Temporal-Difference Value Estimation via Uncertainty-Guided Soft Updates

Litian Liang, Yaosheng Xu, Stephen McAleer, Dailin Hu, Alexander Ihler, Pieter Abbeel, Roy Fox

University of California, Irvine

NEURAL INFORMATION PROCESSING SYSTEMS

Introduction

In this work, we estimate β by maintaining an ensemble of Q functions to approximate the uncertainty in Q. This estimated β provides a principled temperature schedule that is "softer" when uncertainty is high but gradually becomes "harder" as training proceeds. We experiment in Atari environments to show the effectiveness of our method in practical discrete action space environments, and find that our method outperforms the Rainbow DQN (Hessel et al., 2018) and PPO (Schulman et al., 2017) algorithms in most Atari environments.

Contributions:

- We provide a principled numerical method for estimating a β term that makes the maximum-entropy target value approximately unbiased in SQL.
- We provide extensive experimental results demonstrating that reducing bias in SQL via our estimated β improves performance.
- We provide a proof of convergence of our method.

Algorithm: Unbiased Soft Q-Learning (UQL)

Search for beta by executing a numerical binary search on f. We use an ensemble to represent a collections of Q. This is an extension of the method EQL introduced in Fox. (2019) to discrete multi-action MDP.

$$f_{s;Q}(\beta) = \hat{\mathbb{E}}_Q\left[\frac{1}{\beta}\log \mathbb{E}_{a\sim\pi_0(a|s)}\left[\exp(\beta Q(s,a))\right]\right] - \max_a \hat{\mathbb{E}}_Q\left[Q(s,a)\right]$$

Bellman operator B, given any temperature w, is a contraction in L infinity. The tabular version is provably convergent (detail in Appendix A of paper).

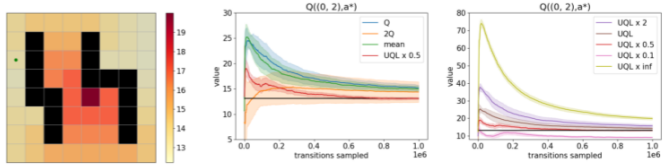
$$\mathcal{B}_w[Q^{(i)}](s,a) \stackrel{\text{def}}{=} \mathbb{E}[r(s,a)] + \gamma \mathbb{E}_{s'|s,a\sim p}\left[w \log \mathbb{E}_{a\sim\pi_0(\cdot|s')}\left[\exp(Q^{(i)}(s',a')/w)\right]\right]$$

Unbiased Soft Update

$$Q^{(i)}(s,a) \leftarrow (1-\alpha_t)Q^{(i)}(s,a) + \alpha_t\left(r(s,a) + \frac{\gamma}{\kappa\beta_Q(s)}\log \mathbb{E}_{a\sim\pi_0(\cdot|s')}\left[\exp(\kappa\beta_Q(s)Q^{(i)}(s',a'))\right]\right)$$

βQ(s) is the searched β from f. Kappa is the correction constant for accounting overestimation of a finite sized ensemble. Kappa is a easier and more interpretable hyperparameter than temperature learning rate for example.

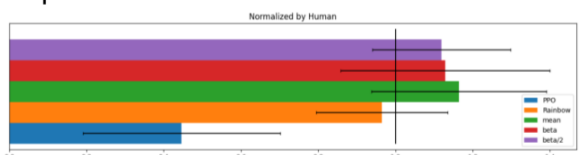
Experiments: Gridworld



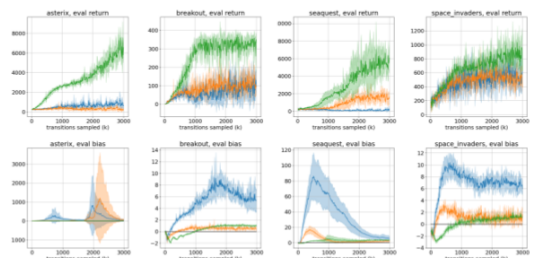
Left: Ground truth value
Middle: Value estimation of different methods and ground truth (black)
Right: Value estimation of different correction constant

Tabular UQL is able to converge to the ground truth value with the exact same training procedure faster than Q and Double Q.

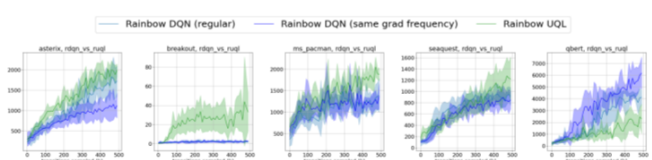
Experiments: Atari



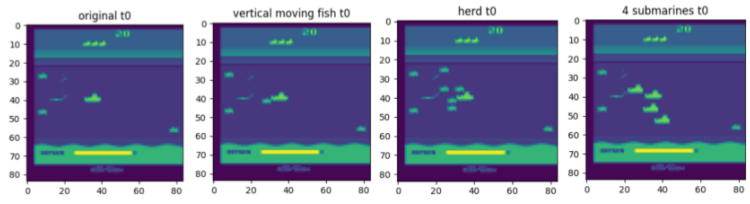
averaged human normalized score at 500k interactions



UQL with no rainbow tricks is able to outperform DQN and Double DQN



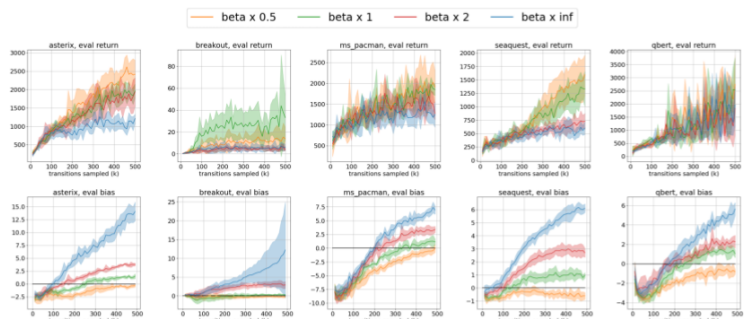
UQL with rainbow tricks added is more sample efficient than Rainbow (regular) and Rainbow with K times gradient updates (K is the ensemble size)



(a) $\log w = -5.2$ (b) $\log w = -5.1$ (c) $\log w = -4.1$ (d) $\log w = -1.7$

Figure 5: Awareness of novel state through unbiased temperature scheduling. $w = \frac{1}{\beta_{Q_\theta}(s)}$.

Trained ensemble is able to predict temperature given the data that it is trained on. This helps the softness of update to be flexible and helps with performance. The predicted temperature increases (from left to right) with the insanity of artificial perturbation to the state observation. This can also be extended to other use cases.



- Bias is monotonic in beta.
- A correction constant should be chosen within (0,1).

Conclusions

- UQL gives a tractable numerical solution to find the temperature of the maximum-entropy target
- Less estimation bias does not necessarily lead to better performance but gives insights and helps performance empirically
- Further separating the introduced methods should give more insights about the source of improvement

References

Roy Fox (2019). "Toward provably unbiased temporal-difference value estimation." In Optimization Foundations for Reinforcement Learning workshop (OPTRL @ NeurIPS), 2019.

Matteo Hessel (2018). "Combining improvements in deep reinforcement learning.". In: Thirty-second AAAI conference on artificial intelligence, 2018.

John Schulman (2013). "Proximal policy optimization algorithms". In: arXiv preprint arXiv:1707.06347, 2017.