Project 1

Nicolas Cofre
nicolas.cofre@gatech.edu

Git hash of last commit

ea488799ffd31ec0e4e02ccde8038baa9a
da259c

***Abstract*—Replication of Sutton 1988 figures 3, 4 and 5. Also some additional minor insights.**

I. The problem

The problem used by Sutton was a bounded random walk with 7 states named: A, B, C, D, E, F and G. The states A and G were the terminal states and Sutton refered to A as the left bound and G as the right bound. The process started in the state D, which is the center, and with a 50% probability the process moved to the right or to the left, finishing when reaching A or G. The rewards were defined as 0 for all transitions except for the transition to G, which gives a reward of 1. This definition implies that the expected reward for each state is equivalent to the probability of reaching the right bound or state G. One realization of this process, starting at D and finishing at A or G defines a sequence.

II. Learning algorithm used: TD(λ)

Given a set of observations $x_1, x_2, x_3, \dots x_m, z$, where x_t represent non-terminal states and z the final outcome of the sequence, we want to predict build predictions P_t of z, for this problem, the prediction model is simply $P_t = w_t^T x_t$ and therefore, we have $\nabla_w P_t = x_t$ which gives us the change in the prediction when we change the underlaying weight w. For each step t in the sequence we generate a weight change according to the equation 4 of Sutton´s paper:

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

When $t = m$ then $P_{t+1} = z$.
In the following experiments we will be accumulating this weight changes over an entire batch or training set of sequences and only changing w after the entire batch has been processed (Experiment I) and also updating the w after each sequence (Experiment II). Note that w remains the same intra-sequence.

III. Experiment 1

In the first experiment, Sutton creates 100 training sets, each one consisting of 10 sequences.

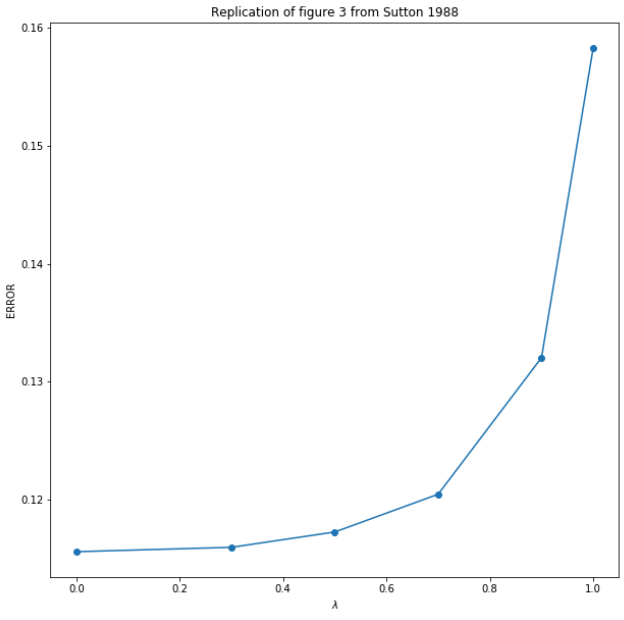
The observation vector was defined as a vector of 0 with a 1 in the component of the state. In this case we have 5 non-terminal states, so for instance the initial observation vector or position D is $x_D = (0,0,1,0,0)$ and $x_F = (0,0,0,0,1)$. According to Sutton´s definition G or A do not need an observation vector as they are terminal states.

The weight vector w together with the observation vector gives the prediction. So if at step t, the position is D and the weight vector is $w = (1,1,1,1,1)$, the prediction is $P_t = w^T x_D = 1$.

An estimated prediction was calculated for each training set, for a given λ and $\alpha = 0.01$. The update of w was done after processing all the sequences in the training set, keeping constant w while processing a batch of sequences in the same training set. This procedure was repeated, presenting again the same training set till no further changes in the predictions were seen, the convergence criteria used was:

$$\max(|\Delta w|) \leq 10^{-5}$$

Then for each training set, the prediction was compared to the actual probabilities and the RMSE was computed, then I stored the average of the RMSE over all the training sets. This process was repeated for different levels of $\lambda \in \{0,0.3,0.5,0.7,0.9,1\}$, giving a series of averages of RMSE for each λ. The results can be seen in the following figure,



The conclusion from this figure is the same as Sutton´s, in batch training $\lambda = 0$ gives us the best prediction with respect to the true probabilities. $\lambda = 0$ is not trying to minimize error in the training set, it is giving us the estimates that are consistent with the underlaying Markov process. My intuition

is that $\lambda = 0$ is less likely to overfit compared to $\lambda = 1$ and that explains the better performance when comparing with the true probabilities. A different comparison is the performance in the training set (comparing with the outcome in the training set rather than using the true probabilities), which is what $\lambda = 1$ minimizes.

