Double Learning and TD Methods

2017-Feb-24

Di Wu

ID：260562997

Re-thinking for the double learning concept

- Motivation
- Related work
- Simulation:
  - Implementing the Double Sarsa, Expected Sarsa, Double Expected Sarsa, and Two Step Sarsa.

Motivation

- Why we need double Q learning [1]
- Max operator will cause the positive bias
- In Q learning: Target policy is the greedy policy
- In SARSA: ε-greedy
- Maximization bias: maximum of the estimated values is larger than the maximum of the true values:
- Eg: for state s with many action a, true values for q(s, a) are all zero, but the estimated values: some are negative and some are positive

Double Q Learning

- Problem source: We use the same samples to determine the maximization action and determine the value

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

- Potential solution: Divide the samples to two sets, and learn two estimates
- Core idea: Two estimators: A, and B

Double Q Learning

- Discussion for Lemma 1:

Lemma 1. *Let* $X = \{X_1, \ldots, X_M\}$ *be a set of random variables and let* $\mu^A = \{\mu_1^A, \ldots, \mu_M^A\}$ *and* $\mu^B = \{\mu_1^B, \ldots, \mu_M^B\}$ *be two sets of unbiased estimators such that* $E\{\mu_i^A\} = E\{\mu_i^B\} = E\{X_i\}$*, for all* i*. Let* $\mathcal{M} \stackrel{\text{def}}{=} \{j \mid E\{X_j\} = \max_i E\{X_i\}\}$ *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* $P(a^* \notin \mathcal{M}) > 0$*.*

Proof. Assume $a^* \in \mathcal{M}$. Then $E\{\mu_{a^*}^B\} = E\{X_{a^*}\} \stackrel{\text{def}}{=} \max_i E\{X_i\}$. Now assume $a^* \notin \mathcal{M}$ and choose $j \in \mathcal{M}$. Then $E\{\mu_{a^*}^B\} = E\{X_{a^*}\} < E\{X_j\} \stackrel{\text{def}}{=} \max_i E\{X_i\}$. These two possibilities are mutually exclusive, so the combined expectation can be expressed as

$$\begin{aligned} E\{\mu_{a^*}^B\} &= P(a^* \in \mathcal{M}) E\{\mu_{a^*}^B | a^* \in \mathcal{M}\} + P(a^* \notin \mathcal{M}) E\{\mu_{a^*}^B | a^* \notin \mathcal{M}\} \\ &= P(a^* \in \mathcal{M}) \max_i E\{X_i\} + P(a^* \notin \mathcal{M}) E\{\mu_{a^*}^B | a^* \notin \mathcal{M}\} \\ &\leq P(a^* \in \mathcal{M}) \max_i E\{X_i\} + P(a^* \notin \mathcal{M}) \max_i E\{X_i\} \qquad = \max_i E\{X_i\}\ , \end{aligned}$$

Double Q Learning[1]

- Pseudo code:

Algorithm 1 Double Q-learning

Initialize Q^A,Q^B,s

repeat

    Choose a, based on $Q^A(s,\cdot)$ and $Q^B(s,\cdot)$, observe r, s'

    Choose (e.g. random) either UPDATE(A) or UPDATE(B)

    if UPDATE(A) **then**

        Define $a^* = \arg\max_a Q^A(s',a)$

        $Q^A(s,a) \leftarrow Q^A(s,a) + \alpha(s,a)\left(r + \boxed{\gamma Q^B(s',a^*)} - Q^A(s,a)\right)$

    else if UPDATE(B) **then**

        Define $b^* = \arg\max_a Q^B(s',a)$

        $Q^B(s,a) \leftarrow Q^B(s,a) + \alpha(s,a)(r + \boxed{\gamma Q^A(s',b^*)} - Q^B(s,a))$

