Modeling Humans as Reinforcement Learners: How to Predict Human Behavior in Multi-Stage Games

Ritchie Lee
Carnegie Mellon University Silicon Valley
NASA Ames Research Park MS23-11
Moffett Field, CA 94035
ritchie.lee@sv.cmu.edu

David H. Wolpert
Intelligent Systems Division
NASA Ames Research Center MS269-1
Moffett Field, CA 94035
david.h.wolpert@nasa.gov

Scott Backhaus
Los Alamos National Laboratory
MS K764, Los Alamos, NM 87545
backhaus@lanl.gov

Russell Bent
Los Alamos National Laboratory
MS C933, Los Alamos, NM 87545
rbent@lanl.gov

James Bono
Department of Economics
American University
4400 Massachusetts Ave. NW
Washington DC 20016
bono@american.edu

Brendan Tracey
Department of Aeronautics and Astronautics
Stanford University
496 Lomita Mall, Stanford, CA 94305
btracey@stanford.edu

Abstract

This paper introduces a novel framework for modeling interacting humans in a multi-stage game environment by combining concepts from game theory and reinforcement learning. The proposed model has the following desirable characteristics: (1) Bounded rational players, (2) strategic (i.e., players account for one another's reward functions), and (3) is computationally feasible even on moderately large real-world systems. To do this we extend level-K reasoning to policy space to, for the first time, be able to handle multiple time steps. This allows us to decompose the problem into a series of smaller ones where we can apply standard reinforcement learning algorithms. We investigate these ideas in a cyber-battle scenario over a smart power grid and discuss the relationship between the behavior predicted by our model and what one might expect of real human defenders and attackers.

1 Introduction

We present a model of interacting human beings that advances the literature by combining concepts from game theory and computer science in a novel way. In particular, we introduce the first time-extended level-K game theory model [1, 2]. This allows us to use reinforcement learning (RL) algorithms to learn each player's optimal policy against the level $K-1$ policies of the other players. However, rather than formulating policies as mappings from belief states to actions, as in partially observable Markov decision processes (POMDPs), we formulate policies more generally as mappings from a player's observations and memory to actions. Here, memory refers to all of a player's past observations.

This model is the first to combine all of the following characteristics. First, players are strategic in the sense that their policy choices depend on the reward functions of the other players. This is in

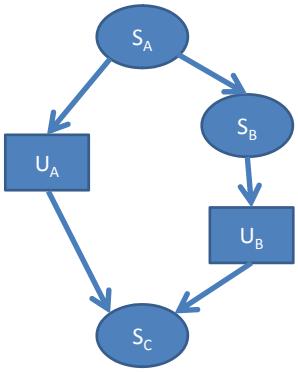


Figure 1: An example semi Bayes net.

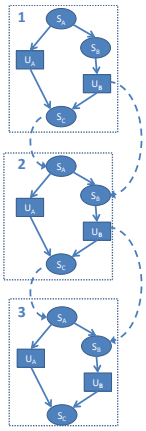


Figure 2: An example iterated semi Bayes net.

contrast to learning-in-games models whereby players do not use their opponents' reward information to predict their opponents' decisions and to choose their own actions. Second, this approach is computationally feasible even on real-world problems. This is in contrast to equilibrium models such as subgame perfect equilibrium and quantal response equilibrium [3]. This is also in contrast to POMDP models (e.g. I-POMDP) in which players are required to maintain a belief state over spaces that quickly explode. Third, with this general formulation of the policy mapping, it is straightforward to introduce experimentally motivated behavioral features such as noisy, sampled or bounded memory. Another source of realism is that, with the level-K model instead of an equilibrium model, we avoid the awkward assumption that players' predictions about each other are always correct.

We investigate all this for modeling a cyber-battle over a smart power grid. We discuss the relationship between the behavior predicted by our model and what one might expect of real human defenders and attackers.

2 Game Representation and Solution Concept

In this paper, the players will be interacting in an iterated semi net-form game. To explain an iterated semi net-form game, we will begin by describing a semi Bayes net. A semi Bayes net is a Bayes net with the conditional distributions of some nodes left unspecified. A pictoral example of a semi Bayes net is given in Figure 1. Like a standard Bayes net, a semi Bayes net consist of a set of nodes and directed edges. The ovular nodes labeled "S" have specified conditional distributions with the directed edges showing the dependencies among the nodes. Unlike a standard Bayes net, there are also rectangular nodes labeled "U" that have unspecified conditional dependencies. In this paper, the unspecified distributions will be set by the interacting human players. A semi net-form game, as described in [4], consists of a semi Bayes net plus a reward function mapping the outcome of the semi Bayes net to rewards for the players.

