Experimental Replication and Analytical review of "Learning to predict by the methods of Temporal Differences" by Richard S. Sutton, 1988

Ashish Panchal *OMSCS Student - Dept. of Computer Sciences*
Georgia Institute of Technology, Atlanta, USA
apanchal33@gatech.edu and ashupanchal.007@gmail.com
Git hash:8c36b5298e71df130d891fcb48d5d99cfd56eecf

***Abstract*—Experiments from Sutton's original work on TD learning were replicated. Simulated results were contrasted with the original and it was concluded that the general idea of TD methods as more efficient learners, proposed by the Author holds true for majority of case, However, Assumptions made by the Author were found to be a subject of the problem being solved, and may not always be true.**

***Index Terms*—TD learning, MGP, Experimental overview, Results and Analytical overview.**

I. Introduction

In many activities in our lives, we take regular feedback and change our actions to get the desired result, many processes we experience can be described in this manner, example: a painter takes regular feedback to understand which stroke adds what effect to his painting, where he can correct misplaced strokes immediately after followed observations, if not done in this way, it may become difficult to identify the mistakes after completion of the painting. Therefore by regular updates it is ensured that every stroke is in the direction of getting to a better version of the painting, with a view of creating a good painting. Additionally a stroke and its results, helps the painter to decide the next best immediate stroke.

Following a similar example, Sutton argued in his paper, that the approach of learning at smaller intervals of time, and therefore called Temporal difference methods (TD), more efficiently use the experience, than the conventional learning methods. Additionally he argues, that TD methods could efficiently learn to predict even arbitrary event and not just goal related ones. [1]

This report aspires to study and explore the two experiments used by Sutton in his paper as a proof of efficiency for TD methods, covering a brief theoretical description of the learning problem, reasoning and experiments as performed by Sutton, Key aspects to replicate the experimental results, Analytical overview of experimental results and comparison with the original work to understand the key differences and similarities and the possible reason for the same. There after as the part of discussion we will explore the challenges in replications of the paper, with suggestive adaptations to better understand the results.

It is to be noted that the report will not cover extensive theory or computations proofs to understand Temporal difference learning, or associated supporting works. And will limit to understand the nature of learning methodologies, from the lens of bounded random walk, as described in the original paper, where the author has focused on predictions which are based on numerical features, which describe the general nature of a environment state, along with associated adjustable parameters, weights, to understand the importance of each feature in association with the immediate or long term goals.

The next section provides a background of Learning processes and theory of reinforcement learning, as an essential part to understand Sutton's work.

II. Background

A. Temporal difference and Supervised Learning methods

As Described in the introductory example, the usefulness of TD methods can only be visualized in the scenario of a multi step processes, since in the case of a single step process the TD methods can only learn from a single observation which does not differentiate from Supervised Learning methods. Therefore, in the study, the original work only explores multi step processes.

B. Computational overview

This section briefly covers the learning mechanisms of TD methods, its similarity with Supervised learning, along with a computational walk-through of how TD methods learns more efficiently than the later.

Each state in a multi step processes describes the environment at the time. A state can be assumed as a collection of numerically presentable features representing different aspect of the environment, example state x = $[1, 0, 0, 0, 2, 2]$, for 5 feature. There could be limited m number of states in the environment, $\{x_1, x_2, x_3, ...x_m\}$. Based on the actions/ steps taken, At different times (t), the environment can transition into a different state based , generating a sequence of states over time $\{x_1, x_2, x_3, ...x_t...x_{T-1}\}$, where $x(t)$ represents as specific visited state at time t, $x_{(time=t)} = x_{(state=k)}$, where$k \ \epsilon \ (1, m)$. At $t = T$, the targeted state is reached, associated with a desired numerical scalar value z. At each visited state, a learning procedure will try to estimate z, based on its defining features. However, each feature do not contribute the same, and therefore, the prediction can be defined as function of state features weighted to their contributions. At every time t, for a visited state this function can be represented as $P_t(x_t, w)$, therefore a sequence of visited states generate a sequence of of predictions,$\{P_1, P_2, P_3....P_t\}$, where any state may or may not be revisited.

