# Project 1: Desperately Seeking Sutton

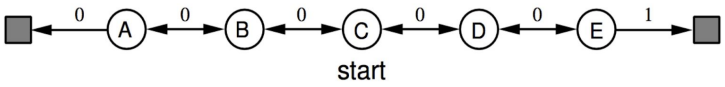
CS 7642 Reinforcement Learning | Spring 2018
Dan Frakes | dfrakes3

## Abstract

In this paper, I discuss my attempts to replicate the results of two temporal difference learning experiments presented in *Learning to Predict by the Methods of Temporal Differences* (Sutton 1988). While Sutton explains the logic behind the structure of his published results, he excludes some specific hyperparameter values used to produce Figures 3, 4, and 5.

## Experiments

Sutton presents two experiments to investigate the behavior of TD($\lambda$) on a trivial random walk problem.[1]




The random walk problem illustrated as an MRP
*Reinforcement Learning: An Introduction*, Sutton & Barto 1998, page 100

The first experiment is to measure the RMSE of TD($\lambda$) for several $\lambda$ values spanning the range of possible values -- more precisely, 0.0, 0.1, 0.3, 0.5, 0.7, 0.9, and 1.0 -- by repeatedly training the TD algorithm on the same series of state sequences (episodes). Using repeated presentation, the state value estimates converge, illustrating the increase in error as the lookahead depth increases (illustration below). Sutton does not explicitly state the learning rate value used to produce Figure 3, so lots of trial and error resulted in a relatively close graph using $\alpha$ = 0.1, decay rate of 0.9, and $\epsilon$ = 1e-2, though the error values varied drastically with subsequent program executions, especially for TD(1) whose error values occasionally spiked to double-digit values.

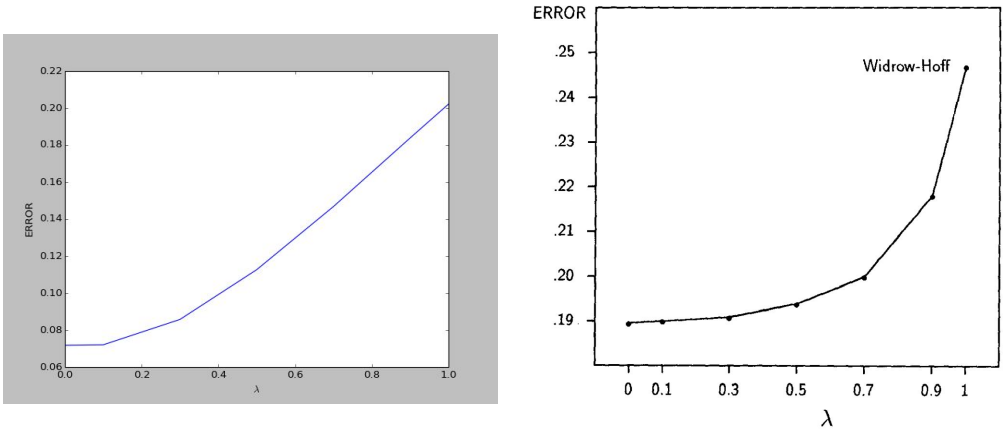



Figure 3. Averaged RMS of converged TD($\lambda$) state value estimates applied to the random walk problem. Frakes (left) and Sutton (right).

[1] Sutton's explanation of the random walk problem in his 1988 paper deviates from that outlined in *Reinforcement Learning: An Introduction* (Sutton & Barto 1998). My experiments, code, and report use the problem description outlined in the textbook, with the addition of naming the left terminal state "0" and the right terminal state "1."

The second experiment involves determining an appropriate learning rate for various $\lambda$ values given TD($\lambda$) is only presented with the training set once (with results averaged over 100 training sets) so as not to converge to estimated state values. Using Sutton's Figure 4 as a reference, I included the same $\alpha$values ranging from 0.0 to 0.6 -- incrementing by 0.05 -- and $\lambda$ values of 0.0, 0.3, 0.8, and 1.0.

While the shapes of the graphs are similar, the most obvious difference between my construction of Figure 4 and Sutton's is the trajectory of error functions beyond their respective minima. This leads me to believe that there is a sum, mean, or other small algorithmic function in my TD implementation that varies from Sutton's, or possibly that Sutton may have interpolated some of the data in the paper. Regardless, I was satisfied with the similarity in the shape of the graph, which indicates the underlying logic of TD is implemented identically.

