Safely Using Predictions in General-Sum Normal Form Games

Steven Damer and Maria Gini
{damer,gini} @cs.umn.edu
University of Minnesota

ABSTRACT

It is often useful to predict opponent behavior when playing a general-sum two-player normal form game. However best-responding to an inaccurate prediction can lead to a strategy which is vulnerable to exploitation. This paper proposes a novel method, Restricted Stackelberg Response with Safety (RSRS), for an agent to select a strategy to respond to a prediction. The agent uses the confidence it has in the prediction and a safety margin which reflects the level of risk it is willing to tolerate to make a controlled trade-off between best-responding to the prediction and providing a guarantee of worst-case performance. We describe an algorithm which selects parameter values for RSRS to produce strategies that play well against the prediction, respond to a best-responding opponent, and guard against worst-case outcomes. We report results obtained by the algorithm on multiple general-sum games against different opponents.

Keywords

Game theory, general-sum normal form games, risk management, prediction errors

1. INTRODUCTION

In many situations agents have to make decisions in a competitive environment, where their interests are directly opposed to their opponent's interests (zero-sum games), but more often some form of collaboration with the opponent can increase the payoff for both players (general-sum games). Examples of such situations can be found in negotiation, cybersecurity, physical security, electronic commerce, and more. In such environments, agents often use a prediction of opponent behavior to guide their decisions.

How should an agent respond when given a prediction of opponent behavior in a general-sum two-player normal form game? Selecting the strategy with the highest payoff against the prediction provides optimal performance if the prediction is correct, but can be arbitrarily bad if the prediction is incorrect. Playing a maximin strategy guarantees a payoff equal to the safety value of the game, but at the cost of performance against the prediction. Playing a Nash equilibrium only makes sense if the prediction is incorrect and the opponent also plays that Nash equilibrium. Furthermore, a maximin strategy or a Nash equilibrium don't use a prediction, so there is no reason for an agent using them to make a prediction.

In our previous work [7], we have shown how an agent can model the opponent's attitude towards cooperation to pursue opportunities for cooperation while limiting exploitation. In this paper we introduce *Restricted Stackelberg Response with Safety* (RSRS), a novel method of choosing a mixed strategy for a general-sum game, given a prediction of the opponent's strategy. RSRS uses a prediction weight parameter, w, to determine how much to guard against a best-responding opponent, and a risk factor parameter, r, to determine how much to guard against a worst-case outcome. RSRS provides a way to make a controlled tradeoff between responding to a (possibly flawed) prediction and dealing with a best-responding opponent while also providing a worst-case performance guarantee. In this paper we use fictitious play to make predictions to demonstrate that RSRS can handle flawed predictions, but RSRS can be used with any prediction method that produces a probability distribution over opponent moves.

If the opponent plays a mixed strategy it is impossible to predict their exact action without knowledge of their randomization device, but even if we settle for predicting their mixed strategy we will only be able to provide predictions for a subclass of all possible opponents. Eventually, it is necessary to accept that one's prediction is as accurate as it can be, and deal rationally with the possibility that it may still be inaccurate.

This leads us to consider the ways in which a prediction can be incorrect, and how we can track and respond to that. In some cases, a prediction may be technically incorrect, but harmlessly so. For example, consider an agent which plays a mixed strategy but uses the digits of π as its randomization device. Such an agent is perfectly predictable in theory (it is playing a fixed sequence of moves), but in practice it is impossible to consider all strategies of that type. In this situation it is reasonable and effective to model the agent as playing the mixed strategy, despite the fact that a more accurate prediction is theoretically possible.

The prediction errors which are of practical concern are those in which the opponent is exploiting the agent's response to an inaccurate prediction, either for their own benefit, or to reduce the payoff to the agent. (Technically speaking it is also a prediction error when the opponent plays to increase the agents score or reduce its own score, but we feel that issue does not represent a significant problem). RSRS allows an agent to deal appropriately with those types of opponents by choosing appropriate parameter values.

The main contributions of this paper are:

1. a novel algorithm, the Restricted Stackelberg Response with Safety (RSRS), which is applicable to general sum games and which calculates a strategy that provides a balance between best-responding to the prediction, avoiding exploitation by a smarter opponent, and guaranteeing the safety value, according to the parameter values it is given,

Appears in: *Proc. of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

2. a method to compute appropriate parameter values to be used for RSRS, and
3. experimental results obtained in several non-zero sum games against different types of opponents: an omniscient opponent who is best-responding, a worst-case opponent, and a simple learning opponent. We compare the performance of RSRS against state-of-art algorithms (Best response, RNR, SPS, and GIGA-Wolf). We also prove the uniqueness of Restricted Nash Response (RNR) in zero-sum games.

Terminology. A game G consists of a set of players $\{1, 2\}$, a set of actions for each player $M_1 = \{m_1^1 \dots m_1^{n_1}\}$, $M_2 = \{m_2^1 \dots m_2^{n_2}\}$, and a set of utility functions $U_1, U_2 : M_1 \times M_2 \to \Re$. $s_1 \in \triangle^{n_1-1}$ and $s_2 \in \triangle^{n_2-1}$ are the mixed strategies adopted by player 1 and player 2 respectively.

We define $U_i(s_1, s_2) = \mathbb{E}_{m_1 \sim s_1, m_2 \sim s_2}[U_i(m_1, m_2)]$ as the expected outcome for player i when actions are drawn from the distributions s_1 and s_2. The Nash equilibrium is a set of strategies s_1, s_2 such that $U_1(s_1, s_2) \geq \max_{m_1 \in M_1} U_1(m_1, s_2)$ and $U_2(s_1, s_2) \geq \max_{m_2 \in M_2} U_2(s_1, m_2)$. The safety value of game G for player i against opponent j is: $V_G^i = \max_{s_i \in \triangle^{n_i-1}} \min_{m_j \in M_j} U_i(s_i, m_j)$. This is the greatest amount player i can guarantee for herself regardless of the opponent's action. Note that for general-sum games, this value may be lower than the expected payoff of any Nash equilibrium of the game.

If player 1 is designated as a *Stackelberg leader* [10] for the game, she selects a mixed strategy which is observed by player 2 before player 2 selects her strategy.

The Stackelberg equilibrium of a game is a set of strategies s_1, s_2 such that

$$\begin{aligned} s_1 &= \operatorname*{argmax}_{s \in \triangle^{n_1-1}} U_1(s, \operatorname*{argmax}_{s' \in M_2} U_2(s, s')) \\ s_2 &= \operatorname*{argmax}_{s \in M_2} U_2(s_1, s) \end{aligned}$$

Demonstration Game. The advantages of RSRS can most easily be observed in competitive general sum games where players have some common interests. In more competitive games, such as Rock/Paper/Scissors, performance is similar to other algorithms for risk avoidance. In more cooperative games, such as Battle of the Sexes, faults in the predictor aren't as significant because the opponent has less motivation to play deceptively.

