Double Q-Learning

Ching-Wei, Tseng
Department of Computer Science
National Chiao Tung University
garry233.iie08g@nctu.edu.tw

1 Introduction

Q-Learning is a well-known reinforcement learning algorithm. However, it performs poorly in some stochastic environments. Q-Learning may over-estimate the action values because it uses the sum of a series of maximum action values as an approximation for the maximum expected action value.

To slove this problem, this paper proposed a novel algorithm called Double Q-Learning, which uses double estimator to update each other. It can avoid overestimation but sometimes under-estimate.

From my perspective, Double Q-Learning is a relatively conservative algorithm. I think "Stand in others shoes" is exactly what Double Q-Learning does.

2 Problem Formulation

Notation that commonly used in this paper :

- $Q(s, a)$: Estimation of action values over state s and action a
- $R_{sa}^{s'}$: The reward over state s and action a and next state s'.
- γ : The discounting value used for caculate cumulative reward.
- α : The learning rate of $Q(s, a)$ update.
- X : A set of M random variables $\{X_1, ..., X_M\}$
- μ_i : an estimator for variable X_i
- f_i : PDF of the i^{th} variable X_i
- F_i : $\int_{-\infty}^{x} f_i(x)dx$, is a CDF

Optimization problem of interest

- Why Double-Q Learning can avoid overestimations ?
- Does Double-Q Learning converge ?
- Is Double-Q Learning an unbiased estimation ?

Technical assumptions

- The MDP is finite ($|S * A| < -\infty$)
- $\gamma \in [0, 1)$
- Q values are stored in table
- Both Q^a and Q^b can get infinite number of updates

Preprint. Under review.

- learning rate α_t
  $\alpha_t \in [0, 1]$
  $\Sigma_t(\alpha_t) = \infty$
  $\Sigma_t(\alpha_t^2) < \infty$
- $\forall s, a, s^{'}$: Var$\{R_{sa}^{s^{'}}\} < \infty$

3 Theoretical Analysis

There are two Lemmas and one Theorem mentioned in this paper. The proofs are already in the paper. I'll just write down my commentary.

Lemma 1. Let $X = \{X_1, ..., X_M\}$ be a set of random variables and let $\mu^A = \{\mu_1^A, ..., \mu_M^A\}$ and $\mu^B = \{\mu_1^B, ..., \mu_M^B\}$ be two sets of unbiased estimators such that $E\{\mu_i^A\} = E\{\mu_i^B\} = E\{X_i\}$, for all i. Let $M = \{j|E\{X_j\} = max_i E\{X_i\}\}$ be the set of elements that maximize the expected values. Let a^* be an element that maximizes μ^A : $\mu_{a^*}^A = max_i \mu_i^A$. Then $E\{\mu_{a^*}^B\} = E\{X_{a^*}\} \leq max_i E\{X_i\}$. Furthermore, the inequality is strict if and only if $P(a^* \notin M) > 0$.
In other words, consider we have double estimators A and B. First, we choose the "best action" a^* by A, which has the maximum expected action value. Then we use B to estimate the action value of a^*. We will find that $E\{B_{a^*}\}$ is less or equal to $max_i E\{X_i\}$. And this result will let $Q_A(s, a)$ make downward corrections when updating. The breif formula of double estimators updating :

$$Q_A(s, a) = Q_A(s, a) + \alpha(r + \gamma Q_B(s^{'}, a^*) - Q_A(s, a)), a^* = argmax_a Q_A(s^{'}, a)$$

We can make a special case to show that Double estimators can lessen overestimate. Assume we train two estimator "A,B" apart, just like Q-Learning. When they converge, we can write down the equation :

$$Q_A^*(s, a) = Q_A^*(s, a) + \alpha(r + \gamma Q_A^*(s^{'}, a^*) - Q_A^*(s, a)), a^* = argmax_a Q_A(s^{'}, a)$$

$$Q_B^*(s, a) = Q_B^*(s, a) + \alpha(r + \gamma Q_B^*(s^{'}, a^*) - Q_B^*(s, a)), a^* = argmax_a Q_B(s^{'}, a)$$

By the equation, we can know that $r + \gamma Q_A^*(s^{'}, a^*) = Q_A^*(s, a)$. Then, we change the L.H.S into $r + \gamma Q_B^*(s^{'}, a^*)$, where $a^* = argmax_a Q_A(s^{'}, a)$. Because "$a^*$ of A" may not be the "a^* of B", we will find that $r + \gamma Q_B^*(s^{'}, a^*) \leq r + \gamma Q_A^*(s^{'}, a^*) = Q_A^*(s, a)$, where $a^* = argmax_a Q_A(s^{'}, a)$. Then $\alpha(r + \gamma Q_B^*(s^{'}, a^*) - Q_A^*(s, a)) \leq 0$, $a^* = argmax_a Q_A(s^{'}, a)$. By this result, $Q_A^*(s, a)$ will be lowered when using double estimators. And $Q_B^*(s, a)$ can get the same reult, too.