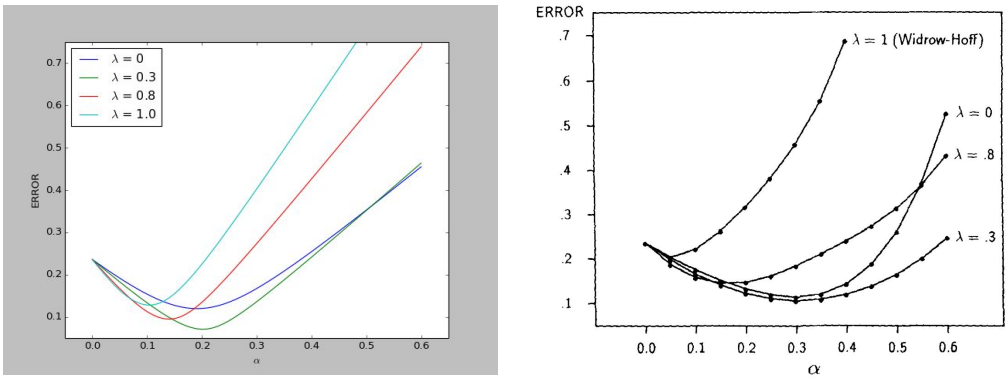
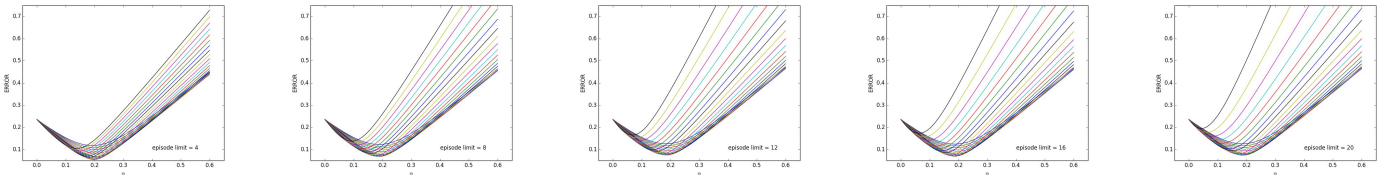



Figure 4. Averaged RMS of TD($\lambda$) state value estimates using various learning rates applied to the random walk problem. Frakes (left) and Sutton (right).

In addition to the similar shape of the converged loss functions, we can see that the error calculations are also identical for all TD($\lambda$) where $\alpha$ = 0. This is because at this extreme of temporal difference learning, no learning occurs: when $\alpha$ is 0, then no matter what weight updates are calculated (which necessarily vary by $\lambda$), the update is multiplied by 0, resulting in a "convergence" at the initial weights (in this case, the initial state values of [0.5, 0.5, 0.5, 0.5, 0.5], which results in an error of 0.235).

As discussed on our student forum, a few suggestions led me to experiment with restricting the length of each episode. Restricting the timesteps to 8 or fewer actually "stretched" Figure 4 horizontally, decreasing the rate of change in each loss function. This effect is illustrated below with graphs of <$\lambda$, $\alpha$> combinations at regular intervals of $\lambda$ (0.05). The result resembles a smooth hyperplane:



Limit = 4 Limit = 8 Limit = 12 Limit = 16 Limit = 20

Figure 4 redrawn with regular intervals of $\lambda$, illustrating effect of episode length limit on RMSE of <$\lambda$, $\alpha$> combination

The second experiment continued using the optimal $\alpha$ values derived from Figure 4 for each $\lambda$ to determine the best [lowest] RMSE for non-converged TD calculation. The $\lambda$ values we consider and their corresponding best $\alpha$ are outlined below:

| $\lambda$ | $\arg\min_{\alpha} RMSE(TD(\lambda))$ | | |
|---|---|---|---|
| | Frakes (±0.05) | Frakes (±0.01) | Sutton |
| 0.00 | 0.20 | 0.21 | 0.30 |
| 0.05 | 0.20 | 0.21 | -- |
| 0.10 | 0.20 | 0.21 | -- |
| 0.15 | 0.20 | 0.22 | -- |
| 0.20 | 0.20 | 0.22 | -- |
| 0.25 | 0.20 | 0.22 | -- |
| 0.30 | 0.20 | 0.22 | 0.30 |
| 0.35 | 0.20 | 0.21 | -- |
| 0.40 | 0.20 | 0.21 | -- |
| 0.45 | 0.20 | 0.21 | -- |
| 0.50 | 0.20 | 0.20 | -- |
| 0.55 | 0.20 | 0.20 | -- |
| 0.60 | 0.20 | 0.19 | -- |
| 0.65 | 0.15 | 0.18 | -- |
| 0.70 | 0.15 | 0.17 | -- |
| 0.75 | 0.15 | 0.15 | -- |
| 0.80 | 0.15 | 0.14 | 0.15 |
| 0.85 | 0.10 | 0.12 | -- |
| 0.90 | 0.10 | 0.11 | -- |
| 0.95 | 0.05 | 0.09 | -- |
| 1.00 | 0.05 | 0.07 | 0.05 |

