

Implementation of Self learning Blackjack player using Markov Decision process

Mahesh Saravanan
Master Artificial Intelligence

University of Applied Sciences Würzburg Schweinfurt
Würzburg, Germany

maheshhss750@gmail.com

Abstract— This paper describes the implementation of a self-learning black jack player using Markov decision process. The environment of the game is developed in python and learning of basic strategy is executed using Q-learning method. The growth of win rate and change in policy for different rules and hyper parameters are experimented.

Keywords— *Q-table, Epsilon greedy method, state-action pair, Environment, Monte Carlo method, Bellman's equation.*

I. INTRODUCTION

Reinforcement learning is one of the three basic machine learning methods, alongside unsupervised and supervised learning. It is a powerful technique that the model doesn't need any training data to learn rather, it learns by interacting with the environment. The performance of the model gets improved over the time. The reinforcement learning algorithms are broadly classified into 3 types i.e., Value based, Policy based and model based. It has a wide range of applications like robotics, data mining, gaming etc.,

II. BLACKJACK

Blackjack is a popular casino game originated from France in 1700's and now being played in many casinos around the world. It is played with cards; each card is associated with a number from 0 to 11 and a collection of 52 cards is called a deck. It can be either played one to one (player to dealer) or with multiple players and dealer. The dealer is the person who manages the cards and rewards and the players will bet against the dealer.

A. Rules of Blackjack

The 52 cards are further grouped into 4 categories, such as Spades, Hearts, Diamonds, and Clubs and each category has 13 cards (9 number cards, Ace, Jack, King, Queen). King, Jack and Queen cards have the value of 10, whereas Ace can be considered as 1 or 11. Initially, all the players and the dealer are provided with one random card and the dealer's card is revealed to all the players. The game is progressed in rounds and at each round the player can draw a card from the dealer till the sum of value of all the cards of a player is less than 21 and this action of drawing a card is referred as "hit". If the all the players in the game stops drawing the card (stick), the dealer will draw cards from the deck until it reaches the value of dealer's threshold and this value is set by the casino (usually 17). The results will be evaluated after dealer stops

drawing the card. If either of player and dealer's total crosses 21, they will lose the game(busted), if player's total is higher than dealer's total, the player wins and vice versa. If the totals are equal, no one will be rewarded. The Rules of black jack might change a bit among casinos. The Fig.1 illustrates the types of cards and its values





















































	Ace	2	3	4	5	6	7	8	9	10	Jack	Queen	King
Clubs													
Diamonds													
Hearts													
Spades													

Fig.1. Classification of cards

B. Strategies of the game

a. Basic strategy

The basic strategy of the game is to make the player's total as close as possible to 21 without crossing it so the player should stop drawing the card if the player's total is close to 21 to avoid getting busted and try to maximize the total to beat the dealer.

b. Card Counting

The probability of the next card in the deck can be calculated if the player keeps track of every used card which could help the player decide hit or stick. The deck will be reset if the count of cards falls below minimum card required for a game. One of the popular methods of card counting is High - Low card counting which assigns value of -1 and 1 to higher value cards and lower value cards respectively. The probability of the next card can be calculated based sum of these values of all used cards.

III. FINITE MARKOV DECISION PROCESS

Markov decision process is a method in reinforcement learning that formalize sequential decision-making process. The major components of Markov Decision process are Environments, Agents, Policy, State, Action and Reward. The Fig. 2 illustrates the working flow chart of Markov Decision Process.

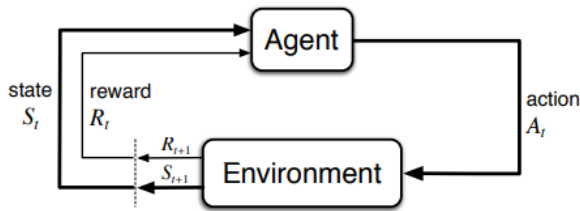


Fig.2 Flow chart of MDP

The Agent is the Decision maker which follows the policy π and takes the action A_t at time step t on the environment. As a result, the environment transformed from state S_t to S_{t+1} and the agent receives the reward of R_{t+1} . This is a sequential process and the policy get updated at each time step. The policy is updated in such a way that it improves the cumulative future rewards. The reward R_{t+1} can be represented as a function of state S_t and A_t .

$$f(S_t, A_t) = R_{t+1}$$

A. Environment

The rules and the algorithm of the blackjack game is defined in environment. It has various functions such as resetting deck, choosing random card for the dealer and player, checking for usable ace, evaluating the final results, returning the current state and reward for the corresponding action, etc.,

B. State

The state is defined by the dealer's first card and the sum of all cards in the player's hand. The possibilities of dealer's first card could be anything between 1 to 10 and sum of all cards in the player's hand is could be any number between 1 to 31. Since the player cannot draw another card when the total reaches 21, the maximum total which the player could obtain is 31 by taking the maximum value card 10 when the total is 21. The maximum possible state combination is 310 (10×31).

