TD(λ)

Zhen Zhou
zzhou602@gatech.edu

1 INTRODUCTION

This paper aims to reproduce the TD(λ) model proposed in Richard Sutton's 1988 paper "Learning to Predict by the Methods of Temporal Differences." I create an implementation and replication of the results found in figures 3, 4, and 5.

The TD(λ) model can be used as a prediction method. The update equation over training set is

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

Δw_t is updated over sequences, which is

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

Where α and λ are hyper-parameters.

The code is in

https://github.gatech.edu/gt-omscs-rldm/7642Sprint2023zzhou602/tree/master/Project_1

2 RANDOM WALK

A bounded random walk game is constructed to test the performance of TD(λ). The rules of bounded random walk game are:

(1) The game has 7 states, which are [A, B, C, D, E, F, G].

(2) The begin state is D, and the terminate state is A and G.

(3) The probability of go right or left for non-terminated state is equal, which is 50%.

(4) If target arrives at G, the reward is 1, otherwise it is 0.

(5) The goal of this game is to find the probability of a right-side termination, which means the probability of [B, C, D, E, F] to arrive at right-side termination. We can calculate the actual probability, which is [1/6, 1/3, 1/2, 2/3, 5/6].

(6) The basic flow chart of bounded random walk game is shown in figure 1.

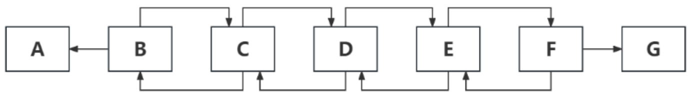

Figure 1 The basic flow chart of bounded random walk game

(7) Experiment 1 and experiment 2 are all conducted over 100 training sets and 10 sequences for every training set.

(8) The metric used in these experiments are RMSE.

3 EXPERIMENT 1

In experiment 1, the hyperparameter alpha is set to be 0.01, and I select 11 different lambdas, which are [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]. The Δw's are accumulated over repeated sequences and finally update the weight vector after the $\triangle w$ is smaller than a threshold $\gamma = 0.001$. The result is shown in Figure 2.

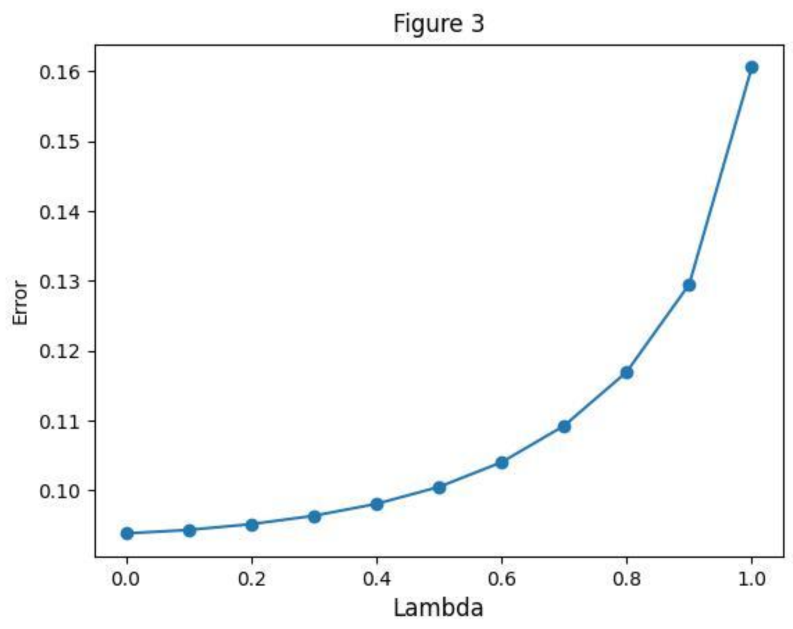

Figure 2 Replication of figure 3 in Sutton's paper

In Figure 2, RMSE increases as the value of lambda increases. the lowest RMSE is obtained by lambda=0, which is 0.092. As we can noticed, the RMSE of different lambdas are lower than Sutton's paper. Because of the fact that Sutton's paper was published in 1988, in the 1980s, computers had very poor computational power and use 16 bit and 32 bit processors. This can explain the higher performance in my experiment.

