Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence

I'm Doing as Well as I Can: Modeling People as Rational Finite Automata

Joseph Y. Halpern **Rafael Pass** **Lior Seeman**
Computer Science Dept.
Cornell University
Ithaca, NY
E-mail: halpern|rafael|lseeman@cs.cornell.edu

Abstract

We show that by modeling people as bounded finite automata, we can capture at a qualitative level the behavior observed in experiments. We consider a decision problem with incomplete information and a dynamically changing world, which can be viewed as an abstraction of many real-world settings. We provide a simple strategy for a finite automaton in this setting, and show that it does quite well, both through theoretical analysis and simulation. We show that, if the probability of nature changing state goes to 0 and the number of states in the automaton increases, then this strategy performs optimally (as well as if it were omniscient and knew when nature was making its state changes). Thus, although simple, the strategy is a sensible strategy for a resource-bounded agent to use. Moreover, at a qualitative level, the strategy does exactly what people have been observed to do in experiments.

1 Introduction

Our goal in this paper is to better understand how people make decisions in dynamic situations with uncertainty. There are many examples known where people do not seem to be choosing acts that maximize expected utility. Various approaches have been proposed to account for the behavior of people, of which perhaps the best known is Kahnemann and Tversky's 1979 *prospect theory*.

One explanation for this inconsistency between expected utility theory and real-life behavior has been that agents are *boundedly rational*—they are rational, but computationally bounded. The most commonly-used model of computationally bounded agents has been finite automata. Finite automata were used by Rubinstein 1986 and Neyman 1985 to explain the behavior of people playing finitely repeated prisoner dilemma. The only Nash equilibrium in finitely repeated prisoner's dilemma is to always defect, but this is clearly not what people do in practice. Rubinstein and Neyman showed (under different assumptions) that if we restrict players to choosing a finite automaton to play for them, there are equilibria with cooperation. (See (Papadimitriou and Yannakakis 1994) for more recent results along these lines.)

Wilson 2002 considers a decision problem where an agent needs to make a single decision, whose payoff depends on the state of nature (which does not change over time). Nature is in one of two possible states, G (good) and B (bad). The agent gets signals, which are correlated with the true state, until the game ends, which happens at each step with probability $\eta > 0$. At this point, the agent must make a decision. Wilson characterizes an n-state optimal finite automaton for making a decision in this setting, under the assumption that η is small (so that the agent gets information for many rounds). She shows that an optimal n-state automaton ignores all but two signals (the "best" signal for each of nature's states); the automaton's states can be laid out "linearly", as states $0, \ldots, n-1$, and the automaton moves left (with some probability) only if it gets a strong signal for state G, and moves right (with some probability) only if it gets a strong signal for state B. Thus, roughly speaking, the lower the current state of the automaton, the more likely from the automaton's viewpoint that nature's state is G. (Very similar results were proved earlier by Hellman and Cover 1970.) Wilson argues that these results can be used to explain observed biases in information processing, such as *belief polarization* (two people with different prior beliefs, hearing the same evidence, can end up with diametrically opposed conclusions) and the *fist-impression bias* (people tend to put more weight on evidence they hear early on). Thus, some observed human behavior can be explained by viewing people as resource-bounded, but rationally making the best use of their resources (in this case, the limited number of states).

Wilson's model assumes that nature is static. But in many important problems, ranging from investing in the stock market to deciding which route to take when driving to work, the world is dynamic. Moreover, people do not make decisions just once, but must make them often. For example, when investing in stock markets, people get signals about the market, and need to decide after each signal whether to invest more money, take out money that they have already invested, or to stick with their current position.

In this paper, we consider a model that is intended to capture the most significant features of a dynamic situation. As in Wilson's model, we allow nature to be in one of a number of different states (for simplicity, like Wilson, in most of the paper we assume that nature is in one of only two states), and assume that the agent gets signals correlated with nature's

state. But now we allow nature's state to change, although we assume that the probability of a change is low. (Without this assumption, the signals are not of great interest.)

Our choice of model is in part motivated by recent work by psychologists and economists on how people behave in such scenarios, and particularly that of Erev, Ert, and Roth 2010a, who describe contests that attempt to test various models of human decision making under uncertain conditions. In their scenarios, people were given a choice between making a safe move (that had a guaranteed constant payoff) and a "risky" move (which had a payoff that changed according to an unobserved action of the other players). Since their goal was that of finding models that predicted human behavior well, Erev et al. 2010a considered a sequence of settings, and challenged others to present models that would predict behavior in these settings.

They also introduced a number of models themselves, and determined the performance of these models in their settings. One of those models is *I-Saw* (inertia, sampling, and weighting) (Erev, Ert, and Roth 2010a); it performed best among their models, with a correlation of 0.9 between the model's prediction and the actual observed results for most variables. I-Saw assumes that agents have three types of response mode: *exploration*, *exploitation*, and *inertia*. An I-Saw agent proceeds as follows. The agent tosses a coin. If it lands heads, the agent plays the action other than the one he played in the previous step (*exploration*); if it lands tails, he continues to do what he did in the previous step (*inertia*), unless the signal received in the previous round crosses a probabilistic "surprise" trigger (the lower the probability of the signal to be observed in the current state, the more likely the trigger is to be crossed); if the surprise trigger is crossed, then the agent plays the action with the best estimated subjective value, based on some sampling of the observations seen so far (*exploitation*).

