Balancing Innovation and Exploitation in a Social Learning Game.

Eric Raboin, Ryan Carr, Austin Parker, Dana Nau
Department of Computer Science, University of Maryland
College Park, MD, 20742, USA
{eraboin,carr2,austinjp,nau}@cs.umd.edu

Abstract

This paper examines the value of innovation within a culture by looking at "innovate" moves in the Cultaptation Project's social learning game (Boyd et al. 2008). We produce a mathematical model of a simplified version of this game, and produce analytic methods for determining optimal innovation behavior in this game. In particular, we provide an efficient algorithm for determining how to balance innovation with the need to exploit one's accumulated knowledge. We create an agent for playing the social learning game based on these results.

Intro

The development of innovations is important to a society, yet the exploitation of those same innovations is clearly also important and many times, a tradeoff must be made between the two. This paper examines the utility of a simple kind of innovation in the Cultaptation Project's social learning game, and shows how we can solve the problem of deciding how to balance innovation with the exploitation of accumulated knowledge.

The European Commission's Cultaptation Project was created to address the evolution of human cultures, and the project's researchers have created a game to examine the relative merits of social learning strategies (Boyd et al. 2008). Each player in the game has three kinds of possible moves: *innovate*, *observe*, and *exploit*. The game's authors devised these moves to be simple analogs of the following three kinds of activities: innovating (spending time and resources learning something new), observing (learning something from another player), and exploiting (using the learned knowledge). At each step of the game, each player must choose one of these three activities, and the algorithm or rules that a player uses for making this choice are the player's "social learning strategy."

Finding a good social learning strategy is clearly a hard question, particularly because the game is defined in such a way that the utility functions are *a priori* unknown. However, hidden within the hard question of determining social strategies is *another* hard question of determining the value of innovation: if one decides to use innovation instead of observation, what is the best way to accomplish this? This second question is the focus of our study.

The contributions of this paper are:

- A mathematical model for a simplified versions of the Cultaptation game called the SRI game.
- A formula for determining optimal strategies in the SRI game that requires access to the underlying probability distribution.
- Experiments detailing the performance of the *smart-innovator* agent, showing it to behave near optimally with wide varieties of underlying probability distribution functions.
- An analysis of an alternative scoring system in the SRI game, which more directly relates to the Cultaptation's version of the social learning game by using per-round payoff rather than total-lifetime payoff.
- A polynomial-time algorithm computing optimal behavior in the SRI under this new scoring system. The algorithm uses state-aggregation and a variant of Dijkstra's algorithm to find an optimal sequence of moves.

Definitions

Social Learning Game

In this section, we lay out the Cultaptation Project's social learning game from (Boyd et al. 2008). We assume a probability distribution π over the integers. There are n_b exploit moves, with each move's utility value being drawn from π. We further assume a *change probability* of c. On each round of game play, with probability c, each move's value will be replaced with another drawn from π. Let $v_{i,j}$ be the value of move i in round j. When a player makes an exploit move i on round j they receive the utility $v_{i,j}$. A player's total utility will be the sum of its utility on every round.

There will be n agents in this environment. In the general game, each agent a_i can make two other moves apart from the n_b exploit moves: innovate (I) and observe (O). Upon making an I move in round r, the value $v_{i,r}$ of a randomly chosen move i gets added to the agent's repertoire as the value of move i. The agent receives no utility on rounds where she makes an I move. Upon making an O move, an agent will get to observe the value received by some other agent who made any exploit move on the last round. Agents

round #	1	2	3	4	5	…	k
I1's move	I	1	1	1	1	…	1
I1's Utility	0	3	6	9	12	…	$3 \cdot (k-1)$
I2O's move	I	I	O	3	3	…	3
I2O's Utility	0	0	0	8	16	…	$8 \cdot (k-3)$

Table 1: The sequence of moves from Example 1 and their utility values.

receive no utility for O moves. When a move is observed, the observed move's value on the last move ($v_{i,r-1}$) is added to the agent's repertoire. If no other agent made an exploit move last round, the observing agent receives no information.

The agent may only make I or O moves and moves from the repertoire (having been put there through either an I or an O move). Notice that because of the probability of change c, the value of $v_{i,r}$ on the current round r may not be the same as what is stored in the agent's repertoire.

