Computer Science 136: Economics and Computation

Prof. David Parkes

Matt Goldberg

Juan C. Perdomo

Final Project

Learning in Games: Fictitious Play and No Regret Dynamics

May 3rd, 2016

1. Introduction

Game theory as a field models scenarios in which agents are involved in strategic interactions with other agents and in which their payoffs depend, not only on their own actions, but also on the actions of other players. Under the traditional view of the subject, games are studied under the lens of Nash Equilibrium (NE) and its many refinements such as subgame perfection. These ideas serve as powerful tools to predict the behavior of rational self-interested agents since they describe scenarios in which every individual is playing a best response to the actions of others.

Nevertheless, there are several problems that arise from using Nash equilibria as the main method of analyzing games. The primary concern when studying these sets of strategy profiles is that they require agents to be infinitely rational and optimizing. Each individual player must consider all possible sets of actions by other agents and figure out on the assumption that every one else is fully rational the expected sets of actions of his opponents. Such an analysis could entail an arbitrarily large number of backward inductions as well as careful consideration of various different subgames. As the number of actions and players in a game increases, such hyper rationality by part of agents might not be a reasonable assumption to make. Moreover, although they are always known to exist, computing them is often intractable. The problem of finding Nash equilibria belongs to the set of PPAD-complete problems. If they are hard to compute using sophisticated algorithms, how can real-world agents readily compute them through simple introspection? Lastly, one final concern is that there might be multiple equilibrium profiles in a games which complicates the issue of predicting which one agents will choose, assuming they choose to play one at all.

Given these concerns, other approaches have been developed to study games. Broadly, these are considered as learning methods in which a set of agents, that while not infinitely rational, repeatedly play a game and employ an algorithm that seeks to improve their payoff over time. In this framework, agents examine the past history of play in order to learn about the game as well as the behaviors of other agents and improve their strategies based on this acquired knowledge. A great deal of research has been devoted to the study of such learning approaches and this paper seeks to investigate two such algorithms, fictitious play and no-regrets learning.

More specifically, our project consisted of implementing both algorithms, fictitious play and no regret dynamics via a variation of the multiplicative weights algorithm and then to test these approaches on a variety of games downloaded from the Stanford GAMUT database. In order to facilitate testing, we created an interface that allows users of the system to quickly download games and run the algorithms on them. The paper is divided into two main sections. The first provides a broad overview of the learning algorithms as they are presented in the literature. More than just discussing their basic mechanics, we aim to discuss any major theorems regarding their dynamics such as the classes of games for which they are guaranteed to converge to Nash equilibrium and any relevant bounds on their performance. The second section discusses empirical results of running the algorithms on several different kinds of games and comparing and contrasting their relative performances.

2. Overview of Methods

1. **Fictitious Play.** In fictitious play, each agent assumes their opponents are selecting their actions from a fixed but unknown distribution. In each iteration of fictitious play, a player's selected action is modeled as an independent draw from the player's unknown distribution. Thus, the player's sequence of play over time can be viewed as a series of i.i.d. multinomial random variables drawn from the same distribution, and so we can use these observations to infer each agent's underlying distribution over actions. As play goes on, each player updates their own estimation of the other agents' underlying mixed strategies; in each step, each player selects an action in order to maximize expected utility assuming that other players will act according to the mixed strategy implied by their history.

More formally, at time t each player i has a history $h_t^i(a)$. This history is initialized to zero for every action a, and is updated in each time step using:

$$h_t^i(a) = h_{t-1}^i(a) + \begin{cases} 1 & \text{if } a_t^i = a \\ 0 & \text{if } a_t^i \neq a \end{cases}$$

where a_t^i is the action played by player i at time t. Then, in each time step t, each agent i plays a best response (i.e., an action which maximizes expected utility) assuming each opponent j plays action $a \in A_j$ with probability

$$p_j(a) = \frac{h_t^j(a)}{\sum\limits_{a' \in A_j} h_t^j(a')}.$$

In other words, each agent assumes their opponents will play according to the mixed strategy given by their normalized history.

