A Note on Double Q-Learning

Po-Chun, Hsu
Institute of Computer Science and Engineering
National Chiao Tung University
hpc.cs08g@nctu.edu.tw

1 Introduction

Q-learning is a classic model-free, off-policy, value-based algorithm come up with Watkins and Dayan [1992], but in the case of stochastic environments or the obtained reward is not fixed. It may cause overestimate problems, which makes the expected total reward that will be higher than the actual reward and it may reduce the effectiveness of Q-learning. So, the author proposes a double q-learning algorithm. Using double estimators instead of one estimator. This method can solve the problem of overestimate. Hasselt [2010] says that this is the first off-policy value based reinforcement learning algorithm that does not have a positive bias in estimating the action values in stochastic environment. So to make an important contribution to reducing overestimate, such algorithms can be used in multi-armed bandit problems, or gambling and other environments with random rewards. I think this is a good improvement to Q-learning. It does not need to significantly modify the algorithm to reduce the problem of overestimate, so that we can train in stochastic environments. Compared with games, the real environment is often accompanied by noise. Double Q-learning can reduce the impact of these noises, making the method of reinforcement learning more widely applicable.

2 Problem Formulation

Q-learning is a reinforcement learning algorithm, which can be used to solve optimally Markov Decision Processes (MDPs). The following formula is the Q-learning update formula.

$$Q_{t+1}\left(s_t, a_t\right) = Q_t\left(s_t, a_t\right) + \alpha_t\left(s_t, a_t\right)\left(r_t + \gamma \max_a Q_t\left(s_{t+1}, a\right) - Q_t\left(s_t, a_t\right)\right)$$

$Q_t(s, a)$ is the state action value that at time t in state s and acdtion a.

The next state s_{t+1} is determined by a fixed state transition distribution $P : S \times A \times S \to [0, 1]$, where $E\left\{r_t \mid (s, a, s') = (s_t, a_t, s_{t+1})\right\} = R_{sa}^{s'}$. P_{sa}^{s} is the probability that transfer from state s_t apply action a_t to s_{t+1}.

The optimal value function to solve problem is:

$$\forall s, a : Q^*(s, a) = \sum_{s'} P_{sa}^{s'}\left(R_{sa}^{s'} + \gamma \max_a Q^*\left(s', a\right)\right)$$

$Q^*(s, a)$ may overestimate.

Preprint. Under review.

3 Theoretical Analysis

3.1 Estimating the Maximum Expected Value

Consider a set of M random variables $X = \{X_1, \ldots, X_M\}$, we want to find the maximum expected value of the variables

$$\max_i E\{X_i\}$$

We can't directly get $\max_i E\{X_i\}$, so we use constructing approximations for $E\{X_i\}$ for all i.

3.2 Use The Double Estimator

If we use the single estimator, overestimation will happen in Q-learning. We avoid this problem that using the double estimator to approximate $\max_i E\{X_i\}$. We use two sets of estimators $\mu^A = \{\mu_1^A, \ldots, \mu_M^A\}$ and $\mu^B = \{\mu_1^B, \ldots, \mu_M^B\}$ We assume that the samples are split in a proper manner so μ^A and μ^B are unbiased. We use $\mu_{a^*}^B$ as an estimate for $\max_i E\{\mu_i^B\}$ to obtain the approximation

$$\max_i E\{X_i\} = \max_i E\{\mu_i^B\} \approx \mu_{a^*}^B$$

If sample is large enough, we can get $\mu_i^A(S) = \mu_i^B(S) = E\{X_i\}$. The expected valeu of the approximation by the double estimator can thus be given by $\sum_j^M P(j = a^*) E\{\mu_j^B\} = \sum_j^M E\{\mu_j^B\} \int_{-\infty}^{\infty} f_j^A(x) \prod_{i \neq j}^M F_i^A(x) dx$ According to Lemma 1, we prove that the estimate $E\{\mu_j^B\}$ is not an unbiased estimate of $\mu_i^A(S)$. Lemma 1. Using two sets of unbiased estimators $\mu^A = \{\mu_1^A, \ldots, \mu_M^A\}$ and $\mu^B = \{\mu_1^B, \ldots, \mu_M^B\}$ and $X = \{X_1, \ldots, X_M\}$ a set of random variables. We can get that $E\{\mu_i^A\} = E\{\mu_i^B\} = E\{X_i\}$ for all i

