Reward Shaping for a Reinforcement Learning Method-Based Navigation Framework

Cubahiro Roland* · Donggyu Choi · Jongwook Jang

Dong-eui University

E-mail : bukuja@gmail.com / dgchoi@deu.ac.kr / jwjang@deu.ac.kr

ABSTRACT

Applying Reinforcement Learning in everyday applications and varied environments has proved the potential of the of the field and revealed pitfalls along the way. In robotics, a learning agent takes over gradually the control of a robot by abstracting the navigation model of the robot with its inputs and outputs, thus reducing the human intervention. The challenge for the agent is how to implement a feedback function that facilitates the learning process of an MDP problem in an environment while reducing the time of convergence for the method. In this paper we will implement a reward shaping system avoiding sparse rewards which gives fewer data for the learning agent in a ROS environment. Reward shaping prioritizes behaviours that brings the robot closer to the goal by giving intermediate rewards and helps the algorithm converge quickly. We will use a pseudocode implementation as an illustration of the method.

키워드

Reward function, Mobile Robot Navigation, Deep Reinforcement Learning

Ⅰ. INTRODUCTION

Reinforcement Learning (RL) is a popular method for an autonomous agent to learn optimal behaviour through repeated interactions with an environment. In general, the RL functions seek to maximize the total reward received by the leanring agent[2]. Modern RL algorithms seek to solve two kinds of problems: episodic and continous tasks. The nature of episodic functions fits well into deterministic models but fails to account for time-horizon or continous models like controlling robots in a non-episodic environment. One way to solve the problem is to apply the discounting approach in which a discount factor ɣ (0, 1) is used to maximize the sum rewards r0+ɣR1+ɣ2R2+⋯.+ɣnRn. However, the discounted has disadvantages in application. A low discount return has a strong bias toward high near-term performances which is not a good solution for long-term behaviours optimization, the solution to this problem would be to increase the discount around 1, however, a large discount returns big numbers which are difficult to learn from.

A suitable solution is to apply the average reward. The average reward doesn't care just as much about delayed rewards as it does about immediate rewards. The average reward is denoted:

$$
\begin{aligned}
r(\pi) &= \lim_{h\to\infty}\frac{1}{h}\sum_{t=1}^{h} E\left[R_t \mid S_0, A_{0:t-1} \sim \pi\right] \\
&= \lim_{h\to\infty} E\left[R_t \mid S_0, A_{0:t-1} \sim \pi\right], \\
&= \sum_{s}\mu_{\pi}(s)\sum_{a}\pi(a \mid s)\sum_{s',r}p(s', r \mid s,a)r,
\end{aligned}
$$
(1)

Here, μπ is a special steady-state distribution μπ = limt→∞ Pr{St = s | A0:t-1 ~ π} under which a selected action according to π remain in the same distribution[3]. Thus, we have a model of reward that allows to order policies according to their average reward per time step, or according to their r(π). In average-reward setting, returns are defined in terms of differences between immediate rewards and the average reward: Gt = Rt+1 – r(π) + Rt+2 – r(π) + Rt+3 – r(π) + ⋯, otherwise known as differential return.

* speaker

Ⅱ. AVERAGE REWARD ON POLICY SEARCH METHODS

When applied to the learning methods, the average reward helps choose the best policy under the conditions of performances based on the maximization of the average reward. The action value function in the average-reward is known as the differential action-value function:

$$V^{\pi}(s) = \lim_{h \to \infty} \frac{1}{h} \sum_{t=1}^{h} E\left[R_t^{\pi}(s) - r(\pi)\right] \quad (2)$$

Traditionally, the policy methods update is define as : where ɣt is the discount, Θ the policy parameter. The goal here is to substitute the discount method with the average reward in the update, which gives us:

$$\theta_{t+1} = \theta_t + \alpha \triangle \ln \pi(A_t \mid S_t, \theta_t) * \left[R_{t+1} - r(\pi) + V(S_{t+1}, w) - V'(S_t, w)\right] \quad (3)$$

The differential return and the estimate value function V(St+1, w) of the next state gives us an approximation of the one-step bootstrap return, based on the TD error method. In this equation, V(St+1, w) is the critic part, giving the immediate feedback. V'(St, w) here is the baseline added to reduce the variance of the update, thus accelerating the learning process [1].

