Comparison of Q-learning and SARSA Reinforcement Learning Models on Cliff Walking Problem

Lv Zhong

Brunel London School, North China University of Technology, Beijing, 100144, China
21190010330@mail.ncut.edu.cn

Abstract. In this work, the performance of two value-based reinforcement learning algorithms is evaluated in the cliff walking problem, including State-Action-Reward-State-Action (SARSA) and Q-learning. This paper uses Python language and Numpy library to implement SARSA and Q-learning algorithms, and compares and analyzes their policy graphs and reward curves. The experimental results show that SARSA is a conservative algorithm, which tends to choose a path away from the cliff, thus reducing the risk but also increasing the steps and time; Q-learning is a greedy algorithm, which tends to choose a path close to the cliff, thus increasing the reward but also increasing the fluctuation and instability. This paper discusses the balance between exploration and exploitation of these two algorithms, as well as their performance under different parameter settings, as well as their adaptability and generalization ability in complex environments. This paper also points out some shortcomings and prospects, such as only using a simple grid world as the experimental environment, without considering more complex or realistic environments; only using two value-based reinforcement learning algorithms, without considering other types of reinforcement learning algorithms; only using one exploration policy, namely ε-greedy policy, without considering other types of exploration policies. This paper provides some valuable contributions and innovations for the reinforcement learning field.

Keywords: SARSA, Q-learning, Reinforcement learning.

1 Introduction

Reinforcement learning is broadly leveraged for exploring how an agent can optimize its action strategies by interacting with the environment [1]. It can be applied to various complex decision and control problems, such as robotics, autonomous driving, games, and finance [2, 3]. Reinforcement learning could be roughly categorized as branches: value-based and policy-based solutions. The former branch aims at forecasting the value function for each state-action pair or state, and guide the agent to select the optimal action. The value function is the expected return of following a specific policy from a certain state or state-action pair. The latter branch denotes those that directly learn the policy function that links each state to an action or its corresponding probability distribution of actions [4].

B. H. Ahmad (ed.), *Proceedings of the 2023 International Conference on Data Science, Advanced Algorithm and Intelligent Computing (DAI 2023)*, Advances in Intelligent Systems Research 180,
https://doi.org/10.2991/978-94-6463-370-2_23

In this project, two value-based reinforcement learning algorithms are compared, including State-Action-Reward-State-Action (SARSA) and Q-learning [5, 6]. They are both solutions that use an epsilon-greedy policy for balancing exploitation and exploration. However, there are also differences, especially on their strategies for updating the value function. SARSA is an on-policy algorithm, which leveraged the current policy to determine the next action, and update the value function according to the actual action taken. Q-learning belongs to a kind of off-policy algorithms, which leverages the optimal policy to select the next action, and updates the value function based on the next state's maximum value function.

In this work, cliff walking problem is leveraged as the evaluation benchmark, as it is a classic reinforcement learning problem that can illustrate the different characteristics and performance of SARSA and Q-learning. The cliff walking problem is a grid world, where the agent has to move from a start state to a goal state, while avoiding falling off a cliff along the edge of the grid. The agent receives different rewards depending on its actions and states [7].

The main objectives of this project are: To compare the exploration and exploitation behaviors of SARSA and Q-learning in this problem. To compare the convergence speed and quality of them for cliff walking problem. To analyze the effects of different parameters, such as learning rate, discount factor, exploration rate, etc., on the learning effect of SARSA and Q-learning in the cliff walking problem.

2 Method

The methodology of this project consists of four main steps: setting up the environment and the task, implementing the algorithms and the parameters, evaluating the metrics and plotting the results, and analyzing and discussing the results.

The environment and the task are the cliff walking problem, which is a classic reinforcement learning problem that can be used to compare different algorithms in terms of exploration and exploitation. The cliff walking problem is a grid world, where the agent has to move from a start state to a target state, while avoiding falling off a cliff along the edge of the grid. The agent receives different rewards depending on its actions and states. The environment and the task are implemented using OpenAI Gym, which includes several environments that is beneficial for validating the reinforcement learning algorithms [8].

