Playing BlackJack with Reinforcement Learning Algorithms

Group G, SDSC4001, School of Data Science, City University of Hong Kong

Introduction

Project Objective : Obtain the possible best policy on playing Blackjack .
-> maximize the rewards of the player in the game

Why we choose this topic ?
Blackjack is a well known game around the world, which is commonly used for developing RL algorithms and their performance evaluation.

RL model perform well in stochastic environments, which matches the structure of Blackjack.
Simplified action and state in the game = Easy MDP formulation

Model-Free reinforcement learning algorithm

GLIE Monte Carlo Control

Algorithm:

Initialization: Q(s,a) = 0, N(s,a) = 0, ∀s ∈S, ∀a ∈A
For loop (looping over episodes i):
Set epsilon <- 1/k, πk = epsilon-greedy(Q)
Get episode observations
Define return G in step t.
For every state-action pair visited in episodes i, and for the first time t that (s,a) is visited in episodes i.
N(s,a) = N(s,a)+1
Q(s,a) = Q(s,a)+(1/N(s,a))*(G-Q(s,a))

ε-greedy policy:

$$\pi'(s) = \arg\max_{a \in \mathcal{A}} Q(s,a)$$

For loop (over episodes):
epsilon_start=1.0
epsilon_decay=0.99999
epsilon_min=0
epsilon = eps_start*(eps_decay^(episodes-1))

SARSA

The formula of SARSA is given as:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \cdot Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t))$$

Algorithm:

The algorithm of SARSA is described as:

1. Initialize the Q value (Q(s,a))
2. Give a observation to the state (S0=s0)
3. Choose an action (At) based on ε-greedy policy π0
4. Take the action A0~π0(S0), and observe the reward, R1, also the new state,S1.
5. Repeat the following steps for each episode until terminate(t=0,1,2…):
   - 5.1. Take action At+1 ~πt(St+1) and observe (Rt+2,St+2)
   - 5.2. Update the Q-value for the state with the observed reward and expected reward for the next state.
   - 5.3. Update the policy πt+1 with εt+1-greedy(Q)

SARSAMAX

The formula of Q-learning is given as:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha_t \cdot \left(R_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(S_{t+1}, a) - Q(S_t, A_t)\right)$$

Algorithm:

1. Initialize the Q value (Q(s,a))
2. Give a observation to the state (S0=s0)
3. Initialize ε-greedy policy π'0
4. Repeat the following step for each episode until terminate(t=0,1,2…):
   - 4.1. Take action At ~π't(St) and observe (Rt+1,St+1)
   - 4.2. Update the Q-value for the state using the observed reward and the maximum reward for the next state.
   - 4.3. Update π't+1 with ε-greedy(Q)

BlackJack

Card type : Poker cards without Jokers
Number of Player : 1 ,   Number of dealer : 1

Actions' option : Hit or Stand

Setting / Rules :
1. All cards in players' hand should be face-up
2. Dealer round starts after player's round is finished
3. Dealer plays with fixed strategy

