Using Iterated Reasoning to Predict Opponent Strategies

Michael Wunder
Rutgers University
mwunder@cs.rutgers.edu

Michael Kaisers
Maastricht University
michael.kaisers@maastrichtuniversity.nl

John Robert Yaros
Rutgers University
yaros@cs.rutgers.edu

Michael Littman
Rutgers University
mlittman@cs.rutgers.edu

ABSTRACT

The field of multiagent decision making is extending its tools from classical game theory by embracing reinforcement learning, statistical analysis, and opponent modeling. For example, behavioral economists conclude from experimental results that people act according to levels of reasoning that form a "cognitive hierarchy" of strategies, rather than merely following the hyper-rational Nash equilibrium solution concept. This paper expands this model of the iterative reasoning process by widening the notion of a level within the hierarchy from one single strategy to a distribution over strategies, leading to a more general framework of multiagent decision making. It provides a measure of sophistication for strategies and can serve as a guide for designing good strategies for multiagent games, drawing it's main strength from predicting opponent strategies.

We apply these lessons to the recently introduced Lemonade-stand Game, a simple setting that includes both collaborative and competitive elements, where an agent's score is critically dependent on its responsiveness to opponent behavior. The opening moves are significant to the end result and simple heuristics have achieved faster cooperation than intricate learning schemes. Using results from the past two real-world tournaments, we show how the submitted entries fit naturally into our model and explain why the top agents were successful.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems

Keywords

Iterated reasoning, cognitive models, multiagent systems, POMDPs, repeated games

1. INTRODUCTION

In many domains where multiple strategic actors are present, it is becoming increasingly common to find computer programs in place of human decision-makers. Algorithmic trading [14], automated ad auctions [12], and botnets [7] are just a few examples of the multiagent problem that have emerged over the past decade. The key challenge of such settings is the deliberate unpredictability of other adaptive agents that can prevent the formation of reliable responses. On the other hand, if others are trying to predict us, there is an opportunity to discover the pattern by which they attempt to do so. Multiagent learning has been motivated by successes in machine learning and several branches of economics to answer the question of how computer agents should make decisions when multiple decision makers are present that may not have the same goals or incentives [13]. The task at hand is actually two separate but related tasks: to *predict* the behavior of other unknown players, and to *respond* in turn. Unlike the single agent case, here agent designers need to recognize that other modelers are changing and attempting to anticipate their agent's actions.

One popular approach to building intelligent agents is to apply reinforcement-learning techniques adapted from single agent environments. Often, for learning to make speedy progress, algorithm designers rely on assumptions about opponents that are not always explicit, and we would like to have a way to explore them and understand how they arise as they do. We might notice that multiagent learning in multiple round games raises similar questions to those of reinforcement learning. Players need to learn how to act in the long-run, how to escape from undesirable locally optimal outcomes, and they need to learn quickly. One difference is that the issue of time can have a big impact on the eventual result of a game with multiple agents, while learners in fixed single-agent environments will typically reach the same policy regardless of the pace of experience. For example, in a game where two players get a high reward for cooperating with each other at the expense of a third, it pays off to be one of the first two cooperators—the third agent may never achieve high reward.

Another option is to model opponent behavior directly, by using recursive modeling [10], Interactive POMDPs [9], or Networks of Influence Diagrams [8]. The famous RoShamBo game is one domain where recursive reasoning has demonstrated its relevance and applicability [1, 6]. The obstacle that disrupts progress in this area is that modeling can go on endlessly, as an agent forms ever more complicated models using simpler models as parts, often in a rather unstructured way. Behavioral economists address very similar issues from a slightly different angle. Using experiments on humans playing games, they have found a great deal of evidence that people use strategic reasoning to make decisions, but only up to a point. Indeed, this reasoning conforms to a well-defined cognitive hierarchy, or a related level-k model,

Cite as: Using Iterated Reasoning to Predict Opponent Strategies, Michael Wunder, Michael Kaisers, John Robert Yaros, Michael Littman, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. 593-600.

composed of levels of thinking [2]. This model can apply to games with two agents or larger population games.

One limitation of previous iterated best response models is that there is a tendency to pick an exact strategy to represent each level. Assuming that agents can be classified as one of a few ideal types is one option for modeling opponents, but does not capture the aspect that players can belong to multiple types. We address the opponent model selection problem by allowing for a distribution of agents to represent each level. The appeal of using distributions is that uncertainty over opponents leads to multiple best responses, and sometimes there is no principled way to choose from among them. We can use this feature to cover uncertainties about implementations or simplified approximate versions of optimal strategies, which play a role in bounded reasoning models. Our enriched model, called a Parameterized I-POMDP, highlights to the user the most important strategies of the game, and identifies their relationships. Given a feasible unknown strategy to test, the framework allows us to directly measure the amount of reasoning that lies behind it by comparing it to constructed strategies derived from a thorough reasoning process.