The winner of the contest was a refinement of I-Saw called *BI-Saw* (bounded memory, inertia, sampling and weighting) model, suggested by Chen et al. 2011. The major refinement involved adding a bounded memory assumption, whose main effect is a greater reliance on a small sample of past observations in the exploitation mode. The *BI-Saw* model had a normalized mean square deviation smaller than 1.4 for estimating the entry rate of the players, and smaller than 1 for estimating the actual payoff they get, which was better than the results of the *I-Saw* model.

I-Saw and BI-Saw seem quite *ad hoc*. In this paper, we show that they can be viewed as the outcomes of play of a resource-bounded agent modeled as a finite automaton.[1] Specifically, we consider a setting where an agent must make a decision every round about playing safe (and getting a guaranteed payoff) or playing a risky move, whose payoff depends on the state of nature. We describe a simple strategy for this agent, and show both theoretically and by simulation that it does very well in practice. While it may not be the optimal strategy if the agent is restricted to n states, we show that as n goes to infinity and the probability of nature changing state goes to 0, the expected payoff of this strategy converges to the best expected payoff that the player could get even if he knew the state of nature at all times. Interestingly, this strategy exhibits precisely the features of (I-Saw and) BI-Saw at a qualitative level. Thus, we believe that (B)I-Saw can be best understood as the outcome of a resource-bounded agent playing quite rationally.

Paper Outline: The rest of the paper is organized as follows. Section 2 describes our stylized model and the agent's strategy. Sections 3 presents our theoretical analysis of the model. Section 4 describes the results of our simulations. We conclude in Section 5.

2 The Model

We assume that nature is in one of two states G ("good") and B ("bad"); there is a probability π of transition between them in each round. (In Section 4, we show that allowing nature to have more states does not affect the results at a qualitative level; similarly, while we could allow different transition probabilities from G to B and from B to G, this would not have an impact on our results.)

The agent has two possible actions S (safe) and R (risky). If he plays S, he gets a payoff of 0; if he plays R he gets a payoff $x_G > 0$ when nature's state is G, and a payoff $x_B < 0$ when nature's state is B. The agent does not learn his payoff, and instead gets one of k signals. Signal i has probability p_i^G of appearing when the state of nature is G, and probability p_i^B of appearing when the state is B. We assume that the agent gets exactly one signal at each time step, so that $\sum_{i=1}^k p_i^G = \sum_{i=1}^k p_i^B = 1$. This signaling mechanism is similar to that considered by Wilson 2002 and Cover and Hellman 1970. However, we assume that the agent gets a signal only if he plays the risky action R; he does not get a signal if he plays the safe action S. We denote this setting $S[p_1^G, p_1^B, \ldots, p_k^G, p_k^B, x_G, x_B]$. We say that a setting is *nontrivial* if there exists some signal i such that $p_i^B \neq p_i^G$. If a setting is trivial, then no signal enables the agent to distinguish whether nature is in state G or B; the agent does not learn anything from the signals. (Note that we have deliberately omitted nature's transition probability π from the description of the setting. That is because, in our technical results, we want to manipulate π while keeping everything else fixed.) One quick observation is that a trivial strategy that plays one of S or R all the time gets a payoff of 0 in this model, as does the strategy that chooses randomly between S and R. We are interested in finding a simple strategy that gets appreciably more than 0.[2]

[1] Although the scenarios in (Erev, Ert, and Roth 2010a) are games rather than decision problems, as observed in (Erev, Ert, and Roth 2010b), learning in a decision problem should work essentially the same way as learning in games, so, for simplicity, we consider the former setting. The winner of a similar contest for single agent decision problems (see Erev et al. 2010) was a predecessor of the I-Saw model with very similar behavior modes. In our setting, the agent's actions do not influence nature, which is similar to assumptions usually made in large economic markets, where a single agent cannot influence the market.

[2] This model can be viewed as an instance of a "one-armed restless bandit" (Whittle 1988) that does not have perfect information about the state of the project. This kind of decision problem was also tested (Biele, Erev, and Ert 2009), and the model that per-

As suggested by the BI-Saw model, we assume that agents have bounded memory. We model this by assuming agents which are restricted to using a finite automaton, with deterministic actions and probabilistic transitions, and a fixed number n of internal states. The agent's goal is to maximize his expected average payoff.

We focus on one particular family of strategies (automata) for the agent. We denote a typical member of this family $A[n, p_{exp}, Pos, Neg, r_u, r_d]$. The automaton $A[n, p_{exp}, Pos, Neg, r_u, r_d]$ has $n+1$ states, denoted $0, \ldots, n$. State 0 is dedicated to playing S. In all other states R is played. The k signals are partitioned into three sets, Pos (for "positive"), Neg (for "negative"), and I (for "ignore" or "indifferent"), with Pos and Neg nonempty. Intuitively, the signals in Pos make it likely that nature's state is G, and the signals in Neg make it likely that the state of nature is B. The agent chooses to ignore the signals in I; they are viewed as not being sufficiently informative as to the true state of nature. (Note that I is determined by Pos and Neg.)

In each round while in state 0, the agent moves to state 1 with probability p_{exp}. In a state $i > 0$, if the agent receives a signal in Pos, the agent moves to $i+1$ with probability r_u (unless he is already in state n, in which case he stays in state n if he receives a signal in Pos); thus, we can think of r_u as the probability the the agent moves up if he gets a positive signal. If the agent receives a signal in Neg, the agent moves to state $i-1$ with probability r_d (so r_d is the probability of moving down if he gets a signal in Neg); if he receives a signal in I, the agent does not changes states. Clearly, this automaton is easy for a human to implement (at least, if it does not have too many states).