When it comes to finding equilibria, fictitious play has many interesting properties of convergence. One such property is that strict Nash equilibria are absorbing states of fictitious play. This means that if Nash equilibrium s is played at time t, then s will also be the action profile selected by agents in all subsequent time periods $t' > t$ (Fudenberg and Levine, 33). Therefore, any pure strategy equilibrium reached by fictitious play must be a Nash equilibrium. Intuitively, this property can be understood by observing that if each player is selecting a best response and an entire action profile is repeated in consecutive time steps, it must be the case that every agent is best responding to each other in this action profile, which by definition means that this action profile is a Nash equilibrium. Moreover, if the empirical distribution of each player i converges to a distribution σ_i, then the set of best responses corresponding to the product of all the distributions is a mixed-strategy Nash equilibrium, since if there were any useful deviations, then the distributions would not be stable and thus would not converge.

Unfortunately, these probability distributions do not always converge. In 1964, Shapley provided an example of a game in which strategies do not converge, but rather cycle around a set of values (Fudenberg and Levine, 34). Thus, fictitious play is not guaranteed to find an equilibrium; however, fictitious play has been shown to always find an equilibrium in certain classes of games. For example, Julia Robinson proved in 1951 that fictitious play always converges to a Nash equilibrium for two-player zero-sum games. Interestingly, Krishna and Sjöström showed in 1998 that for almost all games, fictitious play does not converge, and that Shapley's 1964 example was the norm rather than the exception.

2. **No Regret Dynamics.** As discussed in Parkes and Seuken, regret is defined as the time-averaged difference between the cost of actions played by the agent and the cost incurred by same agent if it has played the optimal action a^{opt} in every period t. An algorithm is considered to implement no regret dynamics if the value of the regret for every agent tends to 0 as T goes to infinity (Parkes Seuken, 668).

$$\text{Definition of Regret} \qquad \frac{1}{T}\Big(\sum_{t=1}^{T} c^t(a^t) - \sum_{t=1}^{T} c^t(a^{opt})\Big)$$

No regret dynamics are useful because it is known that they converge to a set of Coarse Correlated Equilibria (CCE). While not all CCE are Nash equilibrium, all NE are in fact CCE meaning that $NE \subseteq CCE$. Therefore while these algorithms do not always converge to the optimal solution of NE, they are always guaranteed to lead to a solution concept that is within the bounds of being reasonable.

For our learning method we implemented the multiplicative weights algorithm. The algorithm is designed for cost minimization games and works by giving each player a set of weights, one for every action. These weights are updated in accordance with the relative costs of every action and the probability of playing a certain action is determined by its relative weight $p(a) = w(a)/\sum_{a'} w(a')$. Furthermore, in each iteration, each player is allowed to calculate the expectation of playing action a given that all other players employ their current mixed strategies as defined by their current action weights. Let σ denote the set of weights then we have that:

$$c^t(a) = \mathrm{E}_{s_{-i}^t \sim \sigma^t}[C(a, s_{-i}^t)]$$

From calculating this expectation value for every action, each player is given a cost vector c^t which they then use to update the relative weights of each action according to the following rule.

$$w^{t+1}(a) = w^t(a) \cdot (1-\epsilon)^{c^t(a)}$$

From this equation we see that the weight of every action can only decrease as time goes on. One important thing to note is that actions with high costs are discounted exponentially more than better actions which leads to improved convergence properties. Moreover, $\epsilon \in (0, .5)$ is a parameter that can be used to vary the degree of exploitation and exploration the algorithm performs. Large values of ϵ lead to higher exploitation while lower values lead to exploration. This is due to the fact that higher values of ϵ lead to higher decreases in action weights which affects worse actions more than it does better actions.

Analytically, it can be shown that the multiplicative weights algorithm leads to an expected regret with n actions and T iterations of $O(\sqrt{(\ln n)/T})$. From this it follows that an algorithm has regret ϵ after $O((\ln n)/\epsilon^2)$ iterations. This algorithm and analysis is grounded on work done by Tim Roughgarden in his lecture notes on Algorithmic Game Theory.