To illustrate this process, we utilize the recently introduced Lemonade-stand Game (LG) as an example setting where playing one's opponents is more important than playing the game. LG is played by three players, which is more complicated than the simplest 2-player case, but still small enough where the pairwise interactions are major factors. This simple game leads to an elegant analysis, even with the complications of triadic interaction. The main message of our framework is that learning agents can use a number of ways to plan against opponents, but in the end success depends mostly on the distribution of types in the population. The model guides theoretical analysis of the game and its application is demonstrated with actual agents from competitions. Such competitions have a history of focusing researchers on important issues and providing a wide selection of approaches that can be mined for data. This method of mining data to discover aspects of the underlying reasoning model is an exciting emerging branch of computer science [15, 16].

The next section, 2, provides more detail about existing models in both computer science and economics. In Section 3, we introduce our proposed extension to those earlier models. Section 4 explains the LG. Section 5 applies our framework to LG and derives the resulting levels, resulting in a hierarchy over the space of reasonable strategies. Section 6 uses previous tournament submissions as evidence that the new model works in some interesting types of games.

2. BACKGROUND

The cognitive hierarchy model (CH) [3] and its cousin, level-k thinking [4], have been used by behavioral economists to explain observed human behavior. CH consists of an initial level of base strategies combined with a series of levels found by repeatedly taking the best response of lower levels. The level-k model operates by responding to just level $k-1$ instead of levels $0, ..., k-1$. In these investigations, the games are generally simple enough that it is straightfoward to construct the hierarchy. Experimental data then provides knowledge about the frequencies of the various levels, and therefore properties like the average level in a population. CHs are useful in population games or 2-agent games alike, but usually they consist of one-shot experiments, obviating the need to build complex sequential models at each level. While we will primarily consider games played through computer agents and not directly by people, the same underlying process is present in both systems.

From the computer-science or machine-learning perspective, this setting has been formalized as an Interactive Partially Observable Markov Decision Process, or I-POMDP. This development synthesizes the considerable work done on single agent POMDPs with multiagent approaches such as the Recursive Modeling Method (RMM) [11]. This formulation is ideal for sequential or repeated games where unknown opponents have limited reasoning capabilities.

POMDPs are similar to the standard Markov Decision Process except it is not assumed that an agent knows what state it is currently in. A solver must use observations to infer the likely state by updating beliefs over the state space. An I-POMDP is a POMDP that has interactive states in place of states, and joint actions in place of actions [9]. This interactive state is the cross product of environmental states and internal states of agents present in the game. The interactive state space is constructed recursively starting where other agents are represented strictly as a stochastic part of the state. In other words, in the simplest interactive state other agents are assumed to have no reasoning capacity or sensitivity to payoffs, but instead exist as a noisy component of the environment. Then, we build more advanced interactive states in new I-POMDPs to represent further or higher opponent reasoning. We can use this technique to reach any level of sophistication that can be reasonably computed, but in practice only a finite number of nested levels are used.

Associate $\mathsf{I\text{-}POMDP}_i$ with agent i and the only other agent is j. The definition generalizes to more agents. An $\mathsf{I\text{-}POMDP}_i = \langle IS_i, A, T_i, \Omega_i, O_i, R_i \rangle$ has the following features:

- IS_i is the set of interactive states $IS_i = S \times \pi_j$ where S is the set of states from the environment and π_j is the set of policies for agent j.
- A is the set of joint actions $A_i \times A_j$.
- T_i is the transition function $T_i : S \times A \times S \to [0, 1]$. The transition model, along with the internal decisions for policy π_j, determine the next interactive state, but we assume that agent i does not directly control other agents in its environment.
- Ω_i is the set of observations.
- O_i is the observation function $O_i : S \times A \times \Omega \to [0, 1]$.
- R_i is the reward function $R_i : IS_i \times A \to R$.

Policies at each level k are derived from the beliefs $b_{j,k-1}$ over the policies and states of the previous level $k-1$. Define the following spaces.

- $IS_i^0 = S, \qquad \pi_j^0 = IS_i^0 \to A_j \in H_0$
- $IS_i^1 = IS_i^0 \times \pi_j^0, \qquad \pi_j^1 = b_{j,1}(IS_i^1) \to A_j \in H_1$
  .
  .
  .
- $IS_i^L = IS_i^{L-1} \times \pi_j^{L-1}, \qquad \pi_j^L = b_{j,L}(IS_i^L) \to A_j \in H_L$

The strength of this formalism is that it suggests a relatively general algorithm for computing policies and works well with good initial beliefs. A weakness is that the set of opponent policies to include is unspecified and therefore the solution breaks down when those beliefs do not match reality. We attempt to mitigate this flaw by keeping a range of solutions to represent our initial uncertainty. The framework also has the advantage that the solver has some flexibility in selecting the planning problem to attack, which could allow it to select simpler to reduce the computationally demands. Currently there is no predescribed way to achieve this aim, but it is another goal for our extended model.

3. PARAMETERIZED I-POMDPS

Our framework, entitled Parameterized Interactive Partially Observable Markov Decision Processes (PI-POMDPs) is a model for recursively deriving a set of policies that respond to less advanced policies for use in highly structured domains. We extend the I-POMDP framework by building an entire profile of policies at each nested level in place of a single solution. Instead of a single policy, the solution will be the hierarchy of policies computed at each level.

