Project 1: Replicating Sutton 1988

Myles Lefkovitz
gth836x
GitHub Hash: 6cdce28a998d400369b2b116a82aeae7adbba06b

1. INTRODUCTION

In this project, I replicate the experiments used by Sutton in “Learning to Predict by the Methods of Temporal Differences” [Sutton, 1988]. Sutton performs two experiments and produces a number of figures using a bounded Random Walk as a simple example. His experiments show the performance difference between TD(1) (the Widrow-Hoff supervised learning rule) and TD(λ). TD(λ) outperforms TD(1) in both experiments (for some value of $\lambda < 1$). I successfully reproduce the figures and explain the implementation used to recreate the experiments.

2. RANDOM WALK

A bounded random walk is a random set of states along a given path. The bounded random walk generator used in these experiments is shown in Figure 1 below.

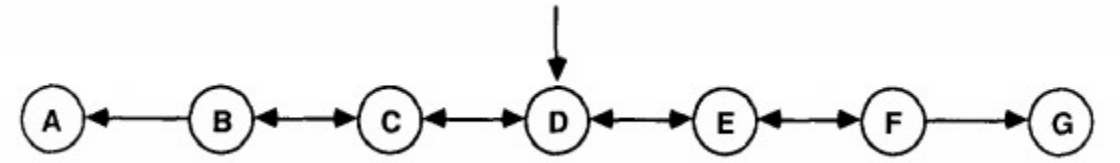

Figure 1. A Random Walk Generator [Sutton, 1988]

A Markov process generates the data sequences uses this generator. The process always starts at state D, then moves randomly (with a 50% chance) to an adjacent state, repeating this process until a terminal state (A or G) is reached. An example walk may be D->E->D->C->B->A.

Each random walk is called a sequence. The experiments below were performed on 100 training sets each with 10 sequences (random walks).

3. TD LEARNING DEFINITION

“Temporal difference (TD) [learning] methods are a class of incremental learning procedures specialized for prediction problems” [Sutton, 1988].

3. 1. DEFINITION IN CONTRAST TO SUPERVISED LEARNING

While other prediction learning methods (like supervised learning) pair a single prediction with an actual outcome, TD learning methods pair a prediction with a successive prediction [Sutton, 1988].

The canonical example that compares TD learning to supervised learning is:

> “For example, suppose a weatherman attempts to predict on each day of the week whether it will rain on the following Saturday. The conventional approach is to compare each prediction to the actual outcome whether or not it does rain on Saturday. A TD approach, on the other hand, is to compare each day's prediction with that made on the following day. If a 50% chance of rain is predicted on Monday, and a 75% chance on Tuesday, then a TD method increases predictions for days similar to Monday, whereas a conventional method might either increase or decrease them depending on Saturday's actual outcome.” [Sutton, 1988]

3. 2. TD LEARNING EQUATIONS

Sutton expresses learning procedures as rules for updating weight *w*. (1) below is the method by which *w* is updated [Sutton, 1988]:

$$w \leftarrow w + \sum_{t=1}^{m} \Delta w_t$$

(1) [Sutton, 1988]

The supervised learning update procedure (2) is [Sutton, 1988]:

$$\Delta w_t = \alpha(z - P_t)\nabla_w P_t$$

(2) [Sutton, 1988]

Converting the supervised learning update procedure (2) to an identical procedure that can be computed incrementally, we get (3) [Sutton, 1988]:

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

(3) [Sutton, 1988]

Adjusting (3) to enable different TD learning rates (λ), we get (4) for 0 ≤ λ ≤ 1 [Sutton, 1988]:

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

(4) [Sutton, 1988]

These λ values adjust the weight of the successive predictive observations. With λ = 1, the update procedure mirrors (3) – all states are weighted equally. With λ = 0, the update procedure adjusts the weight in line with only the most recent prediction.

4. EXPERIMENTS

The experiments used identical (random) inputs: 100 training sets of 10 random walk sequences as described above. The randomizer was seeded to provide identical random walks for each training set for all of the experiments.

The random walks have 7 states [A, B, C, D, E, F, G] as shown in figure 1 above. States A and G are terminal states with a reward of 0 (for state A) and 1 (for state G). The intermediate states (B, C, D, E, and F) offer no reward. As described above, the process always moves randomly (with a 50% chance) to an adjacent state. For convenience, the states were switched from letters to numbers [0, 1, 2, 3, 4, 5, 6] when the algorithm was developed (with state 0 returning 0 reward and state 6 returning 1 reward).