Note that this automaton incorporates all three behavior modes described by the I-Saw model. When the automaton is in state 0, the agent *explores* with constant probability by moving to state 1. In state $i > 0$, the agent continues to do what he did before (in particular, he stays in state i) unless he gets a "meaningful" signal (one in Neg or Pos), and even then he reacts only with some probability, so we have *inertia*-like behavior. If he does react, he *exploits* the information he has, which is carried by his current state; that is, he performs the action most appropriate according to his state, which is R. The state can be viewed as representing a sample of the last few signals (each state represents remembering seeing one more "good" signal), as in the BI-Saw model.

3 Theoretical analysis

In this section, we do a theoretical analysis of the expected payoff of the automaton $A[n, p_{exp}, Pos, Neg, r_u, r_d]$. This will tell us how to optimally choose the parameters Pos, Neg, r_u, and r_d. Observe that the most any agent can hope to get is $\frac{x_G}{2}$. Even if the agent had an oracle that told him exactly what nature's state would be at every round, if he performs optimally, he can get only x_G in the rounds when nature is in state G, and 0 when it is in state B. In expectation, nature is in state G only half the time, so the optimal expected payoff is $x_G/2$. One of our results shows that, somewhat surprisingly, as n gets large, if π goes to 0 sufficiently quickly, then the agent can achieve arbitrarily close to the theoretical optimum using an automaton of the form $A[n, p_{exp}, Pos, Neg, r_u, r_d]$, even without the benefit of an oracle, by choosing the parameters appropriately. More precisely, we have the following theorem.

Theorem 3.1 *Let Π and P_{exp} be functions from $\mathbb{N}$ to $(0,1]$ such that $\lim_{n\to\infty} n\Pi(n) = \lim_{n\to\infty} \Pi(n) = \lim_{n\to\infty} \Pi(n)/P_{exp}(n) = 0$. Then for all settings $S[p_1^G, p_1^B, \ldots, p_k^G, p_k^B, x_G, x_B]$, there exists a partition Pos, Neg, I of the signals, and constants r_d and r_u such that $\lim_{n\to\infty} E_{\Pi(n)}[A[n, P_{exp}(n), Pos, Neg, r_u, r_d]] = \frac{x_G}{2}$.*

Note that in Theorem 3.1, $\Pi(n)$ goes to 0 as n goes to infinity. This requirement is necessary, as the next result shows; for fixed π, we can't get too close to the optimal no matter what automaton we use (indeed, the argument below applies even if the agent is not bounded at all).

Theorem 3.2 *For all fixed $0 < \pi \leq 0.5$ and all automata A, we have $E_\pi[A] \leq x_G/2 + \pi x_B/2$.*

Proof: Suppose that the automaton had an oracle that, at each time t, correctly told it the state of nature at the previous round. Clearly the best the automaton could do is to play S if the state of nature was B and play R if the state of nature was G. Thus, the automaton would play R half the time and G half the time. But with probability π the state of nature will switch from G to B, so the payoff will be x_B rather than x_G. Thus, the payoff that it gets with this oracle is $x_G/2 + x_B\pi/2$. (Recall that $x_B < 0$.) We can think of the signals as being imperfect oracles. The automaton will do even worse with the signals than it will be with an oracle. □

The theorem focuses on small values of π, since this is the range of interest for our paper. We can prove a result in a similar spirit even if $0.5 < \pi < 1$.

The key technical result, from which Theorem 3.1 follows easily, gives us a very good estimate of the payoff of the automaton $A[n, p_{exp}, Pos, Neg, r_u, r_d]$, for all choices of the parameters. We state the estimate in terms of n, π, p_{exp} and four auxiliary quantitites, ρ_u^G, ρ_u^B, ρ_d^G, and ρ_d^B, Intuitively, ρ_u^N is the probability of the automaton changing states from i to $i+1$ (going "up") when nature is in state N and $i \geq 1$, and ρ_d^N is the probability of the automaton changing states from i to $i-1$ (going "down") given that nature is in state N. Thus, $\rho_u^N = (\sum_{i\in Pos} p_i^N) r_u$ and $\rho_d^N = (\sum_{i\in Neg} p_i^N) r_d$. We define $\sigma_N = \rho_u^N/\rho_d^N$. Recall that when the automaton is in state 0, it does not get any signals; rather, it explores (moves to state 1) with probability p_{exp}.

Proposition 3.3

$$E_\pi[A[n, p_{exp}, Pos, Neg, r_u, r_d]] \geq \frac{x_G}{2}\left(1 - \frac{(\rho_u^G - \rho_d^G) + \pi(\sum_{i=1}^n (\sigma_G)^i - n)}{(\rho_u^G - \rho_d^G) + p_{exp}((\sigma_G)^n - 1)}\right) + \frac{x_B}{2}\left(1 - \frac{(\rho_u^B - \rho_d^B) - \pi(\sum_{i=1}^n (\sigma_B)^i - n)}{(\rho_u^B - \rho_d^B) + p_{exp}((\sigma_B)^n - 1)}\right). \quad (1)$$

We sketch a proof of Proposition 3.3 in the next section. Although the expression in (1) looks rather complicated, it

formed best can be be viewed as a predecessor of the I-Saw model.

gives us just the information we need, both to prove Theorem 3.1 and to define an automaton that does well even when n is finite (and small).

