium to find a strategy, unlike RNR, which calculates a Nash equilibrium. This is a necessary extension to handle the problem of multiple equilibria in general-sum games. When there are multiple equilibria, it is tempting to choose the equilibrium with the highest payoff, but this is difficult to justify. If the opponent were aware of which equilibrium was selected by the agent, it should logically also be aware of a choice of non-equilibrium strategy. In that case, it makes the most sense for the agent to maximize its payoff without requiring that it select a strategy which is part of an equilibrium.

It feels problematic to base an algorithm on maximizing payoff when that the opponent is best responding to the mixed strategy chosen by the player when moves are chosen simultaneously, because this is counter to the definition of the game.

I don’t have a simple answer to this, so I’m writing a lot of stuff which addresses it. The first way we can address it is to note that the opponent may be able to predict our agent. If we use a public algorithm, it’s reasonable for the opponent to detect that, and start best-responding. It’s tempting to insist that we will just improve our algorithm to detect an opponent which does that, but it’s important to recognize that there are limits to our own rationality - i.e. no infinite recursion.

Really this comes down to our initial attempt to predict the behavior of the opponent. As soon as we make a prediction in the form of a mixed strategy for the opponent we close off the possibility of
Finally, note that this does not completely solve the problem of multiple equilibria. The RSRS strategy is found by maximizing over the agent’s payoff. This frequently results in a strategy in which two opponent actions are best responses to the strategy chosen, with one action being preferred by the agent. This is easily handled by requiring that the preferred best response be strictly (if infinitesimally) superior to all opponent responses, but this is not very satisfactory. For example, consider the Compromise Stackelberg strategy in general-sum Rock/Paper/Scissors. For the opponent, both Paper and Scissors are best responses, giving it a payoff of $1/9$. However, the agent receives a payoff of $-4/27$ when the opponent plays Scissors, and $-13/27$ when the opponent plays paper. Even if you bias the strategy slightly, so that Scissors is .01 better for the opponent than paper, it is not a strong guarantee to know that the opponent will be costing themselves .01 when the cost to us is $1/3$.

Model Performance

Figure 1 shows the expected payoff of RSRS for our general sum version of Rock-Paper-Scissors where the opponent is predicted to play Rock against a variety of different opponents over the range of possible parameter values. Against the prediction performance is best when $w$ and $e$ are high, against a best-responding opponent performance is best when $w$ is low and $e$ is high, and worst-case performance is best when $e$ is low. In general, changing $e$ results in a continuous change in strategy (and therefore a continuous change in expected payoff) while changing $w$ results in discontinuous jumps between different strategies.

<table>
<thead>
<tr>
<th>Agent</th>
<th>Opponent</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 5 5</td>
</tr>
<tr>
<td>B</td>
<td>1 3 2 2</td>
</tr>
<tr>
<td>C</td>
<td>2 4 8 1</td>
</tr>
</tbody>
</table>

Table 3: A pure strategy equilibrium game

<table>
<thead>
<tr>
<th>Prediction</th>
<th>Weight</th>
<th>PBR</th>
<th>vs. Predicted</th>
<th>Worst Case</th>
<th>vs. Worst Case</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 1, 0)</td>
<td>3/8</td>
<td>(0, 0, 1)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1, 0, 0)</td>
<td>3</td>
<td>B</td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(1, 0, 0)</td>
<td>3/8 – 1</td>
<td>A</td>
<td>2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Effect of PBR in the pure strategy game in Table 3

The values chosen for $w$ and $e$ control the tradeoff between exploitation of the prediction and avoiding being exploited, but they differ in how they affect the strategy of the agent. For a fixed $e$ value there are many $w$ values which produce the same strategy – changes in $w$ either produce no change in strategy, or they produce a discontinuous jump to a completely new strategy. In contrast, changes to $e$ produce a continuous variation between a minimax strategy and a prediction exploiting strategy. Note that the effect of $e$ dominates the effect of $w$ – if $e$ is set to 0, the $w$ value has no effect.

