A Comparison of Value Function Approximation Approaches for Continuous State Space Control Problem in Reinforcement Learning

Zhiwei Han, Dominik Meyer and Hao Shen
Chair of Data Processing
Faculty of Electrical Engineering and Information Technology
Technical University of Munich
Arcisstr. 21, Munich, 80333
Email: hanzw356255531@icloud.com

***Abstract*—In this paper, a comparison of three value function approximation based reinforcement learning algorithms is presented to evaluate their performance on solving continuous state space control problem. They are GQ(λ), $\ell 1$ regularized off-policy TD(λ) and online selective kernel-based TD(λ). For convenience, these algorithms will be called GQ(λ), RO-TD(λ) and OSK-TD(λ) in rest of the paper. The comparison concentrates mainly on the total reward, convergence and efficiency of the algorithms. The classic TD(λ) framework Q-Learning is applied and adapted into new algorithms in parameter learning. Note, other RL framework choice like Sarsa(λ) could also be possible, but won't be discussed here. GQ(λ), which minimize the mean-squre projected bellman error(MSPBE) with gradient descend, is a very powerful and straightforward learning method under small scale or well-defined discrete space setting condition. However, because of the curve of dimensionality, the algorithm could be less computational efficient as the increasing number of state. To overcome this issue, one $\ell 1$ regularity term was inserted to the objective function in RO-GQ(λ) algorithm. The sparsity generated in the learned parameters helps significantly reduce the computational complexity of RO-GQ(λ) and therefore it has a better performance in these problems. Alternatively, OSK-TD(λ) provides another solution for sparsification, which is less computational complex and can be runned online. OSK-TD includes two preocedures, a dictionary consists of feature vector is online generated and refreshed with a criteria for whether and how to add a new feature in dictionary. In the second step, a kernel selective function β(s) selects the best matched feature vector in dictionary for the selective kernel-based Q-function representation.**

***Index Terms*—reinforcement learning, Q-Learning, function approximation, sparsification, slective kernel-based value function**

I. Introduction

Reinforecement learning (RL) approaches to the classic predict and control problems base on dynamic programming (DP) and markov decision process (MDP) model, which proofed to be useful but computational expensive. Sutton and Barto [1] proposed temporal difference (TD) learning framework and its variations as the extension of DP. One of the nice properties of TD learning is its combination with value function approximation and it made the algorithm possible to learn from complicated learning tasks.

But there are still two problems about value function approximation mainteined unsovled. In practical large scale problems, smart value function approximation strategy is alwags required to reduce the computational complexity because large scale RL problems e.g. Go within continous/action spaces suffer from the curse of dimensionality, which means the computational complexity increases exponentially with growing number of states and actions. This problem arises as a main subproblem in RL and is generally considered as the most important step in RL algorithm development. There are serveral value function approximation techniques developed in recent researches, e.g. linear function approximation [1][2], regression tree[3] and kernel-method $[Kernel-basedleastsquarepolicyiteration]$, $[kernel-basedreinforcementlearning]$, which can deal with a specific set of problems. More generally, after deep learning $[deeplearning]$ concept and deep neuro network were introduced into RL, the perforemance has been significantly improved since Alpha Go from Deep Mind, a company owned by Google, beated one of the best human go players, Lee Sedong. Furthermore, deep Q-network (DQN)[Human-level control through deep reinforcement learning] based Q function Approximator was also turned out to be smarter than human player in some Atri-game playing. Regularization plays not only an important role in computational complexity reduction $[RegularizationandFeatureSelectionin]$ and it is also considered as an ideal feature selection technique for traditional tabular algorithmthanks to the sparse property of $\ell 1$ norm $[RegularizedOff-policylearning]$. In this paper, regularization is also applied an the tabular algorithm RO-TD(λ) to compare with the algorithm OSK-TD(λ), which equipped with another feature selection strategy, in term of the efficency of sparsification.

Another widely concerned issure is convergence of the value function approximation. As Tsitsiklis$[AnAnalysisofTDlearning]$ pointed out that even though the linear value function approximation converges in most online learning problem setting, but an algorithm could still diverge if either of the conditions is

not satisfied, when states are sampled from distributions independent of the dynamics of the Markov chain of interst, we will this point of view later in the experiment. It's a servere problem in RL application, however, because of the great needs, value function approximation was still applied in many off-line TD algorithms like Q-learning at that time while ignoring the potential risk of divergent value function approximation. Sutton[$AconvergentO(n)$] [$fastgradientdescendmethodforTDlearning$] solved off-line learning problem of off-policy learning by introducing a new object function, mean-squre projected bellman error (MSPBE) and performing stochastic gradient mehtod an it.