Proof of Theorem 3.1: We want to choose Pos, Neg, r_u, and r_d so that $\rho_u^G > \rho_d^G$—the agent is more likely to go up than down when nature is in state G (so that he avoids going into state 0 and getting no reward) and $\rho_d^B > \rho_u^B$—the agent is more likely to go down than up when nature is in state B (so that he quickly gets into state 0, avoiding the payoff of -1). Suppose that we can do this. If $\rho_u^G > \rho_d^G$, then $\sigma_G > 1$, so the first term in the expression for the lower bound of $E_{\Pi(n)}[A[n, P_{exp}(n), Pos, Neg, r_u, r_d]]$ given by (1) tends to $\frac{x_G}{2}(1 - \frac{\Pi(n)}{P_{exp}(n)})$ as $n \to \infty$. Since we have assumed that $\lim_{n\to\infty} \Pi(n)/P_{exp}(n) = 0$, the first term goes to $x_G/2$. If $\rho_u^B < \rho_d^B$, then $\sigma_B < 1$, so the second term goes to

$$\frac{x_B}{2}\left(1 - \frac{(\rho_u^B - \rho_d^B) + \Pi(n)\frac{\sigma_B}{1-\sigma_B} + n\Pi(n)}{(\rho_u^B - \rho_d^B) - P_{exp}(n)}\right).$$

Since we have assumed that $\lim_{n\to\infty} n\Pi(n) = \lim_{n\to\infty} P_{exp}(n) = 0$, the second term goes to 0.

Now we show that we can choose Pos, Neg, r_B, and r_G so that $\rho_u^G > \rho_d^G$ and $\rho_d^B > \rho_u^B$. By assumption, there exists some signal i such that $p_i^G \neq p_i^B$. Since $\sum_{i=1}^k p_i^G = \sum_{i=1}^k p_i^B (= 1)$, it must be the case that there exists some signal i such that $p_i^G > p_i^B$. Let $Pos = \{i\}$. If there exists a signal j such that $p_j^G < p_i^B$ and $p_j^B > p_i^B$, then let $Neg = \{j\}$ and $r_u = r_d = 1$. Otherwise, let $Neg = \{1 \ldots k\} \setminus \{i\}$, $r_u = 1$, and let r_d be any value such that $\frac{p_i^G}{1-p_i^B} < r_d < \frac{p_i^G}{1-p_i^G}$. It is easy to check that, with these choices, we have $\sigma_G > 1$ and $\sigma_B < 1$. This completes the proof of Theorem 3.1. □

As we said, Proposition 3.3 gives us more than the means to prove Theorem 3.1. It also tells us what choices to make to get good behavior of n is finite. In Section 4, we discuss what these choices should be.

3.1 Proving Proposition 3.3

In this section, we sketch a proof of Proposition 3.3.

Once we are given π and a setting $S[p_1^G, p_1^B, \ldots, p_k^G, p_k^B, x_G, x_B]$, an automaton $A[n, p_{exp}, Pos, Neg, r_u, r_d]$ determines a Markov chain, with states of $(0, G), \ldots, (n, G), (0, B), \ldots, (n, B)$, where the Markov chain is in state (i, N) if nature is in state N and the automaton is in state i. The probability of transition is completely determined by π, the parameters of the automaton, and the setting.

Let $q_i^N(s, t)$ be the probability of the Markov chain being in state (i, N) at time t when started in state s. We are interested in $\lim_{t\to\infty} q_i^N(s, t)$. In general, this limiting probability may not exist and, even when it does, it may depend on the state the Markov chain starts in. However, there are well known sufficient conditions under which the limit exists, and is independent of the initial state s. A Markov chain is said to be *irreducible* if every state is reachable from every other state; it is *aperiodic* if, for every state s, there exist two cycles from s to itself such that the gcd of their lengths is 1. The limiting probaility exists and is independent of the start state in every irreducible aperiodic Markov chain over a finite state space (Resnick 1992, Corollary to Proposition 2.13.5). Our Markov chain is obviously irreducible; in fact, there is a path from every state in it to every other state. It is also aperiodic. To see this, note that if $0 < i \leq n$, there is a cycle of length $2i$ that can be obained by starting at (i, N), going to $(0, N)$ (by observing signals in Neg and nature not changing state) starting $(0, N)$ and going back up to (i, N). At $(0, N)$, there is a cycle of length 1. Thus, we can get a cycle of length $2i + 1$ starting at (i, N). Since we can go from $(0, B)$ to $(0, G)$ and back, there is also a cycle of length 2 from every state $(0, N)$. Since a limiting probability exists, we can write q_i^N, ignoring the arguments s and t.

We are particularly interested in the q_0^B and q_0^G, because these quantities completely determine the agent's expected payoff. As we have observed before, since the probability of transition from B to G is the same as the probability transition from G to B, nature is equally likely to be in state B and G. Thus, $\sum_{i=0}^n q_i^B = \sum_{i=0}^n q_i^G = 1/2$. Now the agent gets a payoff of x_G when he is in state $i > 0$ and nature is in state G; he gets a payoff of x_B when he is in state $i > 0$ and nature is in state B. Thus, his expected payoff is $x_G(1/2 - q_0^G) + x_B(1/2 - q_0^B)$.