We can characterize the change in performance produced by a change in $w$ for a fixed $e$ in terms of the trade-off between performance against the prediction and performance against a best-responding agent. For
example, consider the response to a prediction of Rock in the general sum version of Rock-Paper-Scissors with \( e = 4/3 \) shown in Table 5. Note that when \( w \) transitions

<table>
<thead>
<tr>
<th>( w )</th>
<th>Strategy</th>
<th>vs. Prediction</th>
<th>vs. Best Response</th>
</tr>
</thead>
<tbody>
<tr>
<td>( w &lt; 13/18 )</td>
<td>(4/9, 4/9, 1/9)</td>
<td>1/3</td>
<td>1/3</td>
</tr>
<tr>
<td>13/18</td>
<td>(4/9, 5/9, 0)</td>
<td>14/27</td>
<td>-4/27</td>
</tr>
<tr>
<td>( w &lt; 8/9 )</td>
<td>(0, 1, 0)</td>
<td>2/3</td>
<td>-4/3</td>
</tr>
</tbody>
</table>

Table 5: The effect of \( w \) in general sum Rock-Paper-Scissors with \( e = 4/3 \)

from below 13/18 to above 13/18 it gains 5/27 in expected payoff against the prediction and loses 13/27 against a best responding opponent. The expected gain against the prediction is 5/13 as much as the expected loss against a best responding opponent. This matches the relative probability of those two events expressed by a \( w \) value of 13/18. This is not a coincidence, as we shall prove below.

We are interested in values of \( w \) between two different regions in each of which RSRS for a fixed \( e \) does not change. For those \( w \) values we denote with \( rsrs_{w+} \) and \( rsrs_{w-} \) respectively the restricted Stackelberg response with safety for the region with weight values higher or lower than \( w \).

Assume we are given a game \( G = (A_1, A_2, U_1, U_2) \), a prediction \( \hat{s}_2 \in \Delta A_2 \), and a \( w \) where the RSRS changes. If there is some \( \delta \) such that for all \( 0 < \epsilon < \delta \) the RSRS with weight \( w+\epsilon \) is the same for all \( \epsilon \) and the partial best response with weight \( w-\epsilon \) is the same for all \( \epsilon \) and \( rsrs_{w+} \neq rsrs_{w-} \), we will prove two things. First, we will prove that reducing the weight value used to calculate a RSRS can only improve the expected payoff against a best-responding opponent.

**Lemma 1.**

\[ U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR) < 0 \]

**Proof.** Consider the quantities \( U_1(rsrs_{w+}, \hat{s}_2) - U_1(rsrs_{w-}, \hat{s}_2) \) and \( U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR) \), which represent the relative performance of \( rsrs_{w+} \) compared to \( rsrs_{w-} \) against the predicted distribution and a best responding opponent respectively (note that we are not concerned here with the performance against a worst-case opponent, which will be governed by the chosen \( e \) value). If both are positive or both are negative then \( rsrs_{w+} \) or \( rsrs_{w-} \) would be strictly superior to the other option, which contradicts that they were generated as payoff maximizing distributions.

Because \( rsrs_{w+} \) was found by maximizing performance in \( U^{w+\epsilon} \) we know that \( U^{w+\epsilon}(rsrs_{w+}, BR) > U^{w+\epsilon}(rsrs_{w-}, BR) \). From the definition of \( U^{w+\epsilon} \) this gives us

\[ (w + \epsilon)U_1(rsrs_{w+}, \hat{s}_2) + (1 - w - \epsilon)U_1(rsrs_{w+}, BR) > (w + \epsilon)U_1(rsrs_{w-}, \hat{s}_2) + (1 - w - \epsilon)U_1(rsrs_{w-}, BR) \]

\[ (1) \]

Similarly, for \( rsrs_{w-} \) we have

\[ (w - \epsilon)U_1(rsrs_{w+}, \hat{s}_2) + (1 - w + \epsilon)U_1(rsrs_{w+}, BR) > (w - \epsilon)U_1(rsrs_{w-}, \hat{s}_2) + (1 - w + \epsilon)U_1(rsrs_{w-}, BR) \]

\[ (2) \]

We can manipulate equation 1 to get

\[ (w - \epsilon)U_1(rsrs_{w+}, \hat{s}_2) + (1 - w + \epsilon)U_1(rsrs_{w+}, BR) > (w - \epsilon)U_1(rsrs_{w-}, \hat{s}_2) + (1 - w + \epsilon)U_1(rsrs_{w-}, BR) \]

\[ (3) \]

For this to be true, and equation 2 to be true, we must have

\[ 2\epsilon((U_1(pbr_{w+}, \hat{s}_2) - U_1(pbr_{w+}, \hat{s}_2)) - (U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR))) > 0 \]

\[ (4) \]

We know \( U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR) \) and \( U_1(rsrs_{w+}, \hat{s}_2) - U_1(rsrs_{w-}, \hat{s}_2) \) have different signs. If \( U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR) \) is positive and \( U_1(rsrs_{w+}, \hat{s}_2) - U_1(rsrs_{w-}, \hat{s}_2) \) is negative, then equation 4 will be false, so it must be that \( U_1(rsrs_{w+}, \hat{s}_2) - U_1(rsrs_{w-}, \hat{s}_2) \) is positive and \( U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR) \) is negative.