Any learning algorithm corrects its prediction by understanding the error in previous predictions, defining incorrect

prior beliefs about feature contributions and therefore errors are used to define possible change, Δw_k, in expected feature weights, vector w, at k^{th} timestamp. Therefore at time t, the beliefs can be updated as follows:

$$w = w + \sum_{k=1}^{t} \Delta w_k \tag{1}$$

Supervised-learning only updated w after reaching the terminal state, i.e when z is known. It try to predict z at every visited x_t.Additionally, to minimize the impact of any high variance experiences, learning is controlled by a predefined scalar learning rate α. The change corresponding to a i^{th} feature is weighted to its contribution, it is defined by gradient of the prediction $\nabla_{wi} P_t$, as a derivative of $w_{i^{th} feature}$. Thus update of feature weights, Δw_t, , at time t can be defined as $f(P_t(x_t, w), z, \alpha, \nabla_w P_t)$. As mentioned by Sutton, supervised method updates can be calculated as follows:

$$\Delta w_t = \alpha(z - P_t(x_t, w))\nabla_w P_t \tag{2}$$

As stated above for any supervised model , Δw_t in (2) cannot be computed incrementally, thus, sequences of states and prediction at stored till T. However, By representing the error $(z - P_t)$, as a sum of changes in subsequent predictions, TD methods, produces exactly the same result as (2), and learn computed incrementally. i.e.

$$z - P_t = \sum_{k=t}^{T-1} (P_{k+1} - P_k) \quad where\ P_T = z \tag{3}$$

Based on 2 and 3 , we can say:

$$\begin{aligned} w \leftarrow w + \sum_{t=1}^{T-1} \alpha(z - P_t(x_t, w))\nabla_w P_t \\ = w + \sum_{t=1}^{T-1} \alpha \sum_{k=t}^{T-1} (P_{t+1} - P_t)\nabla_w P_t \\ = w + \sum_{k=1}^{T-1} \alpha \sum_{t=1}^{k} (P_{t+1} - P_t)\nabla_w P_t \\ = w + \sum_{t=1}^{T-1} \alpha (P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_t \end{aligned}$$

where incremental change at time t is:

$$\Delta w_t = \alpha(P_{t+1} - P_t) \sum_{k=1}^{t} \nabla_w P_t \tag{4}$$

The family of TD methods, controls and utilize sensitivity of update for past prediction based on the recent observation, where, special case, (4) updates past predictions equally . Prediction for a visited state, k steps in the past is exponential weighted to its recency i.e. λ^k for $0 \leq \lambda \leq 1$:

$$\Delta w_t = \alpha(P_{t+1} - P_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_w P_t \tag{5}$$

(5) can be calculated incrementally as trace of eligibility e_t, based on (6), therefore, past predictions need not be stored.

$$\begin{aligned} e_{t+1} = \sum_{k=1}^{t+1} \lambda^{t+1-k} \nabla_w P_k \\ = \sum_{k=1}^{t} \lambda^{t+1-k} \nabla_w P_t + \nabla_w P_{t+1} \\ = \lambda e_t + \nabla_w P_{t+1} \end{aligned} \tag{6}$$

C. TD Example : Bounded Random Walk

For a process with states evolving with time, author states, TD methods learn more efficiently than Supervised methods.Explaining through the Dynamical process of bounded random walk with 7 states, *fig*(1), starting from the center stage D and with uniform probability stepping right or left,

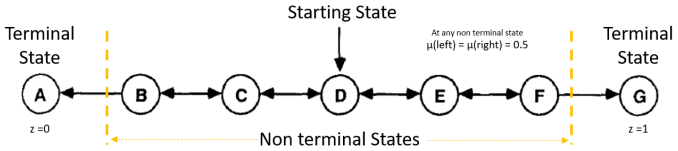


Fig. 1: *Bounded Random Walk*

until a boundary (A or G) is reached, reaching an outcome of $z_A = 0$ or $z_G = 1$. The learning method at any non terminal states,$\{B, C, D, E, F\}$, estimates the expected ideal prediction of z, equal to the probability of terminating at A, i.e. $\{1/6, 1/3, 1/2, 2/3, 5/6\}$, respectively. For simplicity, Authors defines the states feature vector as a unit basis vector, with a size 5, i.e number of non terminal states, with "1" at one unique component, i^{th}, for the state and other components as "0", e.g.$Scale[0.8]x_D = (0, 0, 1, 0, 0)^{Trans.}$. Thus, the value of a state, $Scale[0.8]P_t = w^{Trans}.x_t = w_{i^{th} component}$ Sutton performed two experiments, based on the above process. The following Section discuss the replication process for these experiment, observations, similarities and difference between simulated and Author's Results.

