Q-Learning: Solutions for Grid World Problem with Forward and Backward Reward Propagrations

Antony, S., Roy, R., & Bi, Y. (2023). Q-Learning: Solutions for Grid World Problem with Forward and Backward Reward Propagrations. In *AI-2023 Forty-third SGAI International Conference on Artificial Intelligence CAMBRIDGE, ENGLAND 12-14 DECEMBER 2023*

Link to publication record in Ulster University Research Portal

Published in:
AI-2023 Forty-third SGAI International Conference on Artificial Intelligence CAMBRIDGE, ENGLAND 12-14 DECEMBER 2023

Publication Status:
Published (in print/issue): 01/01/2023

Document Version
Author Accepted version

Document Licence:
Unspecified

General rights
The copyright and moral rights to the output are retained by the output author(s), unless otherwise stated by the document licence.

Unless otherwise stated, users are permitted to download a copy of the output for personal study or non-commercial research and are permitted to freely distribute the URL of the output. They are not permitted to alter, reproduce, distribute or make any commercial use of the output without obtaining the permission of the author(s).

If the document is licenced under Creative Commons, the rights of users of the documents can be found at https://creativecommons.org/share-your-work/cclicenses/.

Take down policy
The Research Portal is Ulster University's institutional repository that provides access to Ulster's research outputs. Every effort has been made to ensure that content in the Research Portal does not infringe any person's rights, or applicable UK laws. If you discover content in the Research Portal that you believe breaches copyright or violates any law, please contact pure-support@ulster.ac.uk

Download date: 03/10/2024

Q-Learning: Solutions for Grid World Problem with Forward and Backward Reward Propagrations

Snobin Antony, Raghi Roy and Yaxin Bi

School of Computing, Ulster University, Belfast, Northern Ireland
{antony-s1, roy-r1,y.bi}@ulster.ac.uk

Abstract. In the area of adaptive and responsive problems, Reinforcement Learning algorithms have made significant progress. This paper presents solutions to the grid world problem using the model-free reinforcement learning method known as the Q-Learning algorithm. The solutions are developed with forward and backward reward propagations under the assumption of a Markov decision process for the grid world problem. This study detail the implementation and comparison of these two reward calculations. The paper also illustrates how an agent interact with the gird environment during both forward and backward propagations and compare their benefits followed by hyperparameter tuning for better understanding of the model's convergence.

Keywords: Reinforcement learning, q-learning, Markov processes, rewards, grid world.

5 Introduction

Machine learning includes supervised, unsupervised, and reinforcement learning, with the latter involving trial and error to discover optimal policies for sequential decision-making. The Markov Decision Process (MDP) is a reframing of Markov chains as shown in Fig. 1, which uses the decision-making process in the stochastic setting. A Markov Decision Process (MDP) is typically used solve the challenges in Reinforcement Learning (RL). MDP aims to offer a mapping of the best course of action for each environmental state via only considering the present and ignoring past knowledge, future state prediction is totally independent from the state in the previous step [7].

The MDP can be further formulated by a 4-tuple (S, A, T, R), where target state $s \in S$ encodes the target's status, action $a \in A$ encrypts actions that can be taken against a target. State transition function $T: S \times A \longrightarrow S$ defines each action's impact on each state. Reward function is defined as R: S × A → R, $R: S \times A \longrightarrow \mathbb{R}$ describes the reward received immediately after performing action a in state s [10].

2.2 Markov Reward Process

The memoryless random process known as the Markov Process comprises of a series of random states $S_1, S_2, \ldots, S_n$, having a Markov Property which is defined with a set of states S and a transition probability matrix P, they define the dynamics of an environment in its totality. The Markov Reward Process (MRP) will result from adding rewards onto the Markov Chain. Rewards are the monetary sums that the agent obtains for taking a certain action a at a certain environmental state s. Depending on the agent's behaviour, the numerical value may be either positive or negative. The Markov chain including value judgement is known as a Markov reward process. In general, an agent will receive a value for each state [8]. The MRP can be expressed mathematically as follows:

$$R_s = \mathbb{E}[R_{t+1} \mid S_t] \quad (1)$$

The agent receives the reward R_s from a certain state S_t is expressed in Equation (3). This indicates the agent's immediate reward for that specific state.

2.4 Exploration and Exploitation

The trade-off among exploration and exploitation is an issue frequently occurring in reinforcement learning [5]. An agent must choose activities that it has proven to be successful in maximising the return of the expectations or cumulative value. The agent must take actions that have not been chosen previously to find such actions and take advantage of what it learns to obtain reward, yet additionally explore to choose its future actions more wisely. The problem is that if a winning strategy was created, neither exploitation nor exploration could be pursued solely. In a volatile environment, the presently exploited solution might no longer be relevant and previously investigated options might have altered in value, which becomes even more difficult [11]. Exploration is a long-term benefit idea because it enables the agent to increase its understanding of each action that may have long-term benefits. Exploitation, on the other hand, essentially takes the advantage of the agent existing estimated value and chose a greedy strategy to gain the greatest rewards.