It remains to compute q_0^B and q_0^G. To do this, we need to consider q_i^N for all values of i. We can write equations that characterize these probabilities. Let $\bar{N}$ be the state of nature other than N (so $\bar{B} = G$ and $\bar{G} = B$). Note that for a time t after (i, N) has reached its limiting probability, then the probability of state (i, N) has to be the same at time t and time $t + 1$. If $i > 0$, the probability of the system being in state (i, N) at time $t + 1$ is the sum of the probability of (a) being in state $(i+1, N)$ (or (n, N) if $i = n$), getting a signal in Neg and reacting to it, and nature not changing state, (b) being in state $(i-1, N)$, getting a signal in Pos and reacting to it (or, if $i = 1$, the system was in state $(i, 0)$ and the agent decided to explore), and nature did not change state, (c) being in state (i, N), getting a signal in I, and nature not changing state, (d) three further terms like (a)–(c) where the system state is $(j, \bar{N})$ at time t, for $j \in \{i - 1, i, i + 1\}$ and nature's state changes. There are similar equations for the state $(0, N)$, but now there are only four possibilities: (a) the system was in state $(1, N)$ at time t, the agent observed a signal in Neg and reacted to it, and nature's state didn't change, (b) the system was in state $(0, N)$ and the agent's state didn't change, and (c) two other analogous equations where nature's state changes from $\bar{N}$ to N.

These considerations give us the following equations:

$$\begin{aligned}
q_0^N = \;& (1-\pi)((1-p_{exp})q_0^N + \rho_d^N q_1^N) \\
& +\pi((1-p_{exp})q_0^{\bar{N}} + \rho_d^{\bar{N}} q_1^{\bar{N}}) \\
q_1^N = \;& (1-\pi)((1-\rho_d^N-\rho_u^N)q_1^N + \rho_d^N q_2^N + p_{exp} q_0^N) \\
& +\pi((1-\rho_d^{\bar{N}}-\rho_u^{\bar{N}})q_1^{\bar{N}} + \rho_d^{\bar{N}} q_2^{\bar{N}} + p_{exp} q_0^{\bar{N}}) \\
\vdots &
\end{aligned}$$

$$
\begin{array}{rl}
q_i^N = & (1-\pi)((1-\rho_d^N-\rho_u^N)q_i^N + \rho_d^N q_{i+1}^N + \rho_u^N q_{i-1}^N) \\
& +\pi((1-\rho_d^{\bar{N}}-\rho_u^{\bar{N}})q_i^{\bar{N}} + \rho_d^{\bar{N}} q_{i+1}^{\bar{N}} + \rho_u^{\bar{N}} q_{i-1}^{\bar{N}}) \\
\vdots & \\
q_n^N = & (1-\pi)((1-\rho_d^N)q_n^N + \rho_u^N q_{n-1}^N) \\
& +\pi((1-\rho_d^{\bar{N}})q_n^{\bar{N}} + \rho_u^{\bar{N}} q_{n-1}^{\bar{N}}).
\end{array} \tag{2}
$$

These equations seem difficult to solve exactly. But we can get very good approximate solutions. Define:

$$
\begin{array}{rl}
\gamma_i^N = & \pi((1-\rho_d^{\bar{N}}-\rho_u^{\bar{N}})q_i^{\bar{N}} + \rho_d^{\bar{N}} q_{i+1}^{\bar{N}} + \rho_u^{\bar{N}} q_{i-1}^{\bar{N}}) \\
& -\pi((1-\rho_d^N-\rho_u^N)q_i^N + \rho_d^N q_{i+1}^N + \rho_u^N q_{i-1}^N) \\
& \text{for } i = 2,\ldots,n; \\
\gamma_1^N = & \pi((1-\rho_d^{\bar{N}}-\rho_u^{\bar{N}})q_i^{\bar{N}} + \rho_d^{\bar{N}} q_{i+1}^{\bar{N}} + p_{exp} \\
& -\pi((1-\rho_d^N-\rho_u^N)q_i^N + \rho_d^N q_{i+1}^N + p_{exp}); \\
\gamma_0^N = & \pi((1-\rho_d^{\bar{N}})q_n^{\bar{N}} + \rho_u^{\bar{N}} q_{n-1}^{\bar{N}}) - \rho_d^{\bar{N}} q_1^{\bar{N}}.
\end{array}
$$

Note that γ_i^N is essentially a subexpression of q_i^N. Intuitively, γ_i^N is the net probability transferred between states of (i,N) from (or to) states of the form $(j,\bar{N})$ as a result of nature changing from N to $\bar{N}$ or from $\bar{N}$ to N. Let $(\gamma_i^N)^+ = \gamma_i^N$ if $\gamma_i^N > 0$ and 0 otherwise; let $(\gamma_i^N)^- = \gamma_i^N$ if $\gamma_i^N < 0$ and 0 otherwise. Intuitively, $(\gamma_i^N)^+$ is the net probability transferred to (i,n) from states of the form $(j,\bar{N})$ as a result of nature's state changing from $\bar{N}$ to N; similarly, $(\gamma_i^N)^-$ is the net probability transferred from (i,N) to states of the form $(j,\bar{N})$ as a result of nature's state changing from N to $\bar{N}$. Since $\sum_{i=0}^N q_i^N = 1/2$, it is easy to check that

$$
\begin{array}{l}
\sum_{i=0}^n \gamma_i^N = 0; \\
-\pi/2 \le \sum_{i=0}^n (\gamma_i^N)^- \le 0 \le \sum_{i=0}^n (\gamma_i^N)^+ \le \pi/2.
\end{array} \tag{3}
$$

We can now rewrite the equations in (2) using the γ_i^N's to get:

$$
\begin{array}{rl}
q_0^N = & (1-p_{exp})q_0^N + \rho_d^N q_1^N + \gamma_0^N \\
q_1^N = & (1-\rho_d^N-\rho_u^N)q_1^N + \rho_d^N q_2^N + p_{exp} q_0^N + \gamma_1^N \\
\vdots & \\
q_i^N = & (1-\rho_d^N-\rho_u^N)q_i^N + \rho_d^N q_{i+1}^N + \rho_u^N q_{i-1}^N + \gamma_i^N \\
\vdots & \\
q_n^N = & (1-\rho_d^N)q_n^N + \rho_u^N q_{n-1}^N + \gamma_n^N.
\end{array} \tag{4}
$$

Although γ_i^N is a function, in general, of q_i^N, q_i^N,q_{i-1}^N,$q_i^{\bar{N}}$, $q_i^{\bar{N}}$, and $q_{i-1}^{\bar{N}}$, we can solve (4) by treating it as a constant, subject to the contraints in (3). This allows us to break the dependency between the equations for $q_0^B,\ldots,q_n^B$ and those for $q_0^G,\ldots,q_n^G$, and solve them separately. This makes the solution *much* simpler.