3. Empirical Results and Discussion

Having now seen the overall properties of the learning algorithms under consideration, in this section we will now discuss their relative successes at arriving at equilibrium profiles for different classes of games. In particular we are interested in evaluating games for which they converge to different sets of profiles and evaluating the reasons for their divergence.

a. *Prisoner's Dilemma and Matching Pennies.* Beginning with perhaps the most common of games, the Prisoner's Dilemma and Matching Pennies, we see that both solutions quickly reach the unique Nash equilibria of these games

Prisoner's Dilemma

Fictitious Play: $((0.0005, 0.9995), (0.0, 1.0))$

Multpilicative Weights: $((0.0, 1.0), (0.0, 1.0))$

Matching Pennies

$$\text{Fictitious Play:} \quad ((0.505, 0.495), (0.508, 0.492))$$

$$\text{Multiplicative Weights:} \quad ((0.5, 0.5), (0.5, 0.5))$$

After just 100 iterations and using an $\epsilon = .1$, the multiplicative weights algorithm found the exact Nash equilibrium for both games. On the other hand, fictitious play also arrived at solutions that are close to the optimal, yet did so after 2000 iterations. Therefore, for this class of simple games, multiplicative weights is significantly more efficient.

In terms of their convergence overall, we see that for games like Prisoner's Dilemma where there is a dominant action both algorithms do very well. Moreover, due to Robinson's result from 1951 we know what fictitious play is guaranteed to converge for Matching Pennies since it is a two player zero sum game. Interestingly enough, multiplicative weights also finds the right solution.

b. *Rock, Paper, Scissors.* Moving on to the slightly more elaborate example of Rock, Paper, Scissors, we find a game for which the two algorithms lead to vastly different solutions. While fictitious play arrives at an equilibrium that is approximately optimal to the Nash equilibrium in which players randomize between all three actions, multiplicative weights leads to a solution where players always play rock and tie.

Rock, Paper Scissors

$$\text{Fictitious Play:} \quad ((0.321, 0.3405, 0.3385), (0.35, 0.319, 0.331))$$

$$\text{Multiplicative Weights:} \quad ((1.0, 1.6992293147573546e{-}270, 0.0), (1.0, 1.6992293147573546e{-}270, 0.0))$$

The reason for convergence for fictitious play is again clear. The game is yet another example of a two player zero sum game. However, the second approach leads to a widely different solution that at first glance seems entirely irrational since both players always choose to play rock and tie. Upon further analysis, the reason for this kind of behavior becomes clear. In the NE of the game, both players place equal weight on every action meaning that they are equally likely to win lose or tie which leads to an expected value of zero for the game. In the case where they always play rock, the agents are guaranteed a payoff of zero. Therefore, we see how this behavior makes sense under the no regrets framework since the agents, even though they are not best responding and mindlessly tying every round, still experience no regret since the time averaged difference between their choice of action and the optimal strategy is zero! It almost seems like the players in the no regrets framework arrive at an equilibrium designed for "risk-averse" agents who wish to remove the randomness from their game. In the real world, we wouldn't expect agents to tie all the time. If one person realized that the other was playing a fixed action all the time, it is reasonable to assume that they would seek to optimize their utility and look to play a winning action. For this reason, fictitious play seems like the more appropriate algorithm for this kind of game.

One last note about this example is that we again see how fictitious play takes significantly more iterations to converge (2000) while multiplicative weights only takes 100. Additionally, multiplicative weights converges to the same profile regardless of the choice of ϵ. For this specific example, we used $\epsilon = .1$ but the solution holds for all $\epsilon \in (0, .5)$.

c. *Battle of the Sexes.* In the case of Battle of the Sexes, we get the opposite behavior to the RPS game. Fictitious play fails to converge to the Nash equilibrium while multiplicative weights does converge.