Define for agent i rule-based policy $H : IS \rightarrow A_i$ to be a basic rule that maps states to actions. One example of a rule would be $A_i^t = A_i^{t-1}$, signifying constant action. Then, a parameterized policy $\pi : \mathbf{R} \rightarrow H$ maps real vector $X \in [0, 1]$ to some rule. X could indeed be used to represent any adjustable feature of an agent, but we will assume that X_r is the probability of playing rule H_r. Note the rule is not fixed for the whole game, but rechosen every time step.

The parameterization of policies begins right away when deciding which beliefs over initial strategies to start with. There may be several options that incorporate the idea of non-reasoning policies, so we end up with $\pi^0(X)$ for agents at level 0 to weight each tactic. In turn, following the I-POMDP mechanism, the $\pi_j(X)$s for each agent j and value X are used to construct an instance of a POMDP problem. Instead of optimizing over all X to arrive at a single policy, compute a range of policies as X changes. If possible, we will attempt to condense all of these rules into a single new parameterized policy π^1 with as few input dimensions as possible, to represent the result of a step of reasoning over level 0. This way, we do not have to make the decision about which strategies are valid for the next level derivation. All of them are kept as a part of the final model. While it is possible for games to take place in states in the environment, we will consider the partial observability to consist of uncertainty over the parameterized policies present in the agent's population.

4. LEMONADE-STAND GAME

Recently, the Lemonade-stand Game was introduced to demonstrate the interaction complexity that can arise in a game from simple rules [17]. The game is played by three lemonade vendors on a circular island with n beaches, where typically $n = 12$, arranged like the numbers on a clock. Each morning, the vendors have to set up on one of the beaches, not knowing where the other vendors will show up. Assuming the beach visitors are uniformly distributed and buy their lemonade from the closest vendor, the payoff for the day is equal to the distance to the neighboring lemonade vendors. For convenience denote $D(A_i, A_j)$ the distance function between agents i and j on the side with no other agent in between. Then, $R_0^t = \sum_{j=1}^{2} D(A_0^t, A_j^t)$.

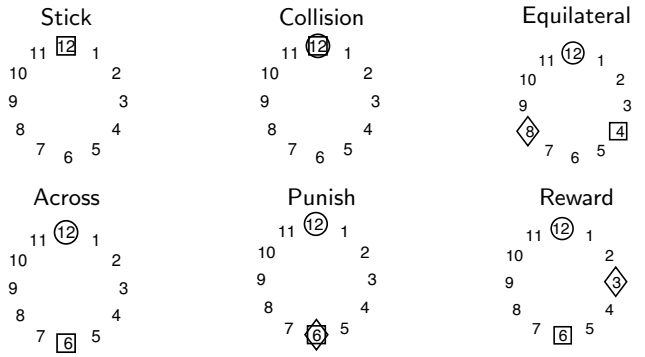

Figure 1: This figure depicts key strategic patterns of the lemonade game. Each of the six diagrams refers to a (partial) joint action, and similarly a strategic move by □, expecting opponents to play ◇ and ○. As the domain is on a ring, the patterns are rotation insensitive.

In game-theoretic terms, LG is a 12-action normal form game on a ring, where the payoff function equals the sum of distances to the right and left neighboring vendor. As a corollary, the cumulative payoff of the three players is 24. The only exceptional formations are when multiple agents conflict by choosing the same action (Collision). If two vendors choose the same action, they receive a reward of 6 and create the most favorable condition for the third agent who receives the maximum of 12. If all three vendors choose the same action, each receives 8. There is no special property about any of the 12 locations on Lemonade Island. The game is played repeatedly for T days and the joint action is observable. T is set to 100 so that agents can learn about the opponents' behavior from previous rounds.

The dynamics of this game are particularly interesting because it involves a sense of competition, as the gains of one always have to be compensated by the loss of others, as well as a sense of cooperation, because two agents can coordinate a joint attack on the third. Figure 1 shows an overview of the key strategic patterns in the LG. Each agent has to choose an action, and the simplest move is to stick with the initial action from then on (Stick). The Equilateral pattern splits the payoff evenly into 8 for each agent, but from worst case perspective is dominated by the cooperative action Across. Once two agents coordinate on the action Across, they will share 18, relegating the third agent to 6 regardless of the action it chooses. As an illustration of its simplicity, □ must only find a predictable player ○ and use the action opposite to it. ○ can be completely oblivious as long as it is predictable (say, a pure Stick player).

If an agent finds its opponents in a consistent Across pattern, it will lose unless it can entice at least one opponent to break formation. In a simple form illustrated in Figure 1, bottom right, ◇ can alternate between using the same action as □ and an action halfway between ○ and □. □ will get the same utility whether it is Across from ◇ or ○ during the Reward phase of ◇, but would choose Across from ◇ if it wants to avoid low utility during its looming Punish phase, essentially switching partners.