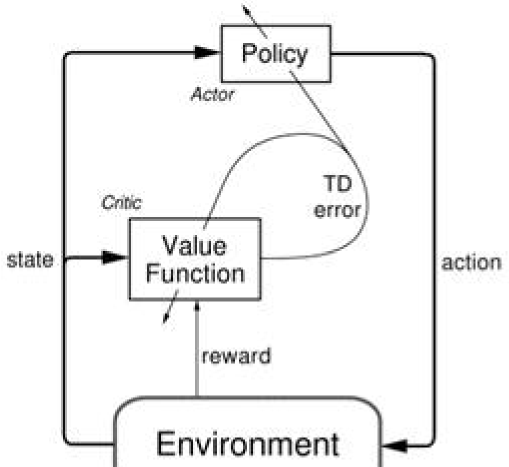

Fig1. Actor-critic model schema

The [Rt+1 – r(π) + V(St+1, w) – V'(St, w)] part can be considered as a TD error δt, which gives us:

$$\theta_{t+1} = \theta_t + \alpha \nabla \ln \pi(A_t \mid S_t, \theta_t)\delta_t \quad (4)$$

The TD error tells us how good an action is compared to the average of that state (Figure 1). If it is positive, the return should change the policy parameters by influencing the probability of that action being to be chosen the next time we are in the same state, because it is better than we expected. A negative TD error produces an opposite result.

Ⅲ. ACTOR-CRITIC PSEUDOCODE EXAMPLE

An actor-critic implementation requires a policy and value function parameterizations and a differential return set to zero. The rest of initializations are similar to a discounted actor critic model and include the weights parameters and the step-size parameters for the weights, the policy and the average reward however we like. From the initial state S, we start the learning process like this:

```
Loop forever (each step)
        $A \sim \pi(. \mid S, \Theta)$

        Take an action A, observe S', R

        $\delta \leftarrow R - r(\pi) + V(S', w) - V'(S, w)$
        //TD error computation

        $r(\pi) \leftarrow r(\pi) + \alpha(r)\delta$
        // update of the average award

        $w \leftarrow w + \alpha(w)\delta \nabla V(S, w)$
        //Update of the value function
        weights with the TD update

        $\Theta \leftarrow \Theta + \alpha(\Theta)\delta \nabla \ln\pi(A \mid S, \Theta)$
        //Update of the policy parameters

        $S \leftarrow S'$
```

Ⅳ. DISCUSSION AND CONCLUSIONS

In this paper we studied a case of reshaping the reward model adapted for continous tasks. The standard model of the policy search based reinforcement learning uses a discounted reward to bring the model to convergence. However, we saw that the discount is biased toward near-term performances and has high variance when maximized. In place of a discounted return we used an average reward formulation which considers all the returns in the same proportion, hence it gives an intuitive value to estimate which policy is better. We later perform a pseudocode to apply the average reward formulation on the policy search based actor-critic method.

Acknowledgement

This research was supported by the BB21plus funded by Busan Metropolitan City and Busan Institute for Talent & Lifelong Education (BIT). And this research was supported by the MSIT (Ministry of Science and ICT), Korea, under the Grand Information Technology Research Center support program (IITP-2022-2020-0-01791) supervised by the IITP (Institute for Information & communications Technology Planning & Evaluation).

References

[1] Jiang, Yuqian & Bharadwaj, Sudarshanan & Wu, Bo & Shah, Rishi & Topcu, Ufuk & Stone, Peter. (2020). Temporal-Logic-Based Reward Shaping for Continuing Learning Tasks.
[2] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. MIT press, 2018
[3] Rati Devidze and Goran Radanovic and Parameswaran Kamalaruban and Adish Singla. Explicable Reward Design for Reinforcement Learning Agents. In Advances in Neural Information Processing Systems, 2021