Second, we will prove that the ratio of the gain in performance against the prediction to the loss in performance against a best-responding opponent is \( \frac{1-w}{w} \).

**Theorem.**

\[ \frac{U_1(rsrs_{w+}, \hat{s}_2) - U_1(rsrs_{w-}, \hat{s}_2)}{U_1(rsrs_{w+}, BR) - U_1(rsrs_{w-}, BR)} = \frac{1-w}{w} \]

We abuse notation to define \( U_1(rsrs_{w+}, \hat{s}_2) \) as the expected utility for player 1 of playing \( rsrs_{w+} \) against \( \hat{s}_2 \). \( U_1(rsrs_{w+}, BR) \) is the expected utility for player 1 of playing \( rsrs_{w+} \) against a best-responding opponent.

**Proof.** From how \( rsrs_{w+} \) is calculated we have

\[ U_1^{w+\epsilon}(rsrs_{w+}, BR) > U_1^{w+\epsilon}(rsrs_{w-}, BR) \]

where \( U_1^{w+\epsilon}(rsrs_{w+}, BR) \) is the expected value of playing \( rsrs_{w+} \) against a best-responding opponent in the modified game \( U^{w+\epsilon} \) created when calculating \( rsrs_{w+} \). Similarly

\[ U_1^{w-\epsilon}(rsrs_{w+}, BR) > U_1^{w-\epsilon}(rsrs_{w-}, BR) \]

From how \( U^{w} \) is constructed we have

\[ (w + \epsilon)U_1(rsrs_{w+}, \hat{s}_2) + (1 - w - \epsilon)U_1(rsrs_{w+}, BR) > (w + \epsilon)U_1(rsrs_{w-}, \hat{s}_2) + (1 - w - \epsilon)U_1(rsrs_{w-}, BR) \]

\[ (5) \]

and

\[ (w - \epsilon)U_1(rsrs_{w+}, \hat{s}_2) + (1 - w + \epsilon)U_1(rsrs_{w+}, BR) > (w - \epsilon)U_1(rsrs_{w-}, \hat{s}_2) + (1 - w + \epsilon)U_1(rsrs_{w-}, BR) \]

\[ (6) \]
By rearranging terms we have:
\[
\frac{U_1(rst^s_w+, \hat{s}_2) - U_1(rst^s_w-, \hat{s}_2)}{U_1(rst^s_w-, BR) - U_1(rst^s_w+, BR)} > \frac{1 - w - \epsilon}{w + \epsilon} \quad (7)
\]
and
\[
\frac{U_1(rst^s_w-, \hat{s}_2) - U_1(rst^s_w+, \hat{s}_2)}{U_1(rst^s_w+, BR) - U_1(rst^s_w-, BR)} < \frac{1 - w + \epsilon}{w - \epsilon} \quad (8)
\]
Note that the inequality in the second equation flips because \( U_1(rst^s_w+, BR) - U_1(rst^s_w-, BR) < 0 \) (see Lemma 1).
Combining the two equations, we get
\[
\frac{1 - w - \epsilon}{w + \epsilon} < \frac{U_1(rst^s_w+, \hat{s}_2) - U_1(rst^s_w-, \hat{s}_2)}{U_1(rst^s_w-, BR) - U_1(rst^s_w+, BR)} < \frac{1 - w + \epsilon}{w - \epsilon} \quad (9)
\]
By taking the limit as \( \epsilon \) converges to 0 we prove the theorem.

**Recursive Model**

An intuitive approach to dealing with an opponent which is exploiting the agent is to alter the prediction model of the agent to include the possibility of an opponent which exploits an agent using the original prediction model. This is the approach adopted by Locaine Powder (Egnor 2000). However, Locaine Powder has one advantage that is not generally available – the looping nature of Rock-Paper-Scissors means that there is a finite number of levels an opponent can be acting at before they begin acting like an opponent with fewer levels. We can estimate the probability of a best-responding opponent by tracking the relative probability of the observed sequence of play given the model, and given a best responding opponent, but that doesn’t allow prediction.

We have developed a recursive model based on fictitious play to study the problem of playing an opponent that can predict the agent’s actions. The lowest level of the model assumes that both players are playing a fixed probability distribution drawn from a uniform prior. Each subsequent level is generated by assuming that the players are responding to the previous level using RSRS with parameters drawn from a uniform distribution over possible values. Computational constraints on the ability of an agent to predict its opponent can be modelled by limiting the number of levels it uses. At the top of the model, the agent assumes that the opponent is either playing according to one of the lower levels, or it’s best-responding to the agent’s actions, or it’s playing a minimax strategy to minimize the agent’s payoff. With this information the agent can form a prediction of opponent behavior and determine appropriate parameters to use with RSRS.