Lemma 2. Consider a stochastic process $(\zeta_t, \Delta_t, F_t), t \geq 0$, where $\zeta_t, \Delta_t, F_t : X \to \mathbb{R}$ satisfy the equations:$\Delta_{t+1}(x_t) = (1 - \zeta_t(x_t))\Delta_t(x_t) + \zeta_t(x_t)F_t(x_t)$, where $x_t \in X$ and $t = 0, 1, 2, ...$ Let P_t be a sequence of increasing σ-fields such that ζ_0 and Δ_0 are P_0-measurable and ζ_t, Δ_t,and F_{t-1} are P_t-measurable, $t = 1, 2, ...$ Assume that the following hold :

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

Then Δ_t converges to zero with probability one. By this Lemma, we can know that the proper ζ_t will surely make Δ_t converges to zero. We can use this lemma to prove convergence of Double Q-learning.

Theorem 1. Assume the conditions below are fulfilled. Then, in a given ergodic MDP, both Q^A and Q^B as updated by Double Q-learning as described in Paper will converge to the optimal value function Q^* as given in the "Bellman optimality equation" with probability one if an infinite number of experiences in the form of rewards and state transitions for each state action pair are given by a proper learning policy. The additional conditions are:

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. $\alpha_t(s, a) \in [0, 1], \Sigma_t \alpha_t(s, a) = \infty, \Sigma_t \alpha_t(s, a)^2 < \infty, \forall (s, a) \neq (s_t, a_t) : \alpha_t(s, a) = 0.$
6. $\forall s, a, s^{'} : Var\{R^{s^{'}}_{s,a} < \infty\}$.

The Theorem 1 can be proof by using lemma 2. There are 6 conditions needed in Theorem 1. From my perspective, condition 2 is common and no need to explain. Condition 1,3 are in a pair. Because $Q(s, a)$ should be stored in a finite lookup table, $|S \times A| < \infty$ must hold. Although condition 4 is impractical, it still conveys that we need to update Q-table sufficient times to make sure estimators are convergent. For condition 5, it is the main condition that warrant estimators converge to the optimal point. Assume we treat each estimator as a "state", and our goal is move estimator from initial state to the optimal state. Since the "distance" between initial state and optimal state may very large, we need $\Sigma_t \alpha_t(s, a) = \infty$ to make sure that we have sufficient power to move. And $\Sigma_t \alpha_t(s, a)^2 < \infty$ make sure that we will finally "stop". Last, I think condition 6 restricts that the reward $R^{s^{'}}_{s,a}$ can not be "total random", or the learning will be meaningless.

4 Conclusion

Potential future research directions

- How to reduce negative bias of Double-Q Learning ?
- Q-Learning may have positive bias while Double-Q may have negative bias. So, is it possible to construct an unbiased off-policy RL algorithm by combining Q and Double-Q ?
- A better algorithm to make two estimator more different (only different in random initial at the beginning)
- Combining Double-Q Learning with Deep Neural Network (However, this is already exist, called Double-DQN)
- Combining Double-Q Learning with other Q-Learning technic.

technical limitations

- Double-Q can avoid over-estimation, but it may under-estimate action values. Hence, in some situation Double-Q Learning may perform pooly.
- For original Double-Q Learning, $Q(s, a)$ must be stored in table. So, when state space or action space is large, this algorithm is not suitable.
- Because Double-Q Learning has two estimators (each time randomly choose one estimator to update), the total training time may be longer then Q-Learning.

latest results on the problem of interest

- Double-DQN : Combine DNN with Double-Q Learning.
- Weighted Double Q-learning : with the goal of balancing between the overestimation in the single estimator and the underestimation in the double estimator.
- Twin-Delayed DDPG (TD3) : The state-of-the-art RL method that combine Policy-Gradient, Actor-Critics, and continuous Double-Deep-Q-Learning.