https://github.gatech.edu/jliu727/RL

Replication of Sutton's Experiment

Jiaxin Liu
Department of Computer Science
Georgia Institute of Technology
Georgia, U.S.
jliu727@gatech.edu

***Abstract*—This article looks into temporal difference learning method introduced from *Learning to Predict by the Methods of Temporal* Differences by Richard S. Sutton in 1988. Here we use the method described in the paper to replicate Sutton's result on bounded random walks mentioned in part 3.2 of the paper.**

Keywords—temporal difference learning, prediction, learning rate.

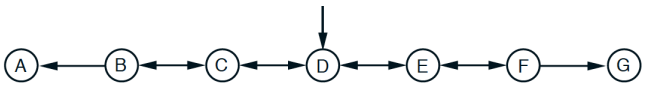
I. INTRODUCTION

In this part of the course, we learned TD learning method and tried to compare it to the conventional supervised learning methods and found it has several advantages when dealing with dynamic system. TD methods tend to calculate incrementally, thus easier in computation and has a better use of past experience when dealing with states that evolve with time. Along with its incremental learning procedure, TD methods generally requires less memory space and milder computational speed compared to Supervised Learning. In this project, we will reproduce some experiments Sutton did in his paper to show performance of TD methods and its advantages.

II. BOUNDED RANDOM WALK PROBLEM

In the paper, Sutton proposed that TD methods should learn more effectively than supervised learning methods if the system predicted is a dynamic system. The problem they looked at is bounded random walk.

Bounded random walk generates a sequence of state by taking random walk to the left or to the right of the current state until a boundary is reached. In the paper, Sutton used the following simple example of the bounded random walk:



Here, we can see that the start state is always state D. For state B, C, D, E, F, it is allowed to go either left or right, each with 50% probability. State A and G are the boundary of this example, once these two states are entered, the walk ends.

In the paper, the walk's outcome (reward) is defined as z = 0 for ending at state A and z = 1 for ending at state G. For each non-terminal state (B, C, D, E, F), it is assigned a vector of length 5. For example, the staring state x_D would be (0,0,1,0,0). x_B would be (1,0,0,0,0).

According to the paper, Sutton wish to estimate the probability of a walk ending at state G given it is at one of the other states using TD Learning methods. The ideal prediction for each of the non-terminal states are $\frac{1}{6}$, $\frac{2}{6}$, $\frac{3}{6}$, $\frac{4}{6}$ and $\frac{5}{6}$.

https://github.gatech.edu/jliu727/RL

III. EXPERIMENT IMPLEMENTATION

The goal is to replicate Fig.3, Fig.4 and Fig.5 from Sutton's paper. For simulation of the above bounded random walk problem, as described above in the problem each non-terminal state is represented by a length-5 vector. For terminal state, I use (0,0,0,0,0) for state A and (1,1,1,1,1) for state G for consistency. For each non-terminal state at time t, it can randomly choose move 'left' or 'right' until it ends in A or G. One sequence is a list concatenation of many states representation starting from (0,0,1,0,0) and ending in either (0,0,0,0,0) or (1,1,1,1,1) as in function *create_training_list*.

For all three experiments, in order to obtain reliable results, we use the same set of training data. Here, as directed in the paper, we generate 100 training sets, each consisting of 10 sequences simulated from the bounded random walk simulation.

For general TD methods, we would like to find the weight vector w that gives goo prediction P of a state eventually ending up in G by calculation $P_t = w^T x_t$. We first set initial value for weight vector w. The we modify w by taking in sequences and generate Δw from function *get_delta_w*. Δw is generated by equation(4) in Sutton's paper:

$$\text{For } 0 \le \lambda \le 1: \quad \Delta w = \alpha(P_{t+1} - P_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_k$$

We evaluate the final result by comparing RMSE between ideal prediction and the prediction generated from the learning procedure. In this problem, the generated prediction equals to the final weight vector w. Now we talk about specific implementation for each experiment.

