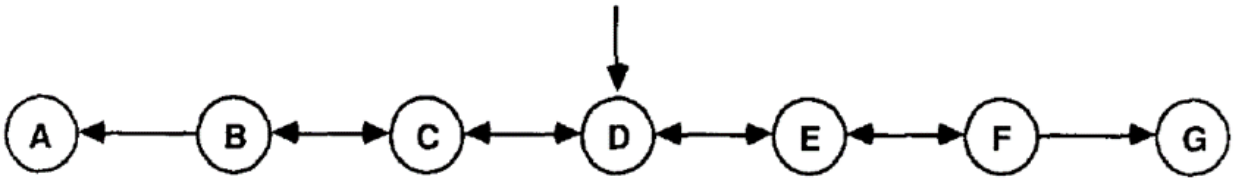
Project 1: TD(λ)

Trung Pham – tpham328@gatech.edu

Abstract – Sutton discussion about Temporal Difference Learning efficiency and less memory requirement compared to conventional method of supervised learning. In his paper, he conducted a Random Walk experiment and used TD learning to point out its superior impact. This report tries to replicate the experiment done by Sutton using TD(λ) using a generated data. It also emphasizes the dependency of the learning model with different parameter like λ and the learning rate.

Introduction

A. Describe Problem – Random Walk



In this problem, there are 7 states line up in a row from A to G alphabetically (each letter represent a state) and a person can only move left or right to transition between two adjacent state with equal probability of left and right (50% chance move left and 50% chance move right). The person started at the center state D and the game ends when the person enters either state A or G. At A, the reward z equal 0, while at state G, the reward z equal 1. Non-terminal states are B,C,D,E,F are represented as a column identity vector of dimension (5x1). For example $X_D = (0,0,1,0,0)^T$. A and G is not represented because game terminated at those state so only reward is recorded. A sequence record the state that a person move until reaching terminated state and ends with the corresponding reward. For example, X_D, X_C, X_D, X_E, X_F,1, represents the person move from D to C to D to E to F to G and received reward = 1 as the game ended. The sequence vector for this sequence is recorded as:

0 0 0 0 0
0 1 0 0 0
1 0 1 0 0
0 0 0 1 0
0 0 0 0 1

And end up with reward z = 1

B. Temporal Difference Learning

For each observation, the learner produces a corresponding sequence of prediction $P_1, P_2,\ldots P_m$ to estimate z. P_t is a function of X_t and modifiable parameter weight w. For the random walk problem, w is a vector 5x1.

$P_t = w^T X_t$

With α is the learning rate, the weight update is learned as

$\Delta w_t = \alpha (P_{t+1} – P_{t+}) * \Sigma \forall_w P_k$

The error term is calculated incrementally as:

$$e_{t+1} = \forall_w P_{t+1} + \lambda e_t$$

Experiment

Discussion of implementation

Widrow-Hoff supervised learning procedure is a special case of TD(λ) where λ =1. The ideal prediction for each of the nonterminal states can be computed as described in section 4.1 of Sutton's paper. It is (1/6,2/6,3/6,4/6,5/6,1) for B,C,D,E, and F respectively. The python script used for this project has tried to replicate this result it is $[(I - Q)^{-1} h]_i$
Where he is $(0,0,0,0,0.5)^T$, I is identity vector and Q is the transition vector.
The training data for this experiment is randomly generate based on 50% chance move left or right, starting at state D.
Discussion of generated data
There are 100 training data set, each consists of 10 sequences.
Set a list of interest lambda and alpha for the TD learning and iterate the learning based on those pre-set value. λ= [0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1]. For figure_3, we set learning rate α = 0.2. For figure 4 and 5, α list is [0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6]. Starting arbitrary weight is [0.5,0.5,0.5,0.5,0.5] because starting arbitrarily weight as extreme value produce no good result.
For the first experiment, the weight vector was not updated after each sequence. It is accumulated over 10 sequences and updated after completion of each training set. Each training data set was presented repeatedly until weight vector converged. For both experiment, we defined convergence as the weight vector change less or equal than epsilon = 0.01. The error is root mean square of the predicted weight and the ideal weight.
For the second experiment, each training set is only presented once. But instead of accumulating change in weight, we update the weight after each sequence is completed. We also used a range of learning rate as defined above instead of 0.2 for figure 3. The second experiment is presented in figure 4 and 5 graph.