In this paper, we consider up-to-date algorithms GQ(λ), RO-GQ(λ) [$RegularizedOff-policylearning$] based on the traditional tabular algorithm framework (without value function approximation), kernel-based feature selective algorithm OSK-TD(λ) [$online-selectivekernel-based..$] and their comparison. The rest of this paper is organized as flollows. In Section II, an introduction on MDPs as well as the TD learning framework is given. An algorithmetic comparison in perspect of objective function, sparsification between three algorithm is presented in the section III. In section IV, the experimental result on the classic RL problem settings Mountain Car and Carto Car are provided to visulization the effectiveness of compared algorithm. At the V section gives a conclussion and future work.

II. RL and Background

A. Framework and Notation

In this paper, we consider infinite markov decision process with discrete state space $\mathcal{S}$ and discrete action space $\mathcal{A}$. A MDP can be defined as tuple ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{P}$, r, γ), where $\mathcal{P}$ represents the transition probability matrix and it is also defined as the one-step dynamics of enviroment by given state and action. [$Introtorl$] Given any state and action, s and a, the probability of each possible state transition to next state s', is a function $Pr : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \mapsto [0, 1]$

$$p(s'|s, a) = Pr\{S_{t+1} = s'|S_t = s, A_t = a\} \tag{1}$$

Similarly, with any current state and action, s and a, and after taking action a the agent reaches the next state s', the expected value of the next reward of tuple (s, a, s') is a function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \mapsto \mathbb{R}$ and $\gamma \in (0, 1)$ is the discount factor.

$$r(s, s', a) = \mathbb{E}[R_{t+1}|S_t = s, A_t = a, S_{t+1} = s'] \tag{2}$$

In RL, main task is either directly to find a convergent value function through like dynamic programming etc., or indirectly approximate a value function (action-value function) through linear value function approximation etc.. A value function maps from each state to a number representation of value function obtained if a policy is exactly followed. Policy can also be analysed and evaluated according its sum of discounted value function. More formally,

$$V^{\pi}(s) = \mathbb{E}[\sum_{t=0}^{\inf} \gamma^t r_{t+1}] \tag{3}$$

$$P^{\pi}(s, s') = \sum_{a \in \mathcal{A}} \pi(s, a) P(s'|s, a) \tag{4}$$

$$R^{\pi}(s) = \sum_{a \in \mathcal{A}} \sum_{s' \in \mathcal{S}} \pi(s, a) p(s'|s, a) r(s, a, s') \tag{5}$$

If V^{π}, R^{π} are vectorized with length $|\mathcal{S}|$ and the dimension of P^{π} is $|\mathcal{S}| \times |\mathcal{S}|$, so the famous $Bellman\ Equation$ can be written in form,

$$V^{\pi} = TV^{\pi} = R^{\pi} + \gamma P^{\pi} V^{\pi} \tag{6}$$

, where T is the bellman operator and V^{π} is the fix point of bellman operation.

In rest of this paper, value function is replaced of action-value function, which maps each state and action to a number representation instead on only state, as we mainly focus on control problem.

B. Tabular Q-Learning(λ)

The basic idea of temporal difference learning is looking back to the previous state, and correcting the previous value or action-value function with experience (Bellman Error). The update of value function is defined as following,

$$V(s_t) = V(s_t) + \alpha e(s_t) \tag{7}$$

, where α is the learning rate and $e(s_t)$ is the bellman error, which defined as

$$e(s_t) = R_{t+1} + \gamma V(s_{t+1}) - V(s_t) \tag{8}$$

As one of the most important extension of TD algorithm, the off-policy learning procedure Q-Learning, whose update form is,

$$\begin{aligned} Q(s_t, a_t) = Q(s_t, a_t) + \alpha[R_{t+1} + \\ \gamma \max_{a \in A} \{Q(s_{t+1}, a) - Q(s_t, a_t)\}] \end{aligned} \tag{9}$$

, directly approximates q^*, the optimal action-value function, independent of the policy being followed. Based on the TD method, eligibility keeps track on the several prior state and how the action-value function should be updated when a new observation is revceived. More specifically, eligibility is a temporal record of occured events e.g. states or actions, then propagates the attenuated TD error back and finally updates the parameters of the previous states. Combined with eligibility trace, In tabular Q(λ), which also used as the main framewok of this paper,

C. TD(λ) with feature dictionary

D. Objective Function

E. Regularization

III. GQ(λ)

IV. RO-GQ(λ)

V. OSK-TD(λ)

VI. Conclusion

Appendix A
Proof of the First Zonklar Equation

Acknowledgment

The authors would like to thank...

REFERENCES

[1] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press Cambridge, 1998, vol. 1, no. 1.

[2] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE transactions on automatic control*, vol. 42, no. 5, pp. 674–690, 1997.

[3] D. Ernst, P. Geurts, and L. Wehenkel, "Tree-based batch mode reinforcement learning," *Journal of Machine Learning Research*, vol. 6, no. Apr, pp. 503–556, 2005.