Table 1: Payoffs for Rock/Spock/Paper/Lizard/Scissors.

	Rock	Spock	Paper	Lizard	Scissors
Rock	0,0	-.5,1.5	-1.5,.5	.5,-1.5	1.5,-.5
Spock	1.5,-.5	0,0	-.5,1.5	-1.5,.5	.5,-1.5
Paper	.5,-1.5	1.5,-.5	0,0	-.5,1.5	-1.5,.5
Lizard	-1.5,.5	.5,-1.5	1.5,-.5	0,0	-.5,1.5
Scissors	-.5,1.5	-1.5,.5	.5,-1.5	1.5,-.5	0,0

The game we will use to show the properties of our RSRS method is a general-sum modification of Rock/Spock/Paper/Lizard/Scissors – a variant of Rock/Paper/Scissors with 5 moves (Table 1).

Rock/Paper/Scissors/Lizard/Spock was presented in the TV show The Big Bang Theory; we have modified it to make it general-sum, and changed the name to reflect the precedence relationship between the moves. Each action beats two other actions, and is beaten in turn by the two remaining actions, as shown in Figure 1.

Players receive a payoff of 1 for a win, -1 for a loss, and 0 for a tie. In addition, both players receive .5 when adjacent moves are played and lose .5 when non-adjacent moves are played. The game has a unique Nash equilibrium at $s_1 = s_2 = (.2, .2, .2, .2, .2)$.

In this game players have conflicting interests but some cooperation is possible, which allows us to distinguish between a best-responding opponent and a worst case outcome. This distinction highlights the properties of our algorithm, which is why we use Rock/Spock/Paper/Lizard/Scissors for a demonstration game.

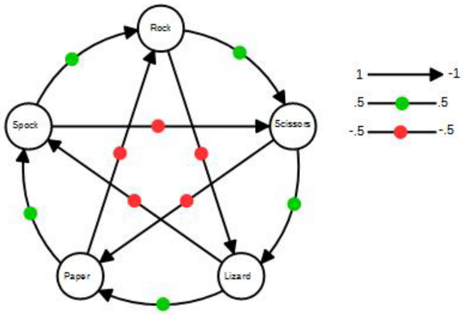

Figure 1: Precedence relationships in Rock/Spock/Paper/-Lizard/Scissors. Arrows point from winning moves to losing moves. Green dots indicate adjacent moves which receive a bonus. Red dots indicate non-adjacent moves which receive a penalty.

2. RELATED WORK

General sum normal form games provide a useful formalization to describe interacting agents. In situations without the public knowledge necessary to justify a Nash equilibrium, a reasonable approach is to form a prediction of the opponent behavior and respond to the prediction.

Our work focuses on finding safe strategies to respond to a prediction. Fictitious play [9] is the simplest way to form a prediction and respond to it. It predicts the most likely opponent strategy under the assumption the opponent is playing a stationary strategy and then plays a best response to that strategy. In self-play the empirical distribution formed by fictitious play can arrive at a Nash equilibrium, but that is not guaranteed. Fictitious play is easily predictable, and can be taken advantage of. De Cote and Jennings [8] describe a method of taking advantage of fictitious play by identifying sequences of moves which to a fictitious player appear to be a stationary distribution, but provide a higher payoff than what that distribution would receive in expectation.

Unlike fictitious play GIGA-WolF [2] does not form a prediction, instead it continually adjusts its strategy towards higher rewards, following the Win-Or-Learn-Fast principle [3]. It is guaranteed to achieve no-regret and to converge to an equilibrium in self-play.

AWESOME [6] forms a prediction of opponent behavior based on differentiating between a stationary distribution, an equilibrium strategy, or some unknown strategy. It best responds to a stationary distribution, and plays an equilibrium strategy otherwise. It guarantees a best-response against a stationary player, but unlike fictitious play it will arrive at an equilibrium in self-play.

TPCM(A) [16] forms a more sophisticated prediction of opponent behavior. It detects whether an opponent can be trained using Godfather [13], if it is willing to cooperate to achieve a Pareto-efficient outcome, or if it plays a fixed strategy conditional on the last k turns. It is optimal against its target set, Pareto-efficient in self-play, and guaranteed to achieve its safety value. CMLeS [4] provides similar guarantees against a larger set of target opponents. It models the opponent as playing a strategy conditional on some number of previous turns, up to a fixed maximum. It uses the Nash equilibrium as a fallback position against opponents it can't predict.

RWYWE [11] plays an equilibrium strategy by default. If the opponent plays a non-equilibrium strategy which gives the agent a higher payoff, it will adjust its strategy to respond better to the opponent's apparent strategy while guaranteeing an expected loss no greater than the already realized expected gain. Wang et al. [17] describe an algorithm which plays a Nash equilibrium in a modified game which constrains the opponent to play a strategy similar to the observed strategy, retreating to minimax if the opponent doesn't play predictably.

One way to handle a sophisticated opponent is to assume that the opponent can accurately predict the strategy the agent will choose. A Stackelberg game has a designated leader and follower. The leader commits to a mixed strategy which is revealed to the follower, who then selects their response. The leader's ability to pre-commit to a strategy can be beneficial, allowing it to pre-commit to a preferred equilibrium, or even a non-equilibrium strategy which has a better outcome for them.

Our work uses the Stackelberg equilibrium. To avoid being deceived by an opponent which plays a distribution near the best response but is not strictly best-responding, we use an exponential response function to our chosen distribution that models an opponent who is biased towards best-responding. A similar problem is addressed in [15] from the other direction – namely being a Stackelberg leader when the opponent has cognitive biases, such as anchoring or bounded rationality. We modify the Stackelberg equilibrium to reflect our prediction of opponent behavior and desired safety value. They modify the Stackelberg equilibria to reflect the fact that the follower may not be strictly best-responding.

The approaches most similar to our work are Safe Policy Selection (SPS) [14] and Restricted Nash Response (RNR) [12], which we describe next.