    end if

    $s \leftarrow s'$

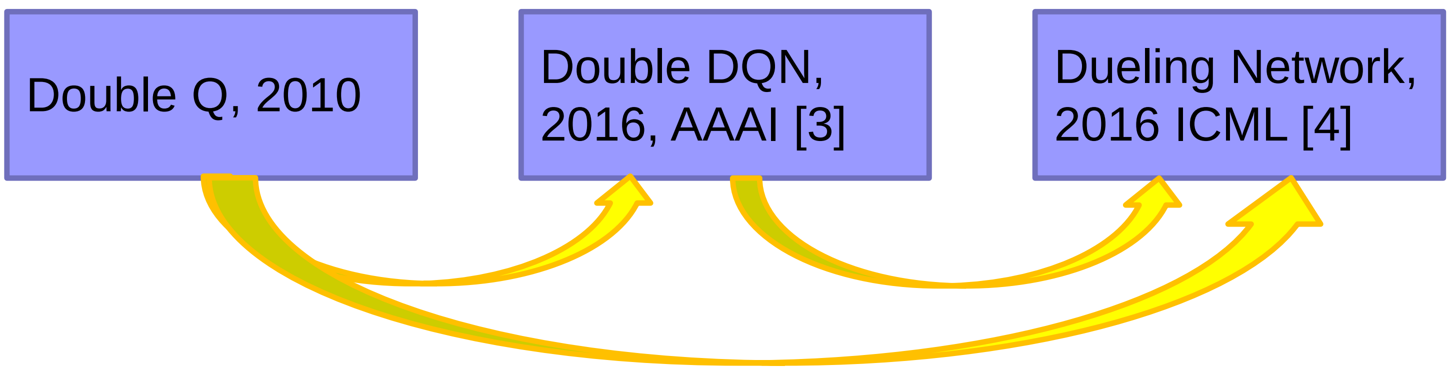
until end

Related work

- Potential work mentioned in the paper:
  - Whether we can an unbiased off-policy RL algorithm without high variance
  - Implement double q learning idea for other Q learning extension algorithms: Delayed Q, Fitted Q Iteration
  - Extending the double learning concepts: Double Sarsa, and Double Expected Sarsa [2].

Related work

- Following work of double Q learning: (Core idea: Two estimators, Asymmetric updating)



Related work: Double DQN[2]

- Extend form tabular to large scale:
  - The idea behind the Double Q-learning algorithm, can be generalized to work with **large-scale function approximation**.
  - Evaluate the greedy policy according to the online network, and using the target network to estimate its value.
  - Shows better performance in “Game Playing”

Related work: Dueling Network[3]

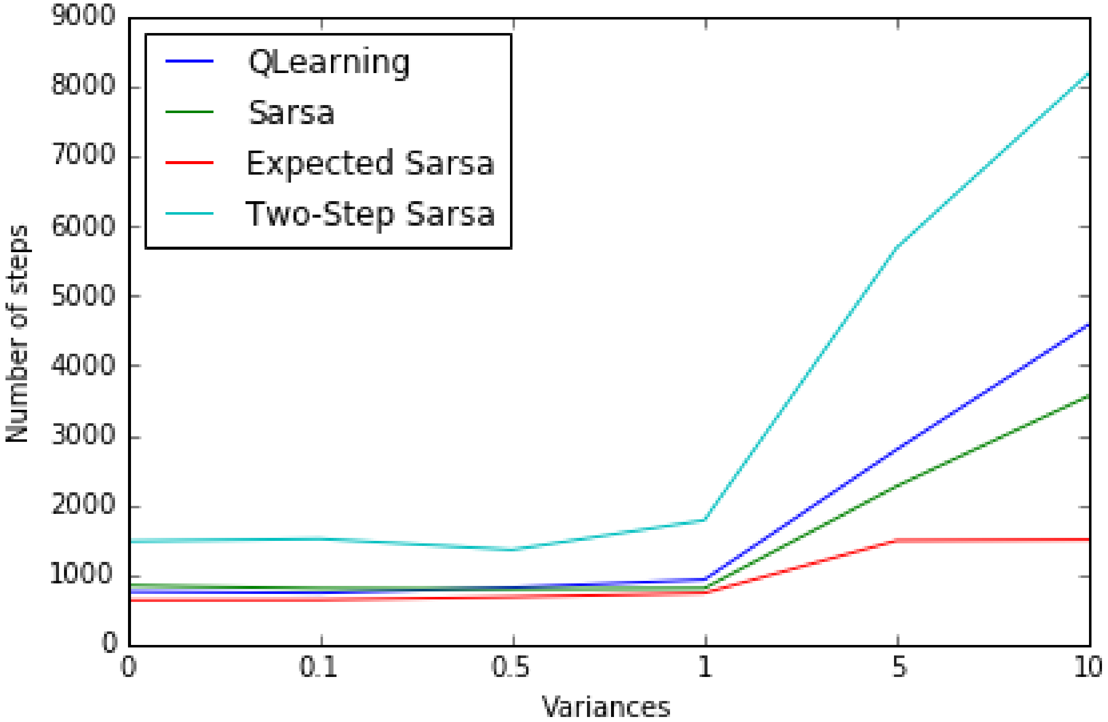
- Following work of double Q learning:
  - Two separate estimators one for the state value function and one for the state-dependent action advantage function.
  - The two streams are **sharing a common convolutional feature learning module**.