By rearranging the arguments, we can express q_n^N as a function of only q_{n-1}^N, ρ_u^N, ρ_d^N, and γ_n^N. By then substituting this expression (where the only unknown is q_{n-1}^N) for q_n^N in the equation for q_{n-1}^N and rearranging the arguments, we can express q_{n-1}^N in terms of q_{n-2}^N (and the constants ρ_u^N, ρ_d^N, γ_n^N, and γ_{n-1}^N). In general, we can compute q_i^N as a function of q_{i-1}^N (and the constants ρ_u^N, ρ_d^N, γ_i^N, and γ_{i-1}^N), and then substitute for it.

These calcuations (which are left to the full paper) give us the following equation for q_0^N:

$$
\begin{array}{l}
(1+\frac{p_{exp}((\sigma_N)^n-1)}{\rho_u^N-\rho_d^N})q_0^N = \\
1/2 + \sum_{i=1}^n \gamma_i^N \frac{i-\sum_{j=1}^i((\sigma_N)^{n-j+1}}{\rho_u^N-\rho_d^N}.
\end{array} \tag{5}
$$

Moreover, all the terms that are multipled by γ_i in (5) are negative, and, of these, the one multiplied by γ_n is the largest in absolute value. Given the constraints on $(\gamma_i^N)^+$ and $(\gamma_i^N)^-$ in (3), this means that we get a lower bound on q_0^N by setting $\gamma_n^N = \pi/2$, $\gamma_0^N = -\pi/2$, and $\gamma_i^N = 0$ for $i \neq 0, n$. This is quite intuitive: In order to make q_0^N as small as possible, we want all of the transitions from N to $\bar{N}$ to happen when the automaton is in state 0, and all the transitions from $\bar{N}$ to N to happen when the automaton is in state n, since this guarantees that the expected amount of time that the automaton spends in a state $i > 0$ is maximized. Similarly, to make q_0^N as large as possible, we should set $\gamma_0^N = \pi/2$, $\gamma_n^N = -\pi/2$, and $\gamma_i^N = 0$ for $i \neq 0, N$.

Making these choices and doing some algebra, we get that

$$
\begin{array}{l}
q_0^N \ge \frac{1}{2}(\frac{(\rho_u^N-\rho_d^N)-\pi(\sum_{i=1}^n \sigma_N^i - n)}{(\rho_u^N-\rho_d^N)+p_{exp}((\sigma_N^i)^n-1)}) \\
q_0^N \le \frac{1}{2}(\frac{(\rho_u^N-\rho_d^N)+\pi(\sum_{i=1}^n \sigma_N^i - n)}{(\rho_u^N-\rho_d^N)+p_{exp}((\sigma_N^i)^n-1)}).
\end{array}
$$

As we have observed before, $E_\pi[A[n,p_{exp},Pos,Neg,r_u,r_d]] = (1/2 - q_0^G)x_G + (1/2 - q_0^B)x_B$. Plugging in the upper bound for q_0^G and the lower bound for q_0^B gives us the required estimate for Proposition 3.3, and completes the proof.

4 Experimental Results

In the first part of this section, we examine the performance of $A[n,p_{exp},Pos,Neg,r_u,r_d]$ with n finite. Using our theoretical analysis, we come up with an estimate of the performance of the automaton, and show that our theoretical estimate is very close to what we observe in simulation. In the second part of the section, we show that the ideas underlying our automaton can be generalized in natural way to a setting where nature has more than two possible states.

4.1 Two states of nature

We focus mostly on scenarios where $\pi = 0.001$, as when nature changes too often, learning from the signals is meaningless (although even for a larger value of π, with a strong enough signal, we can get quite close to the optimal payoff; with smaller π the problem is easier). For simplicity, we also consider an example where $|x_B| = |x_G|$ (we used $x_G = 1, x_B = -1$, but the results would be identical for any other choice of values). We discuss below how this assumption influences the results.

Again, for definiteness, we assume that there are four signals, $1,\ldots,4$, which have probabilities 0.4, 0.3, 0.2, and 0.1, respectively, when the state of nature is G, and probabilities 0.1, 0.2, 0.3, and 0.4, respectively, when the state of nature is bad. We choose signal 1 to be the "good" signal (i.e., we take $Pos = \{1\}$), and take signal 4 to be the "bad" signal (i.e., we take $Neg = \{4\}$), and take $r_u = r_d = 1$. We ran the process for 10^8 rounds (although the variance was already quite

