A Model of Adaptive Reinforcement Learning[*]

March, 2019

Julian Romero[†] Yaroslav Rosokha[‡]

Abstract

We develop a model of learning that extends the classic models of reinforcement learning to a continuous, multidimensional strategy space. The model takes advantage of the recent approximation methods to tackle the curse of dimensionality inherent to a traditional discretization approach. Crucially, the model endogenously partitions strategies into sets of similar strategies, and allows agents to learn over these sets which speeds up the learning process. We provide an application of our model to predict which memory-1 mixed strategies will be played in the indefinitely repeated Prisoner's Dilemma game. We show that despite allowing the mixed strategies, strategies close to the pure strategies always defect, grim trigger, and tit-for-tat emerge – a result that qualitatively matches recent strategy choice experiments with human subjects.

Keywords: Reinforcement Learning, Repeated-game Strategies, Repeated Prisoner's Dilemma, Mixed Strategies, Agent-based Models, Markov Strategies

[†] Eller College of Managment, Univeristy of Arizona • Email:jnromero@gmail.com

[‡] Krannert School of Management, Purdue University • Email:yrosokha@purdue.edu

[*] This paper benefited greatly from discussions with and comments from workshop participants at the 2018 International CEF meetings in Milan and seminar participants at Purdue University.

1 Introduction

Reinforcement learning has a long history as a model of behavior in social sciences and economics. Indeed, early formulations of the underlying principle, known as the "Law of Effect," go back to Thorndike (1898). It states that probability of implementing a strategy will increase (decrease) with the success (failure) of that strategy. While early reinforcement learning models were formalized in the cognitive psychology literature (e.g., Bush and Mosteller, 1955), they have gained traction in economics by the 1990s when several variants of reinforcement learning models were applied to explain behavior in individual choice (Arthur, 1991) as well as strategic games experiments (Mookherjee and Sopher, 1994; Roth and Erev, 1995; Mookherjee and Sopher, 1997; Erev and Roth, 1998).[1] In addition, a more general experience weighted attraction (EWA) model was proposed by Camerer and Ho (1999) which combines reinforcement and belief learning models.

One of the major shortcomings of the reinforcement and EWA models as applied in the early papers, is that agents learned about stage-game strategies.[2] Because of this, more complex behaviors exhibited in human subject experiments, such as tit-for-tat in repeated Prisoner's Dilemma or alternations in the Battle of Sexes were hard to achieve. To overcome this issue several papers took an approach in which subjects reinforced behavioral rules and simple repeated-game strategies instead of stage-game strategies. In particular, Stahl (1996, 1999) propose a general framework in which behavioral rules are reinforced based on their performance. The authors show that level-k type behavioral rules are good at explaining the data from human subject experiments (Stahl, 2000; Haruvy and Stahl, 2012). In the same spirit, Hanaki, Sethi, Erev, and Peterhansl (2005) propose to reinforce repeated-game strategies specified by two-state automata and Ioannou and Romero (2014a,b) generalize it to belief learning frameworks, including EWA.[3] One difficulty of learning over behavioral rules and repeated game strategies is that the set of possible strategies is often large or infinite. The above papers have solved this problem by selecting a small number of reasonable strategies as the learning domain. In this paper we propose a model that endogenously partitions strategies into sets of similar strategies, and agents learn over these sets. This allows us to considerably expand the set of strategies in the learning domain (including multidimensional, continuous strategy sets).

We extend the reinforcement learning literature in economics in two important ways. First, agents in our model learn about sets of strategies rather than individual strategies. That is, we partition the set of strategies into subsets and agents learn the "propensities" of those subsets through a reinforcement process. This allows us to work with large and continuous strategy spaces, rather than having to predefine a finite set of strategies as an input to the learning process. Second,

[1] It is interesting to note that some of the very first economics experiments were done by trained psychologist Sydney Siegel (Smith, 2017). Siegel (1961) used a reinforcement learning formulation of Bush and Mosteller (1955) as a model of human behavior.

[2] For example, Camerer and Ho (1999) note that "Incorporating a richer specification of strategies is important because stage-game strategies are not always the most natural candidates for the strategies that players learn about."

[3] The approach of learning about repeated-game strategies rather than stage game strategies has also been used in combination with evolutionary learning models such as genetic algorithm (e.g., Romero and Rosokha, 2019).

the partition is endogenously refined over time so as to provide more "resolution" in the neighborhood of the strategy space in which the propensities change the most. Specifically, we adopt a recent method on function approximation from the machine learning literature called adaptive tile coding (Whiteson, Taylor, Stone, et al., 2007). Combined, the two developments allow the agents to learn over a continuous, multidimensional set of Markov strategies and overcome the curse of dimensionality inherent to the traditional discretization approach.[4] In addition, the two components of the model provide distinct ways in which agent sophistication can be incorporated into the learning process. In particular, sophistication can be captured by the precision with which propensity differences are evaluated or by the number of partition refinements that agents make. While the former is common to quantal response models (e.g., McKelvey and Palfrey, 1995), the latter is a new development which maps to agent's ability to differentiate between strategies in the strategy space.

We use our model to study learning in the indefinitely repeated Prisoner's Dilemma game. In particular, agents learn over the infinite set of memory-1 Markov strategies. A strategy in this set can be represented as a five-dimensional vector with entries corresponding to the probability of cooperation in the first period and after each of the four possible memory-1 histories. Many learning models have studied mixed-strategies in games with a unique mixed-strategy Nash equilibrium (e.g., Mookherjee and Sopher, 1994; Erev and Roth, 1998; Selten and Chmura, 2008). These papers look at action-learning models and show that these models match experimental choice frequencies. However, not as much attention has been devoted to learning over repeated game mixed-strategies, which have been investigated theoretically and empirically. In particular, Ely and Välimäki (2002); Bhaskar, Mailath, and Morris (2008) show the existence of memory-1 Markov Perfect Equilibria in mixed strategies that cover the range of all folk theorem payoffs. Recent work by Breitmoser (2015) shows that human subjects play mixed strategies in the indefinitely repeated prisoner's dilemma. In this paper, our goal is to develop a learning model to handle continuous sets of repeated-game strategies like those mentioned above.

