How to get escorted out of the casino, with Reinforcement Learning

Kieran Rudd,[1,*] Michael Gamston,[1,†] and Scott Underdown[1,‡]

[1] *School of Physics and Astronomy, University of Nottingham, Nottingham, NG7 2RD, UK*

(Dated: January 26, 2025)

Note: use plain text.
Can prob just use AI to assist in this once done.

I. INTRODUCTION

Blackjack, also known as twenty-one, is the most widely played casino game in the world, largely due to its simple game structure in which a player attempts to get the highest score by drawing cards from a deck. With its long history, various strategies for increasing a players chance of winning, such as card counting, are commonplace. Whilst an optimum strategy for blackjack has been known by statisticians for decades, by drawing on the expected value of future cards [1], training an intelligent agent to learn the optimum strategy is an approach not as often taken. Taking this approach (maths is hard for complex systems so just brute force it) by taking a 'tried and tested' approach (find ref and elaborate)

Reinforcement learning is the subfield of machine learning which aims to train an agent in an environment based on a reward, so that an optimum 'policy' can be obtained. TALK ABOUT REINFORCEMENT LEARNING PLENTY [2]

note the probabilistic nature of the problem The probabilistic nature of this topic means that Markov Decision Processes must be involved.

There are many variations to the game, but for the purposes of this project, the sequence of play was as follows: NOTE ACES RULE

1. A card is dealt to the player with value C_1.
2. For n iterations, or until a total score of 21 is exceeded, the player can make one of two choices;
   (a) Stick, and end the game.
   (b) Hit, and receive another card with value C_{n+1}.
3. The final score is calculated using

$$\text{Score} = \begin{cases} (\sum C_n)^2 & \text{if } \sum C_n \leq 21 \\ 0 & \text{if } \sum C_n > 21 \end{cases} \tag{1}$$

To approach this problem, two situations were considered; infinite, in which the pile of cards being drawn from is infinite, and so, the probability of each card being drawn is equal, and finite, in which the pile of cards being drawn from is finite, meaning unequal probabilities. Doing so allowed for assurance that the agent worked appropriately.

This paper details the methodology taken for both problem situations, the results of each in context of an optimal result, and a conclusion on the efficacy of this approach.

II. METHODOLOGY

In training an agent to play Blackjack, an iterative Q-Learning approach has been taken. Watkins' Q-Learning aims to learn the optimal 'q-value' for given state-action pairs in an environment, i.e., the respective value of making a certain move in a certain environmental state. ... further description

This approach was selected because it does not require direct

The Bellman's equations for Q-Learning is defined as,

$$Q_{new}(s,a) = Q_{old}(s,a) + \alpha(\underbrace{R(s,a) + \gamma MaxQ(s',a') - Q_{old}(s,a)}_{\text{Temporal Difference}}) \tag{2}$$

Where s, a are the current state and action, s', a' are the next state and action, α is the learning rate, γ is the discount factor, $R(s,a)$ is the reward received after taking action a in state s, and $Q(s,a)$ is the q-value for the state-action pair.

Temporal difference somewhere.

Explain Q-tables and the inner functionality of learning in conjunction with the temporal difference.

alpha and how it decreases

To train the model, Python and its basic libraries were employed, i.e., no machine learning libraries like Keras or Tensorflow.

A. Infinite

In the "infinite" situation, the probability of each card being drawn is equal, so retaining prior knowledge of cards drawn poses no advantage, i.e., this situation is purely probabilistic.

The Q-table for the infinite situation is composed of the dimensions,

* efykr2@nottingham.ac.uk
† ppxmg5@nottingham.ac.uk
‡ ppxsu1@nottingham.ac.uk

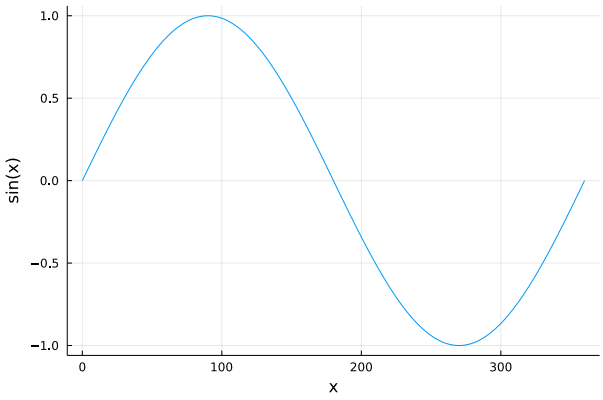

FIG. 1. Shows an example of a figure.

- Card count: 2-20
- Held ace: Y/N
- Action: Hit/Stick

B. Finite

In the 'finite' situation, cards were drawn from a pile of finite number, meaning that the probability of drawing respective cards changed as the game progressed. This posed a new challenge which could ideally be solved by providing the agent with all previously dealt cards, from which it could learn to predict the probabilities of newly dealt cards, and so, how risk-adverse it should play. However, doing so would be at great computational cost, where the Q-table would need to incorporate the dimensions previously described, in addition to some combination of previously dealt cards.

To strike a balance between accuracy and potential advantage, the probability of

... consider other methods (like card counting)

? average of recived cards???

III. RESULTS

Here, one can display figures, such as in Figure 1.

IV. CONCLUSIONS

...

[1] H. M. Roger R. Baldwin, Wilbert E. Cantey and J. P. McDermott, Journal of the American Statistical Association **51**, 429 (1956).

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction* (A Bradford Book, Cambridge, MA, USA, 2018).