5. LEVELS OF REASONING IN LG

The Lemonade-stand game is an ideal example of competitive collaboration. That is, a player able to convince another player to cooperate with it can achieve a higher average score to the disadvantage of the third player. Of course, each player has to choose the "friendlier" player to cooperate with, with the knowledge that any attempts may be tracked by the other players. Ultimately, the two players who work together best will achieve the highest scores.

It appears that players have many repeated turns for observation and experimenting. In reality many matches are settled in the first several rounds, as agents seek partners and mutual history is established. Cooperation, however it is defined, is self-reinforcing. Therefore, strategies in this game put a premium on speed over data collection when finding optimal actions. This property means that traditional learning methods, like gradient ascent or regret matching tend to be outperformed by very simple rules. Because there are many possible Nash equilibria in the game, it is also unclear which ones are optimal and how to reach them. Our aim is a model that can explain such phenomena and yield strategies that at least outperform the simplest heuristics. An alternative approach [5] to this game identifies a stable equilibrium and classifies agents as leaders or followers according to who initiates the equilibrium pattern. While this strategy works in some scenarios, in some cases it is possible to identify several levels of leading and following. It also makes no judgments about whether one is superior to the other, or how one might measure that performance.

5.1 Long-run Optimal Behavior

LG translates into our PI-POMDP model, with several simplifications. In this case, $\Omega \in A$, so that $O : IS \times A \times A \rightarrow [0, 1]$. In addition, there is only one state in the environment, which means there are only pseudo-states depending on the agents' behavior that is conditioned on the current A^t.

- IS_i is the set of interactive states $IS_i = S \times \pi_j$ where S is the set of states from the environment and π_j is the set of policies for agent j.
- S is defined by the time step in the finite horizon case.
- $A_i = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11\}$.
- $H_{\textbf{Stick}}$: A basic action type that repeats the same action as the last turn
- $H_{\textbf{Uniform}}$: Play a random action.
- $\pi_j^0(X_0) = a \in \{H_{stick}, H_{uniform}\}$ s.t. $P(a = H_{stick}) = X_0, P(a = H_{uniform}) = 1 - X_0$.
- Ω_i is equivalent to $\bigcup A_j \forall_j$.
- O_i is determined by the policy of the opposite agent.
- R_i is the sum of distances to the players on either side.

To analyze this game rigorously using a PI-POMDP, we begin like most iterated reasoning models with a base level of non-reasoners, called Level 0 or $L0$. Analogously to an inductive proof, L0 forms the basis for the rest of the hierarchy. First, these base strategies are defined, and subsequently, the higher layers can be constructed by iteratively applying the reasoning step. Here, we define a step of reasoning to be a policy that maximizes the score against either a distribution over previous levels, or a selection of agents from those levels, by solving the POMDP formed by them. To avoid losing information, a parameterized policy (responding to a previous distribution of lower-level policies) represents the next level.

In many games, a base strategy of a single uniform distribution over all actions suffices: H_j^{Uniform}. In repeated games like LG, there exists another trivial action Stick, which leads to the basic notion of sticky strategies. Stickiness, as measured by the likelihood that a player remains in place, plays an important role in this game because it makes action prediction simple. As such, the rule $H_j^{\mathsf{Stick}} : A_j^t = A_j^{t-1}$ deserves a place among base strategies. In typical constant-sum games, a non-changing strategy is easy to defeat. In LG it is a powerful strategy on its own, as it forces other players to take action beneficial to the sticky player, such as to move away from it.

We take the general $\pi_j^0(X)$ base level L0 to be composed of H_j^{Uniform} and H_j^{Stick}, with a single real parameter X_0 to control the relative frequency of each. Consider the two opponents to be named B_k and C_k, where k is the amount of reasoning the agent's strategy contains. The solving agent's perspective is denoted agent I. The L0 strategy for B_0 is defined by an initial random action and the probability X_0 to Stick with the previous action in the following turns, or otherwise pick a new random action. Y_0 is the corresponding value for agent C_0. For $\pi_j^0(0) = H_j^{\mathsf{Uniform}}$ or $\pi_j^0(1) = H_j^{\mathsf{Stick}}$, L0 takes the form of a uniformly random (L0-U) or constant strategy (L0-C) respectively. We refer to this policy π_j^0 as $\pi_j^{\mathsf{Semi\text{-}random}}(X)$. Define $\hat{X}$ as the current estimate of X for an opponent B_0, and $\hat{Y}$ as the estimate of Y for opponent C_0, signifying the same strategy. In other words, given a sequence of observations from two unknown strategies implementing π^0 with values X and Y, we must make statistical conclusions about those values given our experience. We are then faced with finding a long-run strategy given observations $\hat{X}$ and $\hat{Y}$. Although a POMDP solver could be utilized to simulate the reasoning, here proofs are presented because the steps are very open to analysis.