3. BACKGROUND

Safe Policy Selection. SPS [14] is a method of deciding how much to risk against a potentially stronger opponent. Given a game G with safety value V_G^1, an r-safe strategy s_1^r is one whose worst case payoff is within r of the safety value

$$\min_{s_2 \in \triangle^{n_2-1}} U(s_1^r, s_2) \geq V_G^1 - r.$$

SPS selects the r-safe strategy with the best performance against the prediction. Over a series of games SPS adjusts r values to guarantee a payoff close to the value of the game, while also performing well against predictable opponents. It does this by setting r_n (the r value to use in round n) according to $r_n = r_{n-1} + 1/n + U_1(s_1^{n-1}, m_2^{n-1}) - V_G^1$ where s_1^n is the agent's mixed strategy in round n and m_2^n is the opponent's move in round n. Results in Rock/Paper/Scissors demonstrate that SPS can improve the performance of weak players against stronger players.

Restricted Nash Response. RNR [12, 1] exploits a prediction of opponent behavior in a zero-sum game while avoiding exploitation. It finds a strategy by constructing a modified game, and taking the Nash equilibrium of that game. Results in poker demonstrate that it is possible to find strategies which are very effective against the prediction while remaining resistant to exploitation.

To calculate a RNR to a prediction $s_2 \in \triangle^{n_2-1}$ in a zero-sum game G using a weight $w \in [0,1]$ construct a modified game G' with $M_1' = M_1$, $M_2' = M_2$, $U_2' = U_2$, and $U_1'(m_1, m_2) = w \times U_1(m_1, s_2) + (1-w) \times U_1(m_1, m_2)$. RNR returns the Nash equilibrium of G'. Note that although G' is not zero-sum equilibrium selection is not a problem because, as we will show later, all equilibria of G' are interchangeable.

Although RNR and SPS are generated by different procedures, the set of strategies generated by RNR is a subset of the set of strategies generated by SPS. Johanson et al. [12] demonstrate that for any w value, there will always be an r value for which SPS produces the same strategy as RNR.

We will show that the general-sum modified game created to calculate a RNR for a zero-sum game has a unique Nash equilibrium. We consider two equilibria distinct if each player strictly prefers to play their equilibrium strategy in each equilibrium: $U_1(s_1, s_2) > U_1(s_1', s_2)$, $U_1(s_1', s_2') > U_1(s_1, s_2')$, $U_2(s_1, s_2) > U_2(s_1, s_2')$, and $U_2(s_1', s_2') > U_2(s_1', s_2)$. (If the preference is weak then the two equilibria are part of the same connected component and players can play either strategy and achieve the same payoff.)

Theorem 1. *Given a zero-sum game G with utility functions $U_1 = U$ and $U_2 = -U$ let G' be the modified game created to calculate a RNR to a prediction p, with utility function U' where $U_1'(m_1, m_2) = w \times U(m_1, p) + (1-w) \times U(m_1, m_2)$ and $U_2' = -U$. G' doesn't have two distinct equilibria.*

Proof. Assume G' has two distinct Nash equilibria s and s'. Construct a new game G'' from G' with moves $s_1, s_1' \in \triangle^{n_1-1}$ and $s_2, s_2' \in \triangle^{n_2-1}$ and payoffs equal to playing the corresponding strategies in G'. Because s and s' are distinct, we have:

$$\begin{aligned} w \times U(s_1, p) + (1-w) \times U(s_1, s_2) > \\ w \times U(s_1', p) + (1-w) \times U(s_1', s_2) \end{aligned} \tag{1}$$

$$\begin{aligned} w \times U(s_1', p) + (1-w) \times U(s_1', s_2') > \\ w \times U(s_1, p) + (1-w) \times U(s_1, s_2') \end{aligned} \tag{2}$$

$$-U(s_1', s_2') > -U(s_1', s_2) \tag{3}$$

$$-U(s_1, s_2) > -U(s_1, s_2') \tag{4}$$

From 1 and 3 we get:

$$\begin{aligned} w \times U(s_1, p) + (1-w) \times U(s_1, s_2) > \\ w \times U(s_1', p) + (1-w) \times U(s_1', s_2') \end{aligned} \tag{5}$$

From 5 and 2 we get:

$$\begin{aligned} w \times U(s_1, p) + (1-w) \times U(s_1, s_2) > \\ w \times U(s_1, p) + (1-w) \times U(s_1, s_2') \end{aligned} \tag{6}$$

From 6 and 4 we get:

$$\begin{aligned} w \times U(s_1, p) + (1-w) \times U(s_1, s_2) > \\ w \times U(s_1, p) + (1-w) \times U(s_1, s_2) \end{aligned} \tag{7}$$

This is not possible, so there cannot be two distinct Nash equilibria in the modified game created for RNR. □

4. EXTENSION TO GENERAL-SUM GAMES

Safe Policy Selection. SPS was developed for zero-sum games, but it can be extended to general-sum games by treating the value of the game as the amount which the player can guarantee for itself, regardless of the actions of the opponent. In general-sum games SPS may not be an effective method of ameliorating risk. Consider a game in which an opponent has a punishing move which causes the agent to receive a bad outcome regardless of the action the agent chooses. In this case, regardless of the risk value chosen, the algorithm will play a best-response to the prediction, because it will do no worse than any other strategy if the opponent selects the punishing move. We don't use SPS in such games.

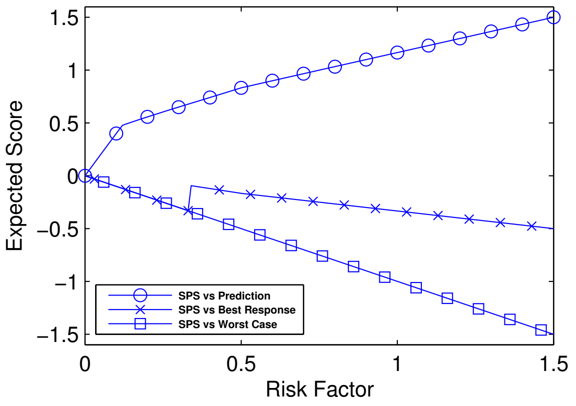

Figure 2: Payoffs of r-safe strategies with a prediction of Rock in the game Rock/Spock/Paper/Lizard/Scissors for different values of r.

Figure 2 shows how the r value affects the performance of SPS in Rock/Spock/Paper/Lizard/Scissors. Performance is measured against the prediction, against a best responding opponent which maximizes its own payoff given the agent's strategy, and against a worst case opponent which minimizes the agent's payoff given the agent's strategy. When $r = 0$ the generated strategy is the maximin strategy $(.2, .2, .2, .2, .2)$. When $r = 1.5$ the generated strategy is $(0, 1, 0, 0, 0)$, which is the best response to the prediction. Intermediate values cause the generated strategy to vary continuously between those two extremes.

