CS 7642 Reinforcement Learning and Decision Making (Project 1): Exploration of Temporal Difference Algorithm with Bounded Random Walk Experiment

Qisen Cheng

Computer Science Department, Georgia Institute of Technology,
Atlanta, Georgia, USA 30318, qcheng35@gatech.edu

Abstract. This work, as part of the course requirement of CS7642, explores the setup, convergence and learning rate of Temporal Difference (TD) algorithm, by reproducing the bounded random walk experiments. TD is a classical reinforcement learning algorithm that has been extensively developed in the last three decades. It has advantages over traditional learning methods in terms of less dynamic memory requirement, easiness of computation, and efficiency of using limited experience data. Bounded random walk, as an interesting experiment included in famous Sutton 88' paper [1] on TD methods, reveals many important properties of the algorithm. Specifically, it shows the on-line and incremental nature of TD method, and discusses the effect of recency parameter (λ) and learning rate (α) on the prediction performance.

Keywords: CS7642, reinforcement learning, Temporal Difference (TD), unbounded random walk, convergence, lambda, learning rate

1 Introduction

The major difference between temporal difference and traditional supervised-learning paradigm is that TD increments the prediction difference step by step and gives adjustable emphasis on the prediction of recent steps. The TD paradigm can be described using following equations [1].

$$w \leftarrow w + \sum_{t=1}^{m} \Delta w_t \tag{1}$$

$$\Delta w_t = \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_k \tag{2}$$

where P_t is the prediction at step t; w is the weight vector in the prediction function of observation vector acquired at certain state. By incrementally updating the weight vector after each observation step with appropriate learning rate α, it is guaranteed that the weight vector can converge close to the optimal value that gives accurate prediction at each state. In practice, a more traceable way to update the weight vector is sequential updating, which updates the

weight vector only after completion of a sequence of steps (e.g. after termination) rather than each step. This makes the updating process more controllable, such that more emphasis can be given to recent steps to reduce the sensitivity of error in long sequences. The recency emphasis is expressed with parameter λ. Different λ generally leads to different properties of convergence and learning results in certain task, thus a group of algorithms differing in choice of λ is named TD(λ), extended from the original TD method (or called TD(1)). The update direction is controlled by the term $\nabla_w P_k$, which is essentially the partial derivative of the prediction function at certain step about weight vector. The updating rate (or learning rate) is determined by α.

1.1 Bounded Random Walk

Example of bounded random walk (BRW) gives a simple but informative setup for exploration of TD(λ) algorithms. BRW can be depicted as following (Fig. 1): 1) a finite number of states is given from A to G, in which A and G are two terminating states; 2) steps are taken to the left or right randomly starting from the central state D; 3) the prediction at each state is the probability of ending the game at the rightmost state G. In this problem, accurate probability value of each state can be found through theoretical analysis, and taken as reference for evaluation of TD(λ) methods.

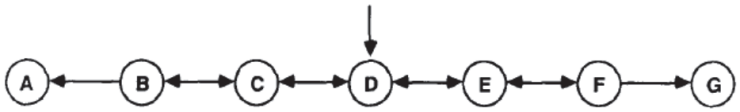


Fig. 1: Illustration of bounded random walk example in Sutton 88' [1].

The goal of the BRW experiment is to explore TD(λ) algorithms in terms of accuracy and learning rate with different λ. The full experiment has two parts. In experiment one, the focus is the effect of λ on the accuracy of prediction. It is assumed that 10 random sequences as a training set of data is given for training, and the weight vector is updated only after every training set repeatedly until convergence. Prediction error can be computed for each converged prediction corresponding to different λ. Then the comparison between errors empirically shows the difference of prediction accuracy between different choice of λ.

Experiment two focuses on learning rate of TD(λ). In this part, one random sequence of walk can only be presented once to the algorithm, but the weight vector is updated right after each random sequence as in equation (1). Since only a limited number (=10) of sequences are used in training of TD method with different λ, only intermediate prediction results would be given for each choice of λ. Furthermore, the difference between these intermediate results empirically shows the different learning speed (rate) corresponding to different choice of λ.