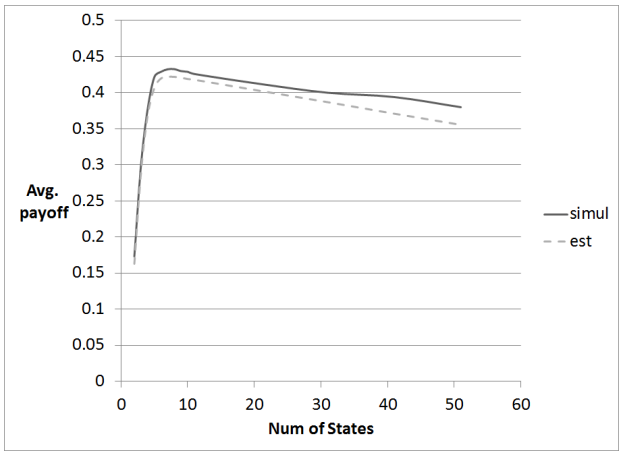

Figure 1: Average payoff as a function of the number of states

small after 10^6 rounds, and we got good payoff even with 10^5, which is approximately 100 switches between states), using a range of p_{exp} values, and took the result of the best one. We call this $p_{exp}^{opt}(n)$. As can be seen in Figure 1, the automaton $A[n, p_{exp}^{opt}(n), \{1\}, \{4\}, 1, 1]$ does well even for small values of n. The optimal expected payoff for an agent that knows nature's state is 0.5. With 4 states, the automaton already gets an expected payoff of more than 0.4; even with 2 states, it gets an expected payoff of more than 0.15.

We also compared the simulation results to the lower bound given by Proposition 3.3 (the "est" line in Figure 1). As can be seen, the lower bound gives an excllent estimate of the true results. This is actually quite intuitive. The worst-case analysis assumes that all transitions from B to G happen when the automaton is in state 0, and all transitions from G to B happen when the automaton is in state n. But when nature is in state B, a "good" automaton should spend most of its time in state 0; similarly, when nature is in state G, a "good" automaton should spend most of its time in state n (as a result of getting good signals). Thus, the assumptions of the worst-case analysis are not far from what we would expect of a good automaton.

Equation (1) suggests that while nature is in state G, as the number of states grow, the loss term (that is, the minus term in the $x_G/2$ factor) decreases rapidly. The exact rate of decrease depends on σ_G. We can think of σ_G as describing the quality of the signals that the automaton pays attention to (those in Pos and Neg) when nature is in state G. From equation (1), we see that as the number of states grows this loss reduces to $\frac{\pi\sigma_G}{p_{exp}(\sigma^G-1)}$. So the agent's optimal choice is to set the parameters of the automaton so that the ratio is as large as possible. This allows him to both decrease the loss as fast as possible (with regards to number of states he needs) and to get to the minimal loss possible.

There is of course a tradeoff between σ_G and σ_B. The loss while nature is in state B also decreases rapidly with the number of states, and the rate is dependent on $1/\sigma_B$. As the number of states grows this loss reduces to $\frac{p_{exp}+\pi(\frac{\sigma_B}{1-\sigma_B}-n)}{p_{exp}+\rho_d^B-\rho_u^B}$.

The graph also shows that, somewhat surprisingly, having too many states can hurt, if we fix Pos, Neg, r_u, and r_d. The lower bound in (1) actually bears this out. The reason that more states might hurt is that, after a long stretch of time with nature being in state G, the automaton will be in state n. Then if nature switches to state B, it will take the automaton a long time to get to state 0. All this time, it will get a payoff of -1. (Of course, if we allowed the automaton a wider range of strategies, including not using some states, then having more states can never hurt. But we are considering only automata of the form $A[n, p_{exp}, Pos, Neg, r_u, r_d]$.) In a sense, this can be viewed as an explanation of the *recency bias* observed in real decision makers—the tendency to place significantly more weight on recent observations. While the recency bias has often been viewed as inappropriate, these experiments can show that it can be helpful. With more states, more can be remembered, and it becomes harder to "convince" the automaton to change its mind when nature's state actually changes. Having less memory, and thus being more easily influenced by the last few signals, may be more adaptive. By way of contrast, in Wilson's 2002 model, nature is static. The optimal automaton in Wilson's setting displayed a strong *first-impression* bias: early observations were extremely influential in the final outcome, rather than recent observations.

We can do a little better by decreasing r_u, thus increasing the amount of time it will take the automaton to get to state n when nature is in state G. While this helps a little, the effect is not great (we provide more details in the full paper). Moreover, we do not think it is reasonable to expect resource-bounded agents to "fine-tune" the value of parameters depending on how many states they are willing to devote to a problem. Fortunately, as our experimental results show, they do not need to do such fine-tuning for a wide range of environment settings. There is another tradeoff when choosing the value of p_{exp}, which lies at the heart of the *exploration-exploitation* dilemma. Clearly, if the automaton is in state 0 and nature is in state G, the automaton wants to explore (i.e., move to state 1 and play R) so as to learn that nature is in state G. There are two reasons that the automaton could be in state 0 while nature is in state G. The first is that the automaton gets a sequence of "bad" signals when nature is in state G that force it to state 0. Clearly this is less likely to happen the more states the automaton has. The second is that nature may have switched from B to G while the automaton was in state 0.

Since nature switches from B to G with probability π, a first cut at the exploration probability might be π. However, this first cut is too low an estimate for two reasons. First, the fewer states an automaton has, the more sensitive it is to "bad" signals. Thus, the fewer states an automaton has, the more it should explore. Second, the cost of exploring while nature is in state B is small in comparison to the gain of exploring and discovering out nature has switched to state G. Again, this suggests an increase in the exploration probability. Indeed, we observe that as π gets smaller the optimal p_{exp} value gets smaller, but not in the same ratio. The optimal agent explores less, but still chooses p_{exp} higher than π. For example, with $n = 6$, when changing π from 0.001 to

0.0001 the optimal p_{exp} only changed from 0.023 to 0.008.