Simulations from our model lead to two main results that relate to regularities observed in human subject experiments. The first result is that agents in our simulations learn to cooperate if they are sufficiently sophisticated. This result relates to several studies that have documented a positive effect of intelligence on cooperation in repeated Prisoner's Dilemma. For example, Jones (2008) survey 36 studies of repeated Prisoner's Dilemma ran between 1959 and 2003 and find that students in universities with higher average SAT scores cooperate more often. In a recent experiment, Proto, Rustichini, and Sofianos (2017) split subjects in groups based on a measure of intelligence and find that groups with higher intelligence are more likely to cooperate. The second result is that agents learn to play strategies that are close to always defect (ALLD), grim trigger (GRIM), and tit-for-tat (TFT). These three strategies are central to the experimental literature on indefinitely repeated Prisoner's Dilemma games (Dal Bó and Fréchette, 2018). The second

[4] For example, consider the spaces of memory-1 mixed strategies which can be represented as a 5-dimensional cube. Even a relatively coarse discretization of 10 points in each dimension would lead to 100,000 strategies.

result is particularly striking because the three strategies arise despite allowing for a continuous, multidimensional set of repeated-game mixed strategies.

The rest of the paper is organized as follows: In Section 2, we provide details of the adaptive reinforcement learning model and present an illustrative example using a two dimensional set of strategies, which nests ALLD and GRIM. Then, in Section 3, we present our main simulation results using memory-1 Markov strategies. Finally, in Section 4, we provide concluding remarks and discuss avenues for future research.

2 Adaptive Reinforcement Learning

There are two main components of the adaptive reinforcement learning model. The first component (Section 2.1) is that agents learn values (i.e., propensities or attractions) corresponding to a set of strategies rather than values corresponding to individual strategies as done in the prior literature (e.g., Roth and Erev, 1995). In particular, the set of all strategies is partitioned into "tiles," where a *tile* refers to one subset from the partition. The agent will not discriminate among the strategies within a tile, rather any strategy chosen from the tile will lead to reinforcement of that tile. Thus, the agent will approximate the value of playing a particular strategy based on the value of the tile that it falls into. The second component (Section 2.2) specifies how the partition changes over time. Specifically, we build on a recent approximation method called adaptive tile coding (Whiteson, Taylor, Stone, et al., 2007). The method begins with a coarse partition which contains only one tile (i.e., the whole strategy set). As the agent learns, the partition is refined over time. The key questions of the refinement process are *when* and *where* to refine the partition. The broad answers to the two questions are – the partition will be refined when the learning with that partition has converged and the refinement will be chosen as to maximize the improvement in the value function. We augment the algorithm by adding a sensitivity parameter, which leads to a more precise refinement of the partition.

2.1 Reinforcement Learning Over Sets of Strategies

The reinforcement learning aspect of the model relies on three elements:

- **Strategy Set Partition**. Let S be the set of strategies that an agent considers, P be the partition of that set, and $T \in P$ be a tile (element of the partition). Instead of individual strategies being assigned a value, all strategies in the same tile of the partition will have a common value. Thus, an agent will keep track of the tile values $V(T)\ \forall T \in P$. Prior to any interaction taking place we will initialize all values to be $V_0(T) = \underline{\mathrm{v}}\ \forall T \in P$, where $\underline{\mathrm{v}}$ is the lowest possible payoff from the stage game for that agent.

- **Strategy Selection**. Before each supergame begins, an agent picks a strategy $s \in S$ to be played in that supergame. The strategy is chosen in two steps. First, a tile T is chosen based on the tile values and the choice function. Specifically, a tile will be selected using the *logit choice function* (also know as Boltzman or softmax):

$$p_t(T) = e^{\lambda^s V_t(T)} / \sum_{t \in P} e^{\lambda^s V(t)},$$

where λ^s is the precision parameter capturing sensitivity to the payoffs. In particular, as λ^s increases the agent will be more likely to select the tile with the higher value. Second, a strategy s will be selected uniformly from the chosen tile so as to provide an approximation of the average value of the tile. The advantage of this approach is that all strategies in the set have some likelihood of being chosen.[5]

- **Reinforcement**. The last element is the reinforcement (or updating) rule. We choose an exponential recency-weighted average rule. Specifically, suppose in supergame t, player i plays $s \in T$ and receives a per period payoff of x, then the tile values will be updated according to

$$V_{t+1}(T) = (1 - \alpha)V_t(T) + \alpha x \tag{1}$$

where α determines forgetting. The value of all other tiles will remain the same from supergame t to $t + 1$.

Note that if the strategy space is discrete and the partition is such that there is only one strategy in each tile, then the above model reduces to the reinforcement learning over strategies - a model similar to Hanaki, Sethi, Erev, and Peterhansl (2005).

