# Temporal Difference Method $TD(\lambda)$

1. Theory

## 1.1 Supervised Learning

Consider a target scalar $z$, an observation vector $x$ (or matrix $X$ with multiple steps), a modifiable parameter vector $w$, and a prediction model constructed by the observation and parameter vector $P(x, w)$

Here we define an error function which will be minimized by modifying vector $w$

$$Error = \frac{1}{2}(z - P(x, w))\text{^}2$$

Use gradient descent to minimize this error function, define learning rate $\eta$, we get updating equation

$$\Delta w = w_{new} - w_{old} = \eta(z - P(x, w))\nabla_w P(x, w)$$

In multiple steps observation, it could be written as

$$\Delta w = w_{new} - w_{old} = \sum_{t=1}^{m} \eta(z - P(x_t, w))\nabla_w P(x_t, w)$$

We can rewrite this equation by representing $z$ as a summation of temporal differences $TD_t = P(x_{t+1}, w) - P(x_t, w)$ between each time step

$$z - P(x_t, w) = \sum_{k=t}^{m} TD_k$$

$$\Delta w = \eta \sum_{t=1}^{m} \left( \sum_{k=t}^{m} TD_k \right) \nabla_w P(x_t, w)$$

$$= \eta \sum_{k=1}^{m} TD_k \left( \sum_{t=k}^{m} \nabla_w P(x_t, w) \right)$$

| | | | | | |
|---|---|---|---|---|---|
| $\nabla_w P(x_m, w)$ | | | | | *1* |
| … | | | | … | … |
| $\nabla_w P(x_{t+2}, w)$ | | | *1* | … | *1* |
| $\nabla_w P(x_{t+1}, w)$ | | *1* | *1* | … | *1* |
| $\nabla_w P(x_t, w)$ | *1* | *1* | *1* | … | *1* |
| | $TD_t$ | $TD_{t+1}$ | $TD_{t+2}$ | … | $TD_m$ |

We can also update this $\Delta w$ incrementally $\Delta w = \sum_{t=1}^{m} \Delta w_t$

$$\Delta w_t = TD_k \sum_{k=1}^{t} \nabla_w P(x_k, w)$$

1.2 $TD(\lambda)$

If we are learning in a non-stationary environment, it is natural to set higher weight on those time step that is close to current step. Based on this idea, we can introduce a type of weighting factor $\varphi_k = \lambda^k$

Plug back to the incremental equation above, we get

$$\Delta w_t = TD_k \sum_{k=1}^{t} \lambda^k \nabla_w P(x_k, w)$$

$$= TD_k(\nabla_w P(x_t, w) + \sum_{k=1}^{t-1} \lambda^k \nabla_w P(x_k, w))$$

2. Random Walk using $TD(\lambda)$

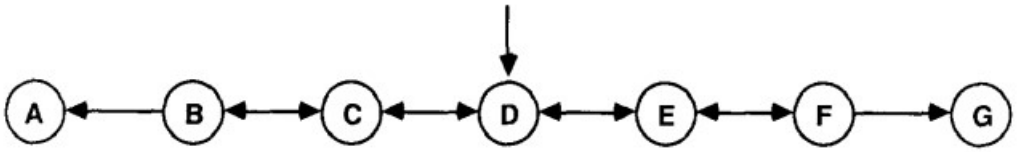



Figure.2.1 A generator of bounded random walks. All walks begin in state $D$. The walk has a 50% chance of moving either to the right or to the left in $B, C, D, E, F$. If either edge state $A, G$ is entered, then the walk terminates.

In this question, we want to evaluate the probability of each non-terminal states $B, C, D, E, F$ that could terminate at stage $G$ using $TD(\lambda)$. We also want to compare the computational efficiency of each $(\lambda, \eta)$ pair.

Using sequence generator to generate 100 training sets, each training set has 10 sequences, each sequence could be treated as an observation.

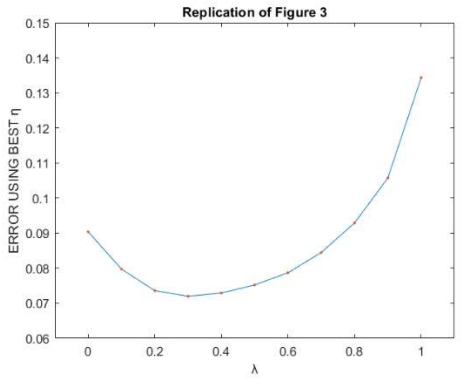
Here we initialize the parameters for $TD(\lambda)$:

- $\lambda = [0.1\ 0.2\ 0.3\ 0.4\ 0.5\ 0.6\ 0.7\ 0.8\ 0.9\ 1.0]$
- $\eta = [0.00\ 0.05\ 0.10\ 0.15\ 0.20\ 0.25\ 0.30\ 0.35\ 0.40\ 0.45\ 0.50\ 0.55\ 0.60]$

Initialize unbiased probability $P_{initial} = [0.0\ 0.5\ 0.5\ 0.5\ 0.5\ 0.5\ 1]$
Notice that for $\eta = 0.00$, there is no update in $TD(\lambda)$ process, it could be treated as a control group.

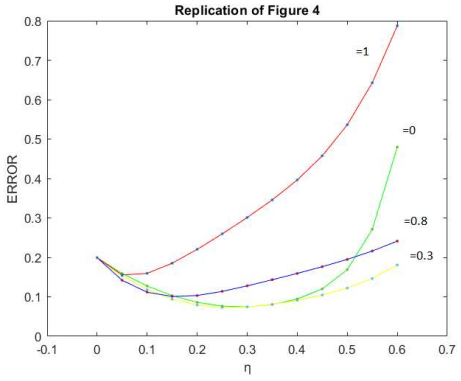
In the first experiment (Figure 3), we use repeated presentation method, which means we would update weight vector after complete presentation of a training set until the procedure no longer produced any significant changes in the weight vector (convergence). The convergence criteria are $|\Delta w| < 0.05|w - w_{exac}|$ in this case.

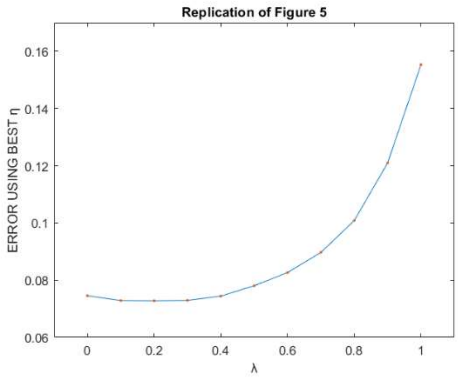
- Note that for improving computational speed, we would break the convergence loop if $|\Delta w| > |w_{initial} - w_{exact}|$, this condition means the learning process would always be better than 'learning nothing'.




In the second experiment (Figure 4 & 5), the training set is presented just once rather than repeatedly until convergence, and we would update weight vector after each sequence has been presented as introduced in theory part.

Figure 4 was generated as the result. It shows the performance of each $(\lambda, \eta)$ pair. Once this has done, we would generate Figure 5 using the best $\eta$ selected from the previous calculations.




These figures are good replication of those shown in paper with same trend but little numerical difference. It is because the learning process is highly dependent on the training sets. If we give the agent different sets, even if all of them have good quality, the number could be different.

**Reference**

Sutton, R.S (1988) Learning to predict by the methods of temporal differences

Sutton, R.S, Barto, A.G (2020) Reinforcement Learning: An Introduction