SARSA and Q-learning are two reinforcement learning algorithms that use temporal difference learning for updating the value function of each state-action pair. Moreover, for balancing the exploitation and exploration, epsilon-greedy policy is leveraged. The parameters are learning rate, discount factor, and exploration rate, which are three important factors that influence the performance of the algorithms. The algorithms and the parameters are implemented using Python language and Numpy library, which are popular tools for scientific computing and data analysis.

The metrics are policy map, reward curve, and parameter analysis, which are used to compare and analyze the performance of SARSA and Q-learning in terms of exploration and exploitation [9]. The metrics are evaluated and plotted using Python language and Numpy library as well [10].

The results are analyzed and discussed based on the policy maps, reward curves, and parameter analysis. The results reveal the characteristics and differences of SARSA and Q-learning in the cliff walking problem, as well as their strengths and weaknesses. The results also provide insights for future research and development of reinforcement learning algorithms.

3 Result

The results show that SARSA algorithm and Q-learning models perform differently in the cliff walking problem. The optimal policy graphs are printed for both SARSA and Q-learning algorithms, based on the cliff walking environment, as demonstrated in Fig 1 and Fig 2 respectively. The graphs show the arrows indicating the optimal action to choose at each state, according to the learned Q values. From the graphs, it could be observed that SARSA and Q-learning have different optimal policies. SARSA learns a safer but longer path, avoiding the edge of the cliff. Q-learning learns a riskier but shorter path, staying close to the edge of the cliff. This is because SARSA belongs to one of the on-policy algorithm, which considers the exploration factor when learning the Q values. Q-learning is off-policy. It ignores the exploration factor and only maximizes the Q values. This result is consistent with the theoretical analysis of SARSA and Q-learning. SARSA converges to an epsilon-greedy optimal policy, which balances exploration and exploitation. Q-learning converges to a greedy optimal policy, which exploits the maximum expected reward. Therefore, SARSA tends to be more cautious and conservative, while Q-learning tends to be more aggressive and optimistic.

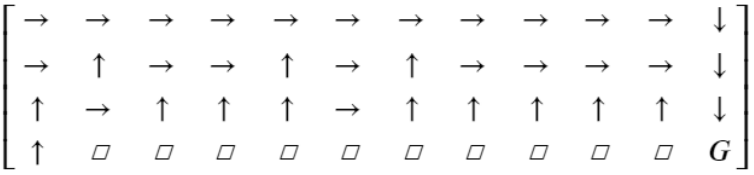

Fig. 1. Policy map of SARSA (Figure credit: Original).

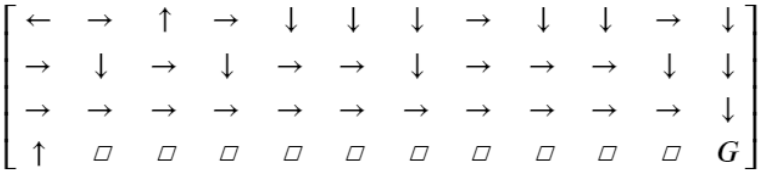

Fig. 2. Policy map of Q-learning (Figure credit: Original).

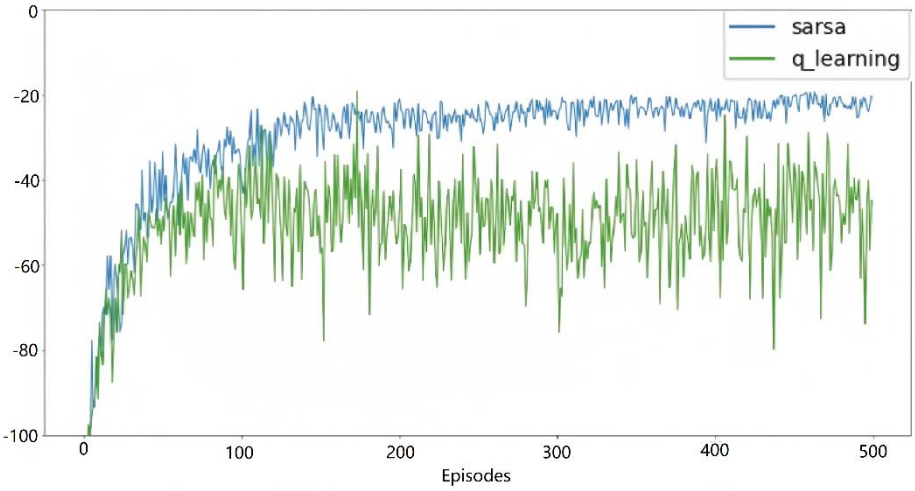

Fig. 3. Curves of return values (Figure credit: Original).