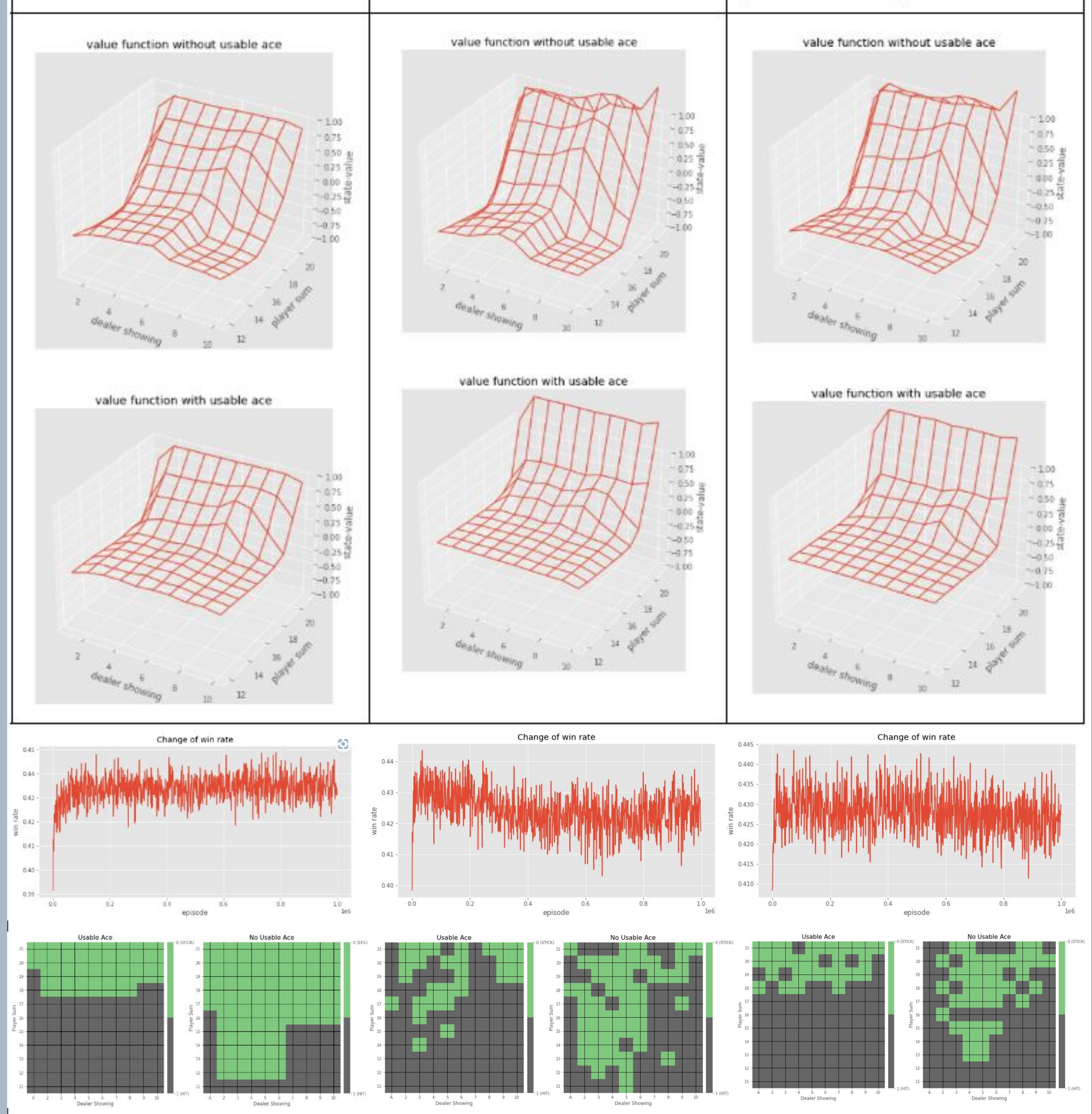
Game Procedure :
Dealer starts with one face-up card and one face-down card
Player starts with two face-up cards.
Player's Round : Option 1 - Hit until bust (exceeding 21)
/ Option 2 : Stick (stop)
Dealer's Round : draw more cards until total card value >= 17

Winning :
-> total card values of player's hand > that of dealers' hand ; otherwise, player loses.
-> dealer bust when player does not bust.

Results

Interpretation:

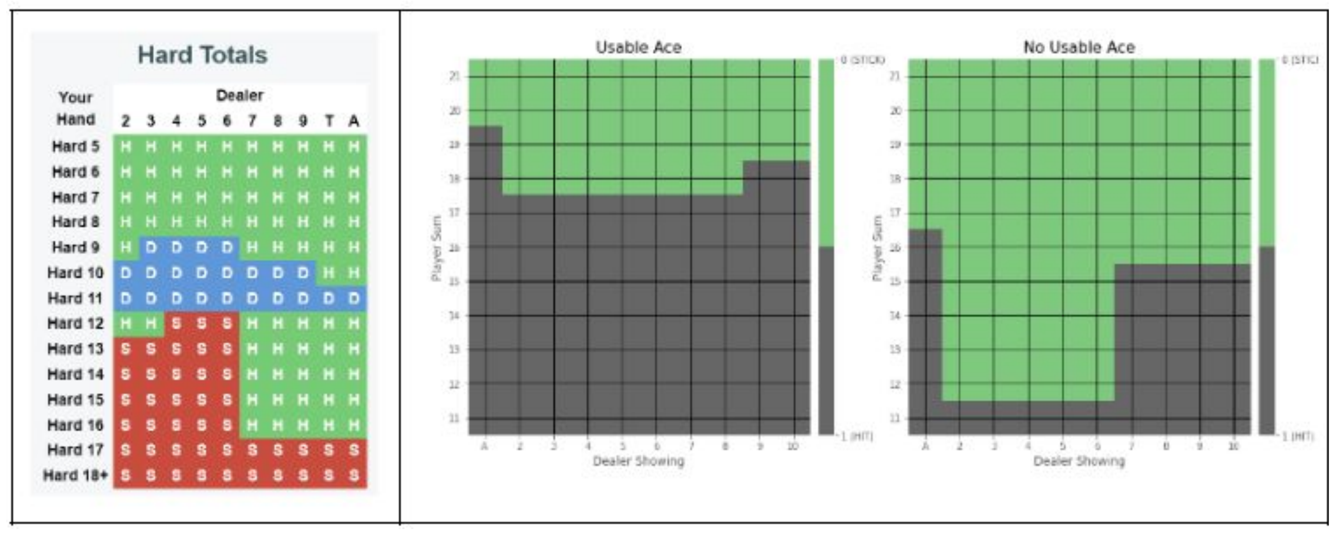
GLIE Monte Carlo control	GLIE SARSA	GLIE Q-Learning (SARSAMAX)