One important assumption in BRW experiments is that observation vector at each state is simply a unit vector, which has element 1 at the position corresponding to each state and elements 0 at all the rest positions. Another assumption is that the prediction function is linear as $\mathrm{P_t} = \mathrm{w^T x}_t$. These two assumptions simplify the problem. So that the $\mathrm{i^{th}}$ component in the weight vector equals the prediction value of the $\mathrm{i^{th}}$ state.

The reproduced experiments and discussions are detailed in following sections 2 and 3, respectively.

2 Experiment One

This section shows the reproduced results of experiment one as in Sutton 88' [1]. The goal is to compare the accuracy of prediction between different choices of λ.

Implementation. 10 random sequences as a training set are generated for training. The sequences are different in number of steps, and terminating state (A or G). 100 training sets are constructed equally, and the comparison metric (error) is averaged over all the training sets for statistical reliability. The weight vector is updated only after each training set with the sum of all the step-wise updates. The update corresponding to certain training set is repeated until convergence of weight vector. The updating procedure is mathematically described as equation (2), in which the partial derivative part is simply unit observation vector x. The weight vector is initialized to be zero in all positions but one in the position corresponding to state G. All the TD(λ) algorithms use the same static learning rate parameter throughout the updating process. The comparison metric is rooted mean square error (RMSE) between the weight vector representing the values of G-ending probability of each state and the theoretical values given in the paper [1] – 0, 1/6, 1/3, 1/2, 2/3, 5/6 and 1 for states from A to G, respectively. Finally, the choice of λ for comparison is λ = 0, 0.1, 0.3, 0.5, 0.7, 0.9, 1. The reproduced result is shown in Fig. 2.

Challenges. The tricky parts in this experiment are the convergence criteria of weight vector and the determination of learning rate parameter (α). For the former item, it is not clarified in the paper [1], but superficially claimed as convergence of weight vector. In this reproduced work, the convergence criteria is assumed as the state-averaged Δw (weight change) between two adjacent updates is smaller than a pre-defined threshold ε = 0.0001.

Determine α is also very challenging in execution of the experiment. α has to be small enough but not too small for keeping balance between ability of convergence and updating efficiency. In this experiment, a static learning rate α = 0.002 is set for all the TD (λ) computations. This value is founded through repeatedly execution of the experiment.

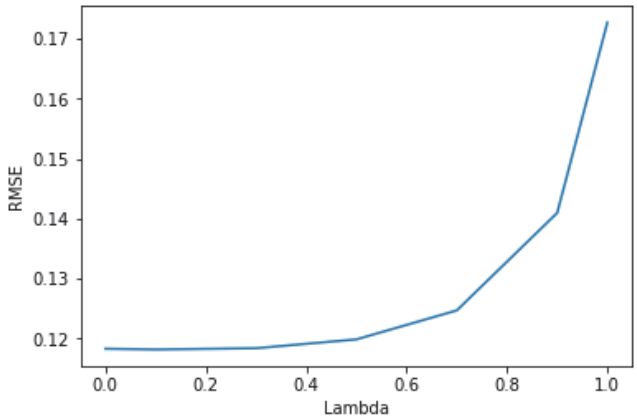


Fig. 2: Results of reproduced experiment one in [1]. The RMSE is minimized at λ = 0, and maximized at λ = 1.

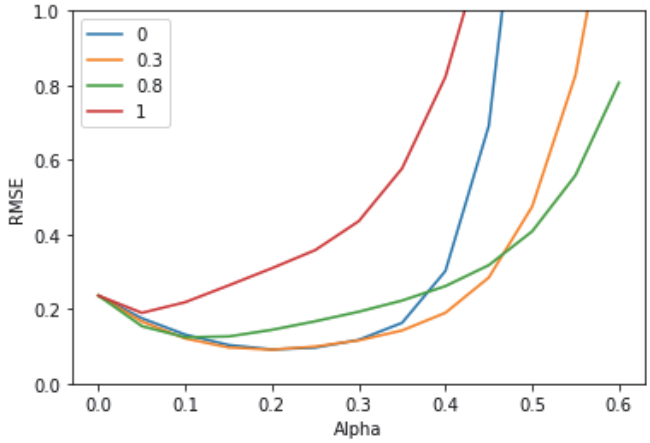
Analysis. The reproduced result has the same trend as Figure 3 in the paper [1]. However, the absolute value of RMSE for each λ is slightly smaller than the errors in the paper [1]. This difference might root in different generation of random sequences for training, different convergence criteria of weight updates, and different setting of learning rate. Besides, as stated in the paper [1], the calculated RMSE stay roughly at same values regardless of initial guess of weight vector, as long as the learning rate α is small enough for convergence.