Restricted Nash Response. RNR can be extended to general sum games by providing a method of selecting an equilibrium when there are multiple equilibria, such as choosing the equilibria with the highest payoff. RNR does not generate multiple equilibria for Rock/Spock/Paper/Lizard/Scissors. Figure 3 shows the effect of the w parameter. The agent predicts Rock and performance is measured against the prediction, against a best responding opponent which maximizes its own payoff given the agent's strategy, and against a worst case opponent which minimizes the agent's payoff given the agent's strategy. When $w = 1$, RNR will play a best response to the prediction, When $w = 0$, RNR will play a Nash equilibrium of the original game. Intermediate values will cause the strategy to abruptly change when increasing the w value prevents the opponent from playing its strategy in the current equilibrium. In general-sum games, increasing the weight value can reduce performance against the prediction when the opponent strategy in the new equilibrium is beneficial to the agent.

5. RESTRICTED STACKELBERG RESPONSE WITH SAFETY

We define the Restricted Stackelberg Response with Safety (RSRS) for player 1 in game G with prediction $p \in \triangle^{n_2-1}$, prediction weight $w \in [0, 1]$, and risk factor $r \in \Re+$ to be the mixed strategy for player 1 that maximizes its expected payoff given the assumption that, with probability w, the opponent will play according to the prediction p and, with probability $1 - w$, it will best-respond to the agent, subject to the constraint that its expected payoff when played against any opponent action is at least $V_G^1 - r$. The Restricted Stackelberg Response (RSR) is identical except that it has no constraint using r (it has no safety factor).

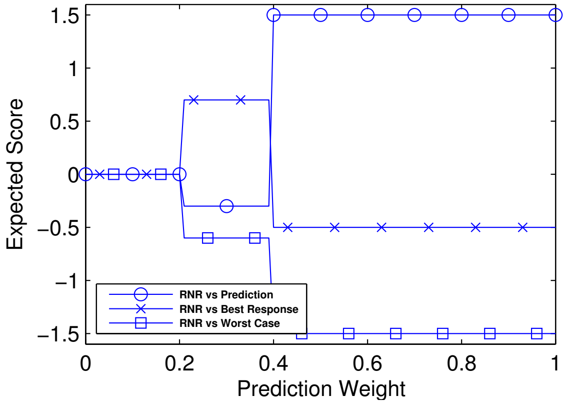

Figure 3: Payoffs of RNR to a prediction of Rock in Rock/Spock/Paper/Lizard/Scissors for different values of w. The chosen strategy changes at weight values .2 and .4, where the changing parameter disrupts the current equilibrium.

The RSRS is calculated by constructing a new payoff function for player 1 which reflects the assumption that the opponent will play the prediction with probability w:

$$U_1'(m_1, m_2) = w \times U_1(m_1, p) + (1 - w) \times U_1(m_1, m_2)$$

The RSRS for player 1 is the probability distribution $s_1 \in \triangle^{n_1-1}$, which maximizes the expected value of U_1' under the assumption that player 2 will best respond, subject to the constraint

$$U_1(s_1, m_2) \geq V_G^1 - r \text{ for all } m_2 \in M_2.$$

Assuming that the opponent is best-responding to the action of player 1 is equivalent to designating player 1 as a Stackelberg leader. The game does not have a Stackelberg leader; that assumption is a convenient way to handle the possibility of a best-responding opponent.

We compute the RSRS using a modification of the technique in [5]. For each opponent action we find a mixed strategy to which the opponent action is a best response and which satisfies the safety value constraint. Then we select the option which performs best against a weighted combination of the prediction and a best responding opponent. More formally, for each $m_2 \in M_2$ maximize over $s_1^{m_2} \in \triangle^{n_1-1}$:

$$s_1^{m_2} = \text{argmax}_{s_1 \in \triangle^{n_1-1}} U_1'(s_1, m_2)$$

$$\text{subject to: } \quad \forall m_2' \in M_2, U_2(s_1^{m_2}, m_2) \geq U_2(s_1^{m_2}, m_2')$$

$$\text{and } \quad \forall m_2' \in M_2, U_1(s_1^{m_2}, m_2') \geq V_G^1 - r$$

Solving this 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 in U_1' against the opponent's best response is the RSRS. The complexity of calculating the RSRS is polynomial in the number of moves in the game.

The values chosen for probability weight (w) and risk factor (r) control the tradeoff between performance against the prediction, performance against a best-responding opponent (w), and performance against a worst-case opponent (r).

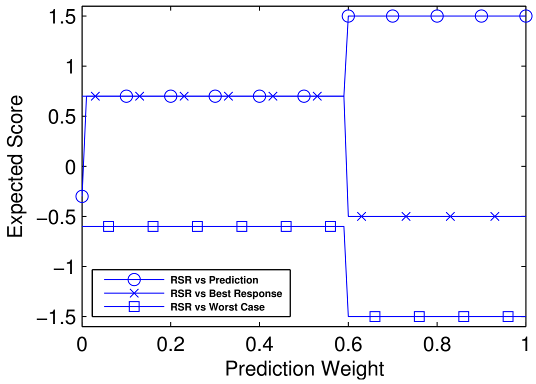

Figure 4: Payoffs of RSRS to a prediction of Rock in the game Rock/Spock/Paper/Lizard/Scissors with w varying from 0 to 1, and r fixed at 1.5.

For a fixed risk factor there are many probability weights that produce the same strategy – changes in w either produce no change in strategy or a discontinuous jump to a new strategy. Jumps occur when confidence in the prediction becomes high enough to justify the additional vulnerability to a best-responding opponent. In contrast, changes to r produce a continuous variation between a minimax strategy and a prediction exploiting strategy. The effect of r dominates the effect of w. If $r = 0$, the w value has no effect. This allows us to provide the same guarantees as SPS by selecting a value for r as SPS does. We select a value for w by calculating the relative posterior probability of the opponent following the prediction and the opponent best-responding.

Figure 4 shows the effects of w on performance, which is measured against the prediction, against a best responding opponent which maximizes its own payoff given the agent's strategy, and against a worst case opponent which minimizes the agent's payoff given the agent's strategy. $w < .6$ produces a Stackelberg equilibrium, $w > .6$ produces a best response to the prediction. The strategy abruptly changes at .6 because at that point a threshold is passed where the increased payoff against the prediction justifies a reduced payoff against a best-responding opponent. Like SPS, r produces a continuous variation of performance (see Figure 2), ranging from the maximin strategy ($r = 0$) to a best response ($r = 1.5$).