One of the main problems here was the convergence, a convergence criteria that was not tight enough, for instance,

$$\max(|\Delta w|) \leq 10^{-1}$$

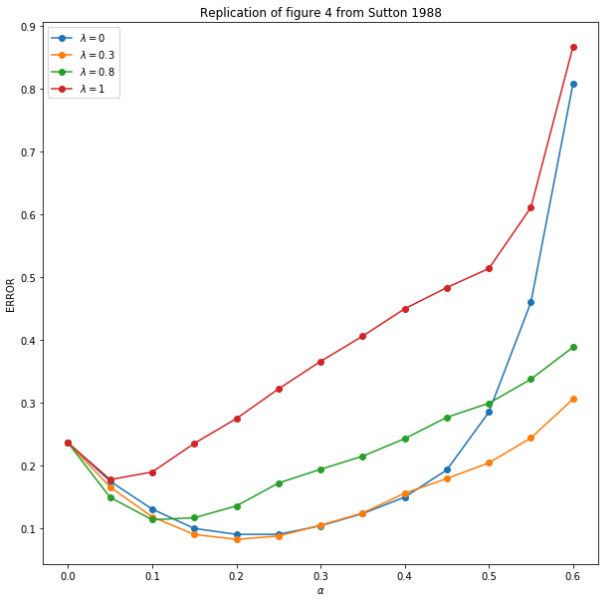
produced very unstable results. Another issue was the selection of the learning rate α, which causes problems if not small enough, specially for bigger λ. The following example illustrates the issue, suppose $\lambda = 1$ and $\alpha = 1$. If a state is visited multiples times, let´s say state E, then the summation term in the learning equation 4 in Sutton 1988 will grow very fast. If the state E is visited 3 times, then x_E will appear 3 times in the following summation of equation 4 of the paper,

$$\sum_{k=1}^{t} \nabla_w P_k = \sum_{k=1}^{t} x_k$$

So there would be an excessively big correction for that state. This is solved choosing a small α, which in this case was $\alpha = 0.01$. Also, to avoid the effect of very unlikely sequences and the accumulation described above, I have used the median instead of the mean over the training sets.

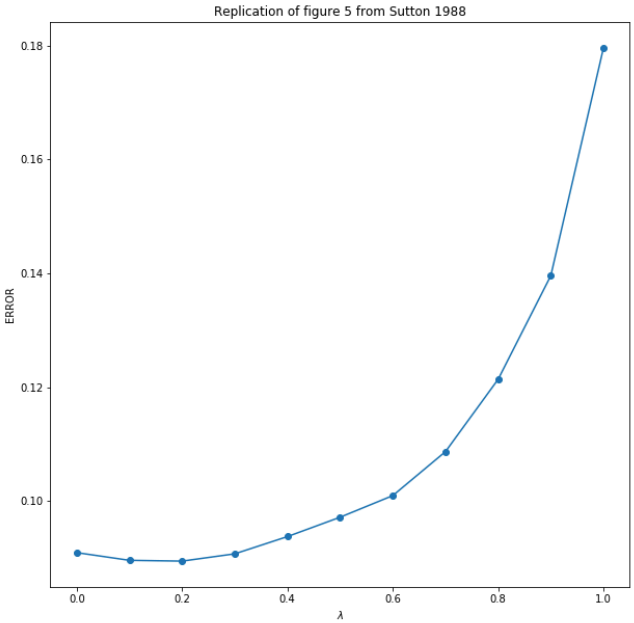
IV. Experiment 2

In this case, the update of *w* was done after each sequence and the training set was presented only once. The main problem was that the error was way higher than in the paper for the region of high α. My explanation for this is similar to my example regarding the problem in the first experiment, if a state is visited multiple times, the update will be excessive if the learning rate is too big. Thus, as we increase α, it is more likely that very unlikely sequences with multiples repeated states overcorrect when they are used only once to update *w*. For instance if we have a sequence including a D,E,D,E,D,E…x_E and x_D will appear multiples times in the summation of equation 4 of Sutton´s paper making the update very big and so the error due to overcorrection. To remove the effect of this very unlikely sequences I have used the median instead of the mean of the RMSE.



As in the paper we can see that $\lambda = 1$ is the worse performer for all the levels of the learning rate. Also we can see that the best learning rates for $\lambda = 0.3$ and $\lambda = 0$ are very close to 0.2, similar to Sutton´s results. Here we can also see the problem with $\lambda = 1$ and a high learning rate. Given the fact that we are correcting all the visited states without any decayment, a high learning rate leads to overcorrection and therefore to high errors in all the predictions for all the states.