III. Experimental Overview

A. Experiment 1

To replicating the experiments, 100 test sets with 10 bounded walk sequences each were generated, as perII-C. With Repeated Presentation of same set of limited experiences, in this experiment, the authors proves $TD(\lambda)$ methods, with small α, convergence criteria and any initial weights, learn more efficiently than Supervised learning methods. Where the later is known to perform.

1) Replication: With Initialization of α at 0.01 and convergence threshold, $\epsilon = 0.000005$, assumed statistically significant, along with weights initialized at 0.5, the process was defined to calculate the Average RMSE error for λ values in $[0, 0.1, 0.3, 0.5, 0.7, 0.9, 1]$, over 100 test sets, as per II-C and II-C.

For a test set and λ, Δw_t was calculated, as per (5) and accumulated in a vector,$Dw = \sum_{k=1,t=1}^{k=10,t=T-1} \Delta w_{k,t}$, over the 10 sequences of the set, with incremental calculation of e_t, as per (6). where k represent k^{th}sequence. If maximum change component in Dw is smaller than ϵ, convergence is said to achieved, with no further change in w, else Dw is added to w, and set is repeatedly presented with w presented and updated for every iteration, till the convergence criteria is met, when RMSE error is calculated and stored for final w and ideal prediction, as per II-C, for the given test set and λ.

2) Results, Sutton and Simulated, fig (2) : Similarities : In Simulation and Sutton's result both, show, with repeated presentation of limited experiences, TD(λ) learn more efficiently, and predict better estimates of the ideal prediction, with lower error, and with TD(0) with least errors, Additionally, in both, the accuracy decreases as we move towards higher λ values, supporting author's conclusions.

Differences : The simulation errors were lower than original study, by 0.06 -0.07. This could be, unclear definition "small α" and "ϵ" , different test sets sequences, and contrary to author's assumption, different initial w, (explored in IV)

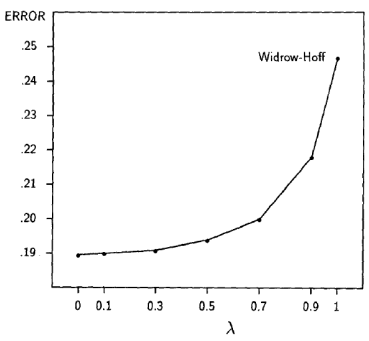


Fig. 2: *Exp1. Original(Figure 3) and Simulated results*

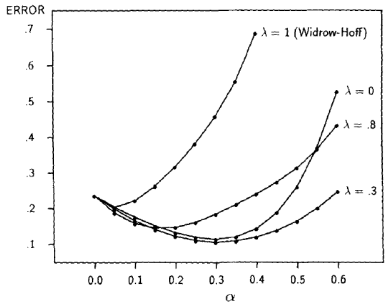


Fig. 3: *Exp1. Original(Figure 4) and Simulated results*

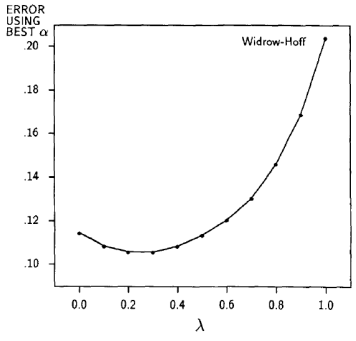
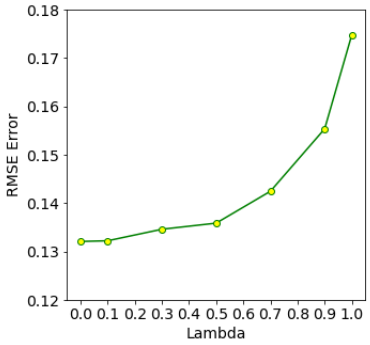
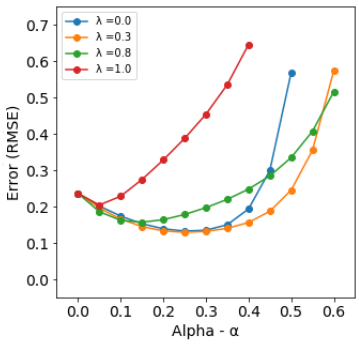
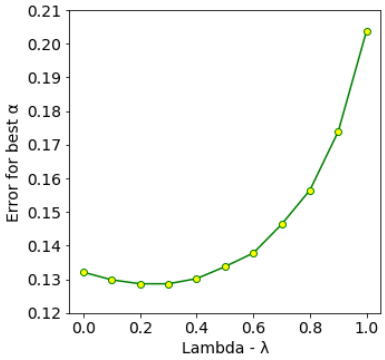