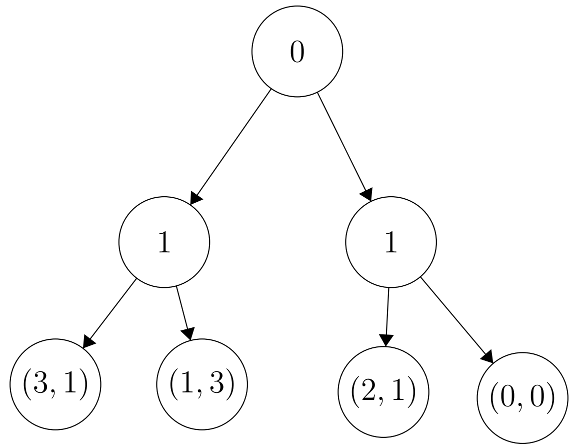
Battle of the Sexes

$$\text{Fictitious Play:} \quad ((0.999, 0.001), (0.998, 0.002))$$

$$\text{Multiplicative Weights:} \quad ((0.6665710836073563, 0.3334289163926437),$$
$$(0.3334289163925545, 0.6665710836074454))$$

While the Battle of the sexes is a two player game, it is no longer zero sum meaning that fictitious play is not guaranteed to converge. On the other hand multiplicative weights (with $\epsilon = .1$) after just 100 iterations finds a solution that is incredibly close to the unique Nash equilibrium of the game.

d. *A Simple Sequential Game.*



**Normal form representation can be found in project folder*

The following sequential, non-zero sum game has two pure strategy Nash equilibrium. One which is not subgame perfect in which player 1 goes right and player 0 plays left and another which is in which player 0 goes right and player 1 goes left. When the learning algorithms are run on this kind of sequential game, we find that both converge not only to Nash equilibria, but also to the one that is subgame perfect.

Simple Sequential Game

$$\text{Fictitious Play:} \quad ((0.002, 0.998), (0.001, 0.0, 0.0, 0.999))$$

$$\text{Multiplicative Weights:} \quad ((0.0021642349201258524, 0.9978357650798741),$$

$$(0.001592166219559499, 1.0586552848380285e{-}06, 0.0006634145785651364, 0.9977433605465905))$$

Although there is still some error, after 2000 and 100 iterations respectively, both fictitious play and multiplicative weights arrive at similar solutions. This is surprising given that there are no guarantees of convergence to NE for either algorithm, especially for fictitious play since the game is neither zero sum nor can be solved through iterative elimination of strictly dominated actions.

Chicken

$$\text{Fictitious Play:} \quad ((0.815, 0.185), (0.815, 0.185)) \text{ or } ((0, 1), (1, 0)) \text{ or } ((1, 0), (0, 1))$$

$$\text{Multiplicative Weights:} \quad ((0.8151486179131, 0.18485138208684), (0.8151486179131, 0.18485138208684))$$

In Chicken, we know that there are two pure-strategy Nash equilibria: $\{(1,0),(0,1)\}$ and $\{(0,1),(1,0)\}$. Additionally, we know there must be one mixed-strategy Nash equilibrium, since there must be an odd number of Nash equilibria; in the random instance of Chicken we got from GAMUT, the mixed-strategy Nash equilibrium is $\{(0.815, 0.185), (0.815, 0.185)\}$. Therefore, both fictitious play and multiplicative weights always converge to a Nash equilibrium; in particular, multiplicative weights always converges to the mixed-strategy Nash equilibrium, which is also a coarse correlated equilibrium, whereas fictitious play finds any of the three Nash equilibria depending on the random initialization.

Multiplicative Gadget

$$\text{Fictitious Play:} \quad ((0.497, 0.503), (0.514, 0.486), (0.514, 0.486), (0.756, 0.244))$$

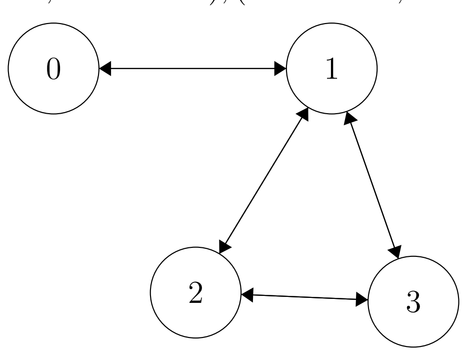
$$\text{Multiplicative Weights:} \quad ((0.5, 0.5), (0.5, 0.5), (0.480718146, 0.51928185), (0.92429339, 0.075706601))$$

We constructed a multiplicative gadget such as the ones we discussed in class. In particular, this game has the property that in any Nash equilibrium, $x_4 = x_1 x_2$, where $x_i = P(a_i = 1)$. In this instance, we constructed the payoffs so $x_1 = x_2 = 0.5$, so a learning algorithm succeeds when $x_4 = 0.25$. As we can see, fictitious play always converges to the Nash equilibrium in which player 4 plays 1 with probability 0.25, whereas multiplicative weights fails to converge to the Nash equilibrium of the game, instead converging to $x_4 \approx 0.075$.