- State value function
  highest state values correspond to when the player sum is something like 20 or 21
  strategy for having a usable ace is more aggressive
- Win rate curve
  Winning rate varies from 0.43 to 0.44, 0.42 to 0.44 and 0.415 to 0.44
- Optimal Policy
  black color region represents the hit action, and the green color region represents the stick action

Dicussion

Blackjack strategy table comparative analysis:



- Left table: If dealer showing 7, 8, 9, T (having a value of 10) -> should hit, risk for high
- For MC control: Similar performance!
- For other algorithms like SARSA and SARSAMAX -> having similar patterns or stick with strategy table of MC control

	Monte Carlo methods	Temporal-Difference learning
Varaince	High	Low
Bias	Zero	Some
Initial Value	Not sensitive	Sentitive
Learning Speed	Wait until the end of the episode	Learn online after every step
How to Learn	Learn from complete sequences	Learn without the final outcome, from incomplete sequences
Environment	Episodic (terminating)	Continuing (non-terminating)

Conclusion

Goal : Obtaining the possible best Hit-Stand policies on the game of Blackjack.

Impremented Algorithm : GLIE Monte Carlo , GLIE Sarsa, and Q-learning.

Winning rate varies from 0.43 to 0.44, 0.42 to 0.44 and 0.415 to 0.44, respectively, which is similar.

Future possible works :
1. Explore the effect of different strategies that already hold together
2. Try some extensions of RL algorithms such as Deep Q-network and Bayesian Q-learning algorithm .

Reference

1. Gautam, A. (2019, March 15). *Introduction to reinforcement learning (coding sarsa) - part 4*. Medium. Retrieved November 17, 2022, from https://medium.com/swlh/introduction-to-reinforcement-learning-coding-sarsa-part-4-2d64d6e37617
2. *Simple reinforcement learning: Q-learning - towards data science*. (n.d.). Retrieved November 16, 2022, from https://towardsdatascience.com/simple-reinforcement-learning-q-learning-fcddc4b6fe56
3. *Reinforcement learning*. Reinforcement Learning - Algorithms. (n.d.). Retrieved November 17, 2022, from https://www.cse.unsw.edu.au/~cs9417ml/RL1/algorithms.html
4. Thoma, M. *What are the advantages / disadvantages of off-policy RL vs on-policy RL?* Data Science Stack Exchange. Retrieved November 17, 2022, from https://datascience.stackexchange.com/questions/13029/what-are-the-advantages-disadvantages-of-off-policy-rl-vs-on-policy-rl
5. *Temporal difference learning (TD learning)*. Engati. (n.d.). Retrieved November 17, 2022, from https://www.engati.com/glossary/temporal-difference-learning
6. Jordan J Hood. (2021, May 10). *Reinforcement learning: Temporal difference (TD) learning*. Jordan J Hood. Retrieved November 17, 2022, from https://www.lancaster.ac.uk/stor-i-student-sites/jordan-j-hood/2021/04/12/reinforcement-learning-temporal-difference-td-learning/
7. *Blackjack - learn the rules, strategy and more at blackjackinfo*. BlackjackInfo.com. (2017, December 7). Retrieved November 17, 2022, from https://www.blackjackinfo.com/
8. *Beating Blackjack - A Reinforcement Learning Approach.* (n.d.). Retrieved November 17, 2022, from https://web.stanford.edu/class/aa228/reports/2020/final117.pdf
9. *Learning to play Blackjack with Deep Learning and Reinforcement Learning* (January 2019). Retrieved November 17, 2022, from https://i.cs.hku.hk/fyp/2018/fyp18013/reports/InterimReport_3035238565.pdf
10. Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.