Artificial Intelligence

ROCK PAPER SCISSORS STRATEGIES

12 January 2022

Rabia Kodal

TABLE OF CONTENTS

INTRODUCTION

Rock, Paper, Scissors is an easy game to play. Each player chooses one of the three things (typically by creating the appropriate hand shape on three counts!) and the following rules are used to determine who won that round:

- Paper wraps (beats) Rock
- Scissors cut (beat) Paper
- Rock blunts (beats) Scissors

The purpose of the game is to predict your opponent's choice and select the proper object to defeat them. People find it difficult to make a flawless random series of selections, therefore any pattern that a player creates might be picked up by the opponent and exploited to win the game.

What Is the Process?

The computer calculates the conditional odds of you selecting each of the three things based on the last object you chose. The computer always selects the thing that it believes you are most likely to select.

In this project different types of algorithms are used for prediction and comparison between each other.

MOTIVATION

In this project, we created the rock-paper-scissors game by working on the algorithms offered by the technology of artificial intelligence. Working on this project allowed us to learn the algorithms we use better. Although the rock paper scissors game seems very simple, it can become a game that works with many algorithms. In this game, we can see the comparison of algorithms of AI agents with each other.

RULES OF GAME

Rule-based

Rule-based artificial intelligence systems realize artificial intelligence with a rule-based model. This system is coded by people based on a certain rule and created by using statements in simple coding such as if-else(if-then). A "set of rules" and "set of facts" are needed to build a basic AI model.

We used 4 rule-based artificial intelligence algorithms in our project and compared them with each other. The first of the compared algorithms are Constant Agent and Nash Equilibrium Agent, which work with mode equal to 1. Other rule-based algorithms compared are "Mirror Shift With Constant Agent" and "Mirror Opponent Agent", which work with mode equal to 2.

Mode 1: Constant Agent (user) VS Nash Equilibrium Agent

When the program is run, you will encounter the question "Which mode do you prefer? (Press 1, 2 or 3) You should press 1 to compare these algorithms.Then the program will ask "How many times do you want to play ?" to find out how many times you want to play the game. asks the question. You are free to enter any number you want.

The first player to start the game is the constant agent. According to this strategy, the player will play the same option continuously and automatically. There are **two different options** for the user to compare the game strategies. If the user wants to compare the Constant Agent with the Nash Equilibrium, he/she must repeatedly repeat the value he/she initially gave as input (1= Rock, 2= Paper, 3=Scissors). **The second option** is about trying to beat the Nash Equilibrium algorithm by giving the values that the user wants this time (This part in the comment line, actual part always move 1. If you want to be user please use comment line).

The Nash Equilibrium is the solution often used in multiplayer and non-cooperative games. In games using Nash equilibrium, players cannot win the game by changing their strategies. Nash equilibria for Rock Paper Scissors means that all 3 options have equal probabilities, and it is played 1/3 at a time. The success analysis of the Nash equilibrium is as follows: Suppose the first player will play "Rock". If the second player adopting the Nash balance responds with "Rock", a draw will be achieved in the 1/3 case. If he counters with "Paper" he will win in 1/3 case. If he counters with "Scissors" he will lose at 1/3. In summary, the probability of winning, drawing and losing is equal.

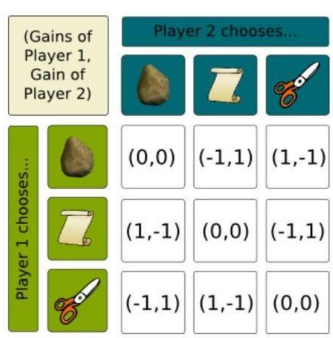

Figure 1: Nash Equilibrium

As a result of this comparison, you will see which Constant Agent or Nash Equilibrium is more successful. If you give the values you want instead of giving constant values for the Constant Agent, you will be measuring the success of the Nash Equilibrium algorithm.

Mode 2: Mirror Shift With Constant Agent VS Mirror Opponent Agent

To see this comparison, you should select the mod value as 2 and you can play as many games as you want. In this strategy, the rule-based algorithm program constantly changes the game with a fixed value at each step. In our project, "Mirror Shift With Constant Agent" always starts with Rock and continues with Scissors twice after Rock. Then it plays Paper twice and returns to the beginning. [R,R,S,S,P,P,R,R,S,S,P ....]

