CS-8803 RLDM Project 1: Temporal Difference Learning (Sutton 88)

Nikhil Gajendrakumar
Georgia Institute of Technology
Master of Science in Computer Science
Email: nikhil.g@gatech.edu

The goal of this project is to experimentally prove that, Temporal-difference methods provide better prediction accuracy that conventional prediction-learning methods in multi-step prediction problems. For this, we replicate the results of Random Walk experiment discussed by Sutton in his paper, Learning to Predict by the Methods of Temporal Differences [1].

1 Temporal-Difference Learning

Temporal-Difference (TD) methods are a class of incremental learning procedures specialized for prediction problems. Conventional prediction-learning methods assign credit by means of the difference between predicted and actual outcomes, TD methods assign credit by means of the difference between temporally successive predictions. TD methods are sensitive to changes in successive predictions. In response to increase/decrease in prediction at time 't' to prediction at time 't+1', predictions for all the previous observation vectors are altered, with greater alterations being made to more recent predictions. In particular, an exponential weighting with recency is considered, in which alterations to the predictions of observation vectors occurring k steps in the past are weighted according to λ^k for $0 \leq \lambda \leq 1$ TD learning is a model-free method that learns from incomplete episodes, by bootstrapping (substituting the reminder of the trajectory with our estimate of what might happen from that point onwards.) All learning procedures will be expressed as rules for updating ω. For each observation, an increment to ω, denoted $\Delta\omega$, is determined.

$$\omega \leftarrow \omega + \sum_{t=1}^{m} \Delta\omega \tag{1a}$$

$$\Delta\omega = \alpha(P_{t+1} - P_t) \sum_{k=1}^{t} \lambda^{t-k} \nabla_{\omega} P_k \tag{1b}$$

$$P_t = \omega^T x_t \tag{1c}$$

For $\lambda < 1$, TD(λ) produces weight changes different from those made by any supervised-learning method. The difference is greatest in the case of TD(0) (where λ = 0), in which the weight increment is determined only by its effect on the prediction associated with the most recent observation:

$$\Delta\omega_t = \alpha(P_{t+1} - P_t) \nabla_{\omega} P_t \tag{2a}$$

2 A random-walk example

Sutton [1] describes the simplest of dynamical systems, to prove the advantages of TD methods over conventional prediction-learning methods, called bounded random walks problem. Random walks problem can be represented as a Markov decision process with states A and G as the termination states, and states B, C, D, E, F as non-termination states. A bounded random walk is a state sequence generated by taking random steps to the right or to the left until a terminal state is reached. Every walk begins in the center state D. At each step the walk moves to a neighboring state, either to the right or to the left with equal probability. If either terminal state (A or G) is entered, the walk terminates. For each non-terminal state i, there was a corresponding observation vector x_i; if the walk was in state i at time t then $x_t = x_i$. Thus, if the walk DCDEFG occurred, then the learning procedure would be given the sequence $x_D, x_C, x_D, x_E, x_F, 1$. The vectors x_i were the unit basis vectors of length 5, that is, four of their components were 0 and the fifth was 1 (e.g., $x_D = (0,0,1,0,0)^T$), with the one appearing at, a different component for each state. The transition probabilities $\forall i \in$ non-terminal state $j \in$ any state $P(x_{t+1} = j | x_t = i) = 0.5$. Sutton assumes that the terminal states have a single transition to themselves with probability 1.

3 Experiment

Sutton [1] describes two different computational experiments (repeated presentations and one presentation) to show that TD methods converge faster and produce predictions with higher accuracy than the conventional prediction-learning methods. In order to obtain statistically reliable results, 100 training sets, each consisting of 10 sequences, were generated as training set for use by all learning methods. The true probabilities of right-side termination, ideal predictions, for each of the states is $[0, \frac{1}{6}, \frac{1}{3}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}, 1]$ for states A, B, C, D, E, F, and G respectively. As a measure of the performance of a learning procedure on a training set, the root mean squared (RMS) error between the procedure's asymptotic predictions using that training set and the ideal predictions has been used. Python has been used to perform the experiment.

3.1 Repeated presentations

Sutton [1] describes that, in this experiment, the weight vector was not updated after each sequence. Instead, the $\Delta\omega$'s were accumulated over sequences and only used to update the weight vector after the complete presentation of a training set. Each training set was presented repeatedly to each learning procedure until the procedure no longer produced any significant changes (ε) in the weight vector. For small α, the weight vector always converged in this way, and always to the same final value, independent, of its initial value.

3.1.1 Results

This experiment was conducted on 100 training sets for each value of λ = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] As we can see from the above Fig. 1, RMS error decreased (or accuracy increased) as λ decreases from 1 to 0, and the best accuracy is obtained at λ = 0. This result completely matches with the result described by Sutton [1].