Example 1 *Consider two strategies: the innovate-once strategy (hereafter I1) which innovates exactly once and exploits that innovated move for the rest of the game, and the innovate-twice-observe-once strategy (hereafter I2O) which innovates twice, observes once, and exploits the higher valued move for the rest of the game. For simplicity of exposition, we allow only four exploit moves: 1, 2, 3, and 4; and exactly two agents, one I1 and one I2O. We suppose a uniform distribution over* $[1, 10]$ *(with mean* 5*) and a probability of change of* 0*. Suppose the initial utility values for the moves are:* $v_{1,0} = 3, v_{2,0} = 5, v_{3,0} = 8, v_{4,0} = 5$*. On the very first move, I1 will make an innovate, which we suppose gives I1 the value of move 1, putting* $v_{1,0}$ *as the value for move 1 in I1's repertoire. On every sequential move, I1 will make move 1, exploiting the initial investment. If the agent dies k rounds latter, then the history of moves and payoffs will be that given in Table 1; giving a utility of* $3 \cdot k - 1$.

In contrast, I2O will make an innovate, giving the value for move 3: $v_{3,1} = 8$*, then makes another innovate giving the value for move 2:* $v_{2,2} = 5$*, and finally observes. On move 2, I1 made move 1, and since these are the only two agents, this was the only exploit move made. Therefore I2O observes that another agent got a value of 3 from move 1 last round. On move 4, I2O's repertoire consists of* $\{v_{1,2} = 3, v_{2,2} = 5, v_{3,1} = 8\}$*. Since the probability of change is* 0*, the obvious best move is move* 3*, which I2O makes for the rest of her life. The average per-round utility of I2O on round k is* $8 \cdot k - 3$*, so for rounds 2 to 4, I2O will actually have a worse than I1, while after round 4, the utility of I2O will actually be higher.*

There are many potential setups for these sorts of games. Generally the *de facto* objective is to acquire more utility than other players, though one can imagine games where the objective is for an individual or group to maximize utility. In the Cultaptation social learning competition, an evolutionary setup is used. Each game starts with 100 agents. Each round each agent has a 2% chance of dying. On death, an agent is replaced, and if there is no mutation, the strategy used by this newborn agent is chosen from the agents currently alive according to their average lifetime utility (the new agent is naïve and shares nothing except social learning strategy with its parent). In this game, the strategy with the highest average per round utility is the most likely to propagate. Mutation happens 2% of the time, and if there is a mutation, then one of the competing strategies is chosen at random and added to the pool (without regard for any strategy's performance). Through mutation, new strategies can be introduced into otherwise homogeneous populations.

In the social learning competition, there are two sorts of games into which an agent may be placed. First is a pairwise contest, where one contestant strategy starts with a dominant population of agents, and an invader strategy is introduced only through mutation. The second is the melee contest, which starts with a dominant strategy of simple asocial learners, and is invaded by several contestant strategies through mutation. We call any instance of either an invasion or a melee game a *Culptaptaion game*.

Simplified Games

In this paper, we will examine a simplified form of the Cultaptation social learning game.

The simplified form is a finite-round innovation game. In this game, we take the original social learning game and eliminate observe moves, the probability of change, and the possibility of death. We further set the number of rounds to some positive integer l. We call this the set-round-innovation game (SRI game). In the SRI game there is only ever one agent. The goal of the SRI game is to achieve maximal utility over the entire game.

Another simplified form of the game, similar to the first, we have called the variable-round-innovation game (VRI game). In this game, agents die with probability 0.02 each round, instead of living a fixed number of rounds. The VRI game was discussed in a previous paper by the same authors (Carr et al. 2008).

Problem Description

The motivation for this work comes from the following result, which implies that innovation moves must be performed even in the version of the social learning game on which the social learning competition will be based.

Proposition 1 *If no agent ever innovates in any form of the social learning game, then the utility of all agents will always be zero.*

Proof: All utility comes from exploiting moves available in an agent's repertoire. Any move in the repertoire due to an observation requires there to have been an earlier exploit. Any move in a repertoire was therefore originally discovered by an innovate move. Thus any non-zero utility implies that at least one agent made at least one innovate at one point.

This implies that innovation is necessary in all social learning games, including the Cultaptation game. Thus even in the full Cultaptation game there is motivation to determine how many times one must innovate in order to optimize one's utility.

A quick analysis of the SRI game determines that only certain kinds of innovations are useful. Consider any innovate move made after an exploit move. It is possible that that innovation discovers a move with higher utility than the previous exploit. In this case, we would want to have innovated before exploiting – in fact, in the general case, all innovates should come before any exploits. Therefore we need only consider strategies which innovate a given number of times.

The question remains, however, how many innovates is best? This is our problem.

