A REVISE ON MAXMIN Q-LEARNING: CONTROLLING THE ESTIMATION BIAS OF Q-LEARNING

20337097 欧建军

ABSTRACT

Maxmin Q-learning stressed on the estimation bias of Q-learning. Q-learning suffers from overestimation bias, because it approximates the maximum action value using the maximum estimated action value. The paper to be revise has provided an understanding of how bias interacts with performance, and the extent to which existing algorithms mitigate bias. It bring up the Maxmin Q-learning, which provides a parameter to flexibly control bias. Also, it showed theoretically that there exists a parameter choice for Maxmin Q-learning that leads to unbiased estimation with a lower approximation variance than Q-learning and proposeed a novel Generalized Q-learning framework. In order to learn more into the paper and implement its theory, in this paper, I 1) retell the theory part of the paper and analysis its rationality;2)rebuild the code of Maxmin Q-learning and use it to train in the given setting of the paper and one of the Benchmark Envirments;3)try to set the parameters to make the code successfully run in local divice without large time consumption.

1 INTRODUCTION

This paper was to revise the paper "MAXMIN Q-LEARNING: CONTROLLING THE ESTIMATION BIAS OF Q-LEARNING"(Qingfeng Lan, Yangchen Pan, Alona Fyshe, Martha White),which provided the Maxmin Q-learning with a parameter to control the bias,and also theoretical prove for the convergence of their algorithm in the tabular case, as well as convergence of several previous Q-learning variants, using a novel Generalized Q-learning framework

We can lean the backgrround of the paper's work from the pape's intoduction

Q-learning (Watkins, 1989) is one of the most popular reinforcement learning algorithms. One of the reasons for this widespread adoption is the simplicity of the update. On each step, the agent updates its action value estimates towards the observed reward and the estimated value of the maximal action in the next state. This target represents the highest value the agent thinks it could obtain from the current state and action, given the observed reward.

Unfortunately, this simple update rule has been shown to suffer from overestimation bias (Thrun Schwartz, 1993; van Hasselt, 2010). The agent updates with the maximum over action values might be large because an action’s value actually is high, or it can be misleadingly high simply because of the stochasticity or errors in the estimator. With many actions, there is a higher probability that one of the estimates is large simply due to stochasticity and the agent will overestimate the value. This issue is particularly problematic under function approximation, and can significant impede the quality of the learned policy (Thrun Schwartz, 1993; Szita L″orincz, 2008; Strehl et al., 2009) or even lead to failures of Q-learning (Thrun Schwartz, 1993). More recently, experiments across several domains suggest that this overestimation problem is common (Hado van Hasselt et al., 2016).

Double Q-learning (van Hasselt, 2010) is introduced to instead ensure underestimation bias. The idea is to maintain two unbiased independent estimators of the action values. The expected action value of estimator one is selected for the maximal action from estimator two, which is guaranteed not to overestimate the true maximum action value. Double DQN (Hado van Hasselt et al., 2016), the extension of this idea to Q-learning with neural networks, has been shown to significantly improve performance over Q-learning.

Several other methods have been introduced to reduce overestimation bias, without fully moving towards underestimation. However, these strategies do not guide how strongly we should correct for overestimation bias, nor how to determine—or control—the level of bias.

The overestimation bias also appears in the actor-critic setting (Fujimoto et al., 2018; Haarnoja et al., 2018). However, there is also no theoretical guide for choosing the number of estimators such that the overestimation bias can be reduced to 0.

In the paper, the author has studied the effects of overestimation and underestimation bias on learning performance, and use them to motivate a generalization of Q-learning called Maxmin Q-learning. Maxmin Q-learning directly mitigates the overestimation bias by using a minimization over multiple action-value estimates. Moreover, it is able to control the estimation bias varying from positive to negative which helps improve learning efficiency. And the author proved theoretically with an appropriate number of action-value estimators, we are able to acquire an unbiased estimator with a lower approximation variance than Q-learning.

For deeper learning ,in this paper, I will try to explain the theory of controlling the estimation bias of Q-learning. Then, I will try to revise the two experiments in the theory part. Also, I shall verify the claims on one of the seven benchmarks used in the paper. All the experiments parameters and envirments have been customized to be tun on local device. And I will also study the convergence properties of the algorithm within a novel Generalized Q-learning framework, which is suitable for studying several of the recently proposed Q-learning variants.

2 PROBLEM SETTING

Q-Learning was a learning algoithm base on Markov Decision Process. Let's formalize the problem as a MDP, (S, A, P, r, γ), where S is the state space, A is the action space, P : S×A×S → [0, 1] is the transition probabilities, r : S×A×S → R is the reward mapping, and γ ∈ [0, 1] is the discount factor. At each time step t, the agent observes a state St ∈ S and takes an action At ∈ A and then transitions to a new state St+1 ∈ S according to the transition probabilities P and receives a scalar reward Rt+1 = r(St, At, St+1) ∈ R. The goal of the agent is to find a policy π : S × A → [0, 1] that maximizes the expected return starting from some initial state. Then, we can put forward the Q-Learning algorithm, which is a off-policy attempts to learn the optimal policy. It tries to solve:

$$Q^*(s,a) = E[R_{t+1} + \max_{a' \in A} Q^*(S_{t+1}, a' | S_t = s, A_t = a)] \tag{2.1}$$

If we could get Q(S,A),then we will get the optimal policy.The optimal policy is to act greedily with respect to these action values.And we can upadte the approximation of Q by a sample$(s_t, a_t, r_{t+1}, s_{t+1})$ as:

$$Q^(s,a) = Q^(s,a) + \alpha(Y_t^Q - Q_t(s_t,a_t))//forY_t^Q =_{t+1} + \gamma \max_{a' \in A} Q(s_{t+1,a'}) \tag{2.2}$$

where α is the step-size.The transition can be generated off-policy, from any behaviour that sufficiently covers the state space.