An iterated semi Bayes net is a Bayes net which has been time extended. It comprises of a semi Bayes net (such as the one in Figure 1), which is replicated T times. Figure 2 shows the semi Bayes net replicated three times. A set of directed edges L sets the dependencies between two successive iterations of the semi Bayes net. Each edge in L connects a node in stage $t-1$ with a node in stage t as is shown by the dashed edges in Figure 2. This set of L nodes is the same between every two successive stages. An iterated semi net-form game comprises of two parts: an iterated semi Bayes net and a set of reward functions which map the results of each step of the semi Bayes net into an incremental reward for each player. In Figure 2, the unspecified nodes have been labeled "U_A" and "U_B" to specify which player sets which nodes.

Having described above our model of the strategic scenario in the language of iterated semi net-form games, we now describe our solution concept. Our solution concept is a combination of the level-K model, described below, and reinforcement learning (RL). The level-K model is a game theoretic

solution concept used to predict the outcome of human-human interactions. A number of studies [1, 2] have shown promising results predicting experimental data in games using this method. The solution to the level-K model is defined recursively as follows. A level K player plays as though all other players are playing at level $K-1$, who, in turn, play as though all other players are playing at level $K-2$, etc. The process continues until level 0 is reached, where the level 0 player plays according to a prespecified prior distribution. Notice that running this process for a player at $K \geq 2$ results in ricocheting between players. For example, if player A is a level 2 player, he plays as though player B is a level 1 player, who in turn plays as though player A is a level 0 player playing according to the prior distribution. Note that player B in this example may not actually be a level 1 player in reality – only that player A assumes him to be during his reasoning process.

This work extends the standard level-K model to time-extended strategic scenarios, such as iterated semi net-form games. In particular, each Undetermined node associated with player i in the iterated semi net-form game represents an action choice by player i at some time t. We model player i's action choices using the policy function, ρ_i, which takes an element of the Cartesian product of the spaces given by the parent nodes of i's Undetermined node to an action for player i. Note that this definition requires a special type of iterated semi-Bayes net in which the spaces of the parents of each of i's action nodes must be identical. This requirement ensures that the policy function is always well-defined and acts on the same domain at every step in the iterated semi net-form game. We calculate policies using reinforcement learning (RL) algorithms. That is, we first define a level 0 policy for each player, ρ_i^0. We then use RL to find player i's level 1 policy, ρ_i^1, given the level 0 policies of the other players, ρ_{-1}^0, and the iterated semi net-form game. We do this for each player i and each level K.[1]

3 Application: Cybersecurity of a Smart Power Network

In order to test our iterated semi net-form game modeling concept, we adopt a model for analyzing the behavior of intruders into cyber-physical systems. In particular, we consider Supervisory Control and Data Acquisition (SCADA) systems [5], which are used to monitor and control many types of critical infrastructure. A SCADA system consists of cyber-communication infrastructure that transmits data from and sends control commands to physical devices, e.g. circuit breakers in the electrical grid. SCADA systems are partly automated and partly human-operated. Increasing connection to other cyber systems creating vulnerabilities to SCADA cyber attackers [6].

Figure 3 shows a single, radial distribution circuit [7] from the transformer at a substation (node 1) serving two load nodes. Node 2 is an aggregate of small consumer loads distributed along the circuit, and node 3 is a relatively large distributed generator located near the end of the circuit. In this figure V_i, p_i, and q_i are the voltage, real power, and reactive power at node i. P_i, Q_i, r_i, and x_i are the real power, reactive power, resistance and reactance of circuit segment i. Together, these values represent the following physical system [7], where all terms are normalized by the nominal system voltage.

$$P_2 = -p_3, \;\; Q_2 = -q_3, \;\; P_1 = P_2 + p_2, \;\; Q_1 = Q_2 + q_2 \tag{1}$$

$$V_2 = V_1 - (r_1 P_1 + x_1 Q_1), \;\; V_3 = V_2 - (r_2 P_2 + x_2 Q_2) \tag{2}$$

