Multiagent Strategies for RPSLS

Bryce Anthony
{Branthony}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

Rock, Paper, Scissors, Lizard, Spock is a popular extension of the classic Rock, Paper, Scissors game that has gained significant attention since its appearance on the TV show The Big Bang Theory. In this game, players have two additional options to choose from, increasing the complexity of the game. Despite there being no optimal winning strategy for Rock, Paper, Scissors, Lizard, Spock, players' tendency to deviate from a uniform distribution of moves allows for opponents to exploit their strategy and win more games on average. In this project, we implemented reinforcement learning techniques to develop Rock, Paper, Scissors, Lizard, Spock bots with the goal of winning a class tournament. We created three different bots, namely LoseToWinBot, comboBot, and RandomBot, by leveraging ideas from existing literature. We evaluated the effectiveness of our models by running matches between our bots and bots provided with starter code. Though the comboBot wasnt able to defeat all other bots. Our study highlights the potential of using reinforcement learning techniques to develop strategies for complex games like Rock, Paper, Scissors, Lizard, Spock.

1 Introduction

Rock, Paper, Scissors, Lizzard, Spock is an extension of the classic Rock, Paper, Scissors game created by Sam Kass and Karen Bryla and ultimately made popular by the tv show The Big Bang Theory. [1]This extension of the game increases the complexity of Rock, Paper, Scissors by adding two more options for players to choose from without adding in a pure strategy that a player could exploit to gain a strategic advantage over their opponent. Despite there being no optimal winning strategy for Rock, Paper, Scissors, Lizard, Spock the tendency of players to deviate from a uniform distribution of moves allows for an opposing player to exploit the others strategy and win more games on average as a result.

For this project I implemented several strategies to create Rock, Paper, Scissors, Lizard, Spock bots with the goal of winning an class tournament by applying reinforcement learning techniques to defeat bots made by classmates. The rationale behind the bots stemmed heavily from existing literature surrounding multiagent learning in the game of rock, paper, scissors. A "LoseToWinBot was implemented based on the findings of (Wang, Xu, and Zhou 2014), A "comboBot" was implemented as an extension of the research from (Wang et al. 2020), and a "randomBot" was built out of speculation that it could perform well. remainder of this paper we provide further background information on the rationale behind the strategies used by our bots along with details about their implementation. We then go on to explain the experimental setup used to evaluate our models and finally we discuss the results of our experiments along with the broader impacts of our work.

[1] https://bigbangtheory.fandom.com/wiki/Rock,_Paper,_Scissors,_Lizard,_Spock

2 Background

- ApeBot- Provided with our starter code, this bot returns the same action as the opponent did the last round.
- MixedBot- Provided with our starter code, this bot randomly plays rock or paper
- NashBot- Provided with our starter code, This bot plays according to the Nash equilibrium for Rock, Paper, Scissors, Lizard, Spock.
- SolidAsARockBot- Provided with our starter code, this bot always plays Rock
- RandomBot- This bot picks a random move, then it picks the opponents last move and repeats.
- LoseToWinBot- This bot randomly returns one of the 2 moves that would have lost to the opponent's previous move but repeats its previous move if it won.
- comboBot- Plays NashBot, RandomBot, or LoseToWinBot based on the opponent's last 100 moves.

Of the 7 bots above 4 were provided with our starter code and the other 3 were created by us using Ideas in existing literature. Two of our bots; the LoseToWinBot and RandomBot were created and used independently while our comboBot implemented a strategy that made use of the LoseToWinBot, RandomBot, and NashBot that was provided with our starter code.

The random bot was inspired by the concept of Nash Equilibrium and the research of (Mohtavipour and Zideh 2019) on the detection of collusive strategies in multiagent games. The idea behind this bot was that it would maintain randomness while also implementing a, fairly simple, strategy.

The LoseToWinBot was implemented based on the paper from (Wang, Xu, and Zhou 2014) as they found a simple conditional response strategy (win-stay lose-shift) to be effective against human rock, paper, scissors players. This was due to the tendency of humans to cycle through the choices in rock, paper, scissors, which we hypothesized would be perpetuated into the models created by others in the class. The LoseToWinBot based its moves solely on the previous move made by itself and its opponent. If the LoseToWinBot won the previous round the same move would be played again. If the opponent won the previous game, the LoseToWinBot would choose a move that would lose to the opponent's last move and due to the tendency to cycle through moves rather than repeat the same move; the LoseToWinBot would be more likely to either win or tie the next round. Although very different functionally from the WoLF PHC algorithm described in (Bowling and Veloso 2001), the concept of learning fast while losing and slowly when winning is captured pretty well by the strategy.