Fig. 4: *Exp1. Original(Figure 5) and Simulated results*







B. Experiment 2

1) Replication: The performance of TD(λ), for α,$\epsilon[0,1]_{interval\ 0.05}$ and $\lambda\ \epsilon[0,1]_{interval\ 0.1}$, were evaluated, similar to exp. 1, but with *single presentation* over test sets, with w initialized to 0.5 , to avoid any terminal bias.

w and P were updated post every sequence, (1), where Δw_t,e_t were calculated as per (5), (6) . The RMSE error for final w and the ideal prediction were calculated and stored for the given test set and λ and α, II-C.

2) Results: ***Sutton and Simulated, fig3,fig 4*** *Similarities* : with single presentation, intermediate λs perform better than$TD(0)$, having comparatively deeper feedback propagation, therefore faster learning, while minimizing impact of variance, unlike TD(1), for all α, with best results at intermediate α and with overall best at λ = 0.3. Additionally, simulated and author's experimental error for larger λs are similar.

Differences : at higher α and smaller λ, error accelerates faster for simulations, as in Fig 2 and 3. Possible reason: In simulation, "all visits" for states were considered for incremental w updates, till T. For long oscillation sequences , TD(0) will repeatedly calculate Δw for oscillating states, with prior w, creating multiplicative effect. resulting in large changes for these stages at T, which is magnified with large α predicting P farther from ideal. However, for larger λ, deeper propagation inherently adjust w for all states. Possibly author's sequences had less oscillation, or with last/first visit updates.

IV. Analytical Overview And Challenges

Simulated experiments support Sutton's statements. However due to certain ambiguous assumptions, replicating the experiment is challenging. Experiment 1 is the focus of this section.

A. Assumptions and Ambiguities in Experiment 1

Following are some ambiguous statements from Sutton's description of Experiment 1:

1) *Statistically reliable* test sets are required to represent unbiased convergence. However, no numerical definition is provided.
2) *Statistically reliable* number of sequences are required per set. But no indication on how many, to give statistically reliable learning experience.
3) Upper limit for small α to maintain *small* learning rate is not defined.
4) Presentation is to be repeated till *significant change* happens, which is not defined clearly. As final prediction is dependant on convergence threshold, replicated experiment may generate significantly different output.
5) Any initial weight with repeated presentation, converge to same value. However this may not always true. Therefore value needs to be clarified.
6) Pattern and length of sequences determine number of previous states to which feedback is shared. Different set of sequences may result in different error values.

1) Statistically reliable number of Test sets: Experiment 1 is run for count training sets in $\{20, 100, 200, 500\}$ and $\{1, 2, 3, 4, 5, 10, 15, 20\}$. For the first range, *fig* (5) (a), given a small α and ϵ, increasing the number of sets does not show any tangible change in results.

For the second range, *fig*(5) (b) : All the procedure, behave as per original experiment, hence, Assumption of "100" test sets can be considered statistically valid. For different λs, single test show largest error range, Adding one more test set reduces variance to a large extent.All other sizes of set of test sets share approximate the same error range.TD(0) performs best regardless of the number of test set, proving to learn more efficiently with limited experience, as compared to larger λs. Overall error of Test set count = 10 is lower than other sets, possibly by chance creation of more suitable sequences for conversion. *Based on the above results and it is possible, that apart from criteria mentioned by Sutton, minimum number of test sets to achieve low variance is dependent on number of states in a MDP, for a specific set of transition probabilities.*