**Learning a Fictitious Play Model**

The simplest level of our model assumes that the opponent plays a stationary probability distribution over all moves. We assume a uniform prior over distributions which allows us to track the posterior distribution by keeping a count of the number of times each action is observed. This results in a prediction of
\[
\hat{s}_2(a_i) = \frac{o_i}{\sum_{j=1}^{n} o_j}
\]
where \( o_i \) is the number of observations of \( a_i \). We initialize the \( o_i \) values to 1 to match our prior over distributions.

**Exponential Smoothing for Best Responses**

Detecting a best-responding or minimax strategy is important for an agent to choose appropriate parameter values for RSRS. A model which predicts that such an agent will only play the strategy which best-responds is very brittle with respect to noise and easily fooled. We use an exponentially weighted model, which predicts that the opponent will play a move with probability proportional to the value of that move.

Given a set of actions \( A = (a_1, ..., a_n) \) with expected payoffs \( P = (p_1, ..., p_n) \), we assume that the opponent will select action \( a_i \) with probability
\[
e^{p_i \lambda} \sum_{j=1}^{n} e^{p_j \lambda}
\]
for some \( \lambda \). \( \lambda = 0 \) describes an opponent which chooses actions uniformly at random. \( \lambda = \infty \) describes an opponent which strictly chooses the maximum expected payoff. Intermediate values represent the degree of bias the opponent displays towards higher expected payoffs. The appropriate value to use is dependent on the range of payoffs. In our experiments we use a value of 10.

**Recursive Model**

The full recursive model is based on a combination of these models (see Algorithm 1). Level 0 predicts the opponent will play according to fictitious play. A higher level predicts that the opponent will either follow a lower level model, or will respond to a lower level model, either by playing a best response (handled with the multinomial logit decision model) or by playing a RSRS.

With the recursive model we can explore the problem of exploitation and examine the efficiency of PBR at avoiding exploitation by more sophisticated agents.

**Partial Best Response Performance**

Figure 2 shows the performance of an agent using the recursive model against opponents of different levels in repeated games of Rock-Paper-Scissors. Agents play a sequence of 100 games of Rock-Paper-Scissors. Results are aggregated over 10 sequences of games played. The stationary opponent always plays Rock, all other opponents use the recursive model to predict
Algorithm 1 Recursive model for one level
1: Observe game
2: Get prediction from lower level
3: Calculate exponentially weighted best response for prediction using $\lambda$
4: Calculate RSRS for prediction using $w$ and $e$
5: Produce weighted average prediction
6: Observe outcome
7: Update child and observe probability
8: Update $\lambda$ for MLD and observe probability
9: Update PBR and observe probability
10: Update weights of child, MLD, and PBR models using observed probability

Figure 2: Performance of an agent using the recursive model against opponents of different levels

Figure 3: Performance of an agent playing PBR against different levels of opponent playing PBR

Figure 4: Effect of using different values of $w$ when playing against an opponent that the agent can predict

Figure 5: Effect of using different values of $w$ when playing against an opponent which can predict the agent

also plays PBR. In this case both the agent and its opponent adjust their $w$ values according to the performance of their prediction. The agent performs particularly well only against the stationary opponent. This occurs because PBR opponents are not exploitable.

Figure 4 shows the effect of using different values of $w$ when playing an opponent the agent can predict. Higher values result in higher expected payoffs. For $w = .2$ the expected payoff is 0 because PBR will always play the Nash equilibrium in Rock-Paper-Scissors with that value. Note that the opponent is not stationary, so the prediction changes as play progresses and the opponent changes its behavior.

Figure 5 shows the effect of using different values of $w$ when playing against an opponent which can predict the agent. Surprisingly, the agent manages to avoid being exploited even with a high value for $w$. This occurs because the best fit it can find for the opponent within the model is a stationary player which plays each strategy roughly equally often. This model is hard to exploit, so it requires high $w$ values to deviate from the Nash equilibrium.
Conclusions and Future Work

We have presented PBR, an algorithm which allows an agent to exploit a prediction of opponent behavior while guarding against the risk of being exploited. By selecting an appropriate value for \( w \), an agent can make itself highly resistant to exploitation while still exploiting its prediction. Further investigation will study how to choose a value for \( w \), and explore performance issues in non-constant sum games, where the best-response assumption is harder to justify.

References