4 EXPERIMENT 2

In experiment 2, lambda is set to be [0, 0.3, 0.8, 1]. Alpha is set be [0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5]. The Δw's are accumulated over repeated sequences and update the weight vector after each sequence. The RMSEs are calculated over each sequence and then calculate the mean of sequences for each training set. The result is shown in Figure 3.

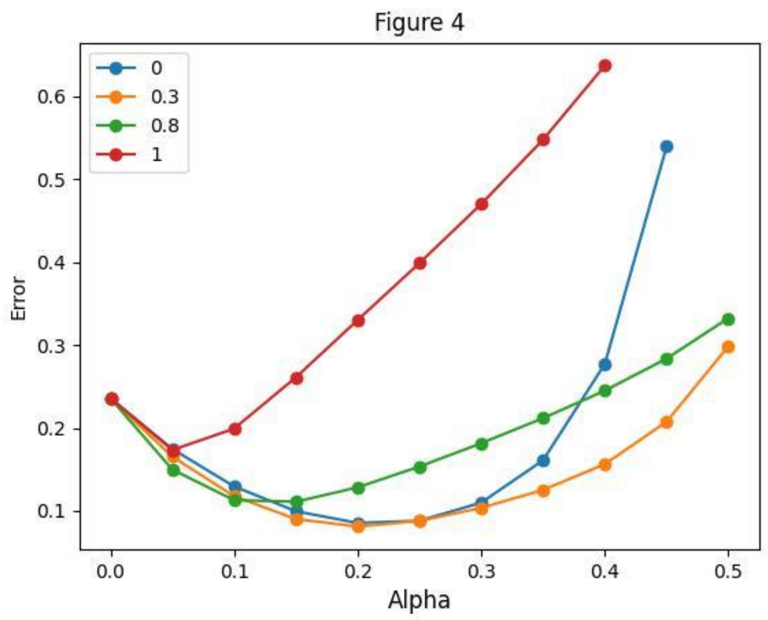

Figure 3 Replication of figure 4 in Sutton’s paper

Figure 3 above is very similar to Sutton’s paper. We can found that the only difference with Sutton’s paper is the line of lambda=0. In my experiments, the line grows exponentially after alpha=0.3, but in Sutton’s paper, the line grows exponentially after alpha=0.4.

In order to further explore the relationship between lambda and RMSE, I use the same procedure in experiment 2, but calculate the minimum RMSE of lambda over different alpha. Alphas are chosen from linspace(0,0.5,21) and lambdas are [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1].

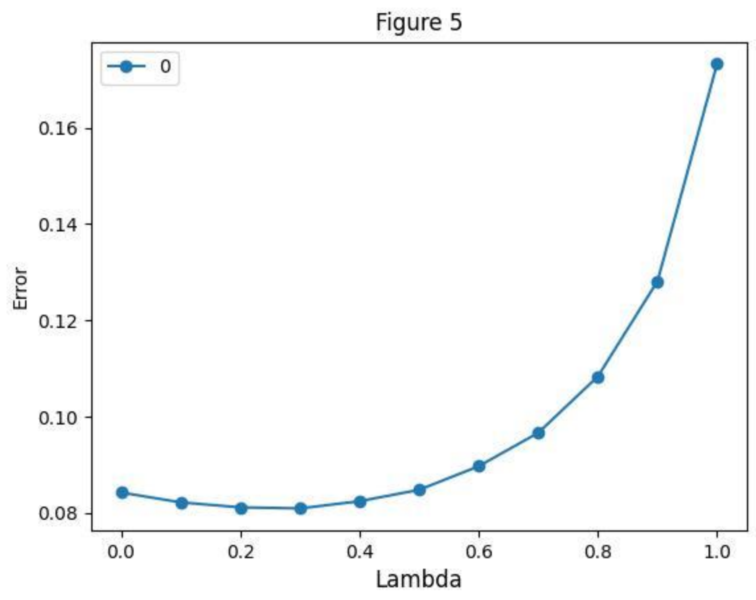

Figure 4 Replication of figure 5 in Sutton’s paper

Similar to Sutton’s figure 5, lambda obtain its minimum point 0.08 around lambda=0.3. When lambda=1, RMSE is around 0.17 and when lambda=0, RMSE is around 0.085. The performance of my experiment is better than Sutton’s paper, this can also be attributed to the computer's computational power which has been explained in section 3.