Best $\alpha$ values for each $\lambda$ (approximated to ±0.05 and ±0.01) that minimize RMSE. Sutton's incomplete column is due to the fact that he only graphed 4 $\lambda$ values in Figure 4.

Invoking TD once more with only these specific <$\lambda$, $\alpha$> combinations, we can derive an illustration similar to Figure 5 in Sutton's paper:

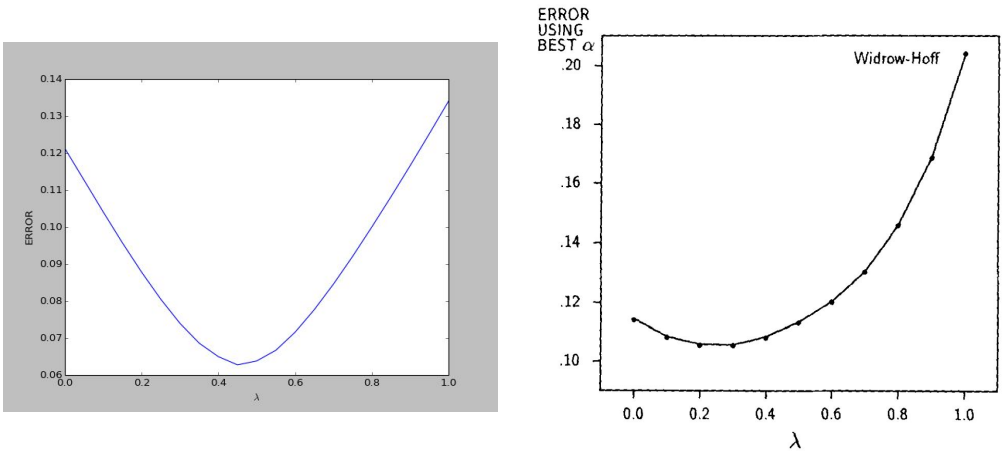
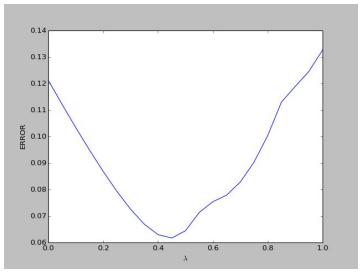



Figure 5. Averaged RMS of TD($\lambda$) state value estimates using approximated optimal learning rate for each $\lambda$ value. Frakes (left) and Sutton (right).

The final graph for my replication of Figure 5 implements the $\alpha$ values from the second column in the chart above (±0.01). Using the coarser approximations resulted, unsurprisingly, in "jumps" in error values -- shown below -- since the optimal $\alpha$ values are not continuous but rather a sequence of discrete values with a minimum interval (±0.05).



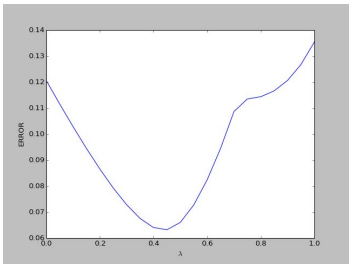



Figure 5 replication using $\alpha$ values varying by ±0.05 (left) and ±0.10 (right

Since my error value for TD(1) using the more coarsely-approximated $\alpha$ value was more consistent with Sutton's, it may be the case that Sutton plotted a few points (e.g. the original 4 points, $\lambda \in \{0.0, 0.3, 0.8, 1.0\}$), then interpolated the remaining data using a line of best fit. Using more precisely calculated optimal $\alpha$ values resulted in a lower error for several TD($\lambda$) calculations, particularly for $\lambda$ > 0.3, which is compatible with this theory of interpolation.