Definition 1 (Optimal Innovation Problem) *Given a probability distribution π, what is the optimal number of innovates in the SRI game?*

Analytic Results

SRI Game

In this section, we introduce and prove a formula which allows for the computation of the optimal number of innovates for a given distribution in the SRI game.

To present these results, we will represent the utilities of each of the n actions as a set $V = v_1, v_2, \dots, v_n$ and we will assume, without loss of generality, that $v_1 \leq v_2 \leq \dots \leq v_n$.

Possible Strategies First, we examine the strategies that players may adopt for agents in this version of the game. Players must choose either I or E for each of l rounds. Thus, there are 2^l possible strategies. Note, however, that there are far fewer intelligent strategies. For instance, I is the only move that makes sense for the first round, since the agent does not know any actions and, thus, cannot choose one to exploit. Similarly, E is the only move that makes sense for the last round, since the agent would not have the opportunity to use any action it learned by choosing I.

Finally, note that, since action utilities do not change, it never makes sense to choose a strategy with an I move following an E move, since this strategy would be guaranteed to do at least as well by swapping the two moves, since the total number of times the agent exploits remains the same, and the utility of the action it chooses to exploit is at least as high. Thus, the only strategies worth considering are those that begin by innovating k consecutive times, where $0 < k < l$, and then exploit $l - k$ consecutive times. For the rest of the analysis, we will refer to the strategy that begins by innovating k times as S_k.

Expected Utility of a Strategy Since all of the strategies we are concerned with contain some number of I moves followed by E moves, we can obtain the expected utility of a strategy by multiplying the utility of the best action found with k innovates by $l - k$, the number of times the strategy exploits:

Proposition 2 *Let $F(k, V)$ be the expected utility of the best action found with k random draws from V. Then the expected utility of S_k is:*

$$E(S_k) = (l - k)F(k, V)$$

Now, however, we need to derive $F(k, V)$. Since F is the expected maximum value of k draws from V, we can obtain it by summing the maximum value of every possible sequence of k innovates, and dividing by the number of possible sequences. Our assumption that $v_1 < v_2 < \dots < v_n$ will help here; we note that, if on a given sequence of innovates the maximum utility discovered is v_i, then the other utilities discovered is some permutation of $k-1$ values from $v_1, v_2, \dots, v_{i-1}$. Since there are $P(i-1, k-1) = \frac{(i-1)!}{(i-k)!}$ of these permutations and the maximum value can be found on any one of the k innovates, there are $k\frac{(i-1)!}{(i-k)!}$ ways to discover a maximum value of v_i. Since there are $P(n, k) = \frac{n!}{(n-k)!}$ possible sequences of discoveries, we know that

$$F(k, V) = \frac{\sum_{i=k}^{n} \left(k\frac{(i-1)!}{(i-k)!} v_i \right)}{\frac{n!}{(n-k)!}}. \quad (1)$$

Now that we have $F(k, V)$, and we know that the only strategies worth considering are ones that innovate for k moves and then exploit for $l - k$ moves, we can easily calculate the optimal strategy by finding the value of k that maximizes the strategy's expected value.

Theorem 1 *Given a distribution V and lifetime l, the optimal strategy for the SRI game is to innovate a number of times equal to*

$$argmax_k\left((l - k)F(k, V)\right) \quad (2)$$

and exploit thereafter.

This theorem follows from the discussion above.

When to Stop Innovating? Now that we can find the expected utility of strategies, we can tell if adding more innovate moves to a strategy will give us a higher expected utility. This will be helpful if, for example, we are building an agent that must decide each turn whether to keep innovating or not. If we want to get the maximum expected utility, we are interested in the value of k such that performing $k + 1$ innovates does not improve out expected utility. In other words,

$$(l - k)F(k, V) \geq (l - k - 1)F(k + 1, V). \quad (3)$$

Experimental Results for SRI Game

To test the effectiveness of Formula 1 and to ensure that no mistakes were made in its calculation, we compared the expected value predicted by the formula to actual expected values achieved in simulation experiments. To do this, we chose two distributions for π: normal and exponential distributions, each with mean 50 and variance 850. For each distribution, we considered games which lasted exactly 100 rounds, and computed the expected average per round utility of an agent which performs i innovations by averaging that agent's utility over a total of 500 thousand simulations. This value is reported as "Measured EV". We then used Formula 1 to compute another expected utility for innovating after i rounds. These computed values are reported in the figure as "Predicted EV". Figure 1 shows the result of this experiment both distributions.

