Reinforcment Learning for Blackjack

Marius Oechslein
Faculty of Computer Science and Business Information Systems
University of Applied Sciences Würzburg-Schweinfurt
Würzburg, Germany
marius.oechslein@study.thws.de

***Abstract*—**

***Index Terms*—Blackjack, Reinforcement Learning, Monte Carlo**

I. Introduction

A. Blackjack and Reinforcement Learning

Blackjack is a popular card game played in casinos worldwide. In 1962 the mathematician Edward Thorb published a Book on how to beat the house in Blackjack [1]. This made is possible for players to win money in the game of Blackjack in casinos. Although the casinos changed their rules to prevent the player from winning in the game of blackjack impossible now, the game is still an interesting subject to mathematical modeling due to its probabilistic nature.

Edward Thorb introduced the idea of card counting with the Complete Point Count System [1] which makes it possible for players to keep track of a score while playing that indicates their chances of winning. The Complete Point Count System keeps track of a counter that adds +1 or -1 based on each card that are seen by the player. This easy counting method makes it possible for players to count cards while playing the game.

B. Reinforcement Learning for Blackjack

The game can be modeled as a Markov Decision Process which makes it possible to apply Reinforcement Learning. With todays computation power it is now possible to let the computer play over 100.000.000 games, which wasn't possible in the 1960s. The baseline goal is therefore to at least achieve the basic strategy of Edward Thorb [1] with Reinforcement Learning methods.

All of the Reinforcement Learning methods play a large number of games while keeping track of a Q-Table containing the expected returns to each of the possible game states [**?**]. The different Reinforcement Learning methods like Sarsa, Q-Learning and Temporal Difference only take a different approach of updating this Q-Table [**?**]. Although the Reinforcement Learning methods are very powerful, they require an increasing amount of games to be able to model the Q-Table as the state-space of the game increases. In this paper we therefore investigate how well the Reinforcement Learning methods Sarsa, Q-Learning and Temporal Difference Learning perform as we increase the state-space by changing the card counting method. We change the card counting method by keeping track of the values of every seen card and not just keeping track of a counter.

C. Deep Reinforcement Learning

In 2013 openAI published the paper "Playing Atari with Deep Reinforcement Learning" [2], combining Deep Learning to Reinforcement Learning. The idea of Deep Reinforcement Learning is to replace the Q-Table with a deep neural network. This makes it possible to model more complex state-spaces since the deep neural network is able to reduce dimensionality of the state while keeping the most important information [**?**].

II. Methods

A. The Blackjack Environment

1) State Spaces (Math)
2) Actions (Math) (Doubling Down, Hitting, Splitting, ...)
3) Which Actions do I permit in my implementation? and why?

B. Reinforcement Learning methods

1) Sarsa:

2) Q-Learning:

3) Temporal Difference Learning:

C. Deep Reinforcement Learning

III. Results

A. Basic Strategy

B. Increased state space - counting all cards

IV. Discussion

V. Conclusion

References

[1] Thorp, E. O. (1966). Beat the dealer: A winning strategy for the game of twenty-one (Vol. 310). Vintage.

[2] Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., & Riedmiller, M. (2013). Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602.