THEOREM 1. *The optimal L1 strategy for agent I_1, π_I^1, is to maximize the distance D from the other two agents, giving $H_I^{\mathsf{Between\text{-}across}} = W_B \mathsf{Across}(A_B) + W_C \mathsf{Across}(A_C)$ where $W_B = \frac{X(1-Y)}{X(1-Y)+Y(1-X)}$ and $W_C = 1 - W_B$ are weights that determine how much $H_I^{\mathsf{Between\text{-}across}}$ should favor each of the Across actions. This strategy will prefer to be Across from the player who Sticks more often.*

PROOF. Since L0 agents do not respond to the actions of agent I_1, the POMDP reduces to a simple MDP. That is, action A_I has no effect on the transitions of the opponents, so the best action is found by calculating the expected utility of each spot, given X, Y, and the current placement of B_0 and C_0. If C_0 Sticks at location 0 and B_0 is random over all locations, the expected value of action a for $a > 0$ is:

$$V(a) = \frac{6}{12} + \frac{12}{12} + \frac{\max(0, a-1)}{12}\left(12 - \frac{a}{2}\right) + \frac{11-a}{12}\left(6 + \frac{a}{2}\right).$$

The first term is the event that B_0 lands on I_1. The second term is the event that B_0 lands on C_0. The third term is the event that B_0 lands in the short distance between C_0 and I_0, and the fourth term is the event of landing on the

large distance side. Taking the derivative, we then find that (when $n > 0$)

$$\begin{aligned} V(a) &= 6 + a - \frac{a^2}{12} \\ V'(a) &= 1 - \frac{a}{6} = 0 \\ a &= 6. \end{aligned}$$

The optimal action is 6, directly across from 0, where the expected value is $V(a) = 6+6-\frac{6^2}{12} = 12-3 = 9$. All actions on the far side of B_0 and C_0 have the same value when both players Stick or act uniformly, so we just care about the case when one of the two switches. Assume that B_0 is at 0 and C_0 is D_{BC} spaces away clockwise, where $D_{BC} \leq 6$ w.l.o.g. The value of action a when C_0 Sticks and B_0 is random is

$$\begin{aligned} V(a) &= 6 + a - D_{BC} - \frac{(a - D_{BC})^2}{12} \\ V'(a) &= 1 - \frac{a - D_{BC}}{6}. \end{aligned}$$

Therefore, the marginal value when B_0 Sticks with probability X and C_0 Sticks with probability Y is

$$V'(a) = \left(1 - \frac{a}{6}\right) X(1-Y) + \left(1 - \frac{a - D_{BC}}{6}\right) Y(1-X) = 0$$

$$a = \frac{6X(1-Y) + (D_{BC}+6)Y(1-X)}{X(1-Y) + Y(1-X)}.$$

Intuitively, the first term of the numerator weights the position directly across from B_0, and the second term does the same for C_0. Therefore the optimal action depends on the relative values of X and Y. □

The implication of this theorem is that an L1 strategy is predisposed toward choosing the action across from the more stable player. In general an agent observing that $X = Y$ causes the agent to always maximize its distance to the closest agent. Of course, in the initial rounds of a game, there are not enough observations to accurately forecast these unknowns. There are various ways to implement this policy, from the method of estimating $\hat{X}$ and $\hat{Y}$ to its reliance on priors of $\hat{X}$ and $\hat{Y}$. Assume $\hat{X}_0 = \frac{X_1 + c_{\mathsf{Stick}}}{2X_1 + c_{\mathsf{Total}}}$ where c_{Stick} is the number of Stick moves and c_{Total} is the current time steps. Here, the new parameter $X_L \in [0, \infty]$ represents the degree of attachment to the prior $\hat{X}_0 = \frac{1}{2}$, such that X_L^{-1} is the learning rate at which this estimate converges to the true value. With few observations, $\hat{X}_0$ will be noisy for low X_L (high learning rate). A high learning rate implies that if one player is constant but the other moves, this strategy will move sharply across from the constant player. When both players have been constant, W is undefined because the random (or half-random) cases does not occur, and therefore in that case there are a range of optimal actions. Since another feasible implementation is to assume both players are constant until there is contrary evidence, there is certainly room for parameterizing the preferred response in this case. However, given asymmetric behaviors, the theorem holds, where the constant player is the preferred partner.

For L2, we are looking for the best strategy given some combination of the first two levels, which is partially observable in the PI-POMDP. L2 optimizes against a distribution of L0s and L1s. The new PI-POMDP is therefore distributed across these two levels, as well as the range of parameters. We have already solved the exclusive L0 case, which will determine the default L2 behavior unless something close to L1 is observed.

When examining the rest of this PI-POMDP, this new type adds two elements to the policy calculation, which again depend on the parameters of the policy. First, the move-away-from-closest-player factor, represented by a slower learning rate and strong commitment to equal priors, exerts an influence on future levels to move directly across from the other player. Second, the punish-movers factor makes this movement less rewarding.