3 THEORY

In this part, I tried to review the theory part that the autho has propose, which includes the understanding of when overestimation bias helps and hurts, the maxmin Q-leaning's structure and its theoretical explaination, as well as the convergence analysis of maxmin Q-learning. The former two parts contain two experiments, which I will go deeper into in the next two section. And the last part is a pure theoretical, whcih means it doesn't contain any experimental part to empirically verify its theory, I will try to review it but with a shorter length.

3.1 UNDERSTANDING WHEN OVERESTIMATION BIAS HELPS AND HURTS

In this part, it is explained that the underestimation and overestimation may helps or hurts while training, and it depends on the envirment. the author briefly discuss the estimation bias issue, and empirically show that both overestimation and underestimation bias may improve learning performance, depending on the environment. This motivates

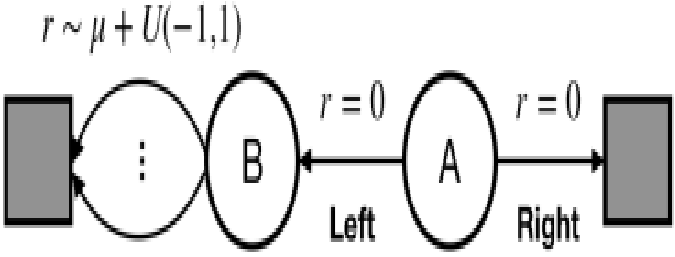

图 3.1: pic1

our Maxmin Q-learning algorithm, which allows us to flexibly control the estimation bias and reduce the estimation variance.

The overestimation bias occurs since the target $\max_{a0\in A} Q(s_{t+1}, a0)$is used in the Q-learning update. Because Q is an approximation, it is probable that the approximation is higher than the true value for one or more of the actions. The maximum over these estimators, then, is likely to be skewed towards an overestimate. For example, even unbiased estimates $Q(s_{t+1}, a0) forall a0$, will vary due to stochasticity. $Q(s_{t+1}, a0) = Q^{\circ}(s_{t+1}, a0) + ea0$, and for some actions, ea0 will be positive. As a result, $E[\max_{a0\in A} Q(s_{t+1}, a0)] \leq \max_{a0\in A} E[Q(s_{t+1}, a0)] = \max_{a0\in A} Q^*(s_{t+1}, a')$.

However, overestimation may not always lead to detrimental results. And underestimation may also be harmful. Since the overestimation may lead to more exploration in the overestimated region. And the underestimation might prevent the agent to explore and learn from the underestimation region. So if the overestimation occours in the actual high-value region, this might encourage the agent to find a better prolicy, and if it occours in a low-value region, this may lead to more useless exploration.

The paper use a very simple example of simple episodic MDP adapted from Figure 6.5 in Sutton Barto (2018)to show that different varients of Q-learning with different overestimation and undedestimation may lead to defferent result. Figure 1 show the simple episodic MDP, and the paper examed that the overestimation may lead to higher convwergence speed, and underestimation may lead to suboptimal results. We will have a closer look into it in the next section.

3.2 符号说明

所使用的符号及说明如表3.1所示。

表 3.1: 符号说明

符号	说明	单位
符号1	这里是符号1的说明。	单位
符号2	这里是符号2的说明。	单位
符号3	这里是符号3的说明。	单位

3.3 符号说明

4 CUSTOMIZED PART AND IMPROVEMNET

4.1 模型1

针对问题1，建立了模型1。

其中，公式的书写方式如下。

$$\mathrm{e}^{i\theta} = \cos\theta + i\sin\theta. \tag{4.3}$$

公式4.3就是大名鼎鼎的Euler公式。

4.2 模型2

针对问题2，建立了模型2。

在论文中可能需要插入图片，在这里插入图片的方式如下。

图??是在实验室中，科学家拿着微生物的照片。

5 EXPERIMENTS

It took a lot of effort to reconstruct the whole running environment of the project. Unfortunately, it is also very time consuming to revise every experiment of the comparing results in all seven Benchmark Envirments. For convenience, I just choose one of the seven Benchmark Envirments to have a closer look.That is LunarLander-v2.And for the theoy proving experiments, since the souce code was not open, I modified the code used in Benchmarks and simplifid then built the envirment by my self. Since all the experiments' result took an average of 500 to 5000 runs, but because of the limited devices and time, I could only run 1 times for each cases, the final result of the rebuilt code might not be so smooth, but we can still extract the trend and difference of different models.

5.1 问题的结果

在这里写问题的结果。

5.2 模型的检验

在这里写对模型的检验。

6 Conclusion

6.1 模型的优点

该模型具有如下的优点。

- 优点1；
- 优点2；
- 优点3。

6.2 模型的缺点与改进

与此同时，该模型也具有如下的缺点。

- 缺点1；
- 缺点2。

同时，在这里给出进一步优化模型的思路。

参考文献

[1] 作者. 文献[M]. 地点:出版社,年份.

[2] 作者. 文献[M]. 地点:出版社,年份.

附录