We notice that in the figure, the predicted values match up exactly with the meausred values for both distributions. This we take as evidence that Formula 1 is correctly computed.

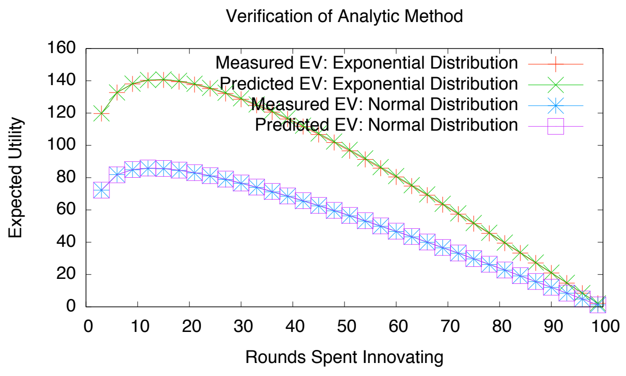

Figure 1: Comparing the utility predicted by equation 1 with the utility achieved in simulations when the underlying distribution is both normal and exponential. Notice that the measured values are almost exactly the same as the predicted values, such that the lines showing the measured expected value are overlaid by the predicted values.

Estimating an Optimal k

Thus far, we have only tried computing the optimal number of innovate moves when the set of actions V is known in advance. For the Cultaptation game described earlier in this paper, as well as in many real world situations, this is an unreasonable assumption. Instead, an agent must estimate the optimal number of innovations given only what it has learned from previous moves in the game.

The *smart-innovator* strategy estimates V based on the actions it has learned from previous innovate moves in the game, and bases its decisions on that. Let $L = l_1, l_2, \ldots, l_k$ be the set of actions that the agent has already innovated by round k.

We could just assume that $V = L$, and decide whether to innovate further by computing $F(k+1, L)$. However, since L has only k values, innovating more than k times would always appear pointless. It is not alway pointless to innovate more than k times, so this would be suboptimal. What we can do instead is use L to estimate π, the underlying distribution, and then generate an approximation (or several) of V from the estimate of π.

For a given assumed family of underlying probability distribution (i.e. normal, uniform, exponential) over V, we can estimate the distribution parameters by measuring the mean and standard deviation of L. By sampling from this new distribution, we can create a new set of actions V' that models the true distribution of V. We can then compute the inequality $(l-k)F(k, V') \geq (l-k-1)F(k+1, V')$ (Formula 3), and decide whether to continue innovating based on that.

No single V' is guaranteed not to yield the same optimal number of innovates as V, but by generating multiple V's and averaging the results we hope to reach a close approximation. Algorithm 1 shows how this procedure works under the assumption that π is some normal distribution.

For situations when the family of the underlying distribution is not known *a priori*, a χ^2 goodness-of-fit test can be used to compare several distributions (normal, uniform, etc), and determine which is best. From this estimation one can form a new distribution that may be representative of V, thereby eliminating the need to assume one particular family of probability distributions.

Algorithm 1 Determine an estimate of the optimal number of innovates with observed values $L = \{l_1, \ldots, l_k\}$, and number of rounds l, under the assumption that π is a normal distribution.

```
Smart Innovator($L$,$l$)
  Let $m$ be the mean of $L$.
  Let $\sigma$ be the standard deviation of $L$
  Let $normal(m, \sigma)$ be a normal PDF with mean $m$ and standard deviation $\sigma$.
  Let $LHS = 0$ {Left hand side of inequality}
  Let $RHS = 0$ {Right hand side of inequality}
  **for** 100 trials **do**
    Create $V' = \{v_1, \ldots, v_{n_b}\}$ where each $v_i$ is a random draw from $normal(m, \sigma)$.
    Let $LHS = LHS + (l-k)F(k, V')$
    Let $RHS = RHS + (l-k-1)F(k+1, V')$
  **end for**
  If $LHS < RHS$ innovate this round.
  Otherwise exploit.
```

Experimental Results for Smart-Innovator

To confirm that the *smart-innovator* strategy produces close to optimal results, we compared the average number of rounds spent innovating by agents performing this strategy, versus the known optimal number of innovate moves computed by the formula earlier in this paper. For simplicity, we assume the agent is given the family of distribution (normal, laplace, etc.), but not its parameters. For each of these distributions, the agent was required to innovate for at least 5 moves before estimating V'.

The results in Table 2 show the average performance for 50 trials of a 100 round SRI game. Notice that the *smart-innovator* strategy is able produce close to optimal results for a range of standard deviation values, even though it was not given those values in advance.