THEOREM 2. *Against* $H_{B_1}^{\textsf{Between-across}}$ *and* $H_{C_1}^{\textsf{Between-across}}$*, the optimal rule for agent* I_2 *is* $H_{I_2}^{\textsf{Stick}}$*.*

PROOF. (Sketch) With two L1s B_1 and C_1, each L1 is trying to move away from its two opponents. L1 is continually estimating $\hat{X}$ and using the estimate to adapt its strategy, which is to follow across from the other two agents, according to relative stickiness. An optimal rule here is just $H_I^{\textsf{Stick}}$ because B_1 prefers to move Across(I_2) over a moving agent, which C_1 certainly is. This tendency means that whenever B_1 registers a move by C_1, it moves a little farther from I_2. This new move then registers as a move for C_1, which in turn updates its action, and so on. This repetitive rule may reach oscillations, but the net effect will be to maneuver away from I_2, to the benefit of I_2. This policy is correct across the range of X_L. □

The interesting case is when the L2 player is up against one L0 and one L1 because essentially π_{C_0} "leads" and π_{B_1} "follows". The asymmetry of strategies allows for a new rule to emerge. In effect, π_{B_1} is constructed to move away from the semi-random π_{C_0}, but also from our agent I_2. We can use this tendency to our advantage in the best response. B_1 tends to play $H_{B_1}^{\textsf{Across}}$ from the C_0 with weighting $W_C = \frac{\hat{Y}_0(1-\hat{Z}_0)}{\hat{Y}_0(1-\hat{Z}_0)+\hat{Z}_0(1-\hat{Y}_0)}$ where $\hat{Z}_0$ is the staying probability of I_2. Thus, in that case we would hope to keep $\hat{Z}_0$ greater than $\hat{Y}_0$.

THEOREM 3. *Against* $H_{B_1}^{\textsf{Between-across}}$ *and* $\pi_{C_0}^{\textsf{Semi-random}}(Y)$*, the optimal rule for agent* I_2 *is* $H_I^{\textsf{Stick}}$ *until the number of moves of* $\pi_C^{\textsf{Semi-random}}(Y)$ *reaches a certain threshold* m*, and then to either move* Across(C_0) *if* C_0 *is too close or* Stick *as* (B_1) *moves closer to* Across(I_2)*.*

PROOF. (Sketch) We will consider the extreme cases where $Y_0 = 0$ or $Y_0 = 1$, and $X_L = 0$ or $X_L = \infty$. If $Y_0 = 0$, then $W_C \to 0$ when $\hat{Z}_0 > 0$ and $B_1 \to$ Across(I_2), regardless of the value of X_L. To accelerate this beneficial response, I_2 needs to Stick. If $Y_0 = 1$, then $H_{B_1}^{\textsf{Between-across}}(X_L)$ depends on the value of X_L. As $X_L \to 0$, B_1 is very sensitive to differences in moving probability. $W_C \to 1$ when $Z_0 < 1$ and $B_1 \to$ Across(C_0). To prevent this harmful response, I_2 should Stick as much as possible, but recognizing that the location of C_0 matters. It is preferable that C_0 be far from I_2 since B_1 will make room for it. In the worst case, if I_2 moves Across(C_0) it gets a minimum reward of 6 and expects a reward of 9, so that action is optimal if the current configuration gives a lower score. As $X_L \to 1$, B_1 retains more committment to its priors and has a high affinity for moving exactly in between I and C, unless a large difference $(\hat{Z}_0 - \hat{Y}_0)$ accrues. Therefore it is safer in that case for I_2 to move Across(C_0) if the learning rate X_L^{-1} is small. Therefore, depending on the relative values of X_L and Y_0, it may be optimal either to Stick or Across. □

We should note here that L2 can only classify opponents in one of two ways. The special case is L1 behavior, which is confined to a window of actions generally across from the two opposing players. The default classification is L0, which is defined by some combination of constant action and uniform random action. The significance of this simple modeling is that L2 would classify itself as L0, albeit with a high staying probability. Because L1 and L2 both have a tendency to move Across from the stickier players given sufficient information, this property will be selected at future levels. In fact, as more reasoning is applied, optimal strategies will start as constant for longer and longer as they attempt to out-wait earlier types. In those cases where all players have been constant from the beginning of the game, the decision about when to move is determined by the cost of remaining in the same location combined with the degree of reasoning ascribed to the opponents. In this case, the higher strategies are discouraged from moving at all due to this tendency to punish moving players. We can therefore consider the parameter X_2 to mean probability of moving into an Across position, especially when the current position is suboptimal.

The iterated best-response methods employed here do not necessarily adhere to the principle of auto-compatibility, whereby players do well against copies of themselves. Evolutionary strategy selection would pursue this goal more closely. A game with two of the same agent and one that is different would take on a new focus, where other forms of cooperation may be attainable that involve breaking the simple delayed across-move found by interated best response.

6. EXPERIMENTS

The levels of LG, while useful, are theoretical constructs. Nonetheless, the basic elements of this account arose in a group of agents developed independently. This section shows the viability of the level-based analysis by applying it to the two rounds of open LG competitions, one in Dec. 2009 and the other in Dec. 2010. The submitted strategies were a diverse collection. No two were alike and ranged from complete uniform action to near constant, to Across-seeking and initiating, and many in between.