2.2 Adaptive Refinement of Sets of Strategies

The agent learns with a given partition until the values of the tiles have converged. Once converged, the agent will refine the partition by selecting one of the tiles to be split. To help with the refinement process the method maintains and reinforces *subtiles*. Importantly, subtiles are not used in the strategy selection process. Instead, subtiles keep track of which neighborhood of the strategy space may benefit from refinement. Next we describe three elements of the refinement process:

- **Convergence Criteria.** We use the convergence heuristic proposed by Whiteson, Taylor, Stone, et al. (2007). Every time a tile is updated, there is a change in the value of that tile, $\Delta(T) = |V_{t+1}(T) - V_t(T)|$. Let $\Delta_{min}(T)$ be the smallest change that has occurred on this tile since the last refinement. The heuristic keeps track of the number of supergames since $\Delta_{min}(T)$ was updated for any of the tiles. If u consecutive supergames fail to produce a change to $\Delta_{min}(T)$ then the learning has converged. The heuristic works because once the

[5] An alternative approach would be to choose strategies from the vertices of the tile. The advantage of this approach is that it speeds up the simulations because some vertices are picked multiple times and hence the value of their interaction can be stored for future use.

learning starts leveling off it will take longer and longer between updates to $\Delta_{min}(T)$.[6]

- **Refinement Selection.** At any given point in the simulation we will have a partition of the strategy space which consists of tiles. For each tile in the partition, we will maintain one subtiling for each dimension of the strategy space. A subtiling is a partition that divides the parent tile in half on one dimension, and consists of two subtiles. Values of subtiles are initialized and updated in the same way as values of tiles. After the convergence criteria has been satisfied (on the values of the tiles, not the subtiles), we find the subtiling the has the largest difference in value between the two subtiles.[7] We then select the tile containing this subtiling and the dimension corresponding to this subtiling for refinement. Thus, the agent will devote more "resolution" to the region of the strategy space where V changes the most.

- **Refinement Precision.** A novel feature that we add to the adaptive tile coding is that we allow the selected tile to be split anywhere along the chosen dimension based on the difference of values of the two subtiles. In particular, we rely on the logit choice function with precision parameter λ^r, to guide the split process. In particular, if the subtiles have values st_1 and st_2 the tile will be split in proportion $e^{\lambda^r st_2} : e^{\lambda^r st_1}$. For example, if $\lambda^r > 0$ then the agent will split closer to the boundary corresponding to the subtile with the higher value. Alternatively, if $\lambda^r = 0$, then the agent will split in the middle, corresponding to the rule in Whiteson, Taylor, Stone, et al. (2007). Thus λ^r captures agent's sensitivity to the payoff during the refinement step, which is similar to the role of λ^s for the strategy selection step.

2.3 2D Example

To demonstrate the model on an example, we will simulate a population of 50 agents that use adaptive reinforcement learning with $\lambda^s = 1.50$ and $\lambda^r = 0.15$. The agents play the indefinitely repeated Prisoner's Dilemma with continuation probability $\delta = .95$ and the stage game payoffs presented in Figure 1a. For this example, we will restrict our attention to a set of Markov strategies of the form $\sigma = (p_1, p_1, p_2, p_2, p_2)$, where $p_1, p_2 \in [0, 1]$. That is, we consider a two-dimensional strategy set:

- **Dimension 1**: probability of cooperation in the first period and after CC.

- **Dimension 2**: probability of cooperation after CD, DC, and DD.

We chose this example for several reasons: First and foremost, the set of strategies is two dimensional which allows us to visualize the partition and the learned values. Second, the set is

[6] As an example of this heuristic consider learning about the mean of a standard normal distribution starting with an initial guess of -5. Each iteration we randomly draw from this distribution and update our guess. In the beginning, when we randomly draw from the distribution, Δ_{min} will likely be large because most draws will have values greater than -5. However, as our estimates get closer to 0, Δ_{min} will get smaller and it will take longer to achieve new Δ_{min}.

[7] We only consider tiles that have been visited a minimum number of times (30 in our simulations). This ensures that differences between values of subtiles are based on large enough sample. As a consequences, this means that tiles that are not visited due to low values are not likely to be refined.

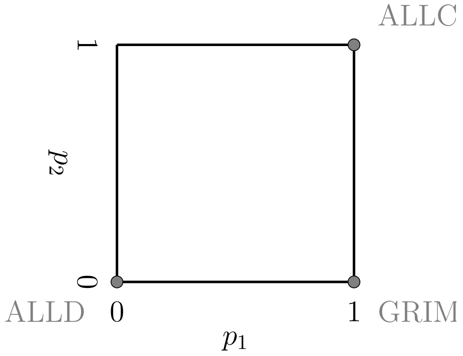
continuous and, therefore, is not a trivial application of the existing models. Third, the set nests ALLD and GRIM (see Figure 1b) – strategies that have been central to game theoretic analysis of the indefinitely repeated games. Finally, the chosen parameters have been used in recent strategy choice experiments by Romero and Rosokha (2018) in which majority of human subjects learned to cooperate.

Figure 1: 2D Example Setup

(a) Stage-game Payoffs

	C	D
C	32,32	12,50
D	50,12	25,25

(b) Strategy Set



Notes: **(a)** Stage-game payoff parameters. Probability of continuation is set to $\delta = 0.95$. **(b)** The unit rectangle represents the 2D set of strategies of the form $\sigma = (p_1, p_1, p_2, p_2, p_2)$, where $p_1, p_2 \in [0, 1]$. Three strategies are marked with gray dots are always defect (ALLD), grim trigger (GRIM), and always cooperate (ALLC).

Figure 2 presents the first three refinement steps of the adaptive reinforcement learning process for one agent in our simulations. The first step involves initialization of the partition, P_0, and the two subtilings (one in each dimension).[8] Notice that as a starting point we chose the most general case in which the partition consists of only one tile that encompasses the whole set of strategies. Using the initial partition we proceed with learning over sets of strategies as described in Section 2.1. Specifically, the agent begins by selecting a strategy for supergame 1. For this example, the randomly chosen strategy is $(0.73, 0.16)$ and the average per period payoff from playing this strategy against one other randomly selected agent is 29.4. In the figure, we mark the selected strategy with a black dot. The payoff is then used to reinforce the value of the tile from which the strategy was selected. In addition, the value of the subtiles that strategy falls into are reinforced. At this point, the convergence criteria is checked, and since it is not satisfied, the learning proceeds to supergame 2.