## Analysis

For the trivial random walk problem used in this project, we can make two assumptions: First, given the observation vectors presented by Sutton used to converge estimated state values are unit vectors, we can conclude that weights and state values are identical in this context. That is, since $P(x_t, w)$ is the dot product $w^T x$ and each observation vector $x_t$ is a unit vector, $P_t = w_t, \forall t \in T$. Second, given the underlying uniformly random transition model (i.e. transition to left or right are of equal probability 0.5), we can assume $P(x_t, w)$ is a linear function. In the case of a random walk with 5 non-terminal states, as presented by Sutton, this equates to the function $P(x_t, w) = \frac{t+1}{6}$.

For the first experiment, I had trouble implementing an appropriate learning rate due to a number of reasons. First, Sutton explains in his paper that he updated the weight vector after each complete iteration of a training set (i.e. after a single run of 100 episodes). Altering this placement of weight updates required changes to the code in order to appropriately store, accumulate, and reset the $\Delta w$ values with each iteration. Additionally, I was confused by the inconsistency in terminology between the course lectures and Sutton's TD representation. As discussed in the course lecture on properties of learning rates[2], in order for the value estimates to converge to the true values (optimal predictions), the learning rate must adhere to both criteria below:

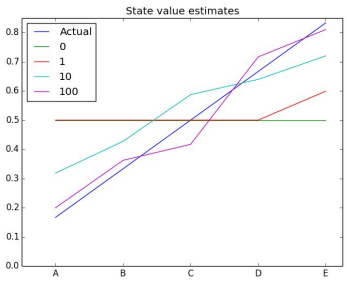
$$\sum_T \alpha_T = \infty \qquad \sum_T {\alpha_T}^2 < \infty$$

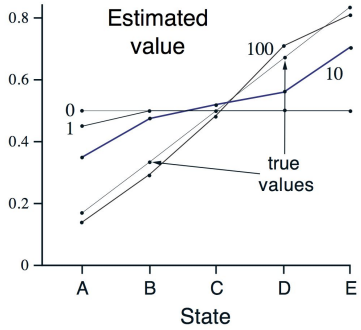
With $\alpha$ as a particular function of T, such as $\alpha_T = T^{-1}$, these conditions are met, but Sutton's $\alpha$ values -- continually decaying constants -- do not adhere to these criteria. Since we are not computing infinitely long episodes, however, we can safely assume that a careful selection of $\alpha$, a decay rate, an appropriate threshold $\epsilon$, and a sufficient number of executions to average will accurately approximate the state value. To justify this assumption, I tested my TD algorithm, using exponentially decaying $\alpha$ values, to approximate the state values, recreating Example 6.2 from Sutton's and Barto's textbook[3].

---

[2] CS7642 Lecture 3 - *TD and Friends* (Littman 2016)

[3] Example 6.2 (p100) from *Reinforcement Learning: An Introduction* (Sutton and Barto 1998).




Example 6.2 from *Reinforcement Learning: An Introduction*
Frakes (left) and Sutton (right)

One major improvement to my replication of Figure 4 was the restriction of episode lengths. As discussed in the class forum, RMSE seems to skyrocket for longer episode sequences. Given the already-limited possible sequences for each episode, this is likely caused by uneven $\Delta w$ values where $\Delta w_C$ (the center state and most likely to repeat) may have a magnitude 3 times (if visited 3 times in an episode) that of another state visited only once. This scenario is particularly noticeable for large values of $\lambda$, where the lookahead includes more states, increasing the likelihood of multiplying the state update (undiscounted for $\gamma$ = 1) unevenly.

## Sources

Isbell, Charles and Littman, Michael. *CS7642 Reinforcement Learning*. College of Computing, Georgia Institute of Technology. Udacity, June 2016.

Sutton, Richard. *Learning to Predict by the Methods of Temporal Differences*. Kluwer Academic Publishers, Boston, 1998. http://incompleteideas.net/papers/sutton-88-with-erratum.pdf

Sutton, Richard. *TD Learning*. Department of Computing Science, University of Alberta. Published July 27, 2017. http://videolectures.net/deeplearning2017_sutton_td_learning/

Sutton, Richard and Barto, Andrew. *Reinforcement Learning: An Introduction (second edition)*. MIT Press. Draft published January 1, 2018. http://incompleteideas.net/book/bookdraft2018jan1.pdf

Silver, David. *[COMPM050/COMPGI13] Lecture 4 - Model-Free Prediction*. University College London. May 13, 2015.

Various contributors. Piazza 2018.

Various contributors. *#cs7642*. Slack 2018.