Experiment 1: In this experiment, we try to replicate figure 3 of the paper. This experiment shows how prediction would vary by different λ value. As instructed in the paper, we update w by repeatedly feeding the learning method each training set (containing 10 sequences) until Δw become too small to have an impact on w. In this experiment, we accumulate Δw throughout a complete training set and update w afterwards. For each final w we get, we calculate the RMSE between it and the ideal prediction. In the end, we would have 100 RMSE from 100 training set and calculate the average of it to be the RMSE for that particular λ. The result of this experiment should be independent of the initial weight vector as it takes runs until convergence according to the paper. Here, I am using initial w = (0.5, 0.5, 0.5, 0.5, 0.5). It turns out that changing initial value does not impact my final result. In this experiment, parameters available to adjust are α and convergence criteria. I am using a small alpha = 0.01 to ensure convergence and a convergence threshold of 0.0001. If the maximum absolute change for w is smaller than the threshold, I consider Δw to small to continue. We expect to get a list of RMSE for each λ value from this experiment and a plot showing relationship between them.

Experiment 2: In this experiment, we try to replicate figure 4 of the paper. This experiment shows how λ and α combination could change the final prediction with limited amount of training data. Here, differing from the above experiment, we only present each training set once to the learning method, therefore, there is no convergence criteria. Also, we update w after each sequence rather than after each training set so we are updating w with a faster frequency. Here we are setting initial w = (0.5, 0.5, 0.5, 0.5, 0.5) as instructed in the paper. The calculation of RMSE is the same and we use it to measure the performance of each λ and α combination. Here we are taking λ in 0, 0.3, 0.8 and 1 while α ranges from 0 to 0.6 for the first three and 0 to 0.4 for λ = 1. In this experiment, I am also excluding RMSE from anomalous training set because we only see the training set once and do not iterate towards convergence, the RMSE can occasionally be very large from large alpha update. Here, I am excluding any RMSE > 2 from being considered in the final average RMSE. To make sure I still have 100 usable RMSE after excluding, I generated a training list of 200 training set and stop whenever the number of usable RMSE reaches 100. We expect to get a list of RMSE for each λ, α combination, in total 48 RMSEs and a plot showing relationship between them.

Experiment 3: In this experiment, we try to replicate figure 5 of the paper. This experiment shows how the performance of each λ and its best α when presented with the training set once. The setup for this experiment is similar to experiment 2, where we set the initial w to (0.5, 0.5, 0.5, 0.5, 0.5) and present TD methods with each training set only once. We get the best α for each λ using method from Experiment 2. We then learn the weight vector w for each λ and plot its RMSE. λ ranges from 0 to 1 and the best α we found ranges from 0.2 to 0.05.

IV. Result Analysis

Experiment 1:

