International Journal of Grid and Distributed Computing
Vol. 9, No. 3 (2016), pp.243-250
http://dx.doi.org/10.14257/ijgdc.2016.9.3.25

Exploration on Obstacle Avoidance and Study of Balance

WANG Qi-ming, Liu Jian-fen and Shi He-sheng

College of Information Engineering, Pingdingshan University, Pingdingshan Henan 467002, China
wqm8157@126.com

Abstract

This paper studies ε-greedy algorithm and softmax algorithm in obstacle avoidance and balance study. In the experiment, Sarsa algorithm and Q-Learning algorithm were used to appropriately simplify and build the model of obstacle avoidance; softmax algorithm was used to address how to balance exploration and utilisation; and two classical algorithms of reinforcement learning were adopted to deal with obstacle avoidance. The results generated by simulation prove that Sarsa algorithm and Q-Learning algorithm can handle obstacle avoidance and balance study in limited time step, which makes the intelligent agent improve the non-maximum estimated value of the value function of the state so as to choose the best action that has been carried out. In addition, Sarsa algorithm and Q-Learning algorithm can also enable the intelligent agent to try new actions and find out the optimal one.

***Keywords**: Obstacle avoidance, Exploration, Balance, Reinforcement learning*

1. Introduction

Learning ability of the intelligent agent is defined as its ability to interact with the unknown environment and acquire information and knowledge in order to adapt to the environment. This issue belongs to machine learning field in which guided learning and reinforcement learning [1] are widely used. Guided learning means the intelligent agent learns from the various and abundant samples provided by the external indicator. However, in this module, the intelligent agent cannot interact with the unknown environment [2] smoothly. By contrast, reinforcement learning enables the intelligent agent to learn from its own experiences through interacting with the environment. Since 1980s, researchers have taken different attitudes towards guided learning and reinforcement learning, and later perfectly combined temporal difference learning and optimal control, which is a breakthrough of reinforcement learning theory. With further studies and application, reinforcement learning has become an important branch in control technology study [3].

Obstacle avoidance of the intelligent agent is related to path planning of the moving robot, aircraft control, developing strategy in robot soccer, and cooperative control of multi-agent systems. In obstacle avoidance, it is expected that the intelligent agent should move as little as possible and reinforcement learning is one of the best ways to find out the optimal strategy [4]. This paper studies obstacle avoidance based on reinforcement learning. First, it will introduce the theory framework of reinforcement learning and two classical algorithms: Sarsa and Q-Learning algorithm. Then, obstacle avoidance is appropriately simplified and the two classical algorithms of reinforcement learning will be used to address it. Finally, this paper analyses simulation results generated by matlab.

2. Reinforcement Learning

2.1. Theory Framework of Reinforcement Learning

As the control technology, reinforcement learning means that the intelligent agent studies how to interact with the environment in order to maximise the value function that represents the long-term goal. The interaction process is illustrated below in Figure 1. The intelligent agent learns a state in the environment and according to the transfer of the last state gets a reward. And then using the strategy that it has learned, the intelligent agent adjusts its action according to the environment and influences it. Correspondingly, the transfer of the state happens in the environment and then the intelligent learns a new state, forming a circle. This indicates that if the intelligent agent receives a positive reward from the environment for certain action, it will be likely to choose this action again in the future; on the contrary, it will not [5]. Generally, Markov decision process, MDP can build the model of issues that need to be solved through reinforcement learning, and I will not explain it further here.

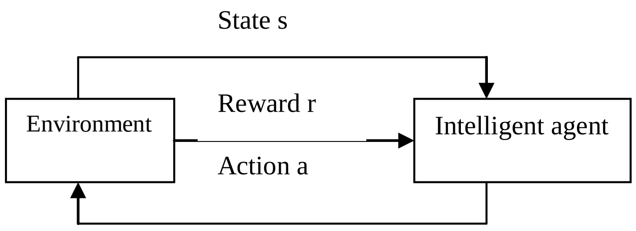

Figure 1. Reinforcement Learning Framework

2.2. Key Elements of Reinforcement Learning

Despite the intelligent agent and environment, reinforcement learning also includes the following four elements: strategy, reward, value function and unnecessary environment model. Strategy is defined as the mapping of the state in the environment on the action that can be carried out in this state. Reward is defined as the instant feedback, representing the intelligent agent’s goal at the moment. Value function is defined as the level of expectation that the intelligent agent gains by accumulating rewards, which can be regarded as the long-term reward and serve as the standard to evaluate the action the intelligent agent chooses in the long run. In terms of environment model, it gives the detailed probability distribution in the transfer of the state. For instance, given a state and an action, this model can predict the next state and reward. However, many algorithms in reinforcement learning do not need environment model thus it is unnecessary [6].