In the experiments, the temporal difference learning algorithm was developed to learn the correct weights for each state based on the randomized input data and the given input variables (λ for both experiments and α for experiment 2). Since we know the true Markov Process for these random walks, we can calculate the correct (true) weights for each of the 5 intermediate states. The true weights are: [1/6, 2/6, 3/6, 4/6, 5/6].

For all experiments, the weight updates are computed using (4) above.

The goal of these experiments is to determine a performance difference between TD(1) (the Widrow-Hoff supervised learning rule) and TD(λ). Performance is calculated by root mean squared error (RMSE). The RMSE is the (square root of the mean of the squared) error between the predicted weights and the true weights for each state. The RMSE is calculated for each training set (of 10 sequences) individually and averaged (over the 100 training sets) to reduce the influence of outliers.

4. 1. EXPERIMENT ONE – REPEATED PRESENTATIONS

The first experiment has the goal of comparing a variety of different lambda values (λ) in the TD(λ) equation once each process has been allowed to run until convergence. The λ values used are 0, 0.1, 0.3, 0.5, 0.7, 0.9, and 1.

In this experiment the weight vector is not updated after each sequence, but instead is updated after each full training set (of 10 sequences). Each training set is repeated until the change in weight was smaller than some chosen value ε (in this case 0.01).

An alpha value of 0.01 was selected, which allowed convergence in all trials.

4. 2. EXPERIMENT TWO – LEARNING RATE ON A SINGLE PRESENTATION

The second experiment has the goal of comparing a variety of different lambda values (λ) and alpha values (α) in the TD(λ) equation with a single presentation of each set of data. The λ values used are 0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, and 1.

In this experiment the weight vector is updated after each sequence.

5. OUTCOMES

I was able to successfully replicate both experiments as described in Sutton's paper. The development occurred in Python 3.6. I implemented each algorithm as described above. Below are the output figures and my findings.

5. 1. EXPERIMENT ONE – REPEATED PRESENTATIONS

In the repeated presentations experiment, Sutton created a figure (Figure 3 in [Sutton, 1988]) that I have successfully replicated below (Figure 2).

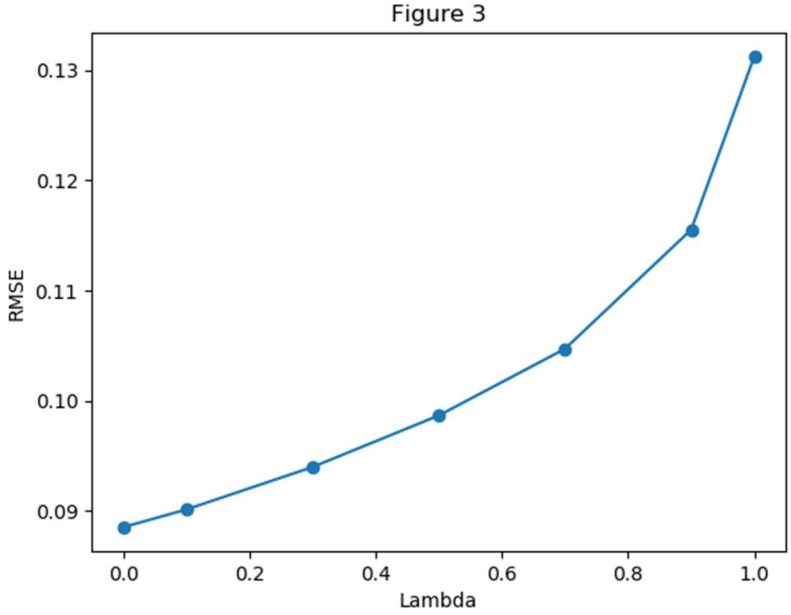

Figure 2. λ vs RMSE – a replication of Figure 3 from Sutton, 1988

In Sutton's Figure 3, we see a similar trend to that in the figure above: RMSE increases with λ. RMSE is lowest where λ = 0. The range of RMSE values (0.08 – 0.13) in the figure above differs from that

in Sutton's paper (0.19 – 0.25). This difference can be accounted to the difference in alpha values (not stated in the paper) and randomized nature of the experiment.

5. 2. EXPERIMENT TWO – LEARNING RATE ON A SINGLE PRESENTATION

In the single presentation experiment, Sutton created two figures (Figure 4 and Figure 5 in [Sutton, 1988]) that I have successfully replicated below (Figure 3 and Figure 4).