To apply the model to real agents, we would like to classify each strategy by level or as a hybrid between levels. If our PI-POMDP model is a good fit for LG, populations consisting of agents that correspond to a similar mix of levels should behave, and score, in roughly the same way as their idealized counterparts. Since each level has its unique strengths and weaknesses, performance depends on the makeup of the population and specifically the relative frequency of each level. For the purposes of this paper, we classify a strategy by inspecting how it scores against idealized strategies from each of the levels we identified. See Figure 2 and Tables 1 and 2, right hand side, for these estimated levels. We ran the submitted agents against strategies over various values for the relevant parameters, such as $X_0, X_1, X_2 \in [0, 0.5, 0.75, 0.9, 0.95, 1.0]$. Using the derived strategies as benchmarks to compare to, we take the squared difference between unknown agent and level representative, and find the smallest difference between two adjacent scorings, say Level 2.95 and 2.975.

The rankings of the players in both tournaments provide a rough correlation to the amount of reasoning. The bottom half of the 2009 performers act like the base assumption strategies. The top half behave like those derived in the higher levels of the PI-POMDP model. From the 2010 dataset we find that on average reasoning has shifted up a level. Players identify the Across position as a goal state, but the top performers are more patient to get there, which implies more reasoning according to the model.

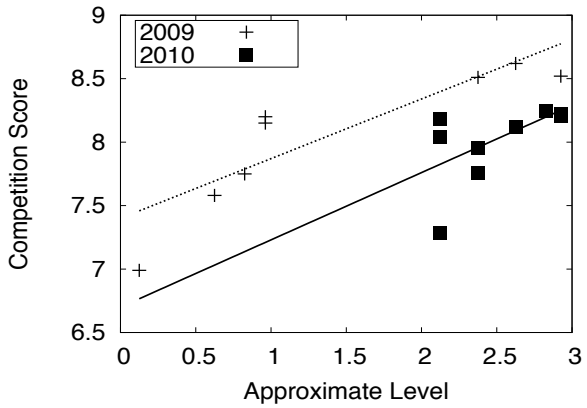

Figure 2: Estimated levels of competitors in two Lemonade-stand Game tournaments. Both sets of agents show positive correlation between reasoning and performance. R^2 values are 0.77 for 2009 and 0.34 for 2010. The more recent agents show a shift to higher reasoning levels, as well as a compression of scores.

Another prediction of this model is that agents that perform too much reasoning do less well than those that go just beyond the average level of the population. We ran tournaments with both sets of agents and two additional agents drawn from the model population. As Tables 1 and 2 show, in the 2009 competition a strategy that is close to the Level 1 ideal (but modified for fast Across) outperforms the rest, while a higher level strategy at Level 2.975 only gets to the middle of the pack. In the 2010 population, this ordering is reversed. Note that winning the competition is, in a sense, easy. Given our analysis, the only missing information is a guess of the average reasoning level of the population. Nevertheless, without access to the complete set of submitted agents, identifying the appropriate reasoning level is a serious challenge.

7. CONCLUSION

This article introduced a PI-POMDP analysis for repeated games and applied it to the Lemonade-stand Game competition. In the competition, simple heuristics outperformed intricate learning schemes, suggesting that PI-POMDP or CH analysis might be preferable to domain-general best responses in strategic interactions. The Lemonade-stand Game rewards strategies that trade off patient exploration for speed and commitment. Those participants who opt for too much exploration over model-based responses suffer against more carefully optimized strategies. The model demonstrates that players must employ some basic heuristics in the early stages of a game. If they do not, they risk getting classified as the less responsive, consistent, or coop-

Table 1: 2009 LSG Tournament results including two agents inspired by the PI-POMDP hierarchy (italicized). The winners are in bold. Level 0.83 would correspond to a player that Sticks with probability of 0.83, but random the rest of the time. An agent that would qualify as Level 2.63 would mean that a player Sticks when in an advantageous starting position. When its initial spot is less beneficial than it is constant with probability equal to 0.63, and the rest of the time moves Across from another player, preferring the more constant one. In cases where it is already Across from a player, it remains in place by choosing the same action.

Rank	Strategy (Affiliation)	Score	Error	Level	Parameterized Level
1.	*PI-POMDP Level 1.0 modified (New addition)*	8.72	± 0.0071	$L1$	1.00
2.	**EA^2 (Southampton/Imperial)**	8.56	± 0.0069	$L2$	2.63
3.	CoOpp (Rutgers)	8.51	± 0.0055	$L2$	2.38
4.	ModifiedConstant (Pujara, Yahoo!)	8.48	± 0.0076	$L2$	2.93
5.	*PI-POMDP Level 2.975 (New addition)*	8.10	± 0.0083	$L2$	2.98
6.	Waugh (Carnegie Mellon)	8.00	± 0.0087	$L0$	0.96
7.	ACT-R (Carnegie Mellon)	7.88	± 0.0086	$L0$	0.96
8.	GreedyExpectedLaplace (Princeton)	7.43	± 0.0086	$L0$	0.83
9.	FrozenPontiac (U Michigan)	7.38	± 0.0075	$L0$	0.63
10.	Kuhlmann (U Texas Austin)	6.94	± 0.0054	$L0$	0.13

Table 2: 2010 LSG Tournament results including two agents inspired by the PI-POMDP hierarchy.