The "Mirror Opponent Agent" strategy will apply the last action of the competitor who started the game first. This strategy is valid when the opponent applies a fixed strategy. It is based on the assumption that if the opponent is executing a strategy, it will be tied and otherwise the probability of losing and winning is equal. In our project "Mirror Opponent Agent" always starts with "rock". The "rock" action here is not important for the initial value. This value could also be "paper" or "scissors". However, our shift value should change as 0,1 or 2 according to the initial state. If we use 0 for Shift, the mirror will be in a similar situation to the opponent. "Mirror Opponent Agent" will copy its next move as the previous move of "Mirror Shift Constant Agent" after entering the initial value rock.

Mirror Shift With Constant Agent		Mirror Opponent Agent
Rock	tie	Rock
Rock	tie	Rock (first move of constant agent)
Scissors	loses	Rock (second move of constant agent)
Scissors	tie	Scissors(third move of constant agent)
Paper	loses	Scissors(fourth move of constant ..)
…..	tie	…….
…….	loses	……….

As a result, the mirror opponent agent always wins by a wide margin against the mirror shift with constant agent.

Knowledge-driven

Focuses on capturing organizational information (data) concisely and draws on this information. This mode of our project has been created to give the best response to these moves by keeping the moves of the opposing player as data. Markov Chain algorithm is used for this mode.

Mode 3: User VS Markov Chain

You should select the mode as 3 to see the performance of the Markov Chain algorithm,. The first player is the user and is free to give whatever value he/she wants.

The Markov chain creates a probabilistic present value by using the values of the previous variable to generate a random variable sequence. More specifically, determining the next variable depends on the last variable in the chain.

Let's give an example chain for the Rock Paper Scissor project.

