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 28, 2025)

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

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. Using intelligent agents like this is known within machine learning communities as 'reinforcement learning'. The advantage of taking a reinforcement learning approach in this circumstance is rooted in its theorem proving nature, i.e., it can validate or indicate mathematical frameworks behind a system [2].

Reinforcement Learning is a subfield of machine learning concerned with teaching an 'agent' to find an optimal set of moves in the context of a wider system. The foundational components compose a policy - mapping possible states to possible actions, a reward signal - defining the agent's goal, an environment - which the agent interacts with, and a value function - indicating the long-term desirability of a sequence of states . A reinforcement learning task is considered a Markov Decision Processes (MDP) if it satisfies the Markov property, defined as the exclusive reliance of the likelihood of changing to a specific state on the present state and elapsed time, not on previous states. MDPs are used in sequential deicion making in probabilistic systems. [3]

Given a finite MDP with state s and action a, the probability of each following state and reward pair is given by

$$p(a', r|s, a) = \Pr\{S_{t+1} = s', R_{t+1} = r|S_t = s, A_t = a\}. \quad (1)$$

Which entirely specifies the dynamics of the system.

There are many variations to the game, typically involving multiple players and a dealer, 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

$$S = \begin{cases} (\sum C_n)^2 & \text{if } \sum C_n \le 21 \\ 0 & \text{if } \sum C_n > 21 \end{cases} \quad (2)$$

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 was 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

Q-learning is a model-free, value-based, off-policy algorithm. Finite Markov Decision Process

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

$$Q_{new}(s, a) = Q_{old}(s, a) + \alpha(\underbrace{R(s, a) + \gamma MaxQ(s', a') - Q_{old}(s, a)}_{\text{Temporal Difference}}) \quad (3)$$

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

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

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. State space, representing all configurations of the game.

alpha and how it decreases

Define reward

$$R(s, a) = S + \delta_{\Delta A} \tag{4}$$

To obtain the optimal policy,

$$\pi^*$$

,

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,

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

In this situation, episodes are defined as hands.

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???

find a reference that justifies using a decreasing alpha for probabilistic problems.

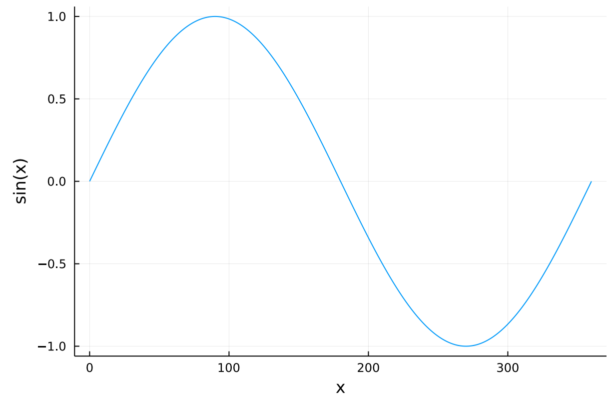

FIG. 1. Shows an example of a figure.

III. RESULTS

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

Ideas; * policy * ideal policy (needs mathematical modelling) * learning rates (and alpha rates?) : absolute and relative changes - derivatives * differences between held ace and no held ace * actual score (moving average?)

IV. CONCLUSIONS

. . . 4 pages, not including references. Allow space for abstract.

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

[2] K. A. Bidi, J.-M. Coron, A. Hayat, and N. Lichtlé, Reinforcement learning in control theory: A new approach to mathematical problem solving (2023), arXiv:2310.13072 [math.OC].

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