Simulation Results

- I use the implementation of Weiwei Zhang's double q leaning, q learning, Sarsa as baselines. Use the same settings of simulation on CliffWalking.
- Provide implementations for: Double Sarsa, Expected Sarsa, Double Expected Sarsa, Two Step Sarsa.

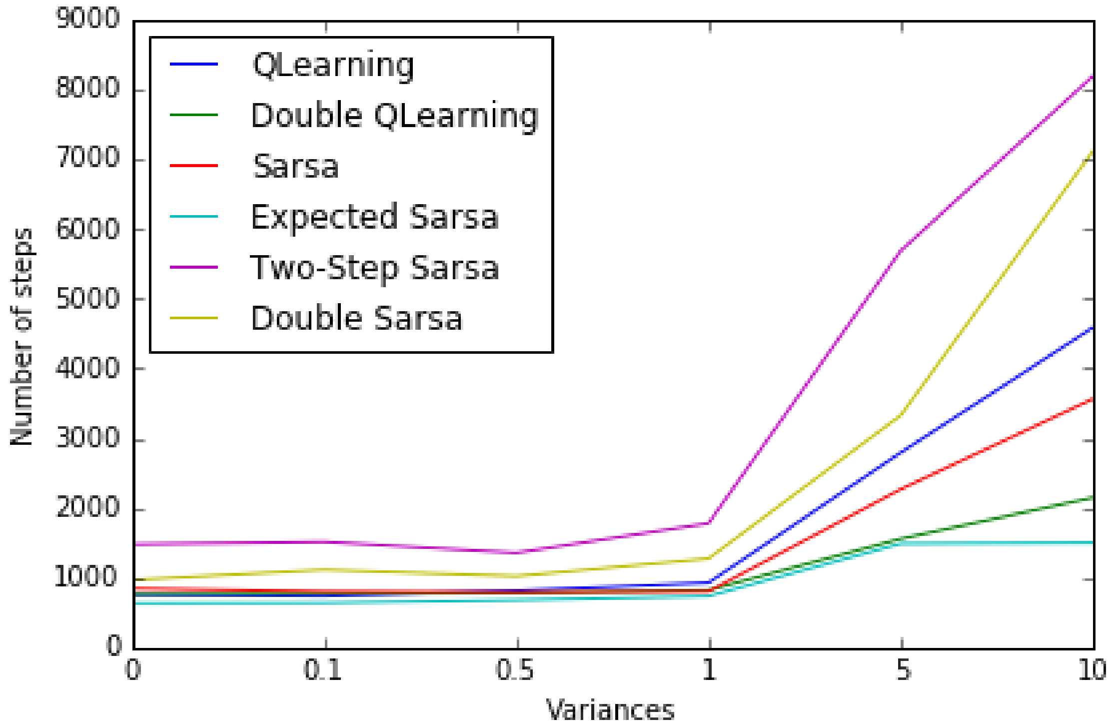
Simulation Results

- Comparison Q, Sarsa, Expected Sarsa, and Two-step Sarsa



Simulation Results

- Comparison for all the algorithms



Reference

- [1] Hasselt H V. Double Q-learning[C], Advances in Neural Information Processing Systems. 2010: 2613-2621.
- [2] Ganger M, Duryea E, Hu W. Double Sarsa and Double Expected Sarsa with Shallow and Deep Learning[J]. Journal of Data Analysis and Information Processing, 2016, 4(04): 159.
- [3] Van Hasselt H, Guez A, Silver D. Deep Reinforcement Learning with Double Q-Learning[C], AAAI. 2016: 2094-2100.
- [4] Wang Z, Schaul T, Hessel M, et al. Dueling network architectures for deep reinforcement learning[J]. arXiv preprint arXiv:1511.06581, 2015.