INF581: Reinforcement Learning Project

Anonymous Authors

***Abstract*—As a popular casino card game, Blackjack has often been studied in order to devise strategies for improving the likelihood of winning. This project aims to perform a comparative analysis of different reinforcement learning algorithm in multiple Blackjack environments.**

I. Introduction

Blackjack[1] is the most widely played casino banking game in the world. It uses 52 cards and and descends from a global family of casino banking games known as Twenty-One. The objective of this game is to win money by obtaining a card point total higher than that of the dealer without exceeding 21. Blackjack possesses an inherent stochastic, which presents a challenge for machine learning algorithms. In fact, such algorithms fail to take into consideration the reward structure of the game. As an optimal Blackjack strategy maximizes the cumulative financial return in the long run, Reinforcement Learning is well suited for this problem.

In this problem, the challenge is centered around the selection of the state space S. Fist, in the game of Blackjack, card counting is a strategy used to determine whether the player or the dealer has an advantage on the next hand. Taking into consideration card counting would result in an extremely big state space, as we would have to account for used and unused card after every round. In addition, the state space also needs to be modeled in respect to the Markov Property.

The contributions of our project are two-fold. First, we perform a comparative analysis of different reinforcement learning (RL) algorithms in multiple Blackjack environments. Second, we provide an improved state parametrization for Blackjack and achieve better performance than the state of the art on the basis of our selected metric (percentage of wins against the dealer).

The limitations

II. Background and Related Work

A. Related Work

Blackjack has been studied through years in the reinforcement learning framework because of its stochastic nature.

The Q-learning algorithm was used to determine a Blackjack strategy that outperform the random strategy in [2].

Value iteration, Q-Learning and SARSA were used in [3] to design an optimal strategy for a simpler version of the Blackjack with just two actions : **HIT** and **STAND**. The paper also contains a comparison between two state parametrizations based on the winning percentage of the agent and shows the superiority of reinforcement learning approaches over the random strategy.

[4] compare the mean rewards per episode in Blackjack environments for six different reinforcement learning algorithms : Deep Q-Network (DQN), Trust Region Policy Optimization, Advantage Actor Critic, Actor Critic with Experience Replay, Proximal Policy Optimization and Actor Critic using Kronecker-Factored Trust Region.

B. Background on Reinforcement Learning

A reinforcement learning problem is characterized by an $agent$ which evolves in an *environment* described by its *state*. The model, which is the *agent* maps the state of the environment to *actions* and obtain *reward* signals. The goal of the agent is to maximize future rewards. We have the following setting :

- S state space, $s_t \in S$
- A action space, $a_t \in A$
- r reward function, $r_t = r(a_t, s_t)$

The environment can be *stochastic* or *deterministic*. In *deterministic* environments, the next state is known for certain when an action is given. *Stochastic* environments have a *transition function* p such that $s_{t+1} \sim p(.|s_t, a_t)$.

The behaviour of the agent is defined by a *policy*. The choice to take the action a_t given the observation s_t is translated by the following equation :

$$\begin{aligned} a_t &= \pi(s_t) \\ \pi &: S \to A \end{aligned} \tag{1}$$

The policy can be *stochastic*, it is the case when it provides a conditional probability distribution over actions given the current state :

$$\begin{aligned} a_t \sim \pi(.|s_t), \pi(s_t) &\in [0,1] \\ \pi : S \times A &\to [0,1] \end{aligned} \tag{2}$$

We can define the return G_t (over an infinite horizon) as :

$$G_t = \sum_{k=0}^{\infty} \gamma^k R_{k+t+1} = R_{t+1} + \gamma . R_{t+2} + \gamma^2 R_{t+3} \dots \tag{3}$$

with a *discount factor* $\gamma \in (0,1)$ which indicates the relative value of closer rewards. We aim at finding the policy which reflects the decision to take actions A_t in order to maximize G_t.

From G_t, we derive the **Value function**

$$V^{\pi}(s) = \mathbb{E}[G_t | S_t = s] \tag{4}$$

and the **Action-Value function**

$$Q^{\pi}(s,a) = \mathbb{E}[G_t | S_t = s, A_t = a] \tag{5}$$

Our optimization problem is to find π^* such that :

$$\pi^*(s) = \underset{a \in \mathcal{A}(s)}{\operatorname{argmax}} Q^{\pi_*}(s,a) \tag{6}$$

Or equivalently :

$$\pi^*(s) = argmax_\pi V^\pi(s) \tag{7}$$

The computation of the optimal policy can be done using iterative algorithms such as **Q-Learning**[7] and **SARSA**[7]. These algorithms are referred to as *model-free* as opposed to *model-based* since they try to estimate the optimal policy without using or estimating the stochastic transitions in the environment.