2.3. Exploration and Utilization

A key issue in studying reinforcement learning is how to balance exploration and utilisation. In the value function of a continuous estimated state (or in a state-action pair), at least there is a maximum estimated value of the value function under one state in any time step. If the maximum estimated value is selected, the action will be greedy action. If greedy action is chosen, the value function of the state in the time step will be utilised. If non-greedy action is chosen, it is called exploration, which allows the intelligent agent to improve the non-maximum estimated value of the state. To sum up, utilisation means that in order to get more instant rewards, the intelligent agent carries out the best action that has been taken before [7]. While exploration means that in order to get better actions, the intelligent agent tries new actions. Therefore, in the whole task, the intelligent agent needs to use exploration and utilisation alternately to obtain the optimal strategy.

Two ways to balance exploration and utilisation are ε-greedy algorithm and softmax algorithm. In terms of ε-greedy algorithm, the probability to choose non-greedy action in the next state is ε ($0<\varepsilon<1$) while choosing greedy action is $1-\varepsilon$. ε-greedy algorithm is easy to carry out but its defect is that the probabilities to choose all non-greedy actions are the same. That is, when the worst action has serious negative impacts on the whole environment, the algorithm convergence is not effective. By contrast, according to the estimated value of the value function of different actions, Softmax algorithm can generate different probabilities. The formula of the probability is as the following:

$$\frac{e^{Q_t(s,a_i)/\tau}}{\sum_{j=1}^{n} e^{Q_t(s,a_j)/\tau}}$$

(1)

τ is a positive parameter. In the formula, $Q_t(s,a_i)$, i=1…n, means in time t, the estimated value of the value function with the action a_i in state S (the estimated value of the value function of the state-action pair) [7]. In this research, Softmax algorithm was adopted.

2.4. Sarsa Algorithm

In 2.1, the framework of reinforcement learning has already been introduced, but it must be applied through specific algorithms. Thus, two classical algorithms are explained: one is strategic temporal difference control Sarsa algorithm, and another is temporal difference control independent from strategy, Q-Learning algorithm [8]. In order to estimate the value function, the strategy of being evaluated must be implemented for many times, and the process from the initial state to the final state is called an episode, which contains numbers of alternate states and state-action pairs. As it is shown in Figure 2, the hollow circle represents the state, and the black filled circle represents the state-action pair. Two algorithms were used to estimate the value function of $Q^{\pi}(s,a)$ of the state-action pair. After study in all time episodes have been finished, the estimated value of the value function $V^{\pi}(s)$ in every state is updated to the maximum estimated value of the value function of the corresponding state-action pair:

$$V^{\pi}(s) = \max_{a} Q^{\pi}(s,a), a \in A(s)$$

(2)

Thus, the reason to estimate $Q^{\pi}(s,a)$ by two algorithms is to obtain the estimated value of the value function $V^{\pi}(s)$. In reinforcement learning, strategy and value function can influence and modify each other. The intelligent agent can evaluate strategy to learn value function and deploy value function to modify strategy, and finally figure out the optimal strategy and the best value function [9].