Public Goods

$$\text{Fictitious Play:} \quad ((0.001, 0.999), (0.9995, 0.0005), (0.001, 0.999), (1.0, 0.0))$$

$$\text{Multiplicative Weights:} \quad ((0.0321209697, 0.96787903), (0.98673290, 0.01326709),$$
$$(0.54881168, 0.45118831), (0.54881168, 0.45118831))$$



All Nash equilibria in the public goods game are maximal independent sets; as we can see, fictitious play finds a maximal independent set in which players 0 and 2 play 1 and players 1 and 3 play 0. On the other hand, multiplicative weights does not find a maximal independent set and instead terminates with mixed strategies for players 3 and 4, in which each plays 1 roughly half of the time. This result resembles a Nash equilibrium in some ways, in that if player 0 plays 1 and player 1 plays 0, then one of players 2 and 3 must play 1 and the other must play 0 in order to be a Nash equilibrium. However, the terminal strategy profile for multiplicative weights has each of these agents playing 1 about half of the time, which does not result in a Nash equilibrium.

Final Remarks. Learning in games provides a novel approach to studying game theory. By gradually optimizing their strategies over time, we hope that agents will eventually reach an equilibrium profile that describes rational play. For multiplicative weights and no regret dynamics, we know that the algorithm will always converge to a Coarse Correlated Equilibrium. The set of all CCE is a superset of NE which explains why the algorithm only sometimes converges to Nash equilibria. On the other hand, fictitious play has no worst case guarantees in the same way multiplicative weights does for CCE, but as seen in our results it is guaranteed to converge to NE for two player zero sum games.

In terms of their asymptotics, the results clearly indicate that multiplicative weights converges to an equilibrium profile much faster than fictitious play. The reason for this behavior is that multiplicative weights has exponential discounting of bad actions which promotes agents to quickly optimize their play. If fast convergence is the primary concern for the choice of algorithm then no regret dynamics and multiplicative weights is the more well suited choice of algorithm. Such a desire for fast asymptotics is justified for scenarios where agents are limited to playing a smaller number of games or when playing each game is a costly endeavor.

More broadly, however, it is unclear which is the superior algorithm. On the one hand, multiplicative weights is always guaranteed to converge to a CCE, but is some cases these equilibria are unintuitive and appear irrational such as in the case of Rock Paper Scissors. Conversely, fictitious play is guaranteed to converge for 2-player zero-sum games but is not always able to find a reasonable solution as evidenced by the case of the Battle of the Sexes. Therefore, the choice of algorithm depends on the class of games that are being tested. In particular, if one is mainly studying 2-player zero-sum games then fictitious play might be the correct choice. However, if the class of games is not as well defined, the multiplicative weights may be the better solution since it runs faster and is always guaranteed to return a CCE.

4. Appendix

1. **Program Description.** The algorithms are implemented in fict_play.py and mult_weights.py. In order to examine our results however, it suffices to simply execute the file results.py which imports all of the necessary games and functions and outputs the results to the terminal. Games can be generated from GAMUT by running `java -jar gamut.jar -g GAME_NAME`, where `gamut.jar` is available for download for free at gamut.stanford.edu.

2. **References.**

(1) Fudenberg, Drew, and David K. Levine. The Theory of Learning in Games. Cambridge, MA: MIT, 1998. Print.

(2) Krishna, Vijay, and Tomas Sjstrm. On the Convergence of Fictitious Play. Mathematics of Operations Research 23.2 (1998): 479511. Web.

(3) Roughgarden, Tim. "No Regret Dynamics." CS364a: Algorithmic Game Theory. California, Palo Alto. 18 Nov. 2013. Theory Stanford. Web. 30 Apr. 2016. ¡http://theory.stanford.edu/ tim/f13/l/l17.pdf¿.

(4) Parkes, David, and Sven Seuken. Economics and Computation. Cambridge: Cambridge UP, 2016. Print.