Q-Learning relies on the update

$$Q(S_t, A_t) = Q(S_t, A_t) + \alpha[(R_t + \gamma \max_{a \in \mathcal{A}(s)}) - Q(S_t, A_t)] \tag{8}$$

III. The Environment

The game begins with two cards dealt to the dealer and the player. One of the dealer's cards is face up and the other is face down. If the sum of the cards in the player's hand is 21, it is called a natural. He wins, unless the dealer also has a natural, in which case the game is a draw. If the total sum is different from 21, he can ask for additional cards, one by one (hits), until he stops (sticks) or exceeds 21. In the latter case, the player loses, and if he decides to stop, it is the dealer's turn. The dealer hits or sticks according to a fixed strategy without choice: he sticks on any sum equal to or greater than 17, and hits otherwise. If the dealer's cards exceed 21, he loses, otherwise the winning player is the one whose sum of cards in hand is closest to 21. If the player holds an ace that he could count as 11 without exceeding 21, then the ace is said to be usable.

Let us now define the problem environment: the state space, the action space, and the reward function. The state of the game is the components that matter and affect the winning chance. Firstly, the most important is card sum, the current value on hand. Secondly, there are two more factors that contribute to game wining, the usable ace and dealer's showing card. As a result, the state would have 3 components player's current card sum, usable ace and dealer's showing card.

$$S = \{0, \ldots, 31\} \times \{1, \ldots, 10\} \times \{0, 1\}$$

Furthermore, actions are clear as one can only have 2 possible actions, either **HIT** or **STAND**.

$$\mathbb{A} = \{0, 1\}$$

Finally, reward would be based on the result of the game, where we give 1 to a win, 0 to a draw and -1 to a loss.

IV. The Agent

The agent is the player who plays against the dealer. He or she has an action space consisting of only two actions: hit (draw another card) or stand (draw no other cards). Although casino blackjack typically includes options to double down, split the hand, or surrender, reducing the action space to these two options simplified the overall complexity of the model.

Our state parameterization included three essential components: the agent's current hand score, the dealer's open card, and whether he has a usable ace.

Since the agent's hand can vary between 32 values and the dealer's up card can vary between 10 different numerical values, and there are two possibilities of having or not having an ace, our state parameterization included 322 total states.

V. Results and Discussion

Test your agent(s) in the environment(s), show the results, – and most importantly – discuss and *interpret* the results. Don't just narrate what you did and observed, but discuss the implications of the results. Method A beats method B – but why? How? In which contexts?

Don't forget: negative results are also results. Your agent didn't perform as expected? If you can explain why this is just as an important contribution.

Always discuss limitations, whether observed in your results or suspected in different scenarios.

Make use of plots, e.g., Fig. 1, tables (e.g., Table I), etc; anything that illustrates the performance of your agent in the environment under different configurations. Make sure to clearly indicate the parametrization behind each result (γ, etc.).

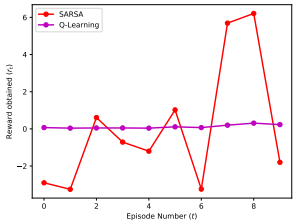

Fig. 1. Your plots should be as 'standalone'/understandable from the labels and the caption as possible: say exactly what the plot is about.

TABLE I
TABLE CAPTIONS SHOULD ADEQUATELY DESCRIBE THE CONTENTS OF TABLES (UNLIKE THIS ONE).

Environment config.	**SARSA**	**Q-Learning**
Simulation 1	10	15
Simulation 2	12	11

VI. Conclusions

Summarize the project briefly (one paragraph will do). Main outcome, lessons learned, suggestions of hypothetical future work. Reflect upon, but don't needlessly repeat, material from the conclusion.

References

[1] Stanford Wong. Professional Blackjack, Pi Yee Press.
[2] Charles de Granville. Applying Reinforcement Learning to Blackjack using Q-Learning
[3] Joshua Geiser, Tristan Hasseler. Beating Blackjack - A Reinforcement Learning Approach
[4] Clifford Mao. Reinforcement Learning with Blackjack
[5] D. Barber. Bayesian Reasoning and Machine Learning, *Cambridge University Press*, 2012.

[6] In Lecture III - Multi-Output Learning. *INF581 Advanced Machine Learning and Autonomous Agents*, 2022.
[7] In Lecture V - Reinforcement Learning II. *INF581 Advanced Machine Learning and Autonomous Agents*, 2022.
[8] D. Mena et al. A family of admissible heuristics for A* to perform inference in probabilistic classifier chains. *Machine Learning*, vol. 106, no. 1, pp 143-169, 2017.
[9] O. Vinyals et al. StarCraft II: A New Challenge for Reinforcement Learning. https://arxiv.org/abs/1708.04782, 2017.

APPENDIX

This is the place to put work that you did but is not essential to understand the paper: additional results and tables, lengthy proofs and derivations, ... Material here does not count towards page limit (but also it will be optional for the reviewer/teacher to work through).