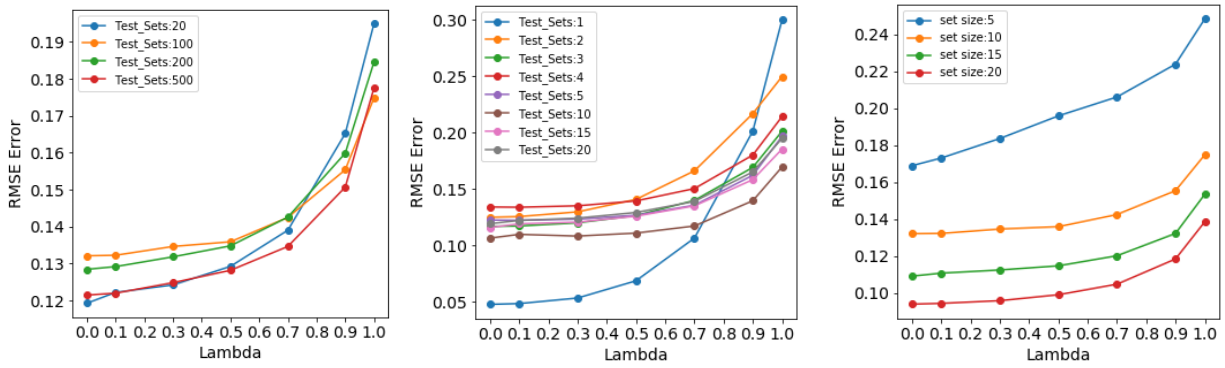


Fig. 5: *"Test-1": Exp1, Performance evaluation of TD(λ) with different # test sets.* Fig. 6: *"Test-2":Exp1, different set sizes*

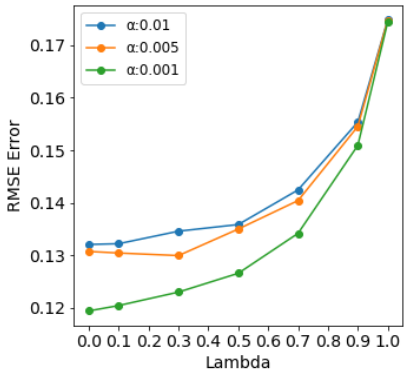


Fig. 7: *"Test-3": Performance evaluation of TD(λ) with different α, and repeated presentation. with smallerα, higher accuracy can be achieved at smaller λ*

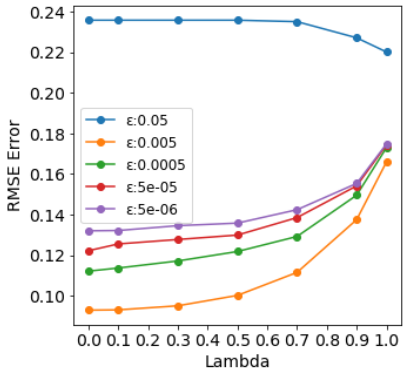


Fig. 8: *"Test-4": Performance evaluation of TD(λ) with different ϵ, and repeated presentation. At Highϵ, TD(1) performs better than other TD(λ) procedures*

2) Statistically reliable number of sequences per sets : The graph in *fig*(6) shows that increase in exposure to the environment (more training sequences) results in more accurate predictions. This increase might be reducing variance in weight updates, normalizing effect of exceptional sequences. This conclusion can be made for all λs. where λ =0 estimates much closer to the ideal prediction, and therefor proving, Sutton's statement of efficient learning. When the procedure "can" have more exposure to the environment,It might be beneficial to use slightly higher values of λ for other set parameters. *λ = 0 will take more repeated presentations, due to its lower range of propagation per episode experience. Other low values of λ (say 0.3) provide almost the same results faster. As depicted in the graph above more sequences per set produce better predictions. Sutton's rational for choosing "10" for learning is unclear.*

3) Small learning rate, fig(7) : Smaller α, produce better predictions over different e_t. For large α, changes in the w are large and learning is quicker but low precision at convergence. It may overshoot the minimum value change required to get to the ideal predictions. Small α eventually reaches higher precision, albeit slowly.Lower λs, results in better prediction, even for lower α, but there is less difference between the predictions made at the higher λs. Possibly because while change in the weights is a factor of λ and α, At higher λ, the impact of α is reduced, since its polynomial relationship gives λ more weight-age. *Therefore, for faster decision-making, it might be beneficial to work with larger α (for faster convergence). It might also be beneficial to use adaptive learning rate, which decreases with time, to make larger changes to initial assumptions for faster convergence and smaller changes later to attain higher precision.*