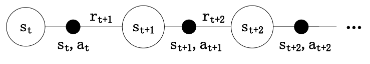

Figure 2. Episode

Sarsa algorithm is a strategic algorithm, which estimates the value function $Q^{\pi}(s,a)$ of all state-action pairs in action strategy π, and changes strategy π into the corresponding action of maximum Q^{π}. Therefore, in the next time step, the intelligent agent will choose action according to the new strategy π. The formula of the estimated value $Q^{\pi}(s,a)$ is updated as:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma Q(s_{t+1},a_{t+1}) - Q(s_t,a_t)] \tag{3}$$

In the formula, α, γ are parameters; a_t is the action chosen with the strategy π; s_{t+1} is the state in the next time step; a_{t+1} is the action chosen with the new strategy π in the next time step; r_{t+1} is the reward in the transfer and the subscript is t+1 for it influences the intelligent agent in the next time step. After each transfer is completed, $Q(s_t,a_t)$ is updated. The upgrade of the formula needs every element in the group $(s_t,a_t,r_{t+1},s_{t+1},a_{t+1})$ which forms the transfer of the state-action pair. In addition, Sarsa algorithm is also named after these five elements [10].

2.5. Q-Learning Algorithm

One breakthrough in reinforcement learning is the development of temporal difference control independent from strategy, that is Q-Learning algorithm [11]. The upgrade formula of its estimated value $Q^{\pi}(s,a)$ is:

$$Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[r_{t+1} + \gamma \max Q(s_{t+1},a) - Q(s_t,a_t)] \tag{4}$$

In this algorithm, the estimated value of the value function of state-action pair Q almost equals to the value function of the optimal state-action pair Q^* (the value function of the best action in certain state), and does not depend on the modified strategy π in the next time step. Specifically, a in formula $\max Q(s_{t+1},a)$ is different from a_{t+1} in the upgraded formula $Q(s_{t+1},a_{t+1})$ in Sarsa algorithm. The independence from the upgraded strategy is the difference between the approach that does not use strategy and the one with strategy. The convergence of the two algorithms has been proved [12].

3. Obstacle Avoidance Model

In this research, grid method was used to simulate environment. When the intelligent agent notices the obstacle, it needs to move along the border of the obstacle from one side to another side. As the obstacle is symmetrical, the probabilities of the intelligent agent moving from each direction to avoid the obstacle are the same. Thus, the environment is simplified as one-direction detour.

Based on the above environment simulation, the environment shown in grid method is in Figure 3. In the 4*5 grid, the six black grids represent the obstacle, the other white grids represent the path, and the bottom left corner is the beginning point while the bottom right corner is the terminal point. The intelligent agent knows the coordinate of the beginning and terminal points and can judge whether the neighbouring four grids are the path or the obstacle. In this environment, there are two paths above the obstacle. The one away from the obstacle is the safe path, and going along this path, the intelligent agent will not come across the obstacle. The one near the obstacle is the optimal path, but going along his path, the intelligent agent may come across the obstacle.

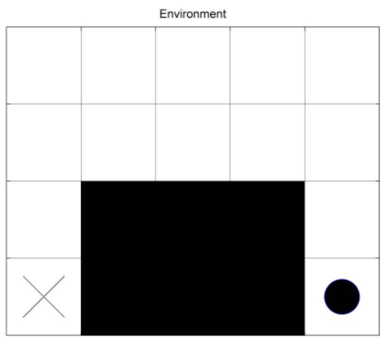

Figure 3. The Environment Simulated by Grid Method

The process that the intelligent agent moves from the beginning point to the terminal point is called the episode. As the intelligent agent learns the environment, the time step in the episode will be converged to a certain value. In reinforcement learning, the intelligent agent explores the whole environment in each episode, and in each time step it can move to one neighbouring grim. If in the next time step, it moves to the grim where the obstacle is, it will receive a reward of -100, stay at the current point and continue to explore in the next time step; if in the next time step, it moves to the edge of the environment, it will stay at the current point, receive a reward of -1, and continue to explore in the next time step; if it moves to the grid of path, it will receive a reward of -1 and continue to explore in the next time step; and if it moves to the terminal point, it will receive a reward of 0 and end the current episode. If the intelligent agent does not come across the obstacle, the nearer it moves around the obstacle, the shorter the path of obstacle avoidance is and the more optimal the path is.

4. The Results of Simulation and its Analysis

In the obstacle avoidance model built in part 3, Sarsa algorithm and Q-Learning algorithm were used respectively to address obstacle avoidance, to prove whether the two algorithms can make the intelligent agent to find the optimal path in the limited time episode.