We can characterize the change in performance produced by a change in w for a fixed r in terms of the trade-off between performance against the prediction and performance against a best-responding agent. For example, in Rock/Spock/Paper/Lizard/Scissors, with a prediction of Rock and $r = 1.5$

RSRS produces $(0, .6, 0, 0, .4)$ when $w < .6$ and $(0, 1, 0, 0, 0)$ when $w \geq .6$. The first strategy produces a payoff of $.7$ when played against the prediction or a best-responding opponent. The second equation produces a payoff of 1.5 when played against the prediction, and a payoff of $-.5$ when played against a best-responding opponent.

When w changes from below .6 to above .6 RSRS gains .8 in expected payoff against the prediction and loses 1.2 against a best-responding opponent. The expected gain is $2/3$ as much as the expected loss. This matches $.4/.6$, the relative probability of those events expressed by a w value of .6. This relationship holds for any game and value of w.

We are interested in values of w between two regions where the RSRS for a fixed r does not change. For those w values we denote with $rsrs_{w+}$ and $rsrs_{w-}$ respectively the RSRS for the region with weight values higher or lower than w. We denote with br_{w+} and br_{w-} the best responses to those strategies. Assume we are given a game G, a prediction $p \in \triangle^{n_2 - 1}$, and a w where the RSRS changes. If there is a δ such that for all $0 < \epsilon < \delta$ the RSRS with weight $w + \epsilon$ is the same for all ϵ, and the RSRS with weight $w - \epsilon$ is the same for all ϵ, and $rsrs_{w+} \neq rsrs_{w-}$, we will prove:

Theorem 2. *The ratio of the performance gain against the prediction to the performance loss against a best-responding opponent is $\frac{1-w}{w}$, i.e.,*

$$\frac{U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)}{U_1(rsrs_{w-}, br_{w-}) - U_1(rsrs_{w+}, br_{w+})} = \frac{1-w}{w}$$

We will begin by showing that reducing w can only improve performance against a best-responding opponent. This may seem obvious, but it doesn't hold for RNR in general-sum games.

Lemma 1. $U_1(rsrs_{w+}, br_{w+}) < U_1(rsrs_{w-}, br_{w-})$

Proof. Consider the quantities $U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)$ and $U_1(rsrs_{w+}, br_{w+}) - U_1(rsrs_{w-}, br_{w-})$, which represent the performance gain of $rsrs_{w+}$ relative to $rsrs_{w-}$ against the prediction and against a best-responding opponent respectively. If both are positive or both are negative then $rsrs_{w+}$ or $rsrs_{w-}$ would be strictly superior to the other, which contradicts the fact that that they were generated as payoff-maximizing distributions. Let $U^{w+\epsilon}$ be the utility function for player 1 in the modified game created with prediction p and weight $w+\epsilon$. Because $rsrs_{w+}$ was found by maximizing performance in $U^{w+\epsilon}$ against a best-responding opponent, we know that $U^{w+\epsilon}(rsrs_{w+}, br_{w+}) > U^{w+\epsilon}(rsrs_{w-}, br_{w-})$. From the definition of $U^{w+\epsilon}$ this gives us

$$\begin{aligned}&(w+\epsilon)U_1(rsrs_{w+}, p) + (1-w-\epsilon)U_1(rsrs_{w+}, br_{w+}) > \\ &(w+\epsilon)U_1(rsrs_{w-}, p) + (1-w-\epsilon)U_1(rsrs_{w-}, br_{w-})\end{aligned} \tag{8}$$

Similarly, for $rsrs_{w-}$ we have

$$\begin{aligned}&(w-\epsilon)U_1(rsrs_{w-}, p) + (1-w+\epsilon)U_1(rsrs_{w-}, br_{w-}) > \\ &(w-\epsilon)U_1(rsrs_{w+}, p) + (1-w+\epsilon)U_1(rsrs_{w+}, br_{w+})\end{aligned} \tag{9}$$

We can manipulate Eq. 8 to get

$$\begin{aligned}&(w-\epsilon)U_1(rsrs_{w+}, p) + (1-w+\epsilon)U_1(rsrs_{w+}, br_{w+}) \\ &+2\epsilon((U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)) \\ &-(U_1(rsrs_{w+}, br_{w+}) - U_1(rsrs_{w-}, br_{w-}))) \\ &> (w-\epsilon)U_1(rsrs_{w-}, p) + (1-w+\epsilon)U_1(rsrs_{w-}, br_{w-})\end{aligned}$$

For this and Eq. 9 to be true, we must have

$$\begin{aligned}&2\epsilon((U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)) > \\ &(U_1(rsrs_{w+}, br_{w+}) - U_1(rsrs_{w-}, br_{w-})))\end{aligned} \tag{10}$$

We know that $U_1(rsrs_{w+}, br_{w+}) - U_1(rsrs_{w-}, br_{w-})$ and that $U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)$ have different signs. If the first term is positive and the second negative, then Eq. 10 will be false, so it must be that $U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)$ is positive and $U_1(rsrs_{w+}, br_{w+}) - U_1(rsrs_{w-}, br_{w-})$ is negative. □

Using Lemma 1, we can prove Theorem 2.

Proof. $rsrs_{w+}$ is calculated by maximizing payoff in the modified game, so

$$U_1^{w+\epsilon}(rsrs_{w+}, br_{w+}) > U_1^{w+\epsilon}(rsrs_{w-}, br_{w-})$$

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

$$U_1^{w-\epsilon}(rsrs_{w-}, br_{w-}) > U_1^{w-\epsilon}(rsrs_{w+}, br_{w+})$$

From how U^w is constructed we have

$$\begin{aligned}&(w+\epsilon)U_1(rsrs_{w+}, p) + (1-w-\epsilon)U_1(rsrs_{w+}, br_{w+}) > \\ &(w+\epsilon)U_1(rsrs_{w-}, p) + (1-w-\epsilon)U_1(rsrs_{w-}, br_{w-})\end{aligned}$$

$$\begin{aligned}&(w-\epsilon)U_1(rsrs_{w-}, p) + (1-w+\epsilon)U_1(rsrs_{w-}, br_{w-}) > \\ &(w-\epsilon)U_1(rsrs_{w+}, p) + (1-w+\epsilon)U_1(rsrs_{w+}, br_{w+})\end{aligned}$$

By rearranging terms we have:

$$\frac{U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)}{U_1(rsrs_{w-}, br_{w-}) - U_1(rsrs_{w+}, br_{w+})} > \frac{1-w-\epsilon}{w+\epsilon}$$

and

$$\frac{U_1(rsrs_{w+}, p) - U_1(rsrs_{w-}, p)}{U_1(rsrs_{w-}, br_{w-}) - U_1(rsrs_{w+}, br_{w+})} < \frac{1-w+\epsilon}{w-\epsilon}$$

These last two equations provide the lower and upper bounds. By taking the limit as $\epsilon \longrightarrow 0$ we prove the theorem. □

6. LEARNING WEIGHT VALUES

RSRS assumes that the opponent will best respond if the prediction is incorrect, so to compute w we estimate the relative probabilities that the opponent played according to the prediction or played a best response in previous rounds. Computing the probability the opponent has played according to the prediction is easy. A naive method of estimating the probability that the opponent played a best response would assign a probability of 1 or 0 (either the opponent played a best response in every previous game or not). This is easy to deceive – for example, an opponent which consistently plays the second-best response would not be considered to be best-responding.