R | P | S | S | R | P | R | P | P | P | R | R | S | P | R

R:	P:	S:
R -> P = 3/5	P->S = 1/6	S->S= 1/3
R -> S = 1/5	P->R = 3/6	S->R = 1/3
R ->R = 1/5	P->P = 2/6	S->P = 1/3

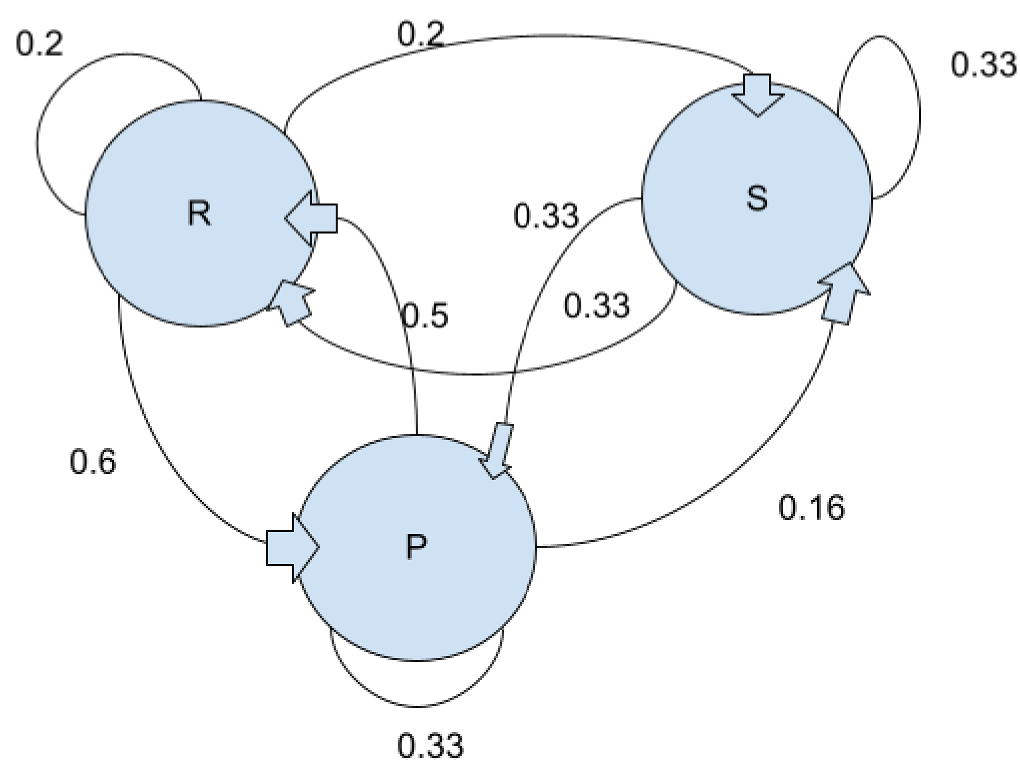

Figure 2: Markov Chain

$$
\begin{array}{c c}
 & \begin{array}{ccc} R & P & S \end{array} \\
\begin{array}{c} R \\ P \\ S \end{array} & \begin{bmatrix} 0.2 & 0.6 & 0.2 \\ 0.5 & 0.33 & 0.16 \\ 0.33 & 0.33 & 0.33 \end{bmatrix}
\end{array}
$$

Figure 3 : Markov Chain Matrix im Rock Paper Scissors

The algorithm keeps which move the user uses after which move in a 3*3 matrix. Let's say the last move of the user is "rock". Let's specify an array that selects a rock value. This goes like this: [1,0,0]. Multiplying this array with a 3*3 matrix will give us the probability with which the user chooses moves after the "rock" value. The player who adopts the Markov Chain algorithm accepts the move corresponding to the largest of these possibilities as the opponent's next move and makes his own move accordingly.

Mode 4: User VS Our Developed Agent

Our developed algorithm is based on gathering the first 30% moves of users. For instance, moves of users: “RRRRSSPP” (R=Rock, P=Paper, S=Scissors) and Our Algorithm creates array then it has “PPPPRRSS”. After then, the algorithm chooses an index from a new array. In this algorithm, we held the users' moves firstly (first, 30% of moves of users.). After then the algorithm creates a new array (the array including the opposite of the user's moves) then the algorithm chooses randomly index from that new array.

AI Progress logic: The AI plays like Nash Equilibrium algorithm until users' all move of 30% and collected data of 30% of the move. After then, the AI starts playing like a knowledge-driven algorithm. Because it creates a winner array moves of the first 30% of move the game. And then the AI play the game over the array (it chooses index from the winner array, randomly).

CODE DESCRIPTION & OUTPUTS

Mode 1: Constant Agent (user) VS Nash Equilibrium Agent

Option 1: If the user wants to play as a constant agent, she/he should constantly enter the value she entered at the beginning. (1 for Rock, 2 for Paper, 3 for Scissors)

```
Which mode do you prefer? (Press 1, 2 or 3)
(1)Constant Agent VS Nash Equilibrium
(2)Mirror Shift Constant Agent VS Mirror Opponent Agent
(3)User VS MarcovChain1
How many times do you want to play ?10
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors1
                 Constant Agent     VS      Nash Equilibrium Agent
    Action :         1                                 2
    Name:           Rock                             Paper
    Result:                                             Win
    Reward:        -1                                   1
```

```
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors1
                 Constant Agent     VS      Nash Equilibrium Agent
    Action :         1                                 2
    Name:           Rock                             Paper
    Result:                                             Win
    Reward:        -4                                   4

What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors1
                 Constant Agent     VS      Nash Equilibrium Agent
    Action :         1                                 3
    Name:           Rock                             Scissors
    Result:          Win
    Reward:        -3                                 3

                >< Nash Equilibrium Agent Won The Game ><
```

User has consistently played Rock, acting as the Constant Agent. The Nash Equilibrium algorithm will always win, as the Constant Agent has a fixed strategy.

Option 2: The user can play any move they want to defeat the "Nash Equilibrium". However, the Nash Equilibrium algorithm always has a 1/3 probability of losing, winning or drawing. Therefore, the user can win, lose or draw the game.