In the two algorithms, parameters were: $\alpha = 0.1, \gamma = 1$; in the softmax that balanced the exploration and utilisation, the parameter was $\tau = 0.1$; the number of episode was 300; and in the experiment, the intelligent agent did not have prior knowledge, and therefore the initial value of the value function in all states was 0.

In Figure 4 and Figure 5, it can be concluded that by using Sarsa algorithm and Q-Learning algorithm, after 180 episodes, the number of time step that the intelligent agent uses to move from the beginning point to the terminal point is within 8 on the whole. And the fluctuations occur because of the exploration in reinforcement learning. That means even if strategy has been converged, the intelligent agent with softmax still has a low probability to choose non-greedy strategy. And Figure 3 indicates that in the environment, the minimum number of step for obstacle avoidance is 8, which means it takes at least 8 times of the transfer for the intelligent agent to move to the terminal point. Thus, two algorithms both figure out the optimal path.

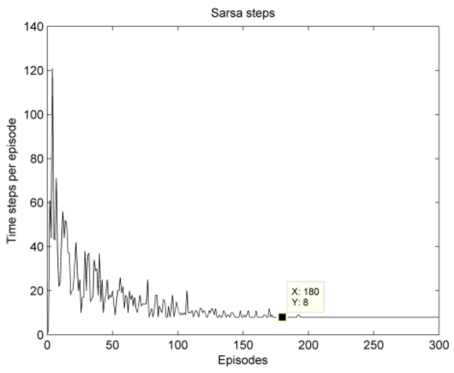

Figure 4. The Results of Using Sarsa Algorithm

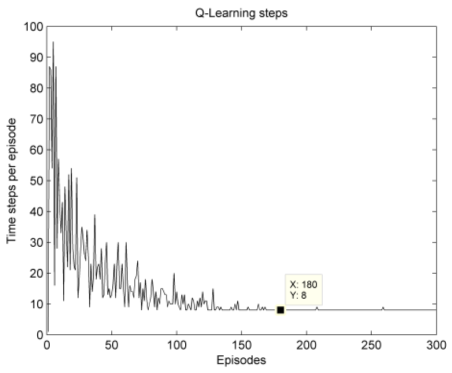

Figure 5. The Results of Using Q-Learning Algorithm

In the following,
the optimal path is illustrated in the strategy graphs, which are produced by the value function of the state.

Table 1.The Estimated Value Function of the State of Sarsa Algorithm

Line / Row	1	2	3	4	5
1	-5.8060	-5.0887	-4.3024	-3.4619	-2.6892
2	-6.0182	-5.0121	-4.0062	-3.0017	-2.0000
3	-7.0284	0	0	0	-1.0000
4	-8.0456	0	0	0	0

Table 2. The Estimated Value Function of the State of Q-Learning Algorithm

Row / Line	1	2	3	4	5
1	-5.3808	-4.7909	-4.0836	-3.3577	-2.6618
2	-6.0000	-5.0000	-4.0000	-3.0000	-2.0000
3	-7.0000	0	0	0	-1.0000
4	-7.9999	0	0	0	0