C. Action

There are two possible action which the agent could take, they are drawing another card (hit), stay with the current total (stick). These actions can be chosen randomly (exploring) or based on the states (exploiting).

D. Reward

For every step the agent takes, a numerical reward of 1 or 0 is given. For every successful step without making the sum of player's total greater than 21, agent receives the award of 1 and vice versa. The agent's role is to maximize the cumulative reward,

$$G_t = R_t + R_{t+1} + R_{t+2} + \dots + R_T$$

where T is the final game.

E. Discounted Reward

The cumulative reward is sum of future rewards from the current state. The agent should give more importance to immediate reward than further future rewards. In order to achieve this, the concept of discounted reward is adopted, in

which reward at each time step is multiplied with some weight called discount factor, γ which is less than 1. This reduces the importance of further future rewards and importance of immediate reward is increased as future rewards are heavily discounted. The agent should update its policy in order to improve the cumulative discounted reward which is given by

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots + \gamma^{T-1} R_T$$

IV. Q – LEARNING

Q-learning policy is defined by q table, which is the table of state action pair. It stores the probabilities of all possible actions at each state. It is formed by states and actions of the environment. The dimension of Q-table for the basic strategy of Blackjack is 310×10 (Maximum states and action) and the dimension of Q-table for card counting strategy is $((310 \times 2) \times 10)$ as each single state have 2 sub states. This table is termed as policy as the agent will follow the table to take action at particular state. The values of Q-table is updated for each reward obtained using Bellman's equation.

a. Bellman's Equation

The value in the q -table is updated for every reward the agent obtains, using bellman's equation.

$$q(s, a)^{new} = (1 - \alpha) q(s, a)^{old} + \alpha (R_{t+1} + \gamma * \max(q(s', a')))$$

R_{t+1} is the reward obtained by taking action a at state s and $\max(q(s', a'))$ is the state at which the maximum future reward is obtained. The learning rate, α is the weight for each reward at which the value of q is updated and γ is the discount factor.

V. EPSILON GREEDY POLICY

The epsilon greedy approach is used to manage the exploration and exploitation trade-off. Exploration is the method of choosing the action randomly from the action set $\{0,1\}$. This helps in exploring the various states and action. Exploitation is the method of following the policy i.e., q -table and maximizing the reward. Epsilon greedy policy is important to increase the knowledge of each state and at the same time maximizing the reward. Initially, the agent should explore a lot to update the knowledge about the environment and it exploits a lot when it gets close to the final time step. This is achieved by the exploration – exploitation ratio ϵ , the agent will explore more when this ratio is higher and vice versa. The value of ϵ will decay at the rate of epsilon decay rate and the value of epsilon reaches close to zero at final time step.

VI. EXPERIMENT AND RESULTS

A. Implementation

The self-learning black jack player is implemented in python. The implementation includes the development of environment and Markov decision process. Initially, an array of zeros for q_table with dimension 10×310 is created. The value of epsilon is set to 1 and game is played for 100000

times. A single game will end only when the sum of agent's total is greater than 21 or when the agent decides to stick. After the termination of one game, the results are evaluated and the dealer and agent total will be reset to 0. Table 1. tabulates the possible results and its corresponding rewards

Result	Reward
Both dealer and Player > 21	0
Dealer > 21 and Player < 21	1
Dealer < 21 and Player > 21	-1
Dealer > Player Dealer and Player < 21	-1
Dealer < Player Dealer and Player < 21	1
Dealer = Player Dealer and Player < 21	0

Table 1. Evaluation table

The values of q table are updated for each game and the value of epsilon is decayed by 0.0001 for each game played. The game is played with 8 decks and the decks will be reset if the number of cards in decks fall below minimum level. The win rate is calculated for each 50 game and visualized. The values of q tables are updated and the policy is represented graphically. The epsilon reaches the specified minimum value after certain games.

B. Card Counting

This is the method of tracking the number of used cards and finding the probability of next card, so that the action can be varied accordingly. The probability of value of cards in the deck is calculated and the sum of probabilities of these values which makes the total of the agent less than or equal to 21 when added is computed. This probability indicates the probability of drawing a card without getting busted. Based on this probability each state will have 2 distinct states. Hence the size of state action table gets doubled i.e., [620x2].

C. Observations

The overall win rate of 40% is observed in 1000000 games. The win ratio is calculated for every 50 games and visualized in Fig.3.