It's possible that the agent won't receive the highest reward since it is being greedy mostly with estimated value rather than the actual value [3]. There are a few different ways to choose what action to be taken. The most straightforward one is greedy selection, where the agent always chooses the action with the higher state-action value. Pure exploitation constitutes this strategy. We can apply a more complex Epsilon ϵ Greedy Policy to solve the Exploration-Exploitation Dilemma. The agent determines the appropriate course of action randomly based on a tiny possibility ϵ, rather than taking the best option [3].

3 Methodology

In a grid world environment, an agent learns to accomplish a mission through movements and several unique tasks. Fig. 2 depicts a configuration of the grid world.

As illustrated in Fig. 2, the environment consists of 5x5 grids, the agent is initially positioned on the (2,1) start cell at the beginning of episode, and the terminal position is on the cell (5, 5). The agent state inputs u_t at each step, and produces an output vectors of action values, $z_t = [Q1_t, Q2_t, Q3_t, Q4_t]$ whose elements are corresponding to" four actions: "go north", "go east", "go south" and "go west", respectively [4]. In addition, there is a special action "jumps" from (2,4) to (4,4) with a +5 bonus reward. The agent will receive a +10 reward when it reaches the end location (5,5) or the win state. Any movements over other grids, the agent will receive a -1 penalty. The agent will move over the grid board and encounter some barriers in the environment. We can apply a Q-learning approach to solve this grid world problem.

3.1 Q-Learning

Q-learning is a type of reinforcement learning algorithms. The seminal work of Q-learning algorithms was developed by Witkins in [15]. The Q-value represents the expected cumulative reward (see Equation 2) the agent will receive by interacting an environment in given states, which can be updated by Equation (6) below:

$$Q_{t+1}(s,a) \leftarrow Q_t(s,a) + \alpha[r + \gamma max_a Q_t(s',a) - Q_t(s,a)] \qquad (6)$$

For any actions to be taken, the agent will receive either a reward or penalty, then accumulate it together using Equation (6), and generate an output $z_t = [Q1_t, Q2_t, Q3_t, Q4_t]$.
To solve the grid world problem using Q-learning, an agent takes actions, receives rewards, and updates Q-values. Two reward accumulation strategies affect convergence time in finding an optimal path.

3.2 Deterministic and Non-Deterministic

Deterministic algorithms yield consistent outcomes, while non-deterministic one's display variability across executions.

Balancing exploration and exploitation are essential in Q-learning, involving the choice of actions in each state to identify the optimal policy [6]. This choice pertains to either selecting the current best action or trying a different one in the expectation of greater future rewards.

We employed the most popular exploration technique "ϵ-greedy" to aid an agent using non-deterministic mode in discovering goals in a grid [9]. It uses $0 \leq \epsilon \leq 1$ as an exploration parameter to determine which action to take $Q_t(s_t, a)$. In the current condition, the agent with probability $1 - \epsilon$ chooses the action with the greatest Q-value. In all other state, the agent selects a random action. A higher value for ϵ indicates that the agent chooses more exploration actions. For an agent moving throughout a stochastic grid to discover the best course of action, randomness is required [9].

In this implementation, the actions are chosen using the stochastic method. The action based on the epsilon greedy value (ϵ), or a random action is returned by the action selection function in the loop.

Epsilon-greedy is used more with lower epsilon values, favouring random action selection otherwise. The best action is chosen based on maximum next reward. The position update function is used when taking actions, and, as the "determine" flag is set to False for non-deterministic behaviour, actions are selected with probabilities using epsilon-greedy. Higher probabilities allow for the same action but also encourage exploration of alternative actions.

4 Implementation

Both deterministic and non-deterministic methods can be used to solve the grid world problem. In this case, we used the Bellman's equation and the epsilon-greedy exploration strategy to construct the non-deterministic method. The most significant role in handling MDP and RL problems is performed by Bellman's famous equation, which was formulated in 1953. Utility function can be calculated using the Bellman's equation. It's a kind of recursion for expected rewards in the future [7].

$$\mathrm{U(s)} = \mathrm{R(s)} + \gamma \max_a \sum \mathrm{T}(\mathrm{s}, \mathrm{a}, \mathrm{s}')\mathrm{U}(\mathrm{s}') \quad (7)$$

The Bellman equation explains that the sum of the agent's benefit for reaching the current state s and the maximum possible reward for state s' equals the maximum future reward. The basic idea behind Q-learning is that by applying the abovementioned Bellman equation, we can iteratively estimate Q^*. The Q-learning formula is given by:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)) \quad (8)$$

where learning rate α is determines how much of the difference between the old and new Q-values are considered [1].