In our simulation, we chose the optimal value of p_{exp} relative to the number of states; this value did not change significantly as a function of the number of states or of the signal profiles. For example, taking $p_{exp} = 0.03$ resulted in payoffs very similar to those with the optimal value of p_{exp} for all $n \geq 5$, and for a wide range of signal profiles while fixing n to 6. This robustness supports our contention that agents do not typically need to fine tune parameter values.

4.2 More states of nature

We now consider a setting where nature can have more than two states. Specifically, we allow nature to have $t+1$ states, which we denote $B, G_1, G_2, \ldots, G_t$. In each state, there is probability of π of transitioning to any other state. Again, we have k signals, and the probability of observing signal i is state dependent. The agent has $t+1$ available actions $\{S, E_1, E_2, \ldots, E_t\}$. As before, S is the "safe" action; playing S gives the agent a payoff 0, but also results in the agent receiving no signal. Playing E_i if the state of nature is B result in a payoff of $x_B < 0$; playing E_i when the state of nature is G_i gives the agent a payoff of $x_G > 0$; playing E_i when the state of nature is G_j for $i \neq j$ gives a payoff of 0.

We generalize the family of automata we considered earlier as follows. The family we consider now consists of product automata, with states of the form $(s_0, s_1, \ldots, s_t)$. Each s_i takes on an integer value from 0 to some maximum n. Intuitively, the s_0 component keeps track of whether nature is in state B or in some state other than B; the s_i component keeps track of how likely the state is to be G_i. If $s_0 = 0$, then the automaton plays safe, as before. Again, if $s_0 = 0$, then with probability p_{exp} the automaton explores and changes s_1 to 1. If $s_1 > 0$, then the automaton plays the action corresponding to the state of nature G_i for which s_i is greatest (with some tie-breaking rule).

We did experiments using one instance of this setting, where nature was in one of five possible states—4 good states and one bad state—and there were six possible signals. We assumed that there was a signal p_i that was "good" for state G_i: it occurred with probability .6 when the state of nature was G_i; in state G_j with $j \neq i$, p_i occurred with probability 0.08; similarly, there was a signal that was highly correlated with state B; the sixth signal was uninformative with probability 0.08 in all states. We considered an automaton for the agent where each of component of the product had five states (so that there were $5^5 = 3125$ states in the automaton. In this setting, the optimal payoff is 0.8. The automaton performed quite well: it was able to get a payoff of 0.7 for an appropriate setting of its parameters. Again, we discuss this in more detail in the full paper.

5 Conclusion

We have shown that observed human behavior that appears to be irrational can actually be understood as the outcome of a resource-bounded agent that is maximizing his expected utility. We plan to use this approach of looking for simple, easy-to-implement strategies with low computational cost that perform well in real scenarios to explain other observed biases in decision making.

Acknowledgments: We thank Ido Erev and the reviewers for useful comments. Halpern and Seeman were supported in part by NSF grant IIS-0812045, and AFOSR grants FA9550-08-1-0438 and FA9550-09-1-0266, and by ARO grant W911NF-09-1-0281. Pass was supported in part by a Alfred P. Sloan Fellowship, Microsoft New Faculty Fellowship, NSF CAREER Award CCF-0746990, AFOSR YIP Award FA9550-10-1-0093, and DARPA and AFRL under contract FA8750-11-2- 0211. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US government.

References

Biele, G.; Erev, I.; and Ert, E. 2009. Learning, risk attitude and hot stoves in restless bandit problems. *Journal of Mathematical Psychology* 53(3):155–167.

Chen, W.; Liu, S.-Y.; Chen, C.-H.; and Lee, Y.-S. 2011. Bounded memory, inertia, sampling and weighting model for market entry games. *Games* 2(1):187–199.

Erev, I.; Ert, E.; Roth, A.; Haruvy, E.; Herzog, S.; Hau, R.; Hertwig, R.; Stewart, T.; West, R.; and Lebiere, C. 2010. A choice prediction competition: Choices from experience and from description. *Journal of Behavioral Decision Making* 23(1):15–47.

Erev, I.; Ert, E.; and Roth, A. 2010a. A choice prediction competition for market entry games: An introduction. *Games* 1:117–136.

Erev, I.; Ert, E.; and Roth, A. 2010b. Market entry prediction competition web site. Available online at https://sites.google.com/site/gpredcomp/.

Hellman, M. E., and Cover, T. M. 1970. Learning with finite memory. *The Annals of Mathematical Statistics* 41(3):pp. 765–782.

Kahneman, D., and Tversky, A. 1979. Prospect theory: an analysis of decision under risk. *Econometrica* 47(2):263–292.

Neyman, A. 1985. Bounded complexity justifies cooperation in finitely repated prisoner's dilemma. *Economic Letters* 19:227–229.

Papadimitriou, C. H., and Yannakakis, M. 1994. On complexity as bounded rationality. In *Proc. 26th ACM Symposium on Theory of Computing*, 726–733.

Resnick, S. I. 1992. *Adventures in Stochastic Processes*. Birkhauser.

Rubinstein, A. 1986. Finite automata play the repeated prisoner's dilemma. *Journal of Economic Theory* 39:83–96.

Whittle, P. 1988. Restless bandits: Activity allocation in a changing world. *Journal of Applied Probability* 287–298.

Wilson, A. 2002. Bounded memory and biases in information processing. Manuscript.