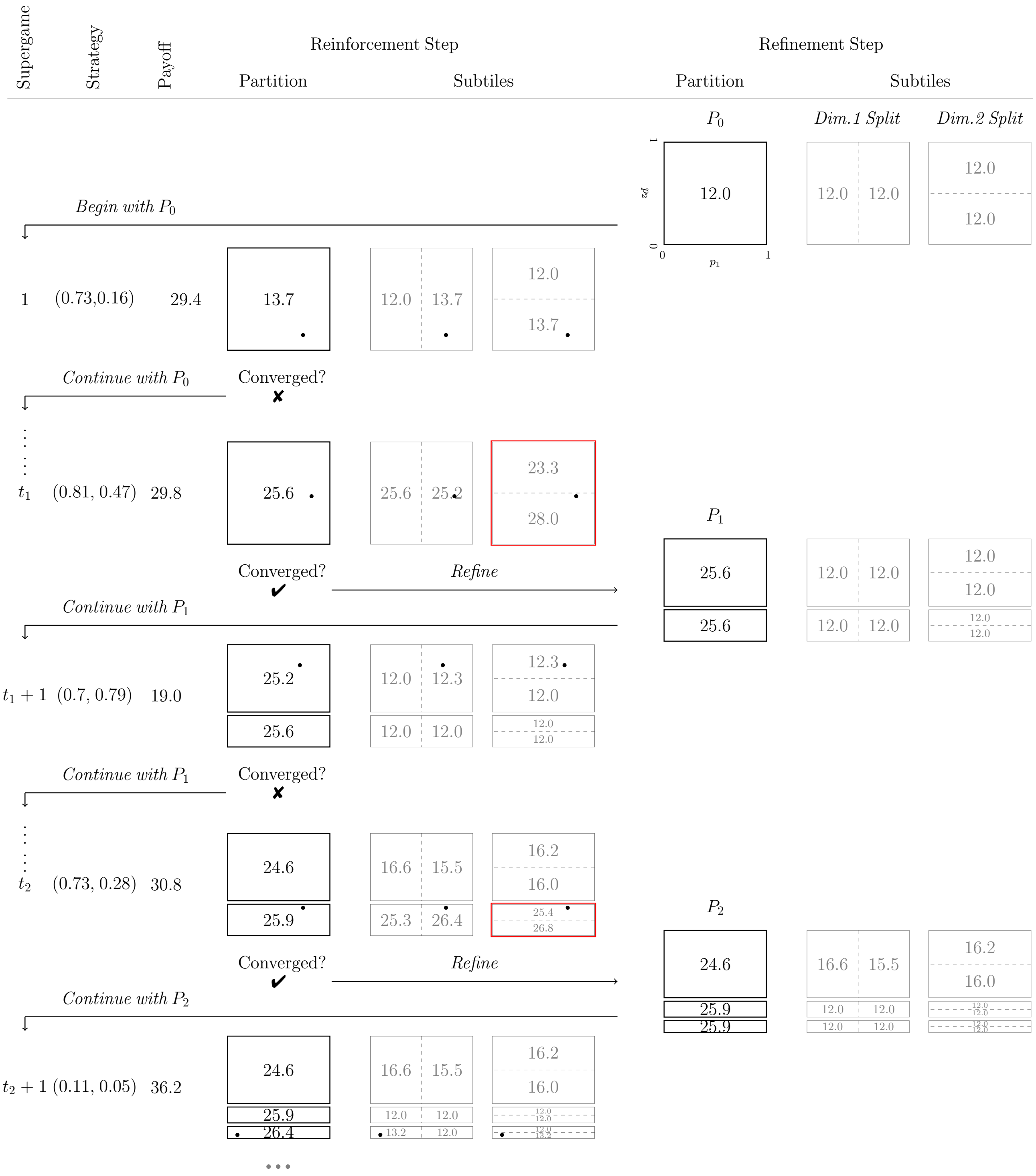
The agent continues with partition P_0 until the learned value has converged (supergame t_1). Once converged, the decision needs to be made as to where to refine the partition. To determine which tile will be split we will use the learned values of the subtiles. In the example, the learned values of the subtiles in dimension 1 are 25.6 and 25.2; and the learned values of the subtiles in

[8] In general, for k tiles there are $2 \times k \times d$ subtiles. Thus, although the number of subtiles increases in the number of dimensions, the increase is not exponential.

dimension 2 are 23.3 and 28.0. Since the difference in the latter subtiling is greater, the agent will split the tile along dimension 2 (highlighted in red). Once the tile and the dimension have been chosen, the agent needs to determine the precision of the split. In our implementation, the split is determined by the refinement precision parameter λ^r described in Section 2.2. Specifically, for this example, $\lambda^r = 0.15$ which leads to $e^{(0.15*23.3)} : e^{(0.15*28.0)}$, which is roughly $\frac{1}{3} : \frac{2}{3}$ split.

There are a couple of additional points worth noting regarding the split. First, during a split, the values of the newly created tiles are set to be the value of the tile that was split, and the values of all other tiles a kept the same. For example, after the initial split, the value of both tiles in P_1 are set to be 25.6, which was the value of the tile in P_0. This leads to the two tiles having equal likelihood of being selected during the strategy selection step for supergame $t_1 + 1$. Second, values of newly created subtiles are set to the lowest possible payoff for the stage game for that agent, while the values of all other subtiles stay the same. For example, after the second split, values of subtiles corresponding to the tile that wasn't split remained the same, but the values of the new tiles were initialized to 12.0. Thus, the tile values are used only during the reinforcement step of the model, while the subtile values are used only during the refinement step of the model.