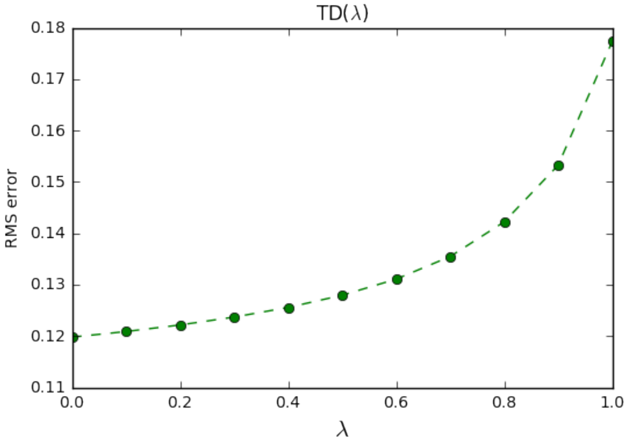

Fig. 1. RMS error of repeated presentation experiment on a random walk problem for above λ values

3.2 One presentations

Sutton [1] describes that, in this experiment, present the same data to the learning procedures, again for several values of λ, with the following procedural changes. First, each training set was presented once to each procedure. Second, weight updates were performed after each sequence. Third, each learning procedure was applied with a range of values for the learning-rate parameter α. Fourth, so that there was no bias either toward right-side or left-side terminations, all components of the weight vector were initially set to 0.5.

3.2.1 Results

As described by Sutton, I have made all 4 procedural changes to the experiment. This experiment is conducted on 100 training sets for each value of $\lambda = [0, 0.3, 0.8, 1]$ As we can see from the above Fig. 2, the value of α had a significant effect on performance. Fig. 2 is almost similar to the results obtained in Sutton [1]. The only difference is, in my result, $\lambda = 0$ performance better over all values of α, but Sutton shows that $\lambda = 0.3$ performs better over all values of α. I think this difference is may be because of the different ε (minimum difference between two consecutive time weight vectors required to stop the procedure) values being used.

3.3 Best error level

Finally, as described by Sutton [1], we find the best error level achieved for each λ value, that is, using the α value that was best for that λ value. To find out the optimal α for each λ value, I did a grid search of all α

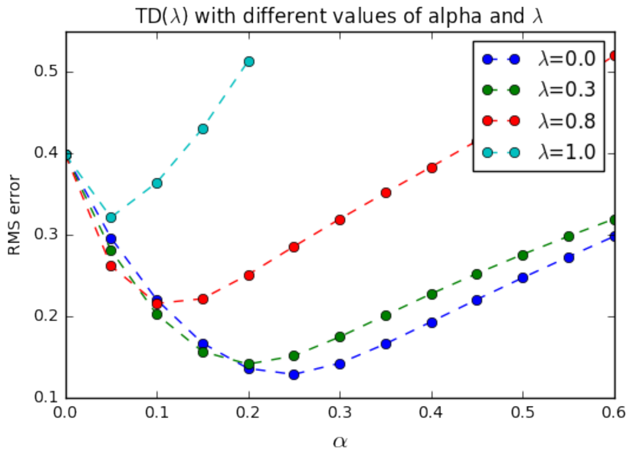

Fig. 2. RMS error of one presentation experiment on a random walk problem for above λ values

values from 0 to 6 at steps of 0.05, and select the α with least average RMS error. I got optimal α = [0.25, 0.25, 0.25, 0.2, 0.2, 0.2, 0.15, 0.15, 0.15, 0.1, 0.05] for λ = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1] respectively.

3.3.1 Results

This experiment was conducted on 100 training sets for each value of λ = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]. As we can see from Fig. 5, RMS error decreased (or accuracy increased) as λ decreases from 1 to 0, and the best accuracy is obtained at λ = 0, but Sutton reports best accuracy for λ = 0.3. I again think this difference is may be because of the different ε (minimum difference between two consecutive time weight vectors required to stop the procedure) values being used.

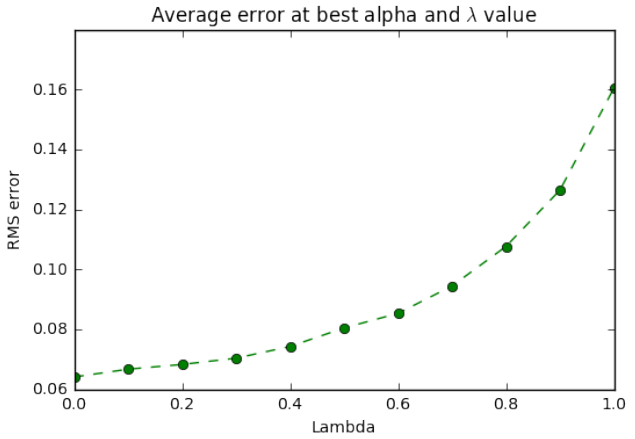

Fig. 3. RMS error on a random walk problem for above λ values with corresponding optimal α value

4 Conclusions

From the results of above 3 experiments, we can clearly observe that TD methods provide better prediction accuracy than conventional-prediction learning methods (example: TD(1)). The results of experiment 2 and 3 deviate a little compared to results given in Sutton, but even with little deviation in result, we prove the TD methods perform better on multi-step prediction problems than other supervised learning procedures.

5 References

[1] Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning, 3(1):9-44, aug 1988.*

6 Link to video presentation

The following github repo has been shared with the TA's (tbail3, ksubramanian6, ahaque34, and mmorales34)
`https://github.gatech.edu/nikhil30/rldm-project-1/blob/master/ProjectPresentation.mp4`
or use the following YouTube link
`https://youtu.be/n5FeLsh-BlU`