For a single run of the SRI game, the *smart-innovator* strategy may produce a higher utility than expected for the optimal number of innovates. This is because the expected value is a prediction of the average case, not a prediction of the best case. It is entirely possible for a single run of *smart-innovator* or any other strategy to draw higher or lower utility than expected. This is exhibited in the standard deviation values column of Table 2.

In (Carr et al. 2008) a similar results was found for the variable-round-innovation game. While introducing probability of death did increase the variance in performance results, on average the *smart-innovator* strategy was still able to approximate the optimal number of innovates.

Distribution	Mean	Stdev	Optimal I_o Innovates	Average SI Innovates	Stdev of SI Innovates	Expected I_o Utility	Average SI Utility	Stdev of SI Utility
Uniform	50	40	11	10	1.15	96.07	95.76	8.18
		80	12	12	1.34	147.53	147.73	16.04
		160	13	13	1.28	250.8	246.4	28.3
Normal	50	40	15	14	1.46	101.59	103.52	17.85
		80	17	16	1.48	160.65	165.01	37.83
		160	18	18	1.95	279.75	277.16	76.41
Laplace	50	40	19	17	1.93	106.5	111.17	44.26
		80	21	21	1.59	172.7	161.86	45.98
		160	23	23	1.36	306.1	304.94	161.61
Exponential	50	40	20	20	0.99	120.1	119.46	24.66
		80	22	23	0.81	200.41	180.24	56.37
		160	24	24	1.28	361.81	373.47	150.03

Table 2: Table showing the optimal number of innovates in the SRI game (I_o), compared with the average number of innovates performed by the *smart-innovator* strategy (SI). Note that regardless of the distribution or its parameters, the SI strategy either correctly estimates, or comes very close to correctly estimating the optimal number of innovate moves.

Optimizing Average Per-round Utility

The strategies presented thus far attempt to maximize the total lifetime utility of an agent in the SRI version of the social learning game. However, in the social learning competition, a winning strategy must achieve the largest agent population across many trials. Therefore, the goal of a winning strategy should be to maximize an agent's probability of reproducing.

In the competition, the probability agent i will reproduce on round k is defined as

$$R_{ik} = \frac{P_{ik}}{\sum_j P_{jk}} \tag{4}$$

Where P_{ik} is the per round utility of agent i on round k, equal to the agent's total utility divided by the number of rounds it has been alive. The average number of times agent i will reproduce during an n round game is simply $\sum_{k=0}^{n} R_{ik}$. Thus, an agent will reproduce at a rate proportional to its average per-round utility, divided by the per-round utility of all the agents in the environment.

During the competition, a single agent cannot know the value P_{ik} for any of the other agents. The agent is also unaware of the respective population densities for each strategy, or the exact round number of the game. This makes it difficult for an agent to know its probability of reproducing during a particular round. For this reason, we make a simplifying assumption that the denominator in equation 4 is, on average, constant over the lifetime of an agent. This will allow us to maximize the average per-round utility, as an approximate means of maximizing the reproduction rate.

The average per-round utility for an agent i in an n round game is defined as the average of its per-round utility for every round in the game

$$A_i = \frac{1}{n} \sum_{k=1}^{n} P_{ik} \tag{5}$$

This formula can be rewritten as an expected value calculation in terms of the sequence of moves M, and a set of actions V

$$A(M, V) = \frac{1}{n} \sum_{k=1}^{n} \frac{1}{k} \sum_{j=1}^{k} M_j * F(K_j, V) \tag{6}$$

Where $M_j \in \{0, 1\}$ is equal to 1 if the move on round j is an exploit, and 0 if the move is an innovate. The value K_j is the number of innovate moves performed by round j, and can be derived from M. The formula $F(k, V)$ returns the expected value of k innovates, and is unchanged from the earlier discussion.

Using equation 6 we can evaluate the average per-round utility of an arbitrary sequence of innovates or exploits. It turns out that maximizing this formula is *not* the same as maximizing the total lifetime utility. The following counterexample shows that these two measurements differ.

Example 2 *Consider a four round SRI game with a single agent, where the set of values learned through innovating is V = {0, 48}. Observe that no more than two innovate moves is ever desired, since there are only two distinct values in V. The expected value of a single innovate is 24, and the expected lifetime utility from one innovate followed by three exploits is 72. The expected lifetime utility from two innovates followed by two exploits is 96. Therefore, to maximize the total lifetime utility, the agent should innovate exactly twice.*

The average per-round utility for one innovate followed by three exploits, according to equation 7, is 11.5. The average per-round utility for two innovates followed by two exploits, according to the same equation, is 10. Therefore, the number of innovates required to maximize average per-round utility is one, different from the number required to maximize total lifetime utility. In this example, maximizing the value of one measure will decrease the value of the other. No other sequence of moves will perform any better.