Here is an output where the user wins the Nash Equilibrium algorithm:

```
Which mode do you prefer? (Press 1, 2 or 3)
(1)Constant Agent VS Nash Equilibrium
(2)Mirror Shift Constant Agent VS Mirror Opponent Agent
(3)User VS MarcovChain1
How many times do you want to play ?10
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors1
                  Constant Agent     VS      Nash Equilibrium Agent
    Action :          1                                1
    Name:           Rock                             Rock
    Result:                           Tie
    Reward:         0                                  0
```

```
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors1
                  Constant Agent     VS      Nash Equilibrium Agent
    Action :          1                                3
    Name:           Rock                             Scissors
    Result:          Win
    Reward:         3                                 -3

What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors2
                  Constant Agent     VS      Nash Equilibrium Agent
    Action :          2                                1
    Name:           Paper                             Rock
    Result:          Win
    Reward:         4                                 -4

                 >< Constant Agent Won The Game ><
```

Mode 2: Mirror Shift With Constant Agent VS Mirror Opponent Agent

Mode is selected as 2 to see the comparison of these algorithms. It can play the algorithms as many times as we want. Here, the user will not enter any values, they can only see the output. This algorithm is shifted so that the mirror opposing agent always wins.

```
Which mode do you prefer? (Press 1, 2 or 3)
(1)Constant Agent VS Nash Equilibrium
(2)Mirror Shift Constant Agent VS Mirror Opponent Agent
(3)User VS MarcovChain2
How many times do you want to play ?10
          Mirror Shift Constant Agent     VS      Mirror Opponent Agent
    Action :        1                                   1
    Name:           Rock                              Rock
    Result:                          Tie
    Reward:         0                                  0

          Mirror Shift Constant Agent     VS      Mirror Opponent Agent
    Action :        1                                   1
    Name:           Rock                              Rock
    Result:                          Tie
    Reward:         0                                  0

          Mirror Shift Constant Agent     VS      Mirror Opponent Agent
    Action :        2                                   1
    Name:           Scissors                            Rock
    Result:                                             Win
    Reward:        -1                                   1
```

```
Action :        1                                   3
Name:           Rock                            Paper
Result:                                             Win
Reward:        -3                                   3

      Mirror Shift Constant Agent     VS      Mirror Opponent Agent
Action :        1                                   1
Name:           Rock                            Rock
Result:                          Tie
Reward:         -3                                  3

      Mirror Shift Constant Agent     VS      Mirror Opponent Agent
Action :        2                                   1
Name:           Scissors                            Rock
Result:                                             Win
Reward:        -4                                   4

      Mirror Shift Constant Agent     VS      Mirror Opponent Agent
Action :        2                                   2
Name:           Scissors                            Scissors
Result:                          Tie
Reward:         -4                                  4

             >< Mirror Opponent Agent Won The Game ><
```

Mode 3: User VS Markov Chain

Note:Please enter a high number of games when running the Markov Chain algorithm.

To see the performance of the Markov Chain algorithm and compete with it, you must enter 3 as the mode. The first player is the user and can make any move to beat the Markov Chain algorithm. After the first two values entered by the user, the Markov Chain algorithm keeps the possibilities of which move the user makes after which move and makes the move accordingly. According to the code written, the player who adopts the Markov Chain makes his first two moves as "rock". "Rock" is a randomly chosen move, as you will have no data while playing the first two moves.If you enter a value such as 15, the Markov Chain algorithm will not be successful because it does not have enough data. It would be reasonable to enter a value like 40 and then make your moves. In order to measure the performance of Markov Chain faster, there is a comment line in the file where the code is written. If you open this comment line, the Markov Chain algorithm will automatically play with the Nash Equilibrium algorithm, not the user.

```
Which mode do you prefer? (Press 1, 2 or 3)
(1)Constant Agent VS Nash Equilibrium
(2)Mirror Shift Constant Agent VS Mirror Opponent Agent
(3)User VS MarcovChain3
How many times do you want to play ?40
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors2
          Mirror Shift Constant(User)     VS                 Markov Chain
    Action :        2                                    1
    Name:           Paper                             Rock
    Result:          Win
    Reward:        1                                  -1
```

```
t role should the constant agent play?
ess (1) for Rock
ess (2) for Paper
ess (3) for Scissors2

, 0, 0], 0.0, 1.0, 0.0]
      Mirror Shift Constant(User)     VS                 Markov Chain
Action :        2                                    3
Name:           Paper                             Scissors
Result:                                              Win
Reward:        -12                                    12


             >< Markov Chain Won The Game ><
```