4.1 Algorithm and Explanation

The implemented algorithm in this study possesses specific characteristics. It operates with four defined actions: South, North, West, and East movements. Q-values are stored in a dictionary of dictionaries, initially set to '-1'. The learning rate, a parameter ranging from 0 to 1, governs the speed of the agent's learning process, with relatively high values accelerating initial learning [2]. The algorithm operates non-deterministically, employing the epsilon-greedy strategy for action selection. The training phase continues iteratively until either the maximum episode limit is reached, or a predefined stopping condition is met, with 1000 episodes used for learning. During this process, actions are chosen using either the epsilon-greedy

strategy or random selection, and the best action is determined based on the maximum anticipated next reward. The reward calculation process differs between backward and forward propagation methods. In back propagation, rewards and next Q-values are computed only upon reaching the episode's destination, while in forward propagation, these calculations occur after every action. A special jump path exists, offering a +5 bonus and a +10 win reward when taken to reach the win state. Equation 8 is employed to calculate the next Q-value. In back propagation, actions are chosen and executed based on a greedy approach, calculating rewards and next Q-values only when reaching the episode's destination. In contrast, forward propagation calculates rewards after each action. Cumulative rewards are determined by summing all rewards within a given episode, and average cumulative rewards are computed by dividing the cumulative reward by the total number of action states in that episode. To conduct a comparative analysis, this study also implemented the State Action Reward State Action (SARSA) for the forward propagation. SARSA shows some similarities to Q-Learning. Q-Learning uses off-policy technique and a greedy approach to learn Q-values. But SARSA uses the on-policy techniques and the value of the action performed using the current policy to learn the Q-Values.

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha(r_{t+1} + \gamma\, Q_t(s_{t+1}, a_{t+1}) - Q_t(s_t, a_t)) \quad (9)$$

Since the Temporal Difference method is implemented in SARSA, it keeps updating the Q-table after every step till it reaches the maximum iteration or converges to an optimal solution [17].

5 Result and Evaluation

To find the optimum path based on the reward function, the Q-learning stochastic method was examined in this study. Using the non-deterministic Q-learning method, the agent achieves the win state in a variety of ways, but eventually determines the most effective route. In the stochastic method, agents utilise exploration technique with an epsilon greedy approach and make decisions based on the long-term rewards or policy method, as opposed to the deterministic Q-learning algorithm where agents only choose those actions that offer larger immediate rewards. This algorithm enables the agent to investigate various potential paths and discover the best route in a smaller number of episodes.

5.1 Comparison between the Forward and Backward Propagation

This section will provide comparative discussions about both forward and backward propagation, and a comparison with SARSA later.

Fig. 3.a shows the state values of each grid cell, which are the highest Q-values

an agent may attain in a specific grid cell if it chooses one of the four actions. The board layout and state data for each grid cell are generated for the backward propagation. The agent's travel path or converging direction to the end state is assumed to be represented by the maximum Q-value of a given state. It is clear from the figure that the agent travelled across every grid cell and discovered the best route, which is the incremental q-values starting from the start cell (1,0). The agent uses several actions and a jump in the optimum path to get the win state with the fewest number of movements.

Fig. 3.b shows the board layout and state data for each grid cell generated for the forward propagation. But in this method agent receives a maximum of 12 reward and it has travelled through different path even if it is finding the optimum path because of the algorithm did not converge within 1000 episodes.

Fig. 3.c depicts the board layout and state data for each grid cell generated using the forward propagation for SARSA. The key difference here is the how they update the Q-values during the learning process. In SARSA, the Q-values are updated based on the observed state-action-reward-state-action transitions. The important characteristic of SARSA is that it uses the current policy to select the next action and estimates the Q-value accordingly. In SARSA, the update considers the Q-value of the next state-action pair (s_{t+1}, a_{t+1}) based on the action actually taken using the current policy. The update depends on the learning rate α, the observed reward r_{t+1}, the discount factor γ, and the current and next Q-values.