We adopt a model in which the opponent plays according to an exponential response function to their expected payoff against the agent's chosen strategy. Given an agent strategy $s_1 \in \triangle^{n_1-1}$, the opponent's exponentially weighted response is

$$P(m_2^i) = \frac{e^{\lambda U_2(s_1, m_2^i)}}{\sum_{m_2^j \in M_2} e^{\lambda U_2(s_1, m_2^j)}}$$

where λ describes how responsive the opponent is to higher payoffs. $\lambda = 0$ describes an opponent that plays uniformly at random. The higher the λ value, the stronger the opponent's preference for higher expected payoff moves.

We calculate λ by finding the value with the maximum likelihood for the prior actions of the opponent. If all the observations have been best responses, the probability maximizing value will be ∞, so we introduce a smoothing observation (see Algorithm 1). We then use λ to compute the probability that a best-responding opponent played the observed move. This allows us to compute the relative probability of a best-responding opponent vs. an opponent playing according to the prediction. We use that value to determine the probability weight to use with RNR and RSRS (see Algorithm 1).

Algorithm 1 Estimate relative probability of prediction and best-responding opponent

```
Initialize $t = 1$, $chosen_1 = .9$, $options_1 = (0, .9, 1)$
$P(Prediction) = .5$, $P(BestResponse) = .5$
while the game continues do
    {Make observations from the previous round}
    Set $s_1' \in \triangle M_1$, the strategy played by the agent
    Set $s_2' \in \triangle M_2$, the predicted strategy for the opponent
    Set $m_2' \in M_2$, the observed opponent move
    Set $u_2' = U_S(s_1', m_2')$, the opponents expected utility
    {Calculate the expected payoff of the opponent's moves}
    for all $m_2^i \in M_2$ do
        Set $u_2^i = U_2(s_1', m_2^i)$
    end for
    Increment $t$
    Set $chosen_t = u_2'$
    Set $options_t = u_2^1..u_2^n$
    Find $\lambda$ maximizing $\prod_{i=1}^t \frac{e^{\lambda chosen_i}}{\sum_{j=1}^n e^{\lambda options_{i,j}}}$ using gradient descent
    {Update the estimated probabilities}
    $P(Prediction) = P(Prediction) \times s_2'(m_2')$
    $P(BestResponse) = P(BestResponse) \times \frac{e^{\lambda u_2'}}{\sum_{i=1}^n e^{\lambda u_2^i}}$
    Renormalize $P(Prediction)$ and $P(BestResponse)$
    Set prediction weight to $\frac{P(Prediction)}{P(Prediction)+P(BestResponse)}$
end while
```

7. RESULTS

We report results obtained in Rock/Spock/Paper/Lizard/Scissors, Traveller's Dilemma, and a simple pursuit/evasion game. Graphs for Battle of the Sexes, Chicken, Stag Hunt, and several other games are omitted for lack of space. In more competitive games all approaches are broadly successful since a best-responding opponent behaves like a worst-case opponent. In more cooperative games, using the Stackelberg equilibrium provides a significant gain against best-responding opponents.

In the experiments agents play a sequence of 100 games. Results are averaged over 100 repetitions. We show the performance of six approaches:

1. a best response to the prediction,
2. SPS,
3. RNR using the calculated weight,
4. RSR using the calculated weight (with no risk factor),
5. RSRS using the calculated weight and SPS to determine risk factors, and
6. GIGA-WolF.

All, except GIGA-WolF, predict the opponent using fictitious play: they assume a stationary opponent playing a distribution drawn from a uniform Dirichlet distribution, and predict using the expected value of that distribution given the observed moves. This flawed prediction technique is chosen to show that RSRS can handle inaccurate predictions.

Figure 5 shows performance of the six different algorithms against three opponents, an omniscient best-responding opponent, a worst-case opponent, and a simple learning opponent (observes for 50 moves, and then plays a best response to the observed distribution).

An omniscient best-responding opponent knows the agent's mixed strategy and plays to maximize its own payoff. A worst case opponent knows the agent's mixed strategy and plays to minimize the agent's payoff. These opponents provide a strong basis for evaluation because they represent the most disadvantageous conditions: the opponent is aware of the agents strategy and uses that to harm the agent or further its own interests.

Against a best-responding opponent, approaches based on the

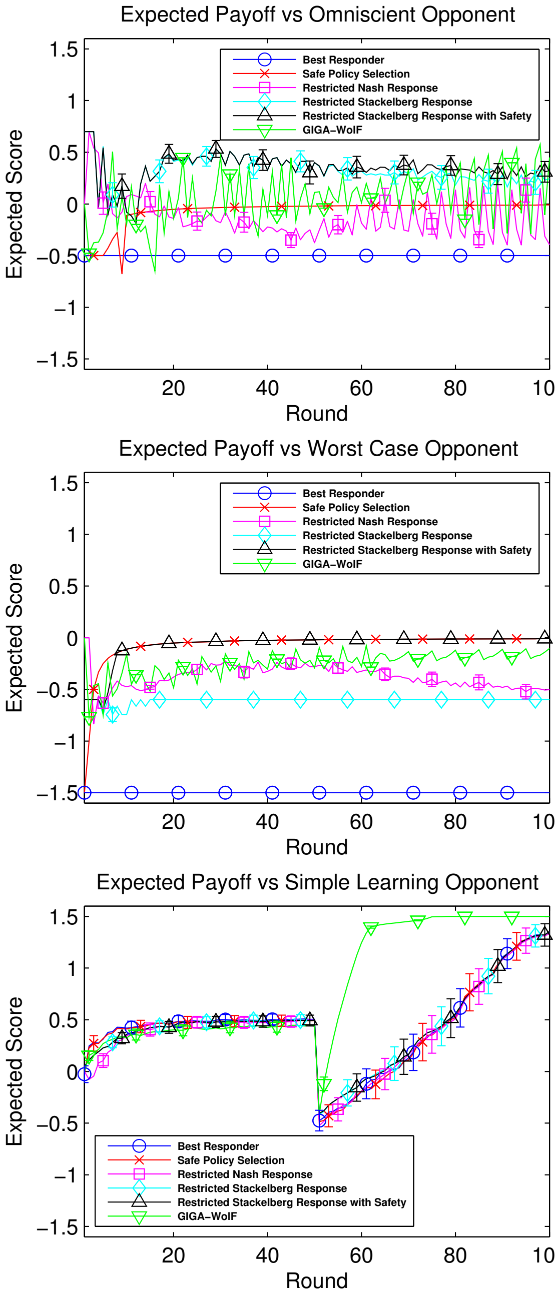

Figure 5: Expected payoff of a player playing different strategies in the game Rock/Spock/Paper/Lizard/Scissors against a best-responding (top), worst case (middle), and simple learning (bottom) opponent, over 100 games. Error bars show 95% confidence interval. Error bounds for the worst case opponent are so tight they are not easily visible.