In this model, r, x, and p_3 are static parameters, q_2 and p_2 are drawn from a random distribution at each step of the game, V_1 is the decision variable of the defender, q_3 is the decision variable of the attacker, and V_2 and V_3 are determined by the equations above. The injection of real power p_3 and reactive power q_3 can modify the P_i and Q_i causing the voltage V_2 to deviate from 1.0. Excessive deviation of V_2 or V_3 can damage customer equipment or even initiate a cascading failure beyond the circuit in question. In this example, the SCADA operator's (defender's) control over q_3 is compromised by an attacker who seeks to create deviations of V_2 causing damage to the system.

In this model, the defender has direct control over V_1 via a variable-tap transformer. The hardware of the transformer limits the defenders actions at time t to the following domain

$$D_{\mathcal{D}}(t) = \langle \min(v_{max}, V_{1,t-1} + v), V_{1,t-1}, \max(v_{min}, V_{1,t-1} - v) \rangle$$

[1] Although this work uses level-K and RL exclusively, we are by no means wedded to this solution concept. Previous work on semi net-form games used a method known as Level-K Best-of-M/M' instead of RL to determine actions. This was not used in this paper because the possible action space is so large.

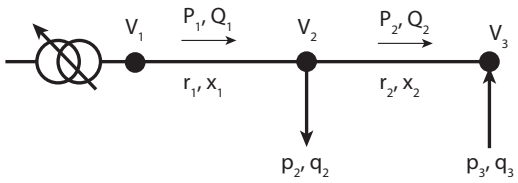


Figure 3: Schematic drawing of the three-node distribution circuit.

where v is the voltage step size for the transformer, and v_{min} and v_{max} represent the absolute min and max voltage the transformer can produce. Similarly, the attacker has taken control of q_3 and its actions are limited by its capacity to produce real power, $p_{3,max}$ as represented by the following domain.

$$D_{\mathcal{A}}(t) = \langle -p_{3,max}, \ldots, 0, \ldots, p_{3,max} \rangle$$

Via the SCADA system and the attacker's control of node 3, the observation spaces of the two players includes

$$\Omega_{\mathcal{D}} = \{V_1, V_2, V_3, P_1, Q_1, M_{\mathcal{D}}\}, \;\; \Omega_{\mathcal{A}} = \{V_2, V_3, p_3, q_3, M_{\mathcal{A}}\}$$

where $M_{\mathcal{D}}$ and $M_{\mathcal{A}}$ are used to denote each two real numbers that represent the respective player's memory of the past events in the game. Both the defender and attacker manipulate their controls in a way to increase their own rewards. The defender desires to maintain a high quality of service by maintaining the voltages V_2 and V_3 near the desired normalized voltage of one while the attacher wishes to damage equipment at node 2 by forcing V_2 beyond operating limits, i.e.

$$R_{\mathcal{D}} = -\left(\frac{V_2 - 1}{\epsilon}\right)^2 - \left(\frac{V_3 - 1}{\epsilon}\right)^2, \;\; R_{\mathcal{A}} = \Theta[V_2 - (1+\epsilon)] + \Theta[(1-\epsilon) - V_2]$$

Here, $\epsilon \sim 0.05$ for most distribution system under consideration, Θ is a Heaviside step function.

Level 1 defender policy The level 0 defender is modeled myopically and seeks to maximize his reward by following a policy that adjusts V_1 to move the average of V_2 and V_3 closer to one, i.e.

$$\pi_{\mathcal{D}}(V_2, V_3) = \arg\min_{V_1 \in D_{\mathcal{D}}(t)} \frac{(V_2 + V_3)}{2} - 1$$

Level 1 attacker policy The level 0 attacker adopts a *drift and strike* policy based on intimate knowledge of the system. If $V_2 < 1$, we propose that the attacker would decrease q_3 by lowering it by one step. This would cause Q_1 to increase and V_2 to fall even farther. This policy achieves success if the defender raises V_1 in order to keep V_2 and V_3 in the acceptable range. The attacker continues this strategy, pushing the defender towards v_{max} until he can quickly raise q_3 to push V_2 above $1 + \epsilon$. If the defender has neared v_{max}, then a number of time steps will be required to for the defender to bring V_2 back in range. More formally this policy can be expressed as

LEVEL0ATTACKER()
1 $V^* = \max_{q \in D_{\mathcal{A}}(t)} |V_2 - 1|$;
2 **if** $V^* > \theta_{\mathcal{A}}$
3 **then return** $\arg\max_{q \in D_{\mathcal{A}}(t)} |V_2 - 1|$;
4 **if** $V_2 < 1$
5 **then return** $q_{3,t-1} - 1$;
6 **return** $q_{3,t-1} + 1$;

where $\theta_{\mathcal{A}}$ is a threshold parameter.