Finally, the comboBot was inspired by the research of (Wang et al. 2020) as they used a mixed strategy that contained several Markov Chain models and selected the move from the best performing Markov chain model along with its prediction based on a specified number of the recent rounds. This "Focus Length" as described in (Wang et al. 2020) was also used in a version of our comboBot allowing it to switch between the 3 strategies it implements to use the one that would be most effective against the most recent strategy being used by the opponent, rather than the just the most recent move. This idea of using a mixed strategy that is informed by the opponents past moves was also echoed in (Bowling and Veloso 2001) and (Wang, Xu, and Zhou 2014); in which both of their rock, paper, scissors agents implement mixed strategies to adapt to the other player's current and potentially changing strategy.

3 Experiments

To evaluate the effectiveness of our models I ran a series of matches between bots that I had created and bots that had been supplied with the starter code. To determine which bot would be entered into the preliminary tournament the NashBot was used as a point of comparison for the two models and both the RandomBot and LoseToWinBot were run against the NashBot 1000 times. The results of the preliminary tournament led me to adjust the strategy being used and to determine my best model the same style of testing from the preliminary tournamnent was used. The comboBot was run against the RandomBot and LoseToWinBot 1000 times to see how the comboBot's pereformance would compare.

4 Results

Overall our best model was the comboModel as it outperformed the Random and LoseToWin bot's when competing directly against them. There was one opportunity to evaluate the effectiveness of the models I created against others in the class and to decide which bot would be entered into this preliminary tournament, I ran a series of matches where the two bots that had been created at the time, RandomBot and LoseToWinBot, played against the NashBot to see which would perform better. Despite both losing to the NashBot, after 1000 rounds the RandomBot performed slightly better than the LoseToWinBot as it won 403 games and tied 191 compared to the 390 wins and 206 ties from the LoseToWinBot. The lack of clarity from this early testing led me to enter the LoseToWinBot into the preliminary tournament, based on intuition, where it ranked fairly poorly (21st out of 23 opponents) scoring only 11 points. These results were obtained by running the LoseToWinBot against each of the opponents bots 10,000 times. An analysis of these matches revealed that the LoseToWinBot was a conquerable strategy as it had less than 50 wins against 6 opponents and no matches where it was defeating an opponent by a significant margin. The results from the preliminary tournament led me to adjust the strategy of the model and I implemented various focus lengths from (Wang et al. 2020) to see how that would impact the performance of the model and suprisingly this made the model perform worse than only remembering the last move so it was removed from the model. Our final model, created after the preliminary round was the comboBot that implemented the Random, LoseToWin, and Nash Bot. The comboBot was run against the RandomBot and LoseToWinBot 1000 times to see how the comboBot's performance would compare. This time, the comboBot was the clear winner after matchups with the two bots. The comboBot defeated the Random bot 218 times and tied 579 times while it defeated the LoseToWinBot 419 times and tied 199 times.

5 Conclusions

In conclusion, Rock, Paper, Scissors, Lizard, Spock is an extension of the classic game that increases its complexity by adding two more options for players to choose from. Although there is no optimal strategy to win at this game, players tend to deviate from a uniform distribution of moves, allowing opponents to exploit their strategy and win more games on average. In this project, we implemented several strategies to create bots for the game, including a LoseToWinBot, a RandomBot, and a comboBot. The strategies were inspired by existing literature on multiagent learning in rock, paper, scissors. The effectiveness of our bots was evaluated through a series of matches with bots provided with starter code. Our comboBot, which implemented a mixed strategy based on the recent moves of the opponent, was our most successful bot The project demonstrates the importance of mixed strategies and the ability to adapt to changing opponent strategies in multiagent learning problems. The work could have broader applications in fields such as game theory and artificial intelligence.

References

Bowling, M., and Veloso, M. 2001. Rational and convergent learning in stochastic games.

Mohtavipour, S. S., and Zideh, M. J. 2019. An iterative method for detection of the collusive strategy in prisoner's dilemma game of electricity market.

Wang, L.; Huang, W.; Li, Y.; Evans, J.; and He, S. 2020. Multi-ai competing and winning against humans in iterated rock-paper-scissors game.

Wang, Z.; Xu, B.; and Zhou, H.-J. 2014. Social cycling and conditional responses in the rock-paper-scissors game.