3.3 Double Q-Learning

Author presents the algorithm called Double Q-learning to avoid these overestimation issues.

```
Algorithm 1 Double Q-learning
1: Initialize $Q^A, Q^B, s$
2: repeat
3: Choose $a$, based on $Q^A(s,\cdot)$ and $Q^B(s,\cdot)$, observe $r, s'$
4: Choose (e.g. random) either UPDATE(A) or UPDATE(B)
5: if UPDATE(A) then
6: Define $a^* = \arg\max_a Q^A(s', a)$
7: $Q^A(s,a) \leftarrow Q^A(s,a) + \alpha(s,a)\left(r + \gamma Q^B(s', a^*) - Q^A(s,a)\right)$
8: else if UPDATE(B) then
9: Define $b^* = \arg\max_a Q^B(s', a)$
10:    $Q^B(s,a) \leftarrow Q^B(s,a) + \alpha(s,a)\left(r + \gamma Q^A(s', b^*) - Q^B(s,a)\right)$
11: end if
12: $s \leftarrow s'$
13: until end
```

We proof this algorithm can convergence according to lemma 2.

Lemma 2. Consider a stochastic process $(\zeta_t, \Delta_t, F_t), t \geq 0$, where $\zeta_t, \Delta_t F_t : X \rightarrow \mathbb{R}$, satisfy the equation:

$$\Delta_{t+1}(x_t) = (1 - \zeta_t(x_t)) \Delta_t(x_t) + \zeta_t(x_t) F_t(x_t)$$

If 1)The set X is finite 2)$\zeta_t(x_t) \in [0,1], \sum_t \zeta_t(x_t) = \infty, \sum_t (\zeta_t(x_t))^2 < \infty$ w.p.1 and $\forall x \neq x_t : \zeta_t(x) = 0$ 3)$|E\{F_t \mid P_t\}|| \leq \kappa| |\Delta_t| | + c_t$, where $\kappa \in [0,1)$ and c_t converges to zero w.p. 1 4)Var $\{F_t(x_t) \mid P_t\} \leq K(1 + \kappa \|\Delta_t\|)^2$, where K is some constant

Theorem 1. If satisfy the follow condition, Double Q-Learning can converge. 1)The MDP is finite, i.e. $|S \times A| < \infty$ 2)$\gamma \in [0,1)$ 3)The Q values are stored in a lookup table. 4)Both Q^A and Q^B receive an infinite number of updates 5)$\alpha_t(s,a) \in [0,1], \sum_t \alpha_t(s,a) = \infty, \sum_t (\alpha_t(s,a))^2 < \infty$ w.p.I, and $\forall(s,a) \neq (s_t, a_t) : \alpha_t(s,a) = 0$. 6) $\forall s, a, s'$: Var $\left\{R_{sa}^{s'}\right\} < \infty$

proof: Define $\Delta_t = Q_t^A - Q^*$ $F_t(s_t, a_t) = r_t + \gamma Q_t^B(s_{t+1}, a^*) - Q_t^*(s_t, a_t)$, where $a^* = \arg\max_a Q^A(s_{t+1}, a)$ $F_t^Q = r_t + \gamma Q_t^A(s_{t+1}, a^*) - Q_t^*(s_t, a_t)$

We can get

$$F_t(s_t, a_t) = F_t^Q(s_t, a_t) + \gamma\left(Q_t^B(s_{t+1}, a^*) - Q_t^A(s_{t+1}, a^*)\right)$$

This is the expected of F_t.