This result shown in Fig 3 also illustrates the balance between exploration and exploitation. Exploration can help the agent discover new states and actions, but it can also lead to suboptimal or dangerous choices. Exploitation can help the agent optimize its performance, but it can also lead to overfitting or missing opportunities. Depending on the environment and the goal, different levels of exploration and exploitation may be appropriate. SARSA algorithm is a conservative algorithm, which considers the long-term impact of each action, thus avoiding high-risk actions. This makes SARSA algorithm choose a path away from the cliff in the cliff walking problem, which reduces the risk of falling off the cliff, but also increases steps and time to reach the end point. The reward curve of SARSA algorithm shows a smooth and low convergence value, indicating that SARSA algorithm can stably learn a good policy, but it may also miss some better policies. Q-learning algorithm is a greedy algorithm, which only focuses on the maximum immediate reward of each action, thus choosing high-reward but also high-risk actions. This makes Q-learning algorithm choose a path close to the cliff in the cliff walking problem, which improves steps and time to reach the end point, but also increases the risk of falling off the cliff. The reward curve of Q-learning algorithm shows a fluctuating and high convergence value, indicating that Q-learning algorithm can quickly learn an optimal policy, but it may also be affected by noise and delay, resulting in an unstable learning process. The experimental results of this paper also show that SARSA algorithm and Q-learning algorithm have different performances under different parameter settings. Learning rate, discount factor and exploration rate all affect the learning speed and effect of the algorithm. Generally speaking, higher learning rate and discount factor make the algorithm adapt to environmental changes faster, but they may also cause overfitting or divergence; lower learning rate and discount factor make the algorithm converge more stably, but they may also cause underfitting or local optimum. Higher exploration rate makes the algorithm try more new actions, thus increasing the possibility of finding better policy.

To balance exploitation and exploration, the ε-greedy policy is leveraged, which means Q-learning has a certain probability of choosing random actions instead of the ones that correspond to the maximum Q value. This way, Q-learning might choose a wrong action near the edge of the cliff, causing it to fall and get a reward of -100. Therefore, the reward for Q-learning will be lower and more fluctuating than that for SARSA. The ε-greedy policy is a straightforward and highly effective way for balancing the exploitation and exploration by adjusting the value of ε. Generally, a larger ε means a higher exploration rate; a smaller ε means a higher exploitation rate. This is implemented by a function which selects chooses an action based on the current state. Meanwhile, according to the Q table. A constant ε is defined, which represents the exploration rate, or the probability of choosing a random action. In this function, an if-else statement is leveraged to choose a random action if the generated random number is larger than ε, or if all the actions for the current state have zero value; otherwise, this work selects the action with the maximum value for the current state according to Q table. In this way, this work achieves the ε-greedy policy, which means a random action is selected with a certain probability and an optimal action with another probability.

4 Discussion

This paper studies the performance and effect of two value-based reinforcement learning algorithms: SARSA and Q-learning in the cliff walking problem. This paper uses Python language and Numpy library to implement SARSA and Q-learning algorithms, and compares and analyzes their policy graphs and reward curves. The experimental results show that SARSA is a conservative algorithm, which tends to choose a path away from the cliff, thus reducing the risk but also increasing the steps and time; Q-learning is a greedy algorithm, which tends to choose a path close to the cliff, thus increasing the reward but also increasing the fluctuation and instability. This paper discusses the balance between exploration and exploitation of these two algorithms, as well as their performance under different parameter settings. This paper also discusses the adaptability and generalization ability of SARSA algorithm and Q-learning algorithm in complex environments, as well as their advantages, disadvantages, and limitations.

The main contributions and innovations of this paper are: using Python language and Numpy library to implement SARSA and Q-learning algorithms, and conducting experiments in the cliff walking problem. Comparing and analyzing the policy graphs and reward curves of SARSA and Q-learning algorithms, revealing their differences and characteristics in exploration and exploitation. Discussing the performance of SARSA and Q-learning algorithms under different parameter settings, as well as their adaptability and generalization ability in complex environments.

