Time Horizon Minority Game

Bryan Yu
yu.bryan.j@gmail.com

Abstract

NOTE: *This was reproduced as per Risks-X Prof. Didier Sornette's request.*

We reproduce the time horizon minority game[1] as per Satinover and Sornette [4] and study the performance of optimizing versus random agents in the period before a stationary state is reached. Our results is in agreement and also show the following order of performance: mean score of agents with one strategy $>$ mean score of random agents $>$ mean score of optimizing agents. In other words, we also find that agents optimizing based on historical information in periods where the system has not stabilized under performed random agents. We conclude with a brief discussion of how optimizing agents relates to reinforcement learning agents.

1 Introduction

Whether at the gambling table, stock market, or settings that mix skill and chance, individuals exhibit a tendency to overestimate their influence over an uncontrollable outcome; otherwise known as the 'illusion of control'[2]. Satinover and Sornette [4] demonstrate this tendency in agents playing the time horizon minority game)[1] (a subclass of market-entry games characterized by multiple decision-making agents, with bounded use of historical information, making a binary decision and only those whose actions are not in the majority are rewarded) in the period before the system stabilizes.

2 Environment

Chapter: Environment In the time horizon minority game, *N* agents are each endowed with 1) a *m* bit memory and 2) *S* strategies $(s_1, s_2..., s_S)$ each of size 2^m bits. Agents' endowed strategies are determined by sampling uniformly at random with replacement from the universe of 2^{2^m} strategies and fixed. At each timestep, *t*, each agent, a_i for $i \in \{1...N\}$, selects an action $u_i(t) \in \{-1, 1\}$. The joint action across all agents produces the aggregated sum, $U(t) = \sum_{i=1}^{N} u_i(t)$, which is then added to $\mu(t)$, a *m* bit history $(D(t-m), ..., D(t-1))$ where $D(t) = 0.5[sign(2U(t) - N) + 1] \in \{0, 1\}$. Agent a_i then receives a reward $r_i(t) = -sign[u_i(t)U(t)]$. That is, if agent a_i is in the minority, it receives a reward of +1; -1 otherwise. The environment terminates after T time steps.

2.1 Optimizing Agents

Optimizing agent, a^o, determines actions based on her most successful strategy in terms of accumulated payoff in a rolling window of length τ. She will then play the opposite of the best strategy's prediction given the history, $\mu(t)$. If the strategy predicts 1 (resp. 0), she will play -1(resp. 1).

After each timestep, agent a_i^o updates the valuation of each of her strategies, $v(s_k^{a_i}, t) = \sum_{j=t-\tau}^{t} \mathbb{1}_{s_k^{a_i}(\mu(j))=D(j)}$ for $k \in \{1...S\}$, as the sum of the strategy's ability to predict the aggregate action in the previous τ time steps.

2.2 Random Agents

Random agent, a^r, determines actions by first selecting a strategy from its set of endowed strategies, uniformly at random. Then, as per optimizing agents, she will play the opposite of the strategy's prediction given $\mu(t)$. Random agents do not keep a valuation of strategies.

2.3 System Characteristics

In the minority game proper ($\tau = \infty$), the system arrives at a stationary state at $\tau_{\text{eq}} \geq 2^m \times 200$. At this state, for a subset of agents, one strategy's virtual score remains permanently higher than others and as such agents henceforth choose this strategy.[4]

Satinover and Sornette [4] also notes the existence of a phase transition as control parameter approaches, $\alpha_c = 2^{m_c}/N \approx 0.34$. In this region, the normalized variance of $A(t)$ falls below that of a random fair coin flip for large *m* and as such, optimizing agents can outperform non-optimizing agents.

3 Setup

We study three settings as per Table 1. Each setting is repeated 100 time, each initialized with a random seed. Experiments A and B evaluate the performance of optimizing agents and random agents respectively. Experiment C evaluates the performance of agents endowed with only one strategy. Note, that we study the time horizon minority game in the period before a steady state is reached and as such $\tau \ll \tau_{\text{eq}}$.

Experiment	N	m	S	τ	T	Agent type
A	31	2	2	1	100	All Optimizing
B	31	2	2	1	100	All Random
C	31	2	1	1	100	All Optimizing

Table 1: Experiment Parameters

Additionally, we conduct a parameter search with optimizing and random agents operating in $N \in \{11, 21, 31\}, m \in \{2, 3, 4, 5\}, S \in \{2, 3\}, \tau \in \{1\}$. Each setting was ran 100 times; each initialized with a random seed.