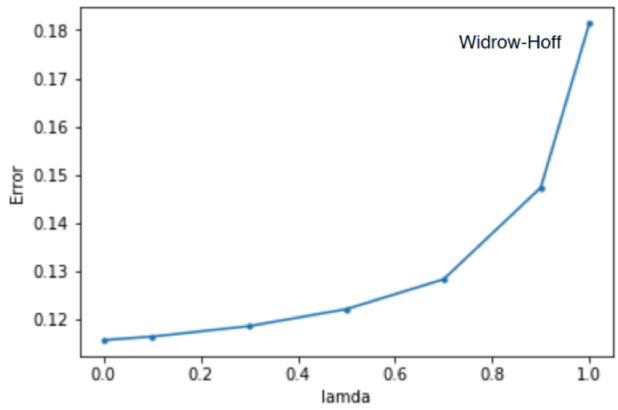


Figure 3 Replication

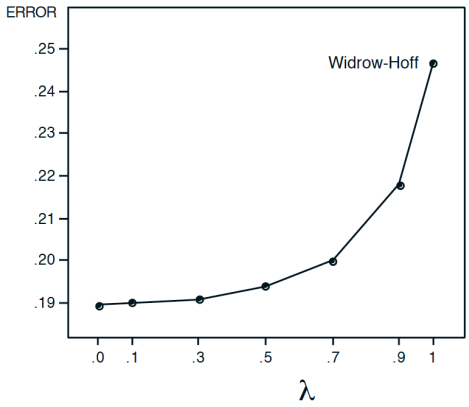


Figure from Sutton's

In this experiment, we want to investigate relationship between prediction performance and λ. The graph on the left is the replication of Sutton's figure 3. Same as the original figure from Sutton's, we can see that $\lambda = 0$ performs the best when training set is presented repeatedly until convergence and the error (RMSE) increases as λ increases. It matches the fact in lecture and paper that TD(0) converges to maximum likelihood result which gives best estimate for future prediction while TD(1) gives best estimate for the training set itself, thus not performing well in generating future estimate.

Despite the expected upwards trend, we can see that our replicate has in general lower error than Sutton's figure. Our error ranges from ~0.12 to ~0.18 while Sutton's ranges from ~0.19 to ~0.25. It is unlikely that this shift is caused by the randomness of the training set as we averaged over 100 training set. It is probably caused by Sutton using a different convergence criterion from what we are using here. Looking from the graph, a higher error indicates that Sutton's learning method terminated earlier and hence not reaching the best prediction. My assumption is that Sutton probably also took into account the number of iterations it allows the different TD method to run while having a less strict convergence criterion which resulted in its higher error.

Experiment 2:

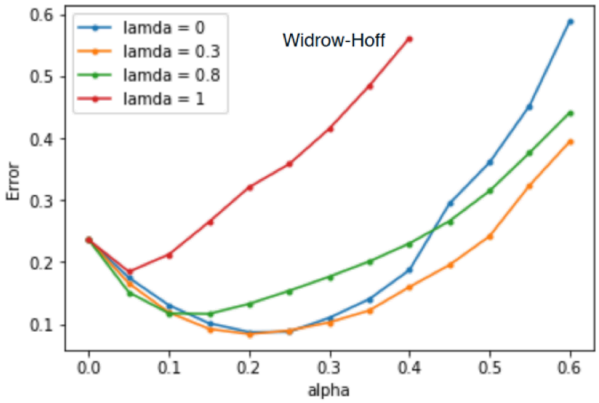


Figure 4 Replication

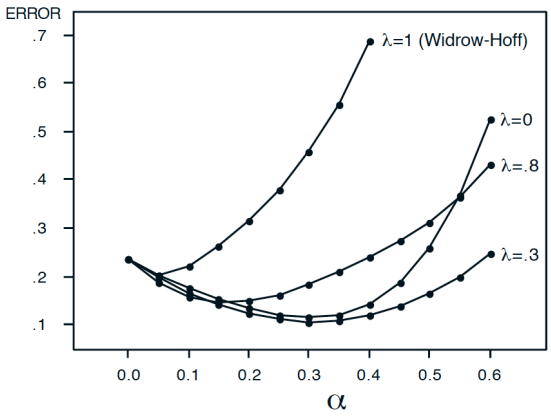


Figure from Sutton's

In this experiment, we want to investigate the relationship between prediction performance and (λ, α) combination. The graph on the left shows replicate of Sutton's figure 3. We can see that the the trend is very similar to Sutton's graph on the right. All lines start from 0.2357 because when α = 0. This makes sense as on that point, there is no update to the initial w and the final error is the RMSE between (0.5, 0.5, 0.5, 0.5, 0.5) and ($\frac{1}{6}$, $\frac{2}{6}$, $\frac{3}{6}$, $\frac{4}{6}$, $\frac{5}{6}$). We can also see that TD(1) is in general worse than other λ value in any alpha value which makes sense as Widrow-Hoff method only minimizes error on training set and while only presented with one training set containing 10 training sequences, it can easily fail when the sequences presented is biased or anomalous. We can see for all λ, the trend shown on the replicate is very similar to the one shown on Sutton's graph with $\lambda = 0$ and 0.3 performing well in small α value (< 0.3). We can see that $\lambda = 0$ increases a lot in larger α value. It makes sense as TD(0) updates each states slower than other λ value. with a large α, it can modify only limited states with large updates and resulting in inaccurate predictions.