This paper also has some shortcomings, such as: this paper only uses a simple grid world as the experimental environment, without considering more complex or realistic environments, such as with obstacles, noise, multiple agents, etc. This paper only uses two value-based reinforcement learning algorithms, without considering other types of

reinforcement learning algorithms, such as policy-based methods, model-based methods, deep reinforcement learning methods, etc. This paper only uses one exploration policy, namely ε-greedy policy, without considering other types of exploration policies, such as softmax policy, upper confidence bound (UCB) policy, information-theoretic policy, etc.

Future work can be extended from the following aspects: using more complex or realistic environments to test the performance and effect of SARSA algorithm and Q-learning algorithm, such as maze navigation, autonomous driving, robot control, etc. Using other types of reinforcement learning algorithms to compare and analyze with SARSA algorithm and Q-learning algorithm, such as policy-based methods, model-based methods, deep reinforcement learning methods, etc. Using other types of exploration policies to adjust the balance between exploration and exploitation of SARSA algorithm and Q-learning algorithm, such as softmax policy, upper confidence bound policy, information-theoretic policy, etc.

5 Conclusion

This paper reveals the different characteristics and differences of SARSA algorithm and Q-learning algorithm in exploration and exploitation by comparing and analyzing them in the cliff walking problem. This paper finds that SARSA algorithm is a conservative algorithm, which considers the long-term impact of each action, thus avoiding high-risk actions; Q-learning algorithm is a greedy algorithm, which only focuses on the maximum immediate reward of each action, thus choosing high-reward but also high-risk actions. This paper discusses the trade-off between exploration and exploitation of these two algorithms, as well as their performance under different parameter settings. This paper also discusses the adaptability and generalization ability of these two algorithms in complex environments, as well as their advantages, disadvantages, and limitations. This paper provides some valuable contributions and innovations for the reinforcement learning field, such as using Python language and Numpy library to implement SARSA and Q-learning algorithms, and conducting experiments in the cliff walking problem; using policy graphs and reward curves to evaluate and display the performance of SARSA and Q-learning algorithms; using different parameter settings to analyze the influencing factors of SARSA and Q-learning algorithms. This paper also points out some future work directions, such as using more complex or realistic environments to test SARSA algorithm and Q-learning algorithm; using other types of reinforcement learning algorithms to compare and analyze with SARSA algorithm and Q-learning algorithm; using other types of exploration policies to adjust the balance between exploration and exploitation of SARSA algorithm and Q-learning algorithm.

References

1. Levine, S., Kumar, A., Tucker, G., & Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. arXiv preprint arXiv:2005.01643 (2020).

2. Canese, L., Cardarilli, G. C., Di Nunzio, L., Fazzolari, R., Giardino, D., Re, M., & Spanò, S. Multi-agent reinforcement learning: A review of challenges and applications. Applied Sciences, 11(11), 4948 (2021).
3. Qiang, W., & Zhongli, Z. Reinforcement learning model, algorithms and its application. In 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC), 1143-1146 (2011).
4. Zhang, H., & Yu, T. Taxonomy of reinforcement learning algorithms. Deep Reinforcement Learning: Fundamentals, Research and Applications, 125-133 (2020).
5. Clifton, J., & Laber, E. (2020). Q-learning: Theory and applications. Annual Review of Statistics and Its Application, 7, 279-301.
6. Zhao, D., Wang, H., Shao, K., & Zhu, Y. Deep reinforcement learning with experience replay based on SARSA. In 2016 IEEE symposium series on computational intelligence (SSCI) 1-6 (2016).
7. Jiang, Z., & Luo, S. Neural logic reinforcement learning. In International conference on machine learning, 3110-3119 (2019).
8. OpenAI Gym. URL: https://github.com/openai/gym. Last accessed 2023/08/24
9. Alharin, A., Doan, T. N., & Sartipi, M. Reinforcement learning interpretation methods: A survey. IEEE Access, 8, 171058-171077 (2020).
10. Home page of NumPy. URL: https://numpy.org/. Last accessed 2023/08/24.