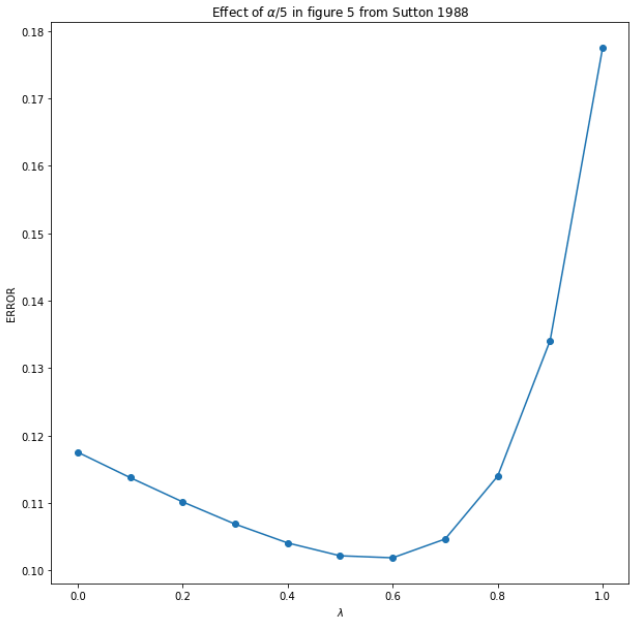
The figure 5 of the paper of Sutton is based on the same experiment but with more values of λ. For each value of λ we store the best error, among those using the alpha candidates of the figure 4. The results are the following,



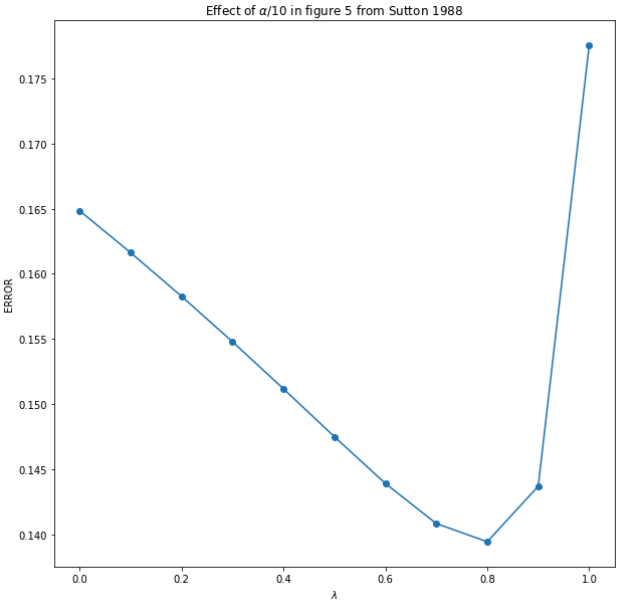
We can see that the best λ is around 0.2, not far from what Sutton mentioned which was 0.3.

As mentioned by Sutton, $\lambda = 0$ is not the best option when we present the training set only once, due to the fact that we will be updated only the current or visited state and the rest of the states visited previously would not be updated at all. When $\lambda > 0$ those states previously visited are updated, despite the

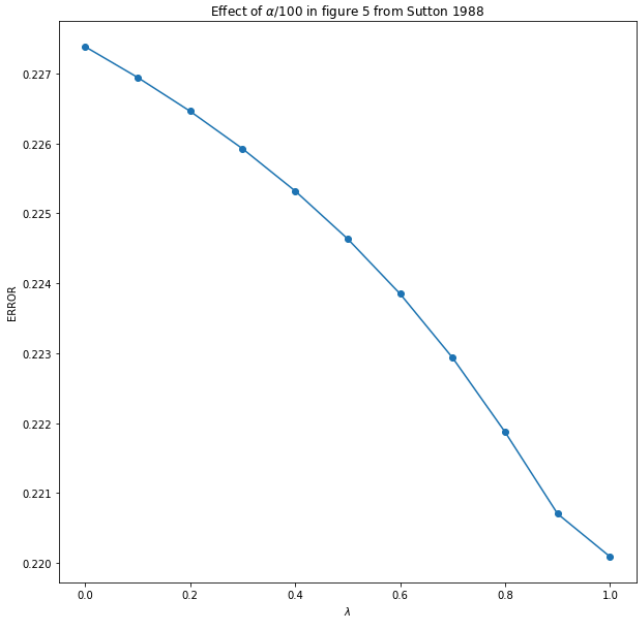
fact that this update can be very minor compared to the currently visited states if they were visited long ago. My explanation of why bigger λ values are not optimal is because they would require a way smaller learning rate as the one I have used in the experiment 1 to be competitive. Thus for the given learning rates the optimal λ is around 0.2. If we decreased the learning rate, diving them by 10 or 100 then probably the optimal λ would shift. We can see this in the experiment below in which I have modified the α using $\alpha' = \alpha/5$ and $\alpha' = \alpha/10$.



As expected, when we decrease the learning rate, the rate of propagation of the update must be higher affecting more previous visited states. We can see in the previous figure that the new optimal λ is around 0.6.



If we further decrease the learning rates, we can see that the behaviour continues as expected, an even lower learning rate makes it better to affect more previous visited states with an increased $\lambda = 0.8$. This can continue if we make the learning rate excessivily small as below,



V. Conclusion

The results of Sutton appear to be fairly replicated, Sutton does not explicity talk about the convergence criteria used or about the divergence for bigger learning rates, we can see evidence of that behaviour in his figure 4, where for $\lambda = 1$ the higher learning rate showed is approximately 0.4 while for the other curves is 0.6. Also, Sutton does not talk on the relation between λ and α, for smaller α the optimal λ is bigger, at least in the bounded random walk example.

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