3.1 Reinforcement Learning Implementation

Using defined level 0 policies as the starting point, we now bootstrap up to higher levels by training each level K policy against an opponent playing level $K-1$ policy. To find policies that maximize reward, we can apply any algorithm from the reinforcement learning literature. In this paper, we use

an ϵ-greedy policy parameterization (with $\epsilon = 0.1$) and SARSA on-policy learning [8]. Training updates are performed epoch-wise to improve stability. Since the players' input spaces contain continuous variables, we use a neural-network to approximate the Q-function [9]. We improve performance by scheduling the exploration parameter ϵ in 3 segments during training: An ϵ of near unity, followed by a linearly decreasing segment, then finally the desired ϵ.

3.2 Results and Discussion

We present results of the defender and attacker's behavior at various level K. We note that our scenario always had an attacker present, so the defender is trained to combat the attacker and has no training concerning how to detect an attack or how to behave if no attacker is present. Notionally, this is also true for the attacker's training. However in real-life the attacker will likely know that there is someone trying to thwart this attack.

Level 0 defender vs. level 0 attacker The level 0 defender (see Figure 4(a)) tries to keep both V_2 and V_3 close to 1.0 to maximize his immediate reward. Because the defender makes steps in V_1 of 0.02, he does nothing for $30 < t < 60$ because any such move would not increase his reward. For $30 < t < 60$, the p_2, q_2 noise causes V_2 to fluctuate, and the attacker seems to randomly drift back and forth in response. At $t = 60$, the noise plus the attacker and defender actions breaks this "symmetry", and the attacker increases his q_3 output causing V_2 and V_3 to rise. The defender responds by decreasing V_1, indicated by the abrupt drops in V_2 and V_3 that break up the relatively smooth upward ramp. Near $t = 75$, the accumulated drift of the level 0 attacker plus the response of the level 0 defender pushes the system to the edge. The attacker sees that a strike would be successful (i.e., post-strike $V_2 < 1 - \theta_{\mathcal{A}}$), and the level 0 defender policy fails badly. The resulting V_2 and V_3 are quite low, and the defender ramps V_1 back up to compensate. Post strike ($t > 75$), the attackers threshold criterion tells him that an immediate second strike would would not be successful, however, this shortcoming will be resolved via level 1 reinforcement learning. Overall, this is the behavior we have built into the level 0 players.

Level 1 defender vs. level 0 attacker During the level 1 training, the defender likely experiences the type of attack shown in Figure 4(a) and learns that keeping V_1 a step or two above 1.0 is a good way to keep the attacker from putting the system into a vulnerable state. As seen in Figure 4(b), the defender is never tricked into performing a sustained drift because the defender is willing to take a reduction to his reward by letting V_3 stay up near 1.05. For the most part, the level 1 defender's reinforcement learning effectively counters the level 0 attacker drift-and-strike policy.

Level 0 defender vs. level 1 attacker The level 1 attacker learning sessions correct a shortcoming in the level 0 attacker. After a strike ($V_2 < 0.95$ in Figure 4(a)), the level 0 attacker drifts up from his largest negative q_3 output. In Figure 4(c), the level 1 attacker anticipates that the increase in V_2 when he moves from $m = -5$ to $m = 5$ will cause the level 0 defender to drop V_1 on the next move. After this drop, the level 1 attacker also drops from $m = +5$ to -5. In essence, the level 1 attacker is leveraging the anticipated moves of the level 0 defender to create oscillatory strikes that push V_2 below $1 - \epsilon$ nearly every cycle.