Here, I set a restriction to the RMSE eligible to be used in the final average calculations because for large α while $\lambda = 0$, occasionally, the RMSE can be large enough to be identified as outlier due to anomalous data. Therefore, I am using RMSE as an indicator to exclude anomalous training set to be used in the final evaluation. Whenever RMSE > 2, this training set is considered outlier and not usable for that particular λ. On average, we excluded ~2% of training set.

Experiment 3:

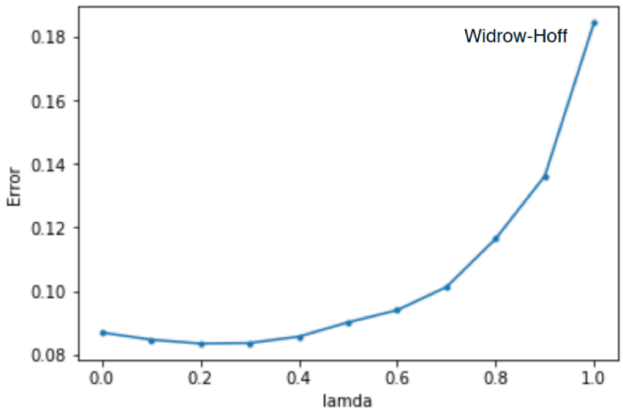


Figure 5 Replication

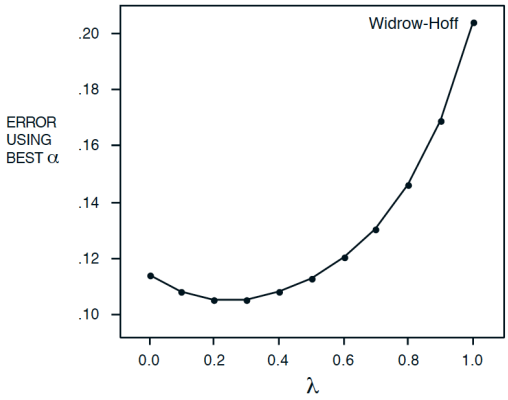


Figure from Sutton's

In this experiment, we want to investigate the relationship between prediction performance and (λ, best α) combination. The graph on the left shows the replicate of figure 5 in Sutton's paper. The best α we are

using here for λ = 0, 0.1, 0.2……1.0 are 0.2, 0.2, 0.2, 0.2, 0.2, 0.2, 0.15, 0.15, 0.15, 0.1, 0.05. Here we can see that the best alpha decreases as λ increases which makes sense as with a small λ, the learning speed through summation of past states is small and correspondingly the learning rate need to be larger to achieve best Δw when only presented with training set once because the number of iterations to change w is small. Here, the best RMSE for each λ value are 0.0870, 0.0847, 0.0835, 0.0837, 0.0857, 0.0902, 0.0940, 0.101, 0.116, 0.136, 0.184. We can see that the best performance is not at TD(0) any more. It is now around λ = 0.2 and λ = 0.3. This is because TD(0) make changes to each states the slowest as it propagates changes slower than other methods. When provided with the training set only once, TD(0) does not have enough time (chance) to update w thoroughly. If the training sets are provided repeatedly until convergence as in experiment 1, this would not be an issue.

We can also see by comparing the two graphs, the trend is very similar with the only difference that with small λ, we get slightly better RMSE compared to Sutton. This small difference could be due to different training sets we used from Sutton or the best α we found differ from Sutton by a little. We have not put any checks on training sets we randomly generated. However, 10 sequences are a rather small amount of training sequences to train on so it could be that Sutton has some anomalous training sets used in the procedure, resulting in a slightly higher RMSE.

V. Conclusion

From this replication, we can clearly see the advantages of TD methods (λ<1) compared to classical supervised learning procedure, the Widrow-Hoff rule in multi-step prediction. TD methods is better in using past experience to improve prediction for each states incrementally compared to all updates at the end of the sequence by supervised learning.

References

[1] S. Richard, “Learning to Predict by the Methods of Temporal Differences,” Machine Learning 3: 9-44, 1988