Example 2 shows that a new way to compute the optimal number of innovate moves is needed. It would be relatively easy to extend the work from the SRI games to include this

Moves	Per-Round Utility	Total Lifetime Utility
$IEIEEE...$	69.25	1782.5
$IEEIEE...$	69.08	1765.0
$IIEEEE...$	69.02	1800.0

Table 3: Table showing the top three strategies for maximizing the per-round utility in a 20 round SRI game, with $V = \{65, 100\}$.

new calculation. However, this will not work, because a previously held assumption is no longer true.

Proposition 3 *There exists a set of actions V and number of rounds n such that the sequence of moves that maximizes the average per-round utility contains an exploit move before an innovate move.*

This proposition differs from before, where the goal was to maximizing total lifetime utility, and we could guarantee that all optimal strategies were of the form of k innovates followed by $n - k$ exploits. When attempting to maximize the average per-round utility, a simple 20 round SRI game with innovate values $V = \{65, 100\}$ is an exception to this previously held rule. The strategy $innovate$, $exploit$, $innovate$, and then $exploit$ 17 times outperforms all other sequences of moves. The search can no longer be limited to strategies that innovate k times then exploit.

This problem could be solved by performing a brute force search over all sequences of innovate and exploit moves, and choosing the sequence that maximizes the per-round utility. However, there are 2^n such sequences for an n round game, so for large values n this method is simply intractable.

Finding M in Polynomial Time

Given an n round SRI game with set of actions V, there is a polynomial time algorithm for computing the sequence of innovate and exploit moves M that maximizes the average per-round utility of an agent. While there are 2^n possible sequences to evaluate, exponential time complexity can be avoided by reducing the search to a polynomially sized graph problem. The algorithm takes advantage of the fact that the expected value of an exploit move depends only on the number of innovates performed and the round number.

First, observe that equation 6 can be simplified by factoring out the summation over k

$$A(M, V) = \frac{1}{n} \sum_{j=1}^{n} M_j * F(K_j, V) * \sum_{k=j}^{n} k^{-1} \qquad (7)$$

Several parts of equation 7 can now be pre-computed. Let F be set of values returned by the expected value computation $F(k, V)$ for all values $1 \leq k \leq n$, and let S be the set of values computed by the finite series $\sum_{k=j}^{n} k^{-1}$ for all rounds $1 \leq j \leq n$. Both sets are at most length n, and can be computed in polynomial time. Let $A = F \times S$ (the Cartesian product of F and S). The set A is also polynomially bounded, and represents the expected utility of any exploit move given k innovates and j rounds.

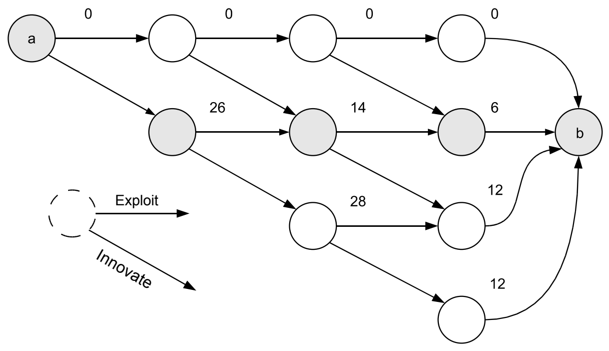

Figure 2: The graph constructed to find M for the four round SRI game described in example 2. The sequence $innovate$, $exploit$, $exploit$, $exploit$ maximizes the average per-round utility. A variant of Dijkstra's algorithm is able to find this in polynomial time, by finding the path with greatest total edge weight. Only edge weights for the horizontal exploit moves are shown.

According to equation 7, innovate moves provide no direct utility, while exploit moves provide the utility derived directly from each element in A. That is, for $(F(k, V), \sum_{i=j}^{n} i^{-1}) \in A$, an exploit move at round k after j innovates contributes exactly $F(k, V) \cdot \sum_{i=j}^{n} i^{-1}$ to the sum in $A(M, V)$ (Equation 7). For this reason we will focus only on the states associated with elements in A. We can construct a graph G that represents the $(n^2 + n)/2$ possible states of this kind. Let there be a vertex in G for every element in A, connected by edges representing innovate and exploit moves.