Acknowledgments

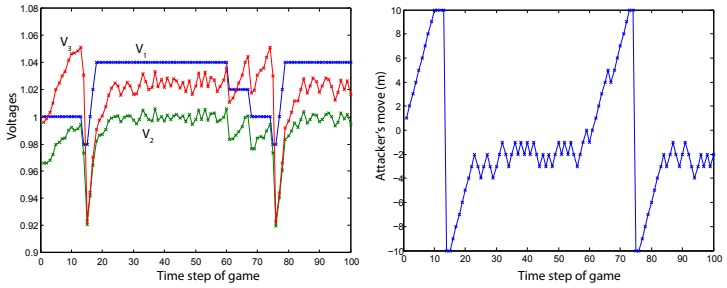
This research was supported by the NASA Aviation Safety Program SSAT project, and the Los Alamos National Laboratory LDRD project Optimization and Control Theory for Smart Grid.

References

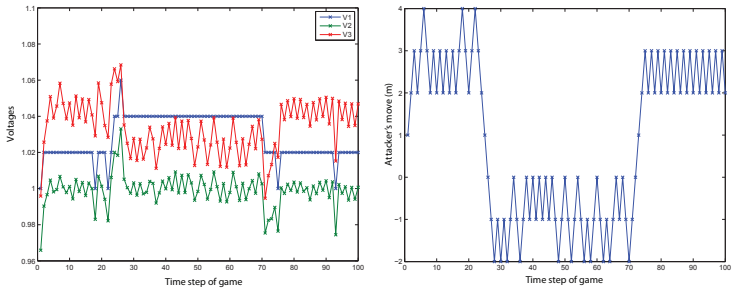
[1] Miguel Costa-Gomes and Vincent Crawford. Cognition and behavior in two-person guessing games: An experimental study. *American Economic Review*, 96(5):1737–1768, December 2006.

[2] Dale O. Stahl and Paul W. Wilson. On players' models of other players: Theory and experimental evidence. *Games and Economic Behavior*, 10(1):218 – 254, 1995.

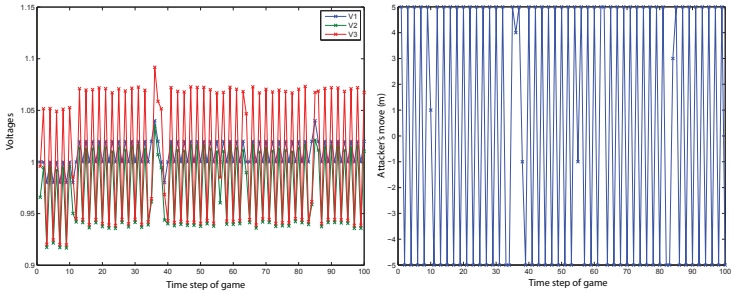
[3] Richard Mckelvey and Thomas Palfrey. Quantal response equilibria for extensive form games. *Experimental Economics*, 1:9–41, 1998. 10.1023/A:1009905800005.



(a) Level 0 defender vs. level 0 attacker



(b) Level 1 defender vs. level 0 attacker



(c) Level 0 defender vs. level 1 attacker

Figure 4: Voltages and attacker moves of various games.

[4] Ritchie Lee and David H. Wolpert. *Decision Making with Multiple Imperfect Decision Makers*, chapter Game Theoretic Modeling of Pilot Behavior during Mid-Air Encounters. Intelligent Systems Reference Library Series. Springer, 2011.

[5] K. Tomsovic, D.E. Bakken, V. Venkatasubramanian, and A. Bose. Designing the next generation of real-time control, communication, and computations for large power systems. *Proceedings of the IEEE*, 93(5):965 –979, may 2005.

[6] Alvaro A. Cárdenas, Saurabh Amin, and Shankar Sastry. Research challenges for the security of control systems. In *Proceedings of the 3rd conference on Hot topics in security*, pages 6:1–6:6, Berkeley, CA, USA, 2008. USENIX Association.

[7] K. Turitsyn, P. Sulc, S. Backhaus, and M. Chertkov. Options for control of reactive power by distributed photovoltaic generators. *Proceedings of the IEEE*, 99(6):1063 –1073, june 2011.

[8] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

[9] Lucian Busoniu, Robert Babuska, Bart De Schutter, and Ernst Damien. *Reinforcement Learning and Dynamic Programming Using Function Approximators*. CRC Press, 2010.