First experiment Result (outcome, how it is different, etc)

Result for experiment 1 is presented in figure 3 graph. TD(1) or Widrow-Hoff which is supposed to be optimal method produce the worst than all TD method for λ < 1. One of the reason mentioned in Sutton's paper is that Widrow-Hoff only minimize error on the training set, it does not necessarily minimize error for future experience. Supervised learning also tends to be more overfit.

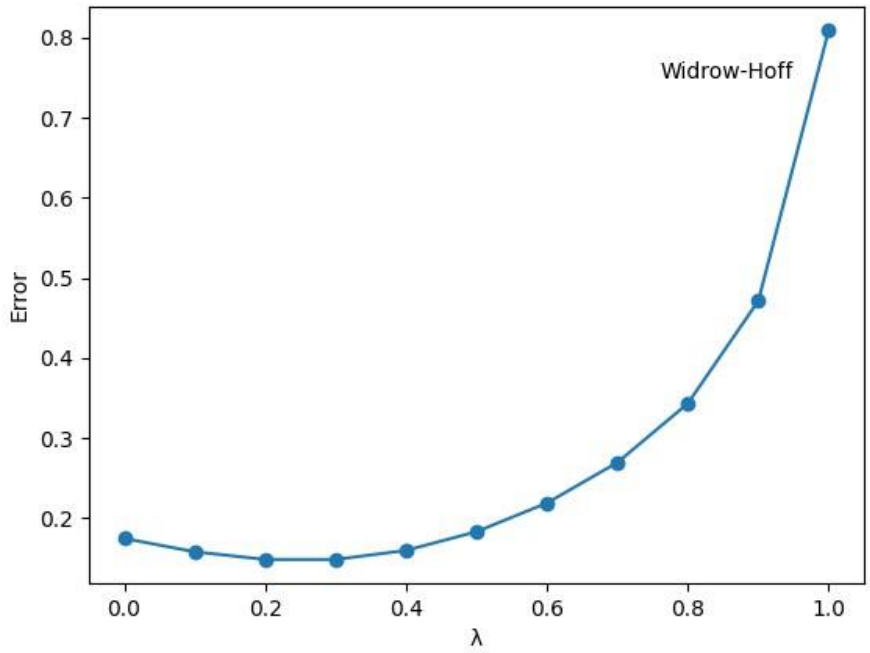


Figure 3 – alpha = 0.2. The RMS error is averaged over 100 training data set.

We are able to produce a graph that very similar to figure 3 from Suttons' paper. After testing over many learning rate alpha, it is showed that alpha around 0.1-0.3 produce a match graph. Extreme value of learning rate like 0.05 produce non-match graph. Although the value of error is not matched, this is because Sutton does not specify his data set, initial weight value or learning rate, so it is not possible to produce matching in value, but the shape of the graph is almost exact match.

Second experiment Result

For the second experiment, a pair-wise value of λ and α is used, with each training set is presented only once not infinitely so the computation time is less than experiment 1, but more because of more learning rate is tested. Similar to experiment 1, the Widrow-Hoff TD(1) produce worst result, error is higher than any other TD learners with λ<1. The learning rate α significantly impact the error or the learners' performance. Alpha = 0.2 produce better result than most in figure 4 graph. Also λ = 0. 3 produces best result not λ = 0. The reason for this is that TD(0) is relatively slow at propagating prediction levels back along a sequence. Since training data sets are presented only once instead of unlimited, this handicapped TD(0).

