Online Markets Project 2

Online Learning

April 2022

1 Introduction

This purpose of this project is to examine online learning algorithms in a game. Generally, in auctions, we have examined bidders (players) whose value distributions are independent. This project aims specifically to evaluate online learning algorithms in a game where player's value distributions and available actions are dependent on other player's actions. The way this works is that first player draws their value independently, takes their action, and this action affects the second player's value distribution which affects their available actions. This trend continues until all players have played an action. The specifics of the game we examined are in the following section.
We simulated the game in the context where each player's decisions are made by the Exponential Weights Algorithm. In addition, we attempted to see whether or not the third player was able to manipulate the results of the game in their favor. Our results showed that no-regret learning algorithms will tend to vote as close to possible to their value, and that the third player has no ability to manipulate the game in their favor.

2 Preliminaries

Consider the following game:
3 players: $\{P_1, P_2, P_3\}$.
P_1 has value $v_1 \sim U[0,1]$.
P_1 takes an action $a_1 \in \{\frac{1}{2} + \sigma_1 j \mid j = 0, \pm\frac{1}{2}, \pm 1\}$, where $\sigma_1 = \frac{1}{2}$.
Then, calculate P_2's value distribution as follows:

$$\begin{cases} U[0,1] & a_1 = \frac{1}{2} \\ U[0, \frac{1}{2} + a_1] & a_1 < \frac{1}{2} \\ U[a_1 - \frac{1}{2}, 1] & a_1 > \frac{1}{2} \end{cases}$$

Letting this calculated distribution be $U_2 = U[x_2, y_2]$ and $m_2 = \frac{y_2 - x_2}{2}$, P_2 gets value $v_2 \sim U_2$.
Then P_2 takes an action $a_2 \in \{m_2 + \sigma_2 j \mid j = 0, \pm\frac{1}{2}, \pm 1\}$, where $\sigma_2 = \frac{1}{2}(y_2 - x_2)$.
In the same fashion, calculate P_3's value distribution as follows:

$$\begin{cases} U[x_2, y_2] & a_2 = m_2 \\ U[x_2, m_2 + a_2] & a_2 < m_2 \\ U[a_2 - m_2, y_2] & a_2 > m_2 \end{cases}$$

Letting this calculated distribution be $U_3 = U[x_3, y_3]$ and $m_3 = \frac{y_3 - x_3}{2}$, P_3 gets value $v_3 \sim U_3$.

P_3 then takes an action $a_3 \in \{m_3 + \sigma_3 j \mid j = 0, \pm\frac{1}{2}, \pm 1\}$, where $\sigma_3 = \frac{1}{2}(y_3 - x_3)$.

Given all three actions, let M be their median, and given this, give each player P_i payoff $1 - |v_i - M|$.

Let us now explain the reason this game was implemented this way. The idea is that this game is a crude approximation of the distribution of news on the internet. Think of the first player as news organizations. Generally, they learn about news first, derive an opinion of the event, and share the news with inherent bias (we model bias in this game as varying along one dimension). Think of the second player as informed activists. They learn about the news through news organization, form an opinion that we assume to be dependent on the news organization's bias, and share the news to their followers with their own opinion. The third player can then be thought of as your average social media consumer, they get the news from the social media influencers, with their compounded bias, and form their own opinion. In our representation, we are making the assumption that the player's opinions are influenced by bias in a positive fashion, that is, their value distribution is skewed so that their value is more likely to be closer to the bias they observe. In reality, lots of people are capable of reasonably independent critical thinking, and some even have skeptical mindsets that would cause them to be influenced by bias in a negative fashion. We don't examine these possibilities in this report, but we acknowledge that these ideas should be addressed in following research.

Given this set up, we imagine a theoretical situation where the actions of these players correspond to "votes" and the median of these votes establishes the "prevailing opinion" of the vote. Accordingly, the payoff for each player is inversely proportional to the difference between their value and the "prevailing opinion".

Now that we have introduced the game, let's explain how we're going to analyze the game in the context of online learning algorithms.

First, we are going to consider the situation where each player's actions are determined by the Exponential Weights (EW) learning algorithm.

We decided to consider the optimal learning rate, a learning rate of 0 (equivalent to random guessing), and a learning rate of ∞ (equivalent to Follow-The-Leader (FTL)).

Because we have three learning rates and three players, there are 9 different competitions we can consider when running these algorithms against each other.

We decided to run $n = 100$ trials, because we found that are results did not change significantly after this point.

We conduct trials as follows:

First, for each player, create an instance of EW for each learning rate.

Then, for each trial $1, ..., n$:

1. draw $v_1 \sim U[0, 1]$
2. have each EW instance (for P_1) pick action a_1
3. draw $v_2 \sim U_2$
4. have each EW instance (for P_2) pick action a_2
5. draw $v_3 \sim U_3$
6. have each EW instance (for P_3) pick action a_3
7. calculate payoff vectors and update each EW instance accordingly

This will allow us to understand if any of the players have an inherent advantage by comparing their average payoff per round.
We now examine the Nash equilibria of this game so that in the following section we can see if our learning algorithms converge to any of them. First, note that while players with later orders will have more values that their actions can correspond to, we define actions relative to the distribution the player's value was chosen from so that each player will effectively have the same possible actions each round. Because of this, given the way we defined actions above, our game has a finite action space, so we must have at least one (possible mixed) Nash equilibrium.
Because of the inherent complexity of this game, we wrote code to determine the Nash equilibria given certain values for the players. To determine these values, we first calculated all the possible medians we could observe given our game's constraints, and then we made the following assumption: a player's strategy will not change if their value changes as long as the new value has the same closest possible median as the old value. Given this assumption, the only other time a player's strategy will change is if their value is halfway between two possible medians. To make up for this, we simple added all of these values to our list of values to test. We were then able to calculate, for each notable tuple of three values, all of the sets of actions that satisfy the requirements of a Nash equilibrium. This code can be found in the appendix.

Given this setup, we will determine, in the following section, which learning rates are optimal compared to others (for EW) and whether any of our EW instances converge to the Nash equilibria we found.

Following this, we will establish two no-regret learning algorithms for the first two players and attempt to devise a strategy that gives us a relatively higher payoff.

3 Results

Our results were fairly straight-forward. We found that both uniform guessing and FTL consistently under performed compared to our theoretically optimal learning rate. In addition, we found that the first player had a consistent advantage, and that all no-regret learning algorithms tended towards voting as close to their value as possible, even for the other two players. In light of this, we ultimately failed to derive a strategy that could manipulate either of the first two payoffs, even though in our schema the third player gets to decide which vote becomes the median (assuming the first two votes weren't identical and their voting range covers the other votes).

4 Conclusions

There are some potential routes for future studies to evaluate. First, it would be interesting to consider the mixed Nash equilibria of this game. In addition, we think it be valuable for future studies to test different kinds of learning algorithms (i.e. not EW) to see if they fair differently. Finally, we think it would be interesting to see if this model could be applied to help model the spread of political opinions online.

5 Appendix

Code