4 Results

Figure 1 illustrates the results of experiments A, B, and C in Figure 1a, 1b, 1c respectively. Hypothetical score is computed as the mean reward (assuming as if a strategy was played) averaged over agents and strategies while actual score is computed as the mean reward averaged over agents.

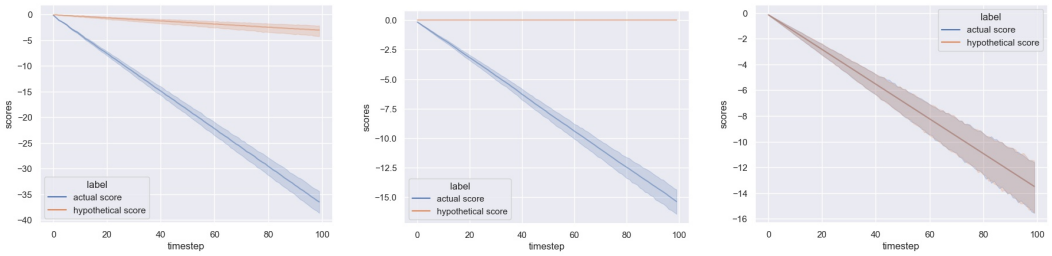
Our findings agrees with Satinover and Sornette [4] and show the following order of performance: mean score of individual strategies (Fig. 1c) $>$ mean score of random agents (Fig. 1b) $>$ mean score of optimizing agents (Fig 1a).

Figure 2 illustrate the mean per step change in wealth, $\Delta W = \frac{1}{N}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{N} r_i(t)$, resulting from the parameter search. Our empirical results show that random agents outperform optimizing agents across these settings.

5 Optimizing Agents and RL Agents

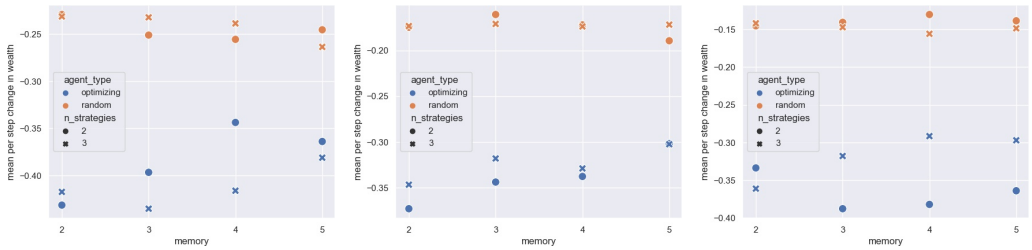
Optimizing agents that value strategies based on a rolling window share many commonalities with reinforcement learning agents (i.e. Deep Q Learning agents[3] that maintain a replay buffer to learn Q values, a mapping of state and action pair to a valuation). Furthermore, the time horizon minority game highlights challenges faced by multi agent reinforcement learning agents such as overestimating valuations[5] and non-stationarity.

As agents strategies are randomly endowed, the optimization mechanism as per [4] allows agents' decision boundaries to change significantly between time-steps by switching between strategies with



(a) Optimizing Agents (b) Random Agents (c) Individual Strategy

Figure 1: Scores are illustrated as a solid line bounded by a 95% confidence interval. Blue shows actual score (averaged across all agents) while orange shows hypothetical score (averaged over all agents and strategies). Figure 1a, 1b, 1c depicts experiments A, B, C respectively as per Table 1. Random agents do not keep a hypothetical score and as such, the orange line in figure 1b is correctly flat. In Figure 1c, when agents only have one strategy, the actual score is equivalent to the hypothetical score.



(a) N=11 (b) N=21 (c) N=31

Figure 2: Mean per step change in wealth of optimizing(blue) and random agents(orange) across a range of parameters. Our results show that random agents consistently achieve higher scores than optimizing agents.

a large Hamming distance. This mechanism relates to optimization under back propagation with a large learning rate without annealing and as such, can compound the non-stationarity challenge as past experience is not indicative of future behaviors. Deep reinforcement learning mitigate this challenge by allowing for incremental adjustments to the decision boundary with an adaptive learning rate or limiting updates to a trust region[7, 6].

Albeit the differences in detail, the lesson of the time horizon minority game - in a changing environment, that there may not be enough time for optimization to be rewarding - holds as true for reinforcement learning as it did for optimizing agents. That is, there are environments where agents that optimize based on historical data, will perform worse than chance due to continual distributional shift as a steady state has not been reached. In other words, it may not be a good idea to drive only looking at the rear view mirror.

6 Response

In my humble opinion.

First, let us take a step back from "deep" agents. Deep networks was introduced to address the issue of exploding state and action space in tabular RL approaches and to take advantage of new machine learning techniques. Lets refocus on the larger idea in reinforcement learning, that agents can be designed to learn policies from feedback signals.

