Machine Learning: Reinforced Learning

Robert Horton[*]
University of Colorado Springs Colorado
rhorton2@uccs.edu

Gaberial Yeager[†]
University of Colorado Springs Colorado
gyeager@uccs.edu

September 23, 2021

Abstract

ABSTRACT HERE!!!!!

1. Introduction

Blackjack is a commonly known game that can be played with different types of methods of approach.

2. Background

Reinforcement learning, although considered a part of machine learning, is very different from the other types of machine learning. Two main categories that the different types of machine learning fall under are supervised learning[1] and unsupervised learning [2] [Sutton, R. S., & Barto, A. G. , 2018]. Although supervised learning can be used to speed along the process of machine learning this is not always an option when deploying a machine learning process into an unexplored environment. These types of implementations call upon reinforcement learning to perform an unsupervised learning process. Now when performing these unsupervised approaches of machine learning there are many different methods that have been used and studied over the years. Finding out which method is the most efficient and works best for training a machine to learn how to play blackjack will be the topic of this paper.

When considering which method of RL[3] to use when wanting to teach a machine how to play the game blackjack, there are many different options to consider. When considering the game blackjack the next cards that will be drawn are always unknown to the players and dealers. The hole card[4] is also a card that is not known to any player including the dealer, making this game an even more inherently stochastic model. Since bets are placed while playing this game, maximizing the profit on return for each game played is very important and is hard to do with the uncertainty of what the next card dealt will be and what hole card has been dealt to the dealer. This models the perfect Markov Decision Process where each game promises states and actions that are unknown and have not been explored yet.

When considering reinforcement learning it can be broken down further into two categories “On Policy” or “Off Policy” learning. On policy RL approaches are considered as methods that involve learning agents that learn from the value function denoted as v_π or q_π according to the current occupied state which can also

[*] "Denim"

[†] "Gabe"

[1] Explanation of supervised learning

[2] Explanation of unsupervised learning

[3] Reinforcement Learning

[4] Face down card that is dealt to dealer at the beginning of each game

Table 1: *Example table*

Blackjack RL Methods		
Name	On/Off-Policy	Greedy
Q Learning	Off-Policy	True
SARSA	On-Policy	True
Temperoal Differnece	Off-Policy	False

be denoted as $v_\pi(s)$ or $q_\pi(s)$ for $s \in S$ where S is the set of all possible states. Off policy works when the learning agents learn from the value function of the previous occupied state [GeeksforGeeks]. Above is a table that will be referred to when discussing what methods will be used for approaching these problems.

2.1. Q-Learning

One of the most well known and oldest methods listed is the "Q-Learning" approach. This approach is considered an off-policy TD control algorithm that can be defined as

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\big[R_{t+1} + \gamma \binom{max}{a} Q(S_{t+1}, a) - Q(S_t, A_t)\big]$$

[1]

, where the steps taken are independent of what the next updated policy might be based upon that action. Depending upon the value of ϵ the frequency of random steps taken can vary and make the model even more stochastic. What is important to note is that although it might be able to ensure a steady increase in profit after each game faster a not so greedy approach could in the long run ensure more consistent and bigger profits over a longer period of multiple games. With each action taken the next one could be random. The two actions are independent in the sense that the previous step was not taken based on the reward given from the action performed to move into now occupied state.

2.2. SARSA

The "SARSA" [5] is a slightly modified version of the Q -learning approach where the modification of the policy differs. Q-learning is considered an off policy where SARSA is an on policy [GeeksforGeeks].

2.3. Temporal Difference

The third and last method that will be considered for application in this problem is "Temporal Difference" suggested by Dr. Andrew G. Barto [Sutton, R. S., & Barto, A. G. , 2018]. Temporal difference involves adjusting the policy based on a estimation of the sum of future actions to be made. This method, unlike the rest, takes a greedy approach that will form decisions on what actions to take next based off of the newly calculated policy from known possible actions in states. All of these are important and vary in the approaches for adjusting and reacting to the policy that helps are self-learning machines decide what moves to make next and at what rate they learn new things and ultimately uncover the best policy to solving a problem.

3. Related Work

4. Methodology

A statement requiring citation [Figueredo and Wolf, 2009].

5. Conclusion

[5] State Action Reward State Action

References

[Sutton, R. S., & Barto, A. G. , 2018] Sutton, R. S., & Barto, A. G. (2018). Reinforcement Learning: An Introduction. Cambridge (Mass.): *The MIT Press.*, .

[GeeksforGeeks] "Sarsa Reinforcement Learning." *GeeksforGeeks*,
24 June 2021,
www.geeksforgeeks.org/sarsa-reinforcement-learning/.

[Figueredo and Wolf, 2009] Figueredo, A. J. and Wolf, P. S. A. (2009). Assortative pairing and life history strategy - a cross-cultural study. *Human Nature*, 20:317–330.