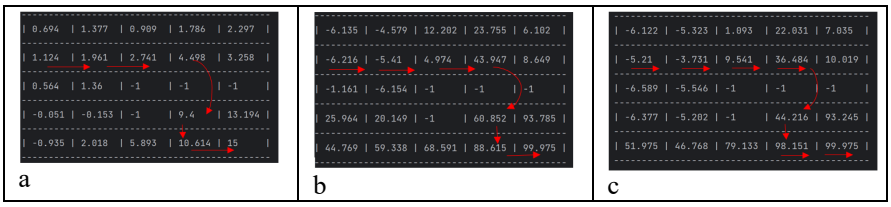

Fig. 3. Board layout of grid world state values (a. Backward, b. Forward, c. SARSA)

5.2 Performance comparison

Overall, forward propagation updates rewards with Q-values more frequently, considering rewards after each action. On the other hand, backward propagation updates rewards with Q-values only when the agent successfully reaches the win state per episode. However, SARSA specifically employs an on-policy learning approach and uses a different update equation to calculate the next Q-value. The choice between these approaches depends on the specific requirements and dynamics of the grid world problem being addressed.

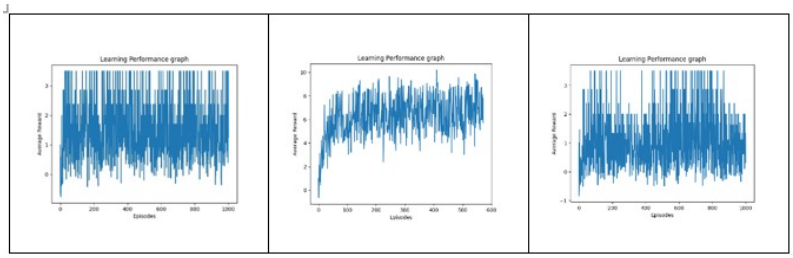

Fig 6. Performance graph of Forward Propagation, Backward Propagation, and SARSA

6 Conclusion

This study develops a Q-learning solution for the grid world problem. It demonstrates that the agent is taking the actions and predicting what it might do next based on its experience using the non-deterministic Q-learning technique. A grid board layout and a Q-table are used to show the agent's performance. The agent is required to travel from the beginning to the end while earning rewards. The environment contains a special jump action that offers a bonus as well as certain obstacles that could result in penalties. By accumulating the most reward points, including the jump rewards, the agent identified the most effective route to the win state.

To the extent possible, we have presented the bellman's equation and exploration technique based on the back propagation approach for the stochastic q-learning algorithm. The epsilon greedy exploration method never stops looking for new paths and keeps exploring even after it finds the best one. Hence, the convergence term average cumulative reward fluctuated even if it reaches the maximum episode. In an order to move ahead, we have analysed how exploration and exploitation strategies perform in both backward and forward propagation approaches. Backward propagation algorithm reaches a converged path with a high cumulative average reward compared with the forward propagation algorithm does.

References

1. Campbell JS, Givigi SN, Schwartz HM (2014) Multiple-model Q-learning for stochastic reinforcement delays. In: Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics. Institute of Electrical and Electronics Engineers Inc., pp 1611–1617
2. Chen Y, Schomaker L, Wiering M (2021) An investigation into the effect of the learning rate on overestimation bias of connectionist q-learning. In: ICAART 2021 - Proceedings of the 13th International Conference on Agents and Artificial Intelligence. SciTePress, pp 107–118
3. Coggan M Exploration and Exploitation in Reinforcement Learning
4. IEEE Computational Intelligence Society, International Neural Network

Society, Institute of Electrical and Electronics Engineers, IEEE World Congress on Computational Intelligence (2020 : Online) 2020 International Joint Conference on Neural Networks (IJCNN) : 2020 conference proceedings.

5. Macready WG, Wolpert DH (1998) Bandit problems and the exploration/exploitation tradeoff. IEEE Transactions on Evolutionary Computation 2:2–22. doi: 10.1109/4235.728210
6. Manju MS, Punithavalli MM An Analysis of Q-Learning Algorithms with Strategies of Reward Function
7. Naeem M, Rizvi STH, Coronato A (2020) A Gentle Introduction to Reinforcement Learning and its Application in Different Fields. IEEE Access 8:209320–209344. doi: 10.1109/ACCESS.2020.3038605
8. Sutton RS, Barto AG Reinforcement Learning: An Introduction Second edition, in progress
9. Tijsma AD, Drugan MM, Wiering MA (2017) Comparing exploration strategies for Q-learning in random stochastic mazes. In: 2016 IEEE Symposium Series on Computational Intelligence, SSCI 2016. Institute of Electrical and Electronics Engineers Inc.
10. Yang XH, Yin F, Liu CL (2018) Online video text detection with markov decision process. In: Proceedings - 13th IAPR International Workshop on Document Analysis Systems, DAS 2018. Institute of Electrical and Electronics Engineers Inc., pp 103–108
11. Yen G, Yang F, Hickey T, Goldstein M (2001) Coordination of exploration and exploitation in a dynamic environment. In: Proceedings of the International Joint Conference on Neural Networks. pp 1014–1018