Then we show that $\Delta_t^{BA} = Q_t^B - Q_t^A$ converges to zero. If Q^B update

$$\Delta_{t+1}^{BA}(s_t, a_t) = \Delta_t^{BA}(s_t, a_t) + \alpha_t(s_t, a_t) F_t^B(s_t, a_t)$$

else if Q^A update

$$\Delta_{t+1}^{BA}(s_t, a_t) = \Delta_t^{BA}(s_t, a_t) - \alpha_t(s_t, a_t) F_t^A(s_t, a_t)$$

Define $\zeta_t^{BA} = \frac{1}{2}\alpha_t$. We can get that

$$\begin{aligned} E\left\{\Delta_{t+1}^{BA}(s_t, a_t) \mid P_t\right\} &= \Delta_t^{BA}(s_t, a_t) + E\left\{\alpha_t(s_t, a_t) F_t^B(s_t, a_t) - \alpha_t(s_t, a_t) F_t^A(s_t, a_t) \mid P_t\right\} \\ &= \left(1 - \zeta_t^{BA}(s_t, a_t)\right)\Delta_t^{BA}(s_t, a_t) + \zeta_t^{BA}(s_t, a_t) E\left\{F_t^{BA}(s_t, a_t) \mid P_t\right\} \end{aligned}$$

Rewrite $E\left\{F_t^{BA}(s_t, a_t) \mid P_t\right\}$

$$E\left\{F_t^{BA}(s_t, a_t) \mid P_t\right\} = \gamma E\left\{Q_t^A(s_{t+1}, b^*) - Q_t^B(s_{t+1}, a^*) \mid P_t\right\}$$

If $E\left\{Q_t^A(s_{t+1}, b^*) \mid P_t\right\} \geq E\left\{Q_t^B(s_{t+1}, a^*) \mid P_t\right\}$, we know that $Q_t^A(s_{t+1}, a^*) = \max_a Q_t^A(s_{t+1}, a) \geq Q_t^A(s_{t+1}, b^*)$, Then,

$$\begin{aligned} \left|E\left\{F_t^{BA}(s_t, a_t) \mid P_t\right\}\right| &= \gamma E\left\{Q_t^A(s_{t+1}, b^*) - Q_t^B(s_{t+1}, a^*) \mid P_t\right\} \\ &\leq \gamma E\left\{Q_t^A(s_{t+1}, a^*) - Q_t^B(s_{t+1}, a^*) \mid P_t\right\} \leq \gamma\|\Delta_t^{BA}\| \end{aligned}$$

If $E\left\{Q_t^B(s_{t+1}, a^*) \mid P_t\right\} > E\left\{Q_t^A(s_{t+1}, b^*) \mid P_t\right\}$, we know that $Q_t^B(s_{t+1}, b^*) \geq Q_t^B(s_{t+1}, a^*)$. Then, we can get

$$\begin{aligned} \left|E\left\{F_t^{BA}(s_t, a_t) \mid P_t\right\}\right| &= \gamma E\left\{Q_t^B(s_{t+1}, a^*) - Q_t^A(s_{t+1}, b^*) \mid P_t\right\} \\ &\leq \gamma E\left\{Q_t^B(s_{t+1}, b^*) - Q_t^A(s_{t+1}, b^*) \mid P_t\right\} \leq \gamma\left\|\Delta_t^{BA}\right\| \end{aligned}$$

Finally, $\left|E\left\{F_t^{BA} \mid P_t\right\}\right| \leq \gamma\left\|\Delta_t^{BA}\right\|$ apply lemma 2, Δ_t^{BA} convergence to zero.

4 Conclusion

This concept that using double estimator was later used by DeepMind on DQN, called double DQN, which can also reduce the situation of the overestimate. This method of using two estimators should be implemented on most value-based reinforcement learning algorithms, so it can be used in a variety of value-based reinforcement learning algorithms that would produce overestimates. When the estimated value becomes accurate, we can know what the actual expected total reward is, not just the optimal policy. It may be helpful for analyzing the experimental results, and we can know how much it differs from the actual maximum value.

References

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3-4):279–292, 1992.

Hado V. Hasselt. Double q-learning. In J. D. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 2613–2621. Curran Associates, Inc., 2010. URL `http://papers.nips.cc/paper/3964-double-q-learning.pdf`.