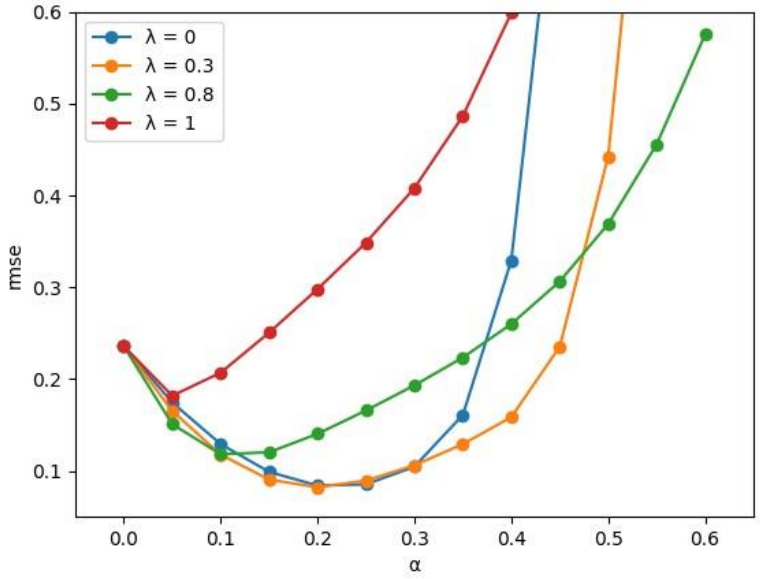
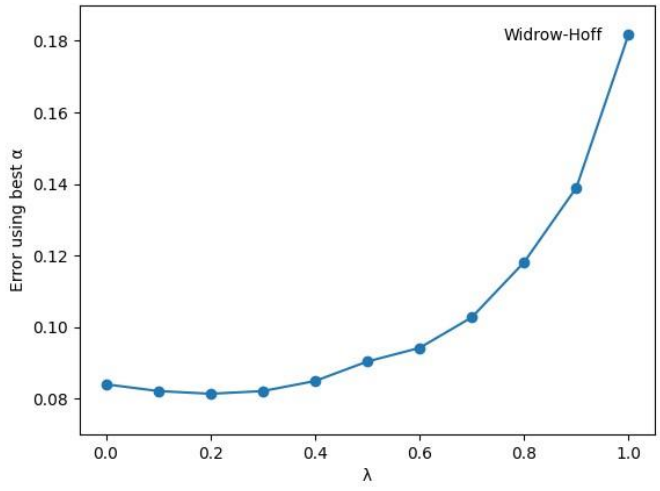


Figure 4: Average error on random walk after experiencing 10 sequences.

Our figure 4 looks very similar to Suttons' graph. One of the difference is the value of error but similar to figure 3, this could be because of different training data used or different initial weight. Another different is the λ = 0.3 is not always the best. It start going bad after alpha > 0.5. This indicates how important the selection of learning rate in TD learning procedure. The difference here could be because of randomness data, where learning rate in my experiment impact the result more than the learning rate in Sutton's experiment. Our experiment indicate that alpha within 0.15-0.25 produce the best result. This is also the reason we set alpha 0.2 in figure 3. When we set the best learning rate, figure 5 show exact match with Sutton's.



In figure 5, we presented the best error achieve in each lambda value. This produced the best graph for error. Learning rate in the range of 0.1-0.4 generally produce the best result here. The best lambda for TD learner is either 0.2 or 0.3. Similar to other graph, λ = 1 produced the worst result. The figure 5 we produced match almost exact as Sutton's except the error value. Like mentioned above, probably because of randomness.

Drawback – What could be better.

Overall, our replication of Suttons' experiment produced very similar results and conclusion to his paper. The only difference is the value of error. If we can have access of how he generates his data or how he collected his data, we might produced a similar error value. But overall the shape of the graph should not change much. Sutton also do not specify how he choose learning parameter and other arbitrary value he set. Knowing this could also help us produce a more matching result.

Not knowing his assumptions or his data, we have to try to generate many data sets with different seeds value, where we see a while range of error value. Our results also not produce good result when we try to select extreme value for our parameter.

Conclusion

When we conduct the Sutton's experiment for Random Walk problem. We realized the important of selecting a learning rate alpha as well as initial starting weight values. We also see how different training set can result in different error range. We also confirmed his points of how TD learning can be more effective and efficient compared to conventional supervised learning method. In general, we produce a very similar sharp of the graph and could have match exactly if knowing Sutton's training data set.

References:

[1] Richard S Sutton and Andrew G Barto. Reinforcement learning: An Introduction, 2020

[2] Richard Sutton. "Learning to Predict by the Method of Temporal Differences". In Machine Learning 3 (Aug 1988), pp 9-44