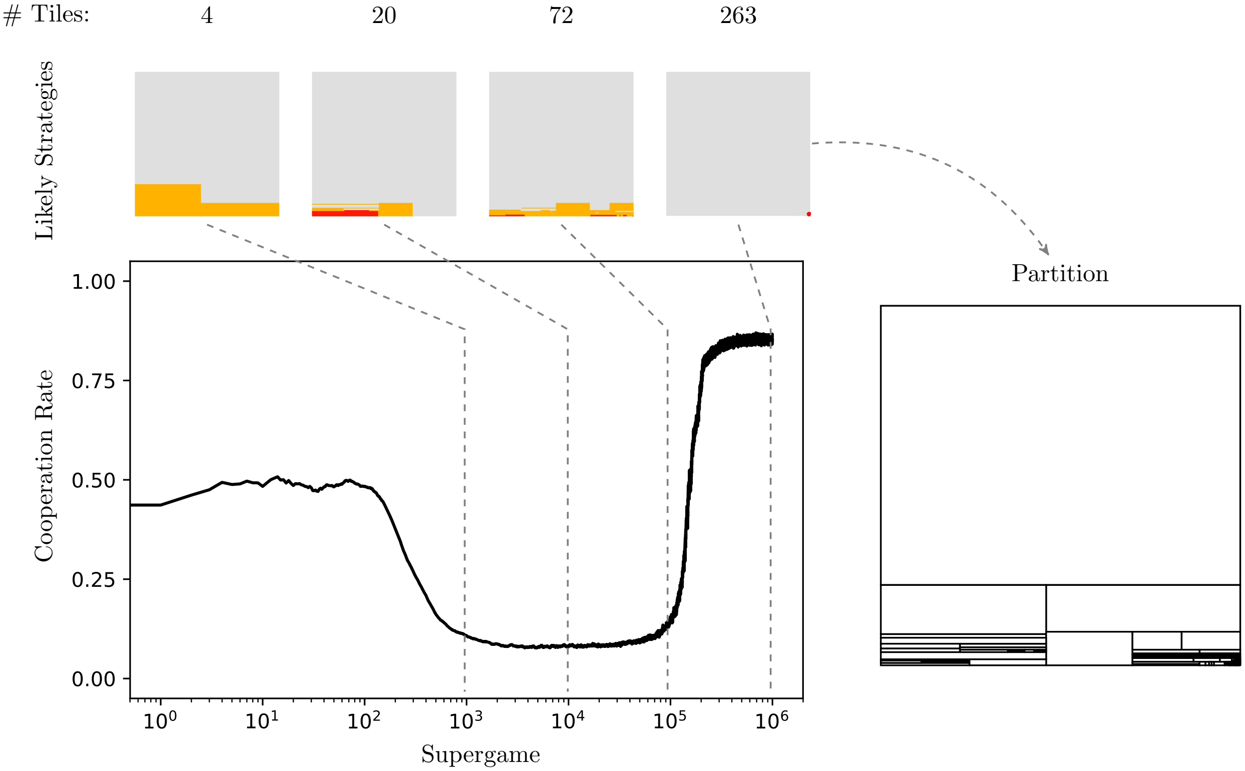
Figure 2: 2D Example: Step-by-step



Notes: An example of the first two refinements. Tiles and their values are in black. Subtiles and their values are in gray. Strategies chosen for each supergame are marked with black dots within the tiles and subtiles. The chosen split at each refinement step is highlighted in red.

Figure 3 presents evolution of cooperation and four examples of the learned partitions throughout the simulations. In terms of cooperation, agents initially start out by playing randomly, but then quickly learn to defect after CD, DC, and DD histories. This results in the low cooperation around supergame 10^3. What is striking is that the low cooperation is achieved with only 4 tiles. As the learning progresses agents learn to play strategies closer to ALLD. This can be seen in the figure as the most likely strategies (highlighted in red) for supergame 10^4 are close to the bottom-left corner of the strategy space, which correspond to ALLD. Notice that at this point, the partition consists of 20 tiles, with much finer tiles around ALLD and along the ALLD-GRIM (bottom) edge. Once agents have learned to defect, they slowly start to transition to playing strategies closer to GRIM. This can be seen in the figure as the most likely strategies around supergame 10^5 are mixed strategies along the ALLD-GRIM (bottom) edge. Finally, by supergame 10^6 subjects have learned to cooperate by playing strategies close to GRIM.

Figure 3: 2D Example: Evolution of Cooperation



Notes: **Top:** Number of tiles in the partition. **Middle:** Strategies that are likely to be chosen. Tiles in the top 50 percentile are highlighted in red. Next 40% are highlighted in orange. Bottom 10% of tiles are in gray. **Bottom:** Average cooperation rate throughout the simulation horizon. **Right:** Partition after 10^6 supergames. The unit rectangle represents the 2D set of strategies of the form $\sigma = (p_1, p_1, p_2, p_2, p_2)$, where $p_1, p_2 \in [0, 1]$.