An innovate edge in G should connect the vertex representing k innovates and round j to the vertex for $k + 1$ innovates and round $j + 1$. An exploit edge should connect the vertex for k innovates and round j to the vertex for k innovates and round $j + 1$. Innovate moves have an edge weight of zero, while exploit moves have an edge weight equal to the expected utility defined in A.

The above construction should produce a directed acyclic graph with $(n^2 + n)/2$ vertices and $n^2 - n$ edges. Let the vertex representing round one be known as vertex a. There should be n terminal vertices representing the last round in the game, where $j = n$. To complete the graph, connect each of the terminal vertices to a dummy vertex with an innovate and an exploit edge, applying the same rule for edge weights used earlier. Let this dummy vertex be known as vertex b. Since there are only polynomial number of vertices and edges, and A can be computed in polynomial time, complete construction of G should happen in polynomial time.

Lemma 1 *For every unique n-length sequence M, there exists a corresponding path in G from vertex a to vertex b.*

Proof: Every path from a to b in G is of length n, because all vertices corresponding to rounds $j < n$ lead to vertices corresponding to rounds $j + 1$. Every vertex except b has exactly one innovate and one exploit edge, so there should be 2^n possible ways to traverse the graph from vertex a to vertex b.

Lemma 2 *The sum of the edge weights for a path from a to b divided by n is equal to the expected average per-round utility for the corresponding sequence M.*

Proof: Every exploit edge has a weight equal to the expected value of an exploit move for j^{th} round and the k^{th} innovate, as defined in equation 7. Every exploit edge will correctly lead to the vertex for the next round, and the k^{th} innovate. Every innovate edge will correctly lead to the vertex for the next round, and $k+1$ innovates.

The problem of finding an optimal M can now be solved by using a variant of Dijkstra's algorithm on the graph we constructed. We can extract M from the graph by searching for the path from a to b that maximizes the value of the edge weights. Dijkstra's algorithm can be performed in polynomial time, and extracting the sequence M can be done in linear time.

Theorem 2 *Given an n round SRI game with set of actions V, the sequence M that maximizes the expected value of the average per-round utility of an agent can be computed in polynomial time.*

Proof: Sets F and S can both be computed in linear time if the factorial values are pre-computed. Thus, computing the set of states A can be done in $O(n+n^2)$. Construction of G can be done in time linear to the number of states in A, and produces a graph with n^2 edges and $(n^2+n)/2$ vertices. Performing Dijkstra's algorithm on a graph with edges M and vertices N can be done in time $O(|M| + |N|log|N|)$, so the complexity of running the algorithm on G should be $O(n^2 + (n^2+n)log(n^2+n))$. As each individual step can be done in polynomial time, the sum of all the steps should also be polynomial.

Related Work

This work addresses a simplified version of the Cultaptation social learning game, first discussed in (Carr et al. 2008). The authors are aware of no other work on this particular game, although there are related problems in anthropology, biology and economics, where effective social learning strategies and their origins are of particular interest.

The social learning competition attempts to shed light on an open question in behavioral and cultural evolution. Despite the obvious benefit of learning from someone else's work, several strong arguments have been made for why social learning isn't purely beneficial (Boyd and Richerson 1995; Rogers 1988). The answer to how to best learn in a social environment is seemingly non-trivial. Game theoretical approaches have been used to explore this subject, but there is still ongoing research in improving the models that are used (Henrich and McElreath 2003; Enquist, Ghirlanda, and Eriksson 2007).

(Laland 2004) discusses strategies for this problem in detail, and explores when it is appropriate to innovate or observe in a social learning situation. Indiscriminate observation is not always the best strategy, and there are indeed situations where innovation is appropriate. This is largely influenced by the conclusions of (Barnard and Sibly 1981), which reveals that if a large portion of the population is learning only socially, and there are few information producers, then the utility of social learning goes down.

In the social learning competition, an "observe" move introduces new actions into the repertoire of an agent, which the agent may optionally exploit. The relationship between this and social learning in animals is demonstrated by (Galef Jr. 1995), which differentiates social learning from mere imitation. In highly variable environments, socially learned information may not always be the most beneficial, yet animals that learn socially are still able to learn locally adaptive behavior. This is the result of having rewards or punishments associated with expressed behavior, similar to the utility values in the social learning game, which can guide the actual behavior of an animal.