Rank	Strategy (Affiliation)	Score	Error	Level	Parameterized Level
1.	*PI-POMDP Level 2.975 (New addition)*	8.30	± 0.0099	$L2$	2.98
2.	**TeamUP (Southampton/Imperial)**	8.25	± 0.0099	$L2$	2.83
3.	Waugh (Carnegie Mellon)	8.19	± 0.0094	$L2$	2.93
4.	ModifiedConstant (Pujara, Yahoo!)	8.17	±0.0097	$L2$	2.93
5.	Matchmate (GA Tech)	8.15	±0.0095	$L2$	2.13
6.	Shamooshak (Alberta)	8.10	±0.0094	$L2$	2.25
7.	GoffBot (Brown)	7.97	±0.0108	$L2$	2.13
8.	Collaborator (Rutgers)	7.95	±0.0105	$L2$	2.38
9.	Meta (Carnegie Mellon)	7.80	±0.0102	$L2$	2.38
10.	*PI-POMDP Level 1.0 modified (New addition)*	7.80	±0.0096	$L1$	1.00
11.	Cactusade (Arizona)	7.27	±0.0085	$L2$	2.13

erative partner and suffering as a result.

Despite the difficulty of behavior forecasting, there is no question that learning can play a role, even among higher level strategies. However, that learning needs to take place in the proper space, or else a strategy will not have the capacity to react to basic heuristics. For instance, the top three 2009 players did adapt somewhat in response to their opponents. They did so by recognizing that they were not playing against distributions like those found in single-agent domains, but other players who understood the rules and were prepared to leverage them against slower players. The PI-POMDP framework identifies this reasoning process and is able to suggest a strategy that performs much better than previous agents. The resulting population profile gives insight to predict our opponents and respond preemptively.

In sum, the PI-POMDP analysis achieves good predictions of the strategies' performances. Furthermore, it has revealed characteristic properties of the LG. Future work will aim to show its applicability to further domains and establish the method as a framework to understand similar multiagent games of this kind.

8. ACKNOWLEDGMENTS

The authors would like to thank the National Science Foundation for support on this project via NSF HSD-0624191.

9. REFERENCES

[1] D. Billings. The first international roshambo programming competition. *ICGA Journal*, 23, 2000.

[2] C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.

[3] C. F. Camerer, T.-H. Ho, and J.-K. Chong. A cognitive hierarchy model of games. *Quarterly Journal of Economics*, 119:861–898, 2004.

[4] M. Costa-Gomes, V. Crawford, and B. Broseta. Cognition and behavior in normal-form games: An experimental study. *Econometrica*, 69(5):1193–1235, 2001.

[5] E. M. de Côte, A. Chapman, A. M. Sykulski, and N. R. Jennings. Automated planning in adversarial repeated games. *UAI*, 2010.

[6] D. Egnor. Iocaine powder. *ICGA Journal*, 23, 2000.

[7] N. Friess, J. Aycock, and R. Vogt. Black market botnets. *MIT Spam Conference*, 2008.

[8] Y. Gal. *Reasoning about Rationality and Beliefs*. PhD thesis, Harvard University, June 2006.

[9] P. Gmytrasiewicz and P. Doshi. A framework for sequential planning in multiagent settings. *Journal of AI Research (JAIR)*, 24:49–79, 2005.

[10] P. J. Gmytrasiewicz and E. H. Durfee. A rigorous, operational formalization of recursive modeling. *Proceedings of the First International Conference on*

Multi-Agent Systems, pages 125–132, 1995.

[11] P. J. Gmytrasiewicz and E. H. Durfee. Rational communication in multi-agent environments. *Autonomous Agents and Multi-Agent Systems Journal*, 4:233–272, 2001.

[12] P. R. Jordan, M. P. Wellman, and G. Balakrishnan. Strategy and mechanism lessons from the first ad auctions trading agent competition. *Proceedings of the 11th ACM Conference on Electronic Commerce*, 2010.

[13] K. Leyton-Brown and Y. Shoham. *Essentials of Game Theory: A Concise, Multidisciplinary Introduction*. Morgan and Claypool, 2008.

[14] J. Niu, K. Cai, P. McBurney, and S. Parsons. An analysis of entries in the first TAC market design competition. In *Proceedings of the IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, December 2008.

[15] D. Ray, B. King-Casas, P. R. Montague, and P. Dayan. Bayesian model of behaviour in economic games. *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[16] J. R. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal form games. *The Twenty-Fourth Conference on Artificial Intelligence (AAAI-10)*, 2010.

[17] M. Zinkevich. The lemonade game competition. http://tech.groups.yahoo.com/group/lemonadegame/, December 2009.