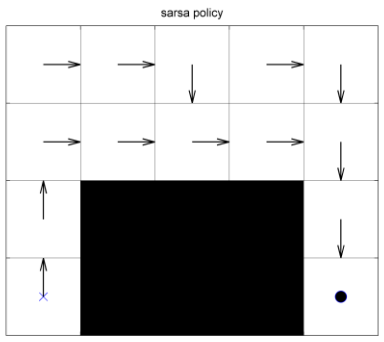

Figure 6. The Strategy Graph of Sarsa Algorithm

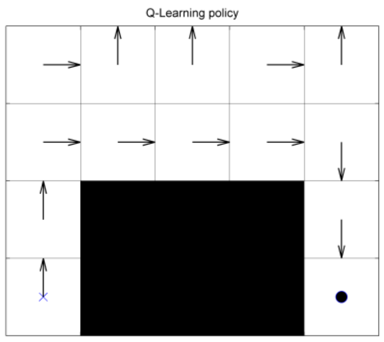

Figure 7.The Strategy Graph of Q-Learning Algorithm

Table 1 and table 2 show the estimated value of the value function of the state after 300 episodes in Sarsa algorithm and Q-Learning algorithm respectively. Different positions in the chart represent different states in the environment. The bottom right corner is the value function of the state of the terminal point. When the intelligent agent arrives at this point, it receives a reward of 0 and, thus the estimated value of the value function at this point is always 0. Other estimated value 0 are the points where the obstacle is. Since the intelligent agent cannot reach these points, it will not receive any reward in any time step at these points. The state that has a higher estimated value means that the intelligent would more likely to arrive at these points. Figure 6 is the corresponding strategy figure of Table 1, and Figure 7 is the corresponding strategy figure of Table 2. In these figures, the position that the arrow points at is the best state where the intelligent agent transfers in the next time step. For example, the intelligent agent is at the second line and first row in Chart 1. According to the path it has learnt, the intelligent agent moves to the right grid with the estimated value of -5.0121 in the next step for the estimated value of the right grid -5.0121 is bigger than that of the above grid -5.8060. With the same theory, the intelligent agent can use the strategy that it has learnt to work out the shortest path from the beginning point to the terminal point. From the above two figures it can be perceived that from the beginning point to the terminal point, the intelligent agent with certain strategy makes transfer of state for 8 times, which is the optimal path mentioned above.

5. Conclusions

This paper introduces the theory framework and related concepts in reinforcement learning, compares the two algorithms used to balance exploration and utilisation, and then analyses the classical Sarsa algorithm and Q-Learning algorithm. After obstacle avoidance was appropriately simplified and its model was built, Sarsa algorithm and Q-Learning algorithm were applied to deal with obstacle avoidance. The results of simulation experiment show that the two algorithms can address obstacle avoidance perfectly. In later studies, referring to the development of reinforcement learning, researchers can study how to combine reinforcement and other control technology; they can also use game theory to study reinforcement learning of the multi-agent system in order to solve more practical problems.

Acknowledgments

This work was supported by Science and Technology Key Project of Henan Province under Grant No. 152102210193.

References

[1] R S Sutton, A G. Barto, “Reinforcement learning: an introduction”, Cambridge, USA: The MIT Press, **1997**.

[2] G Yang, C Shifu, L Xin. Research on Reinforcement Learning Technology:A Review [J]. *Acta Automatica Sinica*, **2004**, vol. 30, no. 1, pp. 86-100.

[3] J.C.H. Watkins, P.Dayan. Q-learning [J]. Machine learning, 1992, 8:, pp. 279-292.

[4] T Ji-hui. A Novel Building Boundary Extraction Method Fof High-resolution Aerial Image [J]. Eview of Computer Engineering Studies, Vol.1, No.2, 2014, pp.19-22.

[5] Z Rubo, Gu Guochang. Reinforcement Learning Theory, Algorithms and Its Application [J], *Control Theory & Applications*, **2000**, vol. 17, no. 5, pp. 637-642.

[6] H Bingqiang. Reinforcement Learning Method and Its Application [D]. *Shanghai: Shanghai Jiao Tong University*, **2007**.

[7] C Xuejiang. Multi-robot Coordination based on Reinforcement Learning [D], *Hangzhou: Zhejiang University of Technology*, **2004.**

[8] Rummery, G. A. and Niranjan, M. On-line Q-learning using connectionist systems. Technical Report CUED/F-INFENG/TR166, Cambridge University Engineering Department. **1994.**

[9] Chengfeng Jian, Wei Zhang and Yue Ying.tr [J].Eview of Computer Engineering Studies , Vol.2, No.2, **2015**, pp.19-24.

[10] S Singh, S., Jaakkola, T., Littman, M., Szepesvari, C.: Convergence results for single-step on-policy reinforcement-learning algorithms [J]. Machine Learning vol. 38, no. 3, **(2000)**, pp. 287–308

[11] XLiangsheng, Yan Weisheng. Study on Mobile Robot Motion Planning Based on Grid Method [J]. Computer Simulation,,vol. 29, no. 12, 2012,pp. 229-233.

[12] Junling Hu, M.P.Wellman, Nash Q-learning for General-Sum stochastic Games, Journal of Machine Learning Research, vol.4, pp.1039-1069, 2003.