4) Significant Convergence Threshold, fig(8) : For the ϵ = 0.05, the model is not converging towards the ideal estimates. Possible because model's learning and weight changes are dependant on α and e_t. If convergence criteria is met without making changes to existing w, learning will stop early, therefore model will not move towards ideal prediction. Maximum-minimum change is defined by the 1st sequence in set and the state before reaching the terminal state, for the experiment,$\alpha(P_T - P_{T-1})\lambda^0 x_{T-1} = 0.01*0.5*1*1 = 0.005$. Any ϵ more than 0.005, might fail in leading to ideal estimate convergence. *As ϵ decreases below 0.005, average error increases. Convergence to ideal estimates, can be achieved with min-max change as the where the lower values may add noise to the data. It is also possible if the α is higher, that change has not made the convergence criteria with minuscule amount and additional change might overshoot the ideal prediction and may converge to a higher error but low change prediction.*

5) Weight initialisation fig(9) : all the models with initial w = 1 for all non-terminal states along with one terminal state, find it hard to converge to the ideal estimate and with worst performance at λ = 0, *fig*(9)(a). Since the all states, except one hold equally high value, model does not see benefit in converging to "G" compared to staying or oscillating between any other state till it observes an example of a sequence ending in "A". For example, when P = [0,1,1,1,1,1,1] , α = 0.01, λ = 0 and for a sequence of [DEFG]. the Δw = [0,0,0,0,0]. As there is no change, there is not learning. The sequences are generated randomly. Therefore there is a chance that out of 100 sequences first 50 end with "G" as a terminal state and there is no learning. This specifically impacts λ = 0, even if it is tries to converge to a better estimate, propagation is slow and even 100 of sets may not be enough to achieve convergence.

In this case, λ = 1 is a better performer as even from low number of sequences ending with "A" change is propagated throughout the states. If λ value is less than 1, but still high (like 0.7), limit the variance partially, but still propagate the change through large number of states, performs better than λ = 1.

Therefore, if a process requires exception identification, higher λ converge faster and perform better, given same number of experience on comparison Procedures with very high initial w (0.99), showed high errors, at $\lambda = 0$, there are less changes for terminating into "G"

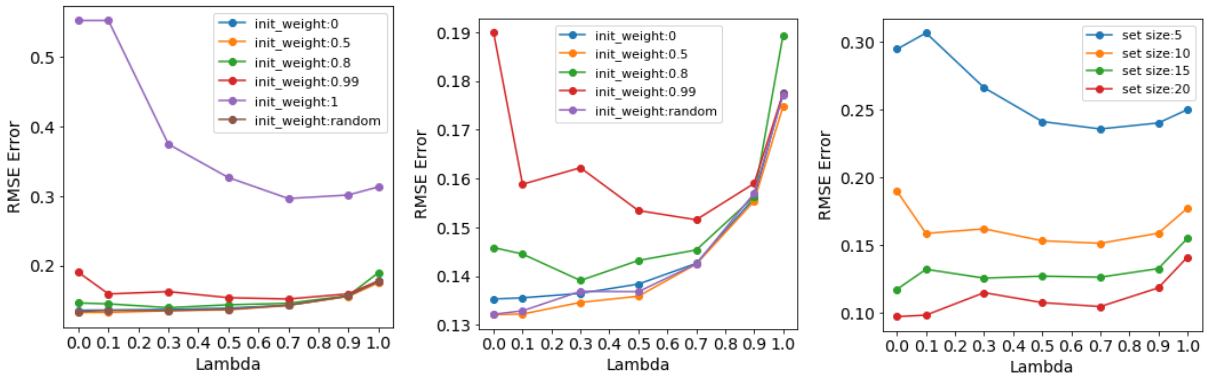


Fig. 9: *"Test-5":Performance of TD(λ) with different initial w, and repeated presentation. Convergence for large initial w.*