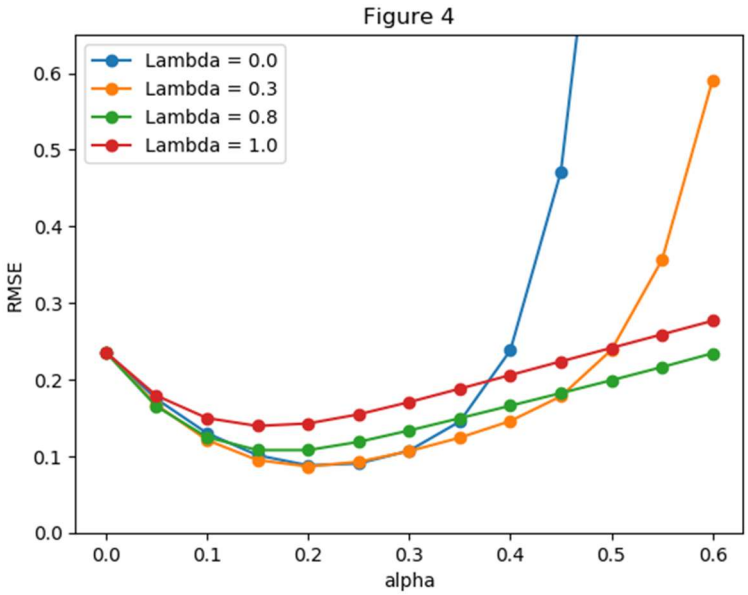

Figure 3. α vs RMSE for selected values of λ - a replication of Figure 4 from [Sutton, 1988]

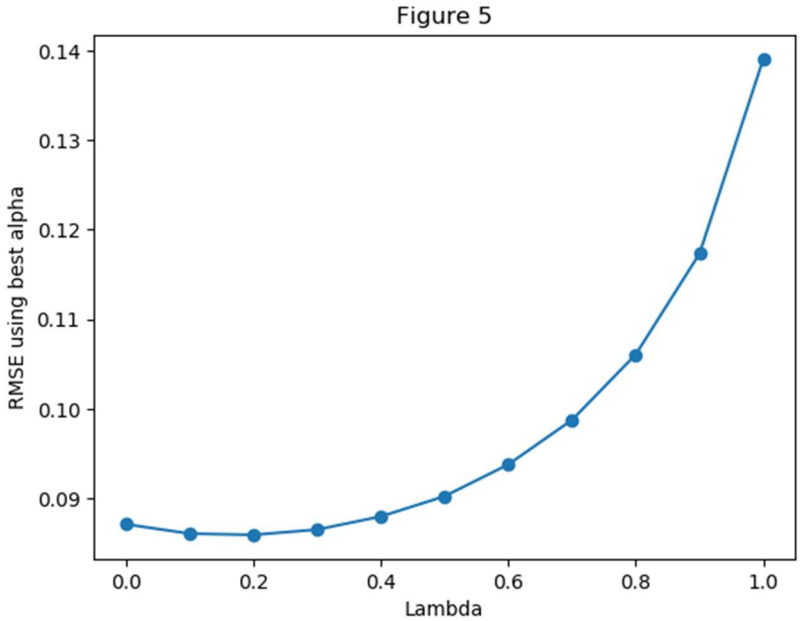

Figure 4. λ vs RMSE with optimal α values – a replication of Figure 5 from Sutton, 1988

For this experiment, many α values were selected. Figure 3 above (a replication of Sutton's Figure 4) shows several representative λ values with 13 α values. It's clear from this figure that each TD(λ) has its error minimized with α between 0.1 and 0.3, usually around 0.2. The figure above is similar, but not identical to the corresponding figure in Sutton's paper. First, TD(0) and TD(0.3) have much higher

errors for large alpha values in my figures than they do in Sutton's paper. This error is likely caused by the random nature of the experiment.

Figure 4 above (a replication of Sutton's Figure 5) shows the minimum RMSE for each of 11 λ values. 18 α values were compared for each value of λ. The figure shows the minimum RMSE for any λ and α combination at 0.859 with $\lambda = 0.2$ and $\alpha = 0.21$. The figure I produced is a fairly close match to the original in Sutton's paper. The biggest differences being the range of errors (0.08 – 0.14 vs 0.10-0.21). I attribute this error difference to randomness and differences in input α values.

REFERENCES

Sutton, R. S. (1988). Learning to predict by the methods of temporal differences. *Machine learning*, *3*(1), 9-44.