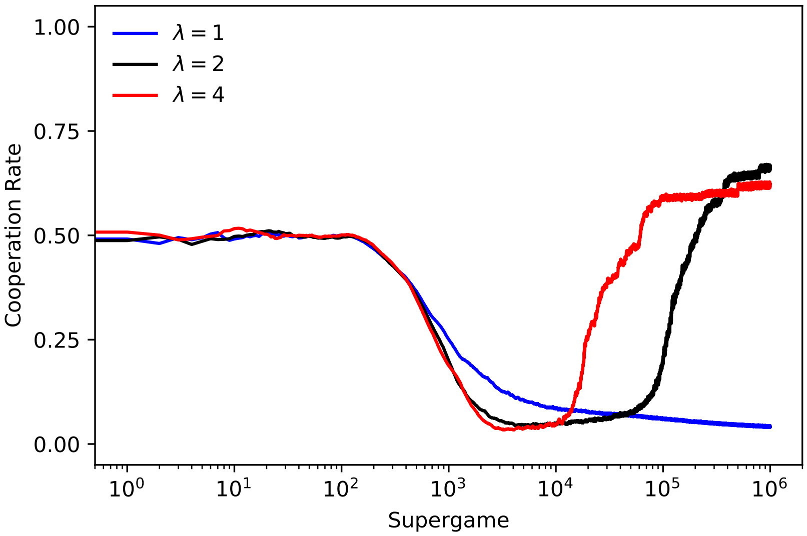
Right panel of Figure 3 shows the final partition after 1 million supergames for one of the agents in our simulation. At this point of the learning process the partition contains 263 tiles. This partition contains parts that are very fine and others that are very coarse. The bottom left and bottom right corners of the strategy space are both very fine as those correspond to commonly studied strategies ALLD and GRIM. The top of the strategy space is very coarse (the top 75% is a single tile). This area contains strategies that cooperate with high probability after CD and DD, which means that they are easily exploitable, and will likely lead to a low payoff, and hence a low value for the corresponding tile. Since this tile has a low value and will not be visited a lot, is is unlikely to be refined (due to the minimum number of visits requirement). The tile in the bottom right near GRIM has an approximate height of .002 and approximate width of .001. To achieve such a fine partition with a discrete grid would require about 1000 points in one dimension and 500 in the other, which translates into 500,000 strategies and makes the learning process prohibitively slow.

3 Results

We apply adaptive reinforcement learning to the indefinitely repeated Prisoner's Dilemma game. In particular, we present the simulation results when the set of strategies consists of all memory-1 Markov strategies. This set of strategies can be represented as a 5-dimensional cube with each element represented by a vector of the form $\sigma = (p_1, p_2, p_3, p_4, p_5)$, where $p_i \in [0, 1]$. Our focus is two-fold: First we are interested in whether agents can learn to cooperate. And second, we are interested in which among the Markov strategies are being played.

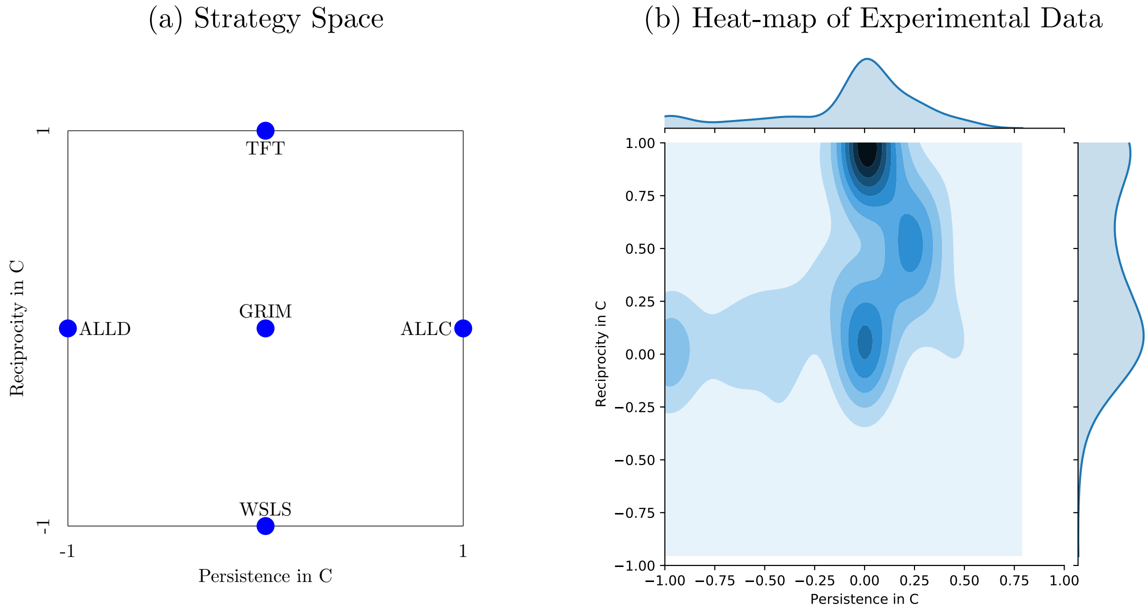
Figure 4 presents the evolution of cooperation for different values of $\lambda^s = \lambda^r = \lambda$. We find that as λ increases the agents in our model are more likely to discover cooperation. Notably, parameter λ captures the level of sophistication of the agent. Which, in our context, would mean that agents with higher sophistication learn to cooperate, while agents with lower sophistication learn to defect. This results relates to prior findings from the experimental literature (e.g., Proto et al. 2018) who find that subjects with higher IQ are more likely to cooperate.

Figure 4: Evolution of Cooperation with Markov strategies



Next we turn to strategies. We present the resulting distribution of strategies on the following domain: *persistence-in-C* x *reciprocity-in-C*. Where, persistence-in-C is measured as the difference between fraction of the time C is played after CC and fraction of the time D is played after CD. That is, we find the likelihood that a subject will maintain mutual cooperation less the likelihood that a subject will retaliate for defection. Reciprocity-in-C is measured as difference between the fraction of time C is played after DC and the fraction of time C is played after DD. That is, we find the likelihood that a subject will reciprocate with cooperative behavior less the likelihood that a subject will try to cooperate after mutual defection. We chose this domain because the key strategies are distinct: ALLC is at (1,0); ALLD is at (-1,0); TFT is at (0,1); GRIM is at (0,0); and finally, WSLS is at (0,-1). Figure 5a presents the strategy space while Figure 5b presents the strategies obtained from a recent experiment by Romero and Rosokha (2018) projected on the same space.