3 Experiment Two

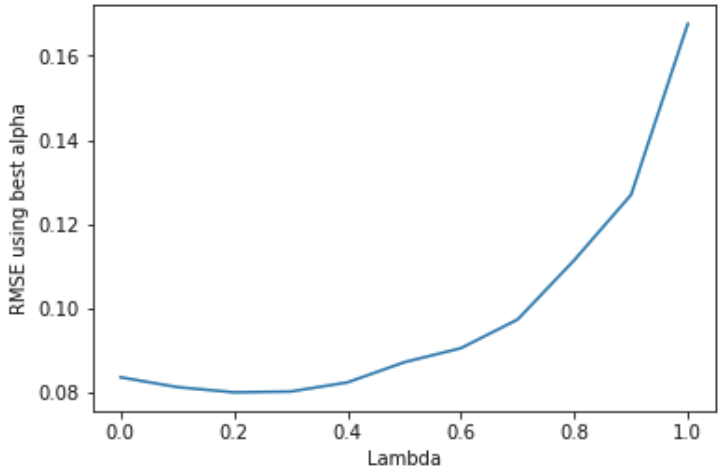
The reproduced results of experiment two in Sutton 88' [1] are detailed in this section. The second experiment is aimed to evaluate the learning speed (rate) between different choices of λ.

Implementation. The experimental setup is nearly the same as in experiment one. The major difference here is that single random sequence is only presented once in the training process. Furthermore, the updates on weight vector are made right after walking through each sequence, rather than after each entire training set. By comparing the error of intermediate training results after limited 10 sequences, we could empirically evaluate the learning speed (rate) of TD corresponding to different choice of λ. Again, to ensure the statistical reliability, the error (RMSE) is averaged over 100 equally constructed training sets. Finally, the choice of λ for comparison is from 0 to 1 with increment of 0.1. The α is swept from 0 to 1 with increment of 0.05. The reproduced result is shown in Fig. 3.

Challenges. The major challenge here is that errors of both TD(0) and TD(1), especially TD(1), blow up when α gets close to 0.6. As intuitively explained in experiment one, when α is getting larger, the updated weight vector might either diverge at certain point or oscillate around optimal value. So that it might result in large error in the intermediate stage of training. To validate the reproduced results, it needs to zoom into appropriate region for display of the curves. A second challenge associated with the first one is that, the plot given in paper [1] does not show any error of TD(1) with α larger than 0.35. This makes comparison a little bit difficult.



(a)



(b)

Fig. 3: Results of reproduced experiment two in [1]. (a) Comparison of RMSE between different λ with learning rate α swept from 0 to 0.6. (b) Comparison of RMS between different λ at corresponding best learning rate α. The best choice of λ is roughly 0.3 with learning rate $\alpha \approx 0.2\text{-}0.3$.

Analysis. The reproduced result has roughly the same trend as Figure 4, 5 in the paper [1]. However, in Fig. 3(a), the shape of the curves is slightly different compared to Figure 4 in the paper, in terms of curvature, bottom location and absolute error values. This is generally because the plot data just represent the intermediate training results after experiencing 10 sequences for each λ. Thus it is not guaranteed the error value is converged close to some certain amount for every execution of experiment. In fact, the errors are considerably different in each execution, though the trend of errors are pretty the same as in Fig. 3(a).

Fig. 3(b) pretty much matches the result in Figure 5 of Sutton's paper [1]. The best choice of λ is around 0.3. This shows TD(0) is not the "fastest-learning" algorithm due to relatively slow propagation of updates.

References

1. Sutton, R. S. Learning to predict by the method of temporal differences. Machine Learning. 3, 9–44 (1988).