Thus, let us consider tabular agents. Tabular agents maintain a table mapping either (1) a state (2) a state, action pair to a valuation. A naive approach in the fashion of (2) would consider mapping a state action pair to the sum of previous rewards with a rolling window, $q_i^t(\mu(t), a_i) = \sum_{k=t-\tau}^{t} r_i(k)$.

If the goal was to re-create optimizing agents, this could be done by endowing tabular agents with *S* fixed strategies. These tabular agents action space would be to select amongst it's endowed *S* strategies based on a counterfactual valuation $q_i(\mu(t), s) \forall s \in S$. As such, optimizing agents behavior can be replicated using tabular agents.

We take a look at how agents make decisions. Recall that agents are endowed with *S* *m*-bit fixed strategies. Thus agents switch between *S* decision boundaries in *m* dimensional space. We note that the hamming distance between strategies may be large, which implies that by design agents may not be able to make incremental refinements to their decision making.

We contrast this to DQN-Agents, agents equipped with a deep network. Firstly, DQN-Agents also learn a mapping between a *m* dimensional input to an output in *u*. However, between input to output, the agent's the deep network transforms the input in a variety of ways (i.e. expanding the dimensions, applying non-linearities). Thus the agent's decision boundary may lie in a higher dimensional space. Secondly, DQN-agents have more control over their decision boundary by choice of learning parameters.

Now we consider the environment and its ability to change over time. An input that resulted in *D*=1 previously, can now result in *D*=0. Thus, predicting future outcomes with past data is now challenged with distributional shifts. Knowledge of prior *D* may not be indicative of future *D*'s, and as such DQN agents may not necessarily perform better. This makes techniques such as opponent modeling ineffective(even more as *N* increases), and benefits from regular re-training of new policies would require the distributional shifts to be sufficiently slow. However, note that the underlying mechanism and processes are invariant. Thus, it would at first suggest that full information of the current state of the system allows us to predict the actions of other agents, and the transitions of the system going forward. This would suggest that if the initial randomness can be predicted, the remainder of the system can be determined. However, this probability decreases as *N* and *M* increases.

So we consider the system and its aggregated output. We ask the following questions, and propose approaches to 4, and 5:

1. I hypothesize that the Minority Game is a negative sum game (only a minority can receive +1, the remainder must receive -1). What if we allowed agents to choose whether they wanted to participate in any round of the game? (e.g. like joining a hand at the blackjack table). Are there pockets of time steps in the minority game where predictability increases?

   Why? It has been shown in blackjack, that it is possible to do better by selectively entering hands. That is, systems may have periods where the outcome is more predictable. As such, inserting an additional action, do nothing, may change the expected value of the game and the lesson learned from the experiment.

   Some Ideas: We introduce the do-nothing action, 0. Thus the updated action space is {-1,0,1} and the reward for taking action 0 is 0. All else the same. If we train RL agents in this setting, I hypothesize one of the strategies it learns is to do nothing. If we insert the RL agent into the Minority Game Proper, will it learn to do nothing until system stability and then participate with a superior policy?

2. Is the Time Horizon Minority Game more complex than the games RL has seen success previously (Go, Chess, Leduc Poker, Stratego)?

Why? The complexity of the game gives insight into whether successful techniques in RL will apply to the minority game. If it is more complex, this raises the question of what new ideas are required?

3. Could the outcome of the time horizon minority game be manipulated by inserting a colluding team of agents into the population? How big would this team have to be? Is it be possible to learn a joint policy that consistently achieves above average team rewards?

   Why? The illusion of control suggests that individuals assume they have more control than they actually do. However, does this still hold true if a group is involved? What is the smallest sized team necessary?

   Analysis It would at first glance suggest that if a majority was in collusion, they could manipulate the outcome of every round. However, if we control all N of the agents, the cost of control exceeds the benefits.

   Some Ideas: We can introduce a single agent, a_{team}, which represents the aggregate decision of ψ agents. We update the action and reward space and train this agent using single agent reinforcement learning.

4. Is it possible to stabilize the game sooner? How much of the system do we need to control before we can determine stability? How much of the system do we need to control if we want to stabilize it over *t* time steps?

   Why? The success of any predictive approach is contingent on the stability of the system. One take away was that the system is unstable for longer than a participant can engage in the market. Is there more to this picture?

   Some Ideas: One hypothesis is that increases in the hamming distance between strategies endowed to an optimizing agent, decreases the predictability of the system. Thus, this can be tested by forcing endowed strategies to satisfy $hamming(s_i, s_j) < k | i, j \in \{1...S\} and i$ for all agents.