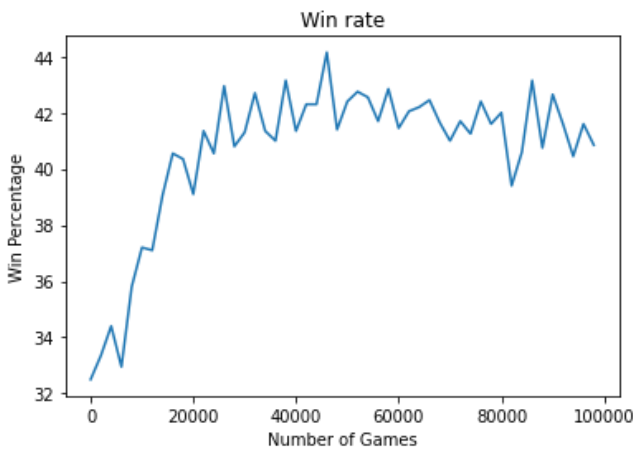


Fig. 3 Win rate over 1000000 games

The gradual growth from 28% to 45% is noticed in the 40000 games and it remains almost constant in rest of games.

The value of epsilon approaches close to 0.1 after 40000 games which shows the agent explore more in first 40000 games and exploits in rest of the game. In total it exploits 92% and explores 2%. The decay of Epsilon over 10^6 games is shown in the Fig. 4

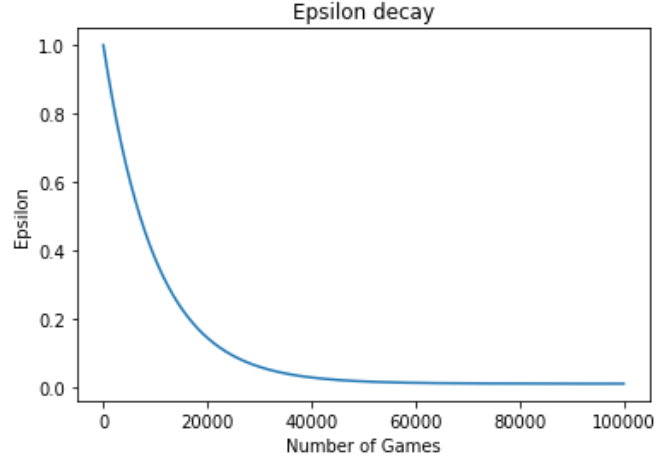


Fig.4 Epsilon decay

The policy (Q-table) converges to constant value after 60000 games and it is visualized in a decision table (Fig.6) and decision graph. 0 denotes stick and 1 denotes hit in the decision table. Each row in the decision table indicates the players total ranging from 1 to 21 and each column represent dealer's face up card 1 to 10.

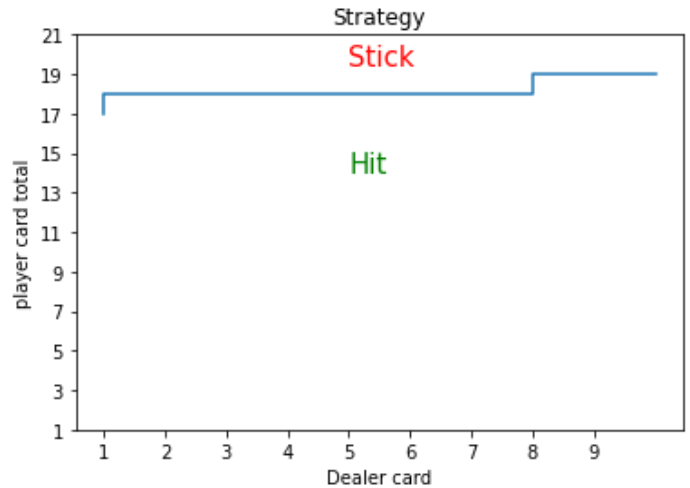


Fig.5. Decision Graph

```
[ [0 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [1 1 1 1 1 1 1 1 1 1]
  [0 0 0 0 0 0 0 0 1 1]
  [0 0 0 0 0 0 0 0 0 0]
  [0 0 0 0 0 0 0 0 0 0]]
```

D. Rule Change

Dealer threshold	Win percentage
1	93
2	92
3	91
4	91
5	90
6	89
7	86
8	83
9	77
10	73
11	55
12	50
13	47
14	46
15	42
16	40
17	40
18	40
19	42
20	52
21	82

References

- [1] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT Press, Cambridge, MA, 2 edition, 2018.
- [2] Edward O. Thorp. Beat the Dealer. Vintage, New York, 1966.
- [3] Watkins, C.J.C.H., Dayan, P. Q-learning. Mach Learn 8, 279–292 (1992).
<https://doi.org/10.1007/BF00992698>
- [4] Brooks, Steve, et al., eds. Handbook of markov chain monte carlo. CRC press, 2011.
- [5] Baldwin, Roger R., et al. "The optimum strategy in blackjack." Journal of the American Statistical Association 51.275 (1956): 429-439.