Stackelberg response perform well. It takes 10-20 observations to learn that the opponent is best-responding, after which the agent takes advantage of that trait. SPS treats the situation as worst case and achieves the value of the game. GIGA-WolF is worse than RSRS and RSR, but still outperforms the safety value. RNR uses the same weight value as RSRS but achieves a worse outcome. Both RSR and RNR are trying to find a strategy which performs well when the opponent is best-responding, but RSR achieves a better outcome because it does not require its own strategy to be a best-response to the opponent strategy.

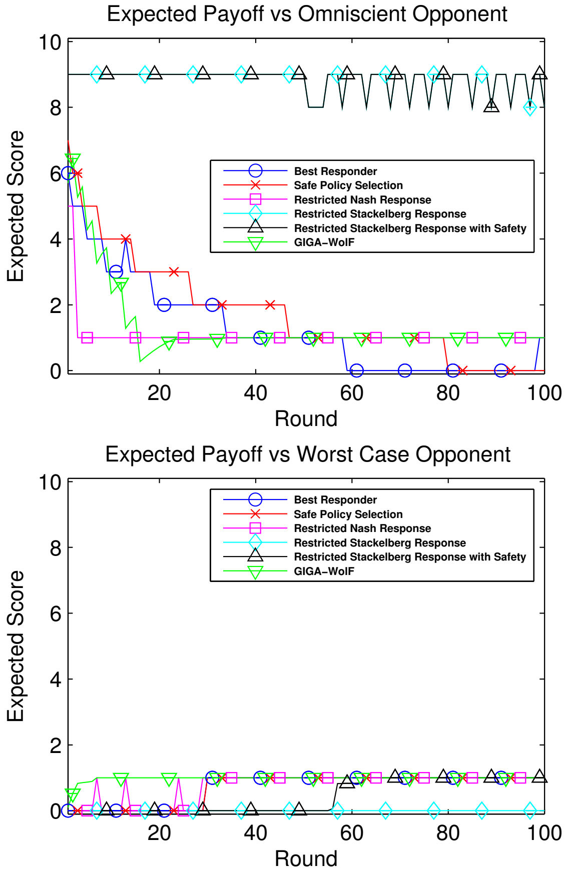

Figure 6: Expected payoff in Traveller's Dilemma against a best-responding (top) and worst case (bottom) opponent, over 100 games.

Approaches which include SPS quickly detect a worst-case opponent and play the maximin strategy (which is the best possible outcome in this situation). Of the agents that don't use SPS, GIGA-WolF approaches the safety value, and RNR does better than RSR (without safety) because assuming a best-responding opponent is inaccurate. The switching opponent reveals what happens when the predictor adjusts to a changing opponent more rapidly than the parameter values. 50 rounds of observations of successful play against the initial strategy builds up a very high risk factor, and a high certainty that the opponent is not best-responding. As a result strategies dependent on those parameters don't adjust to the new strategy until the predictor does. GIGA-WolF reacts faster to the switching opponent because it doesn't use fictitious play as a predictor.

Figure 6 shows the performance of best-responding, SPS, RNR, RSR, and RSRS in the game Traveller's Dilemma.

Traveller's Dilemma is a general-sum game in which both players choose a payoff from 1 to 10. Each player receives the lowest payoff, and if one player chose a lower payoff than the other, that player receives a bonus of 1, while the other receives a penalty of 1. The Nash equilibrium of Travellers Dilemma is for both players to chose the minimum payoff, but the social welfare maximizing strategy is for them both to chose the maximum payoff.

Against an omniscient best-responding opponent methods which use the Stackelberg equilibrium perform well, SPS, GIGA-WolF, and simple best-response gradually converge to the Nash equilibrium, and unsurprisingly, RNR rapidly arrives at the Nash equilibrium. Against the worst case opponent, all agents, except RSR, eventually arrive at the Nash equilibrium (which is also the minimax strategy).

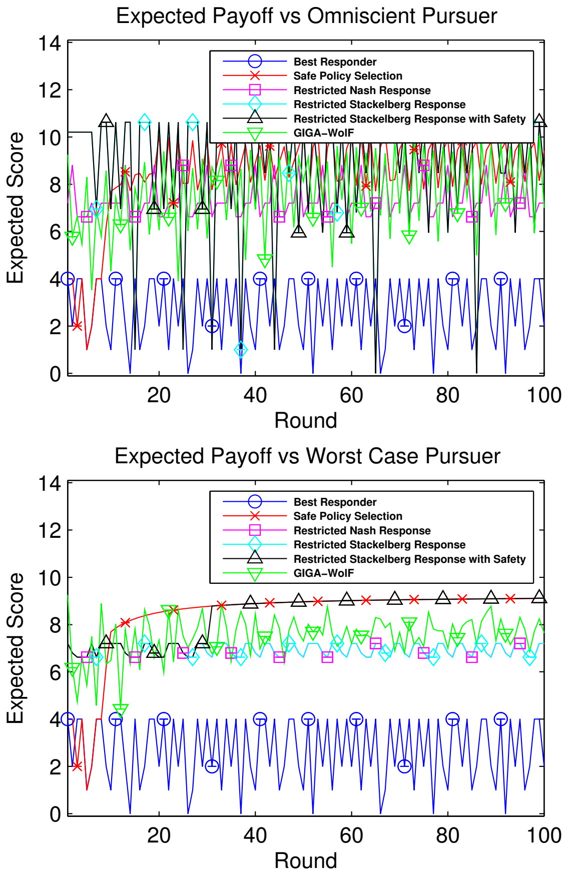

Figure 7: Expected payoff of the evader in the pursuit/evasion game against a best-responding (top) and worst case (bottom) opponent, over 100 games.