After 40 steps, Markov Chain has won the game by far.

```
Which mode do you prefer? (Press 1, 2 or 3)
(1)Constant Agent VS Nash Equilibrium
(2)Mirror Shift Constant Agent VS Mirror Opponent Agent
(3)User VS MarcovChain3
How many times do you want to play ?30
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors2
          Mirror Shift Constant(User)     VS                 Markov Chain
    Action :        2                                   1
    Name:           Paper                             Rock
    Result:          Win
    Reward:        1                                  -1
```

```
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors2
2
[[0, 0, 0], 0.4, 0.13333333333333333, 0.4666666666666667]
          Mirror Shift Constant(User)     VS                 Markov Chain
    Action :        2                                   1
    Name:           Paper                             Rock
    Result:          Win
    Reward:        -4                                  4

                  >< Markov Chain Won The Game ><
```

After 40 steps, Markov Chain has won the game by far.

If Markov Chain plays against the Nash Equilibrium algorithm, there is a possibility of losing because the probability of Nash Equilibrium winning is always 1/3. If the user enters inconsistent values independent of a strategy, the Markov Chain algorithm may lose again, the reason for losing is due to the same as Nash Equilibrium. Markov Chain and Nash Equilibrium are run 25 times with 100 games. According to the results, Markov Chain can achieve 60% - 70% successful performance.

Mode 4: User VS Our Developed Agent

To see the performance of the algorithm we developed, you need to play a large number of games. According to the example in the outputs, it was played 49 times and our algorithm won with 13 differences. The advantages are directly proportional to being played a lot. The disadvantages are that if the player always makes the same move, if the plot later changes its move, ai is defeated. As a result, the algorithm we developed is superior to the Nash Equilibrium algorithm, which shows that it has a success rate of over 30%.

```
Which mode do you prefer? (Press 1 or 2)
(1)Constant Agent VS Nash Equilibrium
(2)Mirror Shift Constant Agent VS Mirror Opponent Agent
(3)MarcovChain
(4)User game vs Our developed Random Agent
4
How many times do you want to play ?
49
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors2
          User play Agent     VS      Our developed Random Agent
    Action :        2                                    2
    Name:           Paper                             Paper
    Result:                          Tie
    Reward:         0                                   0

What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
```

Started

```
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors3
[2, 2, 2, 2, 3, 3, 3, 3, 3, 3, 3, 3, 1, 1, 1, 1]
         User play Agent     VS      Our developed Random Agent
    Action :       3                                   3
    Name:          Scissors                            Scissors
    Result:                         Tie
    Reward:        -5                                  5
```

Created Array by the AI

```
What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors2
         User play Agent     VS      Our developed Random Agent
    Action :       2                                   3
    Name:          Paper                            Scissors
    Result:                                            Win
    Reward:       -13                                   13

What role should the constant agent play?
 Press (1) for Rock
 Press (2) for Paper
 Press (3) for Scissors3
         User play Agent     VS      Our developed Random Agent
    Action :       3                                   3
    Name:          Scissors                            Scissors
    Result:                         Tie
    Reward:        -13                                  13

                 >< Our developed  Agent Won The Game ><
```

End of the game Our algorithm Won

REFERENCES

[1] Preda, G. (2020, November 28). *Rock Paper Scissors Strategies*. Kaggle. https://www.kaggle.com/gpreda/rock-paper-scissors-strategies

[2] Gabrys, P. (2021, June 30). *How to win over 70% matches in Rock Paper Scissors - Towards Data Science*. Medium. https://towardsdatascience.com/how-to-win-over-70-matches-in-rock-paper-scissors-3e17e67e0dab

[3] Brownlee, J. (2019, September 25). *A Gentle Introduction to Markov Chain Monte Carlo for Probability*. Machine Learning Mastery. https://machinelearningmastery.com/markov-chain-monte-carlo-for-probabi lity/

[4] Saurel, S. (2019, March 16). *Creating a Rock Paper Scissors Game in Java with a Markov Chain for the AI*. Medium. https://ssaurel.medium.com/creating-a-rock-paper-scissors-game-in-java-w ith-a-markov-chain-for-the-ai-7672954fd7f6