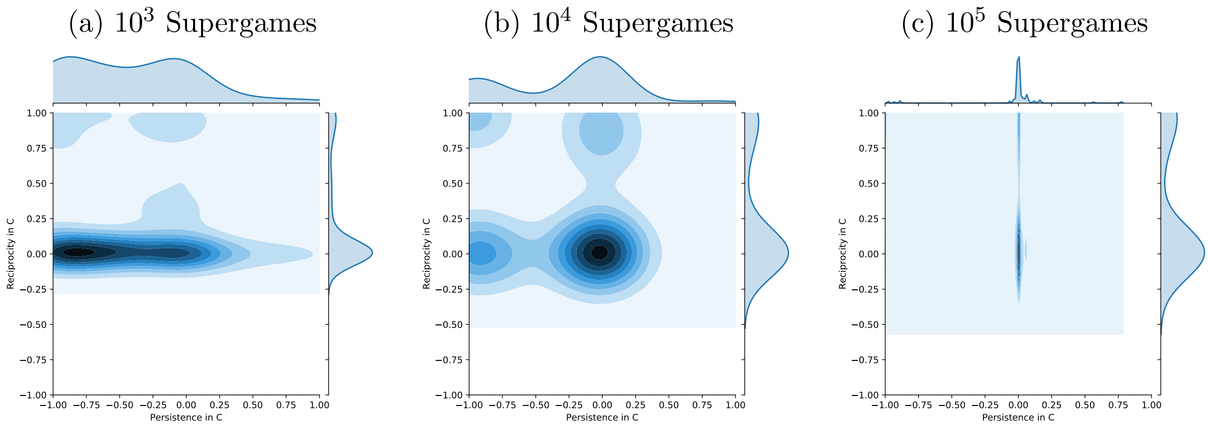
Figure 5: Strategies in Experimental Data



Notes: **(a)** Strategy space. For strategy $s = (p_1, p_2, p_3, p_4, p_5)$, Persistence in C is equal to $p_2 - (1 - p_3)$ and Reciprocity in C is equal to $p_4 - p_5$. **(b)** Experimental data is from Romero and Rosokha (2018).

Figure 6 presents strategies that are played by agents in our simulations at three different time points. In particular, we present a heatmap of strategies played at 10^3, 10^4, and 10^5 supergames. We find that early on agents are more likely to play strategy close to ALLD, but later on strategies that are most commonly played are close to GRIM and TFT.

Figure 6: Strategies During Simulations



4 Conclusion

The two central ideas of the adaptive reinforcement learning model presented in this paper are i) learning over sets of strategies rather than individual strategies; and ii) using a function approximation method to refine the set of strategies over time. In particular, the combination of the two allows us to tackle learning with continuous, multidimensional strategy sets, such as memory-1 Markov strategies that have been of recent theoretical and experimental interest. We find that agents learn to cooperate if they are sufficiently sophisticated. In terms of strategies: the three strategies that regularly appear throughout the learning process are always defect, grim-trigger, and tit-for-tat. These simulation results are qualitatively consistent with recent strategy choice experiments.

There several interesting directions for future work. First, it would be interesting apply the current model to different games. In particular, there are many games with continuous strategy spaces, such as the Cournot duopoly model, that would be interesting to explore. Second, a question remains as to how an agent may choose dimensions of the strategy space to speed up the learning process. Third, while in this paper we considered the case of continuous sets, it would be interesting to extend this approach to large but finite strategy sets, such as memory-2+ forgiving strategies studied in Fudenberg, Rand, and Dreber (2012). In particular the challenge with strategies that condition on longer histories is how to split the sets and how to choose dimensions on which to split.

References

ARTHUR, W. B. (1991): "Designing economic agents that act like human agents: A behavioral approach to bounded rationality," *The American Economic Review*, 81(2), 353–359.

BHASKAR, V., G. J. MAILATH, AND S. MORRIS (2008): "Purification in the infinitely-repeated prisoners' dilemma," *Review of Economic Dynamics*, 11(3), 515–528.

BREITMOSER, Y. (2015): "Cooperation, but no reciprocity: Individual strategies in the repeated Prisoner's Dilemma," *The American Economic Review*, 105(9), 2882–2910.

BUSH, R. R., AND F. MOSTELLER (1955): *Stochastic models for learning.* John Wiley & Sons, Inc.

CAMERER, C., AND T.-H. HO (1999): "Experience-weighted attraction learning in normal form games," *Econometrica*, 67(4), 827–874.

DAL BÓ, P., AND G. R. FRÉCHETTE (2018): "On the determinants of cooperation in infinitely repeated games: A survey," *Journal of Economic Literature*, 56(1), 60–114.

ELY, J. C., AND J. VÄLIMÄKI (2002): "A robust folk theorem for the prisoner's dilemma," *Journal of Economic Theory*, 102(1), 84–105.

EREV, I., AND A. E. ROTH (1998): "Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria," *American economic review*, pp. 848–881.

FUDENBERG, D., D. G. RAND, AND A. DREBER (2012): "Slow to Anger and Fast to Forgive: Cooperation in an Uncertain World," *American Economic Review*, 102(2), 720–49.

HANAKI, N., R. SETHI, I. EREV, AND A. PETERHANSL (2005): "Learning strategies," *Journal of Economic Behavior & Organization*, 56(4), 523–542.