Figures 7 and 8 show the performance of best-responding, GIGA-WolF, SPS, RNR, RSR, and RSRS in a simple pursuit/evasion game. There are 4 locations, with associated payoffs from 0 to 4. Each player receives the payoff of the location they chose. In addition the pursuer receives a payoff of 10 for choosing the same location as the evader, while the evader receives a payoff of 10 for choosing a different location from the pursuer. The Nash equilibrium of the game is $(.075, .175, .275, .475)$ for the pursuer (checking preferred locations more often), and $(.425, .325, .225, .025)$ for the evader (staying away from the preferred locations).

An evader which best-responds to its prediction does particularly poorly regardless of the opponent. For all agents there is considerable instability because minor changes in the prediction can result in relatively large changes in the response. Against a worst case opponent, agents which use a risk factor parameter perform best. Note that RSRS is slower to reach the minimax strategy because its performance prior to reaching that strategy is better, so it takes longer for the initial risk factor to deplete. A pursuer is best-off using RSR or RSRS against a best-responding opponent. Against a worst-case opponent, RSRS and SPS are the only strategies which find the minimax solution.

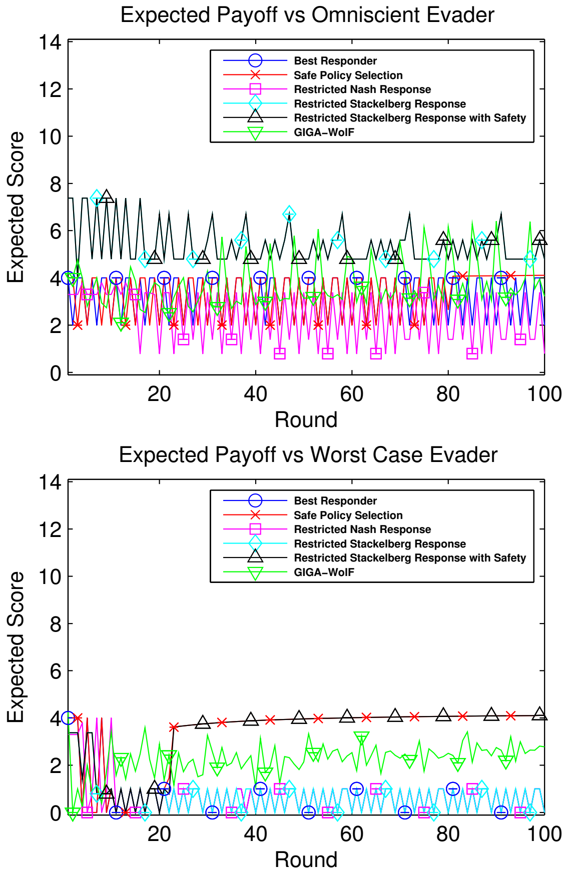

Figure 8: Expected payoff of the pursuer in the pursuit/evasion game against a best-responding (top) and worst case (bottom) opponent, over 100 games.

8. CONCLUSIONS AND FUTURE WORK

We have presented RSRS, a new method for choosing a strategy in a general-sum normal form game. that takes advantage of a prediction of opponent behavior while guarding against exploitation, and shown experimentally that it is effective. RSRS provides a useful basis for acting on a prediction in a general-sum environment.

RSRS deals well with two dangerous opponent types when the prediction is inaccurate, but there are other possible opponents. For example, an opponent which is mildly hostile to the agent will restrict RSRS to the value of the game, but a better strategy could be to offer such an opponent a higher incentive to cooperate. Future work will explore a larger variety of opponents, making more general assumptions about their behavior.

Developing performance guarantees for the weight-learning algorithm is left for future work. Our results show that the weight-learning algorithm is effective, but a more solid theoretical basis is desirable. The algorithm used to assign risk factors for SPS is elegant and effective, and we would like to create a similar algorithm for assigning weight values.

REFERENCES

[1] N. Bard, M. Johanson, N. Burch, and M. Bowling. Online implicit agent modelling. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 255–262. International

Foundation for Autonomous Agents and Multiagent Systems, 2013.

[2] M. Bowling. Convergence and no-regret in multiagent learning. *Advances in neural information processing systems*, 17:209–216, 2005.

[3] M. Bowling and M. Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136:215–250, 2002.

[4] D. Chakraborty. Convergence, targeted optimality and safety in multiagent learning. In *Sample Efficient Multiagent Learning in the Presence of Markovian Agents*, pages 29–47. Springer, 2014.

[5] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proc. Conference on Electronic Commerce*, pages 82–90, 2006.

[6] V. Conitzer and T. Sandholm. AWESOME: A general multiagent learning algorithm that converges in self-play and learns a best response against stationary opponents. *Machine Learning*, 67(1–2):23–43, 2007.

[7] S. Damer and M. Gini. Achieving cooperation in a minimally constrained environment. In *Proc. of the Nat'l Conf. on Artificial Intelligence*, pages 57–62, July 2008.

[8] E. M. De Cote and N. Jennings. Planning against fictitious players in repeated normal form games. In *Proc. of the Ninth Int'l Conf. on Autonomous Agents and Multi-Agent Systems*, pages 1073–1080, 2010.

[9] D. Fudenberg and D. K. Levine. *The Theory of Learning in Games*. MIT Press, 1998.

[10] D. Fudenberg and J. Tirole. Game theory. 1991. *Cambridge, Massachusetts*, 393, 1991.

[11] S. Ganzfried and T. Sandholm. Safe opponent exploitation. *ACM Transactions on Economics and Computation*, 3(2):8:1–8:28, Apr. 2015.

[12] M. Johanson, M. Zinkevich, and M. Bowling. Computing robust counter-strategies. In *Advances in Neural Information Processing Systems (NIPS)*, pages 721–728, 2007.

[13] M. Littman and P. Stone. Leading best-response strategies in repeated games. In *IJCAI Workshop on Economic Agents, Models, and Mechanisms*, 2001.

[14] P. McCracken and M. Bowling. Safe strategies for agent modelling in games. In *AAAI Fall Symposium on Artificial Multi-agent Learning*, Oct. 2004.

[15] J. Pita, M. Jain, M. Tambe, F. Ordóñez, and S. Kraus. Robust solutions to Stackelberg games: addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence*, 174(15):1142–1171, 2010.

[16] R. Powers, Y. Shoham, and T. Vu. A general criterion and an algorithmic framework for learning in multi-agent systems. *Machine Learning*, 67(1–2):45–76, 2007.

[17] Z. Wang, A. Boularias, K. Mülling, and J. Peters. Balancing safety and exploitability in opponent modeling. In *Proc. of the Nat'l Conf. on Artificial Intelligence*, pages 1515–1520. AAAI Press, Aug. 2011.