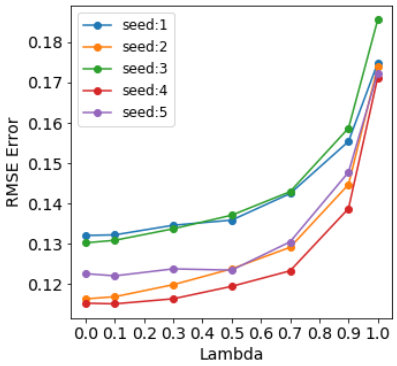


Fig. 10: *"Test-6": Performance evaluation of TD(λ) with random sequence test set, and repeated presentation. Different experience for a MDP, can result in slightly different predictions, however, all the test sets behave as per Sutton's comment of efficient learning* and learning is slow. however, the errors are lower than w =1, and is comparable to λ =1 procedures. Procedures for initial w equal or lower than 0.8 show comparable results throughout the range of λ (between 0 -1) for the set parameters. Only partially aligning with Sutton's statement on page 20, original work.

Contrary to Sutton's following statement : "Each training set if presented repeatedly to each learning procedure until the procedure no longer produced any significant changes in the weight vector. For small alpha, the weight vector always converged, and always to the same final value, independent, of its initial value." ***We fail to prove that weight vectors will always converge to same final values with repeated presentation for any initial weights.*** *Assuming "0.000005" is a statistically significant small change and "0.01" is a small α.As shown in the figure fig 9, the statement can be corrected, with above mentioned assumptions that all initial weights "producing values significantly different from desired terminal state" on repeated presentation are likely to converge near to each other at any λ between 0 and 1. "Significant difference" is subjective to further study, as for this experiment, initial weights below or equal to 0.8 show similar behaviour. Performance did not improve for "all initial weights" as λ is reduced below 1 to 0, contrary to fig 2. Specifically for large initial weights. (0.8,0.99,1). This casts doubts on Sutton's statement, and therefore is required to be tested (Hypothesis 1: - "With asymptotic predictions for a training set of the ideal prediction, Averaging error over training sets,the performance improved rapidly as λ was reduced below 1 (the supervised-learning method) and was best at λ = 0 (the extreme TD method)."*

It is possible, given more data, a high initial w and low λ can converge closer to the ideal prediction. fig (9) (c), It might be useful if computation ability limited, data-set is large and based on past experience one is required to start with high w. With increase in experience, per test set, model came closer to the ideal prediction, over a range of λ. specifically lower λ (like 0) are performing comparatively better, opposed to the low experience procedures. Therefore these observations support Sutton's statement, with a change. For any initial w, given a *"significant"* amount of experience. i.e with asymptotic predictions for a training set of the ideal prediction, averaging error over test sets,the performance improved rapidly as λ was reduced below 1 (supervised-learning) and was best at λ = 0 (the extreme TD method). *Even though Sutton's statement is technically correct, in real life, it is not possible to asymptotically predict due to limited information/experiences, therefore, it may not be possible to achieve predictions close to ideal predictions, for high initial weights as it would be impractical. Hence, initial weights should be carefully selected to make best use of the available data. In this case the weights of concern are "high" but the effect is due to low difference between the desired terminal state and non terminal states.*

6) Different random sequences set: As observed, in fig (10), for 5 different seeds the error over λ values vary. It is possible that Sutton had very specifically generated test sets logic for which is not shared, as it is observed the different patterns can procure slightly different weights. However, the pattern largely remains the same, that is λ=0 gives best performance as compared to the higher λ and which decreases towards λ =1.

Conclusion

The Approach of TD methods as stated by Sutton, based on the observations made through the exploratory analysis, can be said to learns more efficiently than supervised learning, for a general case. However, vagueness in defining the certain assumptions in his paper, added challenges to replicate the his results. Post exploration it can be concluded that the same parameters may not work for process, and therefore these parameters such as learning rate, and convergence threshold could be identified as highly dependent on the nature of the process being explored. Additionally even for the same process, there can an inherent stochasticity in the experiences, which may make effect the learning. And there, it presumably the reason that Sutton in his original work does not provide clarity in defining specification for these terms, as the motivation through the example was to give an initial general idea about the benefits of using TD algorithms for any dynamical process.

References

[1] Sutton, R.S. Learning to predict by the methods of temporal differences. Mach Learn 3, 9–44 (1988).

Github:

Url : https://github.gatech.edu/apanchal33
hashcode : 8c36b5298e71df130d891fcb48d5d99cfd56eecf