Discussing the means of social information transmission, (Nettle 2006) outlines the circumstances in which verbal communication is evolutionarily adaptive, and why few species have developed the ability to use language despite its apparent advantages. The model Nettle describes is similar to the Cultapatation game, where the cost of innovation versus observation can vary depending on the parameters of the system. The modeled population reaches an equilibrium at a point that includes both individual and social learning. The point of equilibrium is affected by the quality of observed information, and the rate of change of the environment.

Other work on similar games include (Giraldeau, Valone, and Templeton 2002), which outlines reasons why social information can become unreliable. Both biological factors, and the limitations of observation, can significantly degrade the quality of information learned socially. (Schlag 1998) explores rules that can be applied in a similar social learning environment which will increase the overall expected payoff of a population, by restricting how and when agents act on information learned through observation.

The structure and intentions of the Cultaptation Institute's social learning game are much akin to those laid out by Axelrod in his Prisoner's Dilemma tournament on the evolution of cooperation (Axelrod and Hamilton 1981). The analysis of the Prisoner's Dilemma has had substantial use in many different areas. It has been used as a model for arms races, evolutionary systems, and social systems. The Cultaptation social learning game has similar potential applications, yet currently no analysis. This work hopes to begin to fill that void.

Conclusion

In this paper, we have provided mathematical models for simplified versions of the Cultaptation game, proven their correctness, and shown their effectiveness in experimentation. We examined two kinds of payoffs in the SRI game: per-round payoff and total-lifetime payoff.

In examining total lifetime payoff, we developed an analytic method for determining the optimal strategy so long as the underlying move values are known. We extended this to an approximate sampling algorithm named *smart-innovator*, which needs only an estimate of the probability distribution, and which performed well in our experiments.

In examining per-round payoff, we discovered a polynomial time algorithm for computing the optimal strategy. This is surprising, as there are an exponential number of potential strategies. The algorithm uses state aggregation to create a weighted graph, and then applies a variant of Dijkstra's algorithm to extract an optimal path through the graph. The path can be shown to correspond exactly to the optimal strategy.

As future work, we will examine the value of observation in the Cultaptation game. Initial analysis leads us to believe that one may posit the existence of a probability distribution over the observed move values, and then apply the framework detailed in this paper to "guess" the optimal number of observe moves. By combining this technique with the techniques given in this paper, we hope to given provable guarantees about the full Cultaptation game.

Acknowledgments

This work was supported in part by AFOSR grants FA95500510298, FA95500610405, and FA95500610295, DARPA's Transfer Learning and Integrated Learning programs, and NSF grant IIS0412812. The opinions in this paper are those of the authors and do not necessarily reflect the opinions of the funders.

References

Axelrod, R., and Hamilton, W. D. 1981. The evolution of cooperation. *Science* 211:1390.

Barnard, C., and Sibly, R. M. 1981. Producers and scroungers: A general model and its application to captive flocks of house sparrows. *Animal Behavior* 29:543–550.

Boyd, R., and Richerson, P. 1995. Why does culture increase human adaptability? *Ethology and Sociobiology* 16(2):125–143.

Boyd, R.; Enquist, M.; Eriksson, K.; Feldman, M.; and Laland, K. 2008. Cultaptation: Social learning tournament. `http://www.intercult.su.se/cultaptation`.

Carr, R.; Raboin, E.; Parker, A.; Nau, D. 2008. When innovation matters: An analysis of innovation in a social learning game. *In International Conference on Computational Cultural Dynamics, 2008*, To appear.

Enquist, M.; Ghirlanda, S.; and Eriksson, K. 2007. Critical social learning: A solution to rogers's paradox of nonadaptive culture. *American Anthropologist* 109(4):727–734.

Galef Jr., B. G. 1995. Why behaviour patterns that animals learn socially are locally adaptive. *Animal Behavior* 49:1325–1334.

Giraldeau, L. A.; Valone, T. J.; and Templeton, J. J. 2002. Potential disadvantages of using socially acquired information. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 357(1427):1559–1566.

Henrich, J., and McElreath, R. 2003. The evolution of cultural evolution. *Evolutionary Anthropology* 12:123–135.

Laland, K. 2004. Social learning strategies. *Learning and Behavior* 32:4–14.

Nettle, D. 2006. Language: Costs and benefits of a specialised system for social information transmission. In Wells, J., and et al., eds., *Social Information Transmission and Human Biology*. London: Taylor and Francis. 137–152.

Rogers, A. R. 1988. Does biology constrain culture? *American Anthropologist* 90(4):819–831.

Schlag, K. 1998. Why imitate, and if so, how?, : A boundedly rational approach to multi-armed bandits. *Journal of Economic Theory* 78:130–156.