HARUVY, E., AND D. O. STAHL (2012): "Between-game rule learning in dissimilar symmetric normal-form games," *Games and Economic Behavior*, 74(1), 208–221.

IOANNOU, C. A., AND J. ROMERO (2014a): "A generalized approach to belief learning in repeated games," *Games and Economic Behavior*, 87, 178–203.

——— (2014b): "A generalized approach to belief learning in repeated games," *Games and Economic Behavior*, 87, 178–203.

JONES, G. (2008): "Are smarter groups more cooperative? Evidence from prisoner's dilemma experiments, 19592003," *Journal of Economic Behavior & Organization*, 68(3), 489–497.

MCKELVEY, R. D., AND T. R. PALFREY (1995): "Quantal response equilibria for normal form games," *Games and economic behavior*, 10(1), 6–38.

MOOKHERJEE, D., AND B. SOPHER (1994): "Learning behavior in an experimental matching pennies game," *Games and Economic Behavior*, 7(1), 62–91.

——— (1997): "Learning and decision costs in experimental constant sum games," *Games and Economic Behavior*, 19(1), 97–132.

Proto, E., A. Rustichini, and A. Sofianos (2017): “Intelligence, Personality and Gains from Cooperation in Repeated Interactions,” SSRN Scholarly Paper ID 2871144, Social Science Research Network, Rochester, NY.

Romero, J., and Y. Rosokha (2018): “Mixed Strategies in the Indefinitely Repeated Prisoner’s Dilemma,” *Available at SSRN 3290732*.

——— (2019): “The Evolution of Cooperation: The Role of Costly Strategy Adjustments,” *American Economic Journal: Microeconomics*, 11(1), 299–328.

Roth, A. E., and I. Erev (1995): “Learning in extensive-form games: Experimental data and simple dynamic models in the intermediate term,” *Games and economic behavior*, 8(1), 164–212.

Selten, R., and T. Chmura (2008): “Stationary concepts for experimental 2x2-games,” *American Economic Review*, 98(3), 938–66.

Siegel, S. (1961): “Decision making and learning under varying conditions of reinforcement,” *Annals of the New York Academy of Sciences*, 89(5), 766–783.

Smith, V. (2017): “Tribute to Sidney Siegel (1916-1961): A Founder of Experimental Economics,” *Southern Economic Journal*, 83(3), 664–667.

Stahl, D. O. (1996): “Boundedly rational rule learning in a guessing game,” *Games and Economic Behavior*, 16(2), 303–330.

——— (1999): “Evidence based rules and learning in symmetric normal-form games,” *International Journal of Game Theory*, 28(1), 111–130.

——— (2000): “Rule learning in symmetric normal-form games: theory and evidence,” *Games and Economic Behavior*, 32(1), 105–138.

Thorndike, E. L. (1898): “Animal intelligence: An experimental study of the associative processes in animals.,” *The Psychological Review: Monograph Supplements*, 2(4), i.

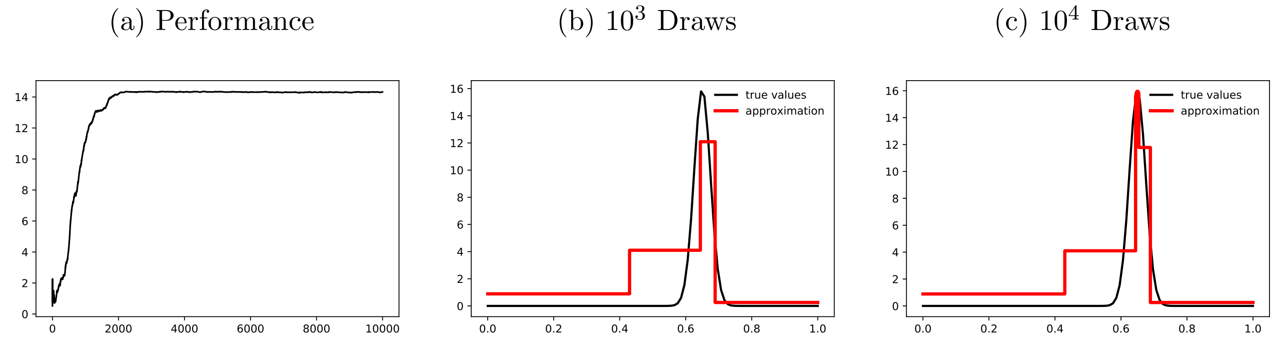
Whiteson, S., M. E. Taylor, P. Stone, et al. (2007): *Adaptive tile coding for value function approximation*. Computer Science Department, University of Texas at Austin.

Appendices

Appendix A Algorithm Test

To demonstrate the algorithm in a non-strategic domain, we apply it to learning the maximum of a normal distribution with a mean 0.65 and a standard deviation of 0.025. Figure A-1 presents the results.

Figure A-1: Learning maximum of a function with non-boundary Max



(a) Performance

(b) 10^3 Draws

(c) 10^4 Draws

Notes: Normal distribution with mean of 0.65 and standard deviation of 0.025. Maximum value is 15.96. **(a)** Running average of draw performance. **(b)** Function approximation after 10^3 draws. **(c)** Function approximation after 10^4 draws.

We find that agents accurately discover the maximum of the function after approximately 2000 draws. This can be seen from Figure A-1a as the average performance plateaus near the true value. The performance is below the maximum of the function due to the logit tile selection. Figure A-1b presents the learned values and the resulting approximation after 1000 draws. We find that even with few partitions the algorithm does well in picking up which strategies are likely to yield the highest payoffs. Figure A-1c presents the learned values and the resulting approximation after 10,000 draws. We find that the improvement in approximation, but more importantly the maximum value is discovered.