Replication of Sutton 1988

Siyuan Li

Date: 2019/09/29
Uni: sli761
Git hash: 5a8a011818d616deba62d162a6b55555a2dc022d
Github repo:
https://github.gatech.edu/gt-omscs-rldm/7642Fall2019sli761/tree/master/project_1

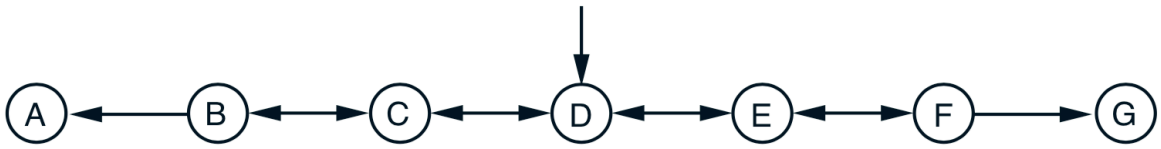


Figure 1: The random walk we used to study. In each node, P(left)=P(right)=50%. Reach G or A will terminate the walk. From Sutton's paper, we know for each node of BCDEF, the theoretical probability of getting G is $[\frac{1}{6}, \frac{1}{2}, \frac{1}{2}, \frac{2}{3}, \frac{5}{6}]$. As this is not what the key point in this project, we just take the probability as given without detailed illustration.

Graph 3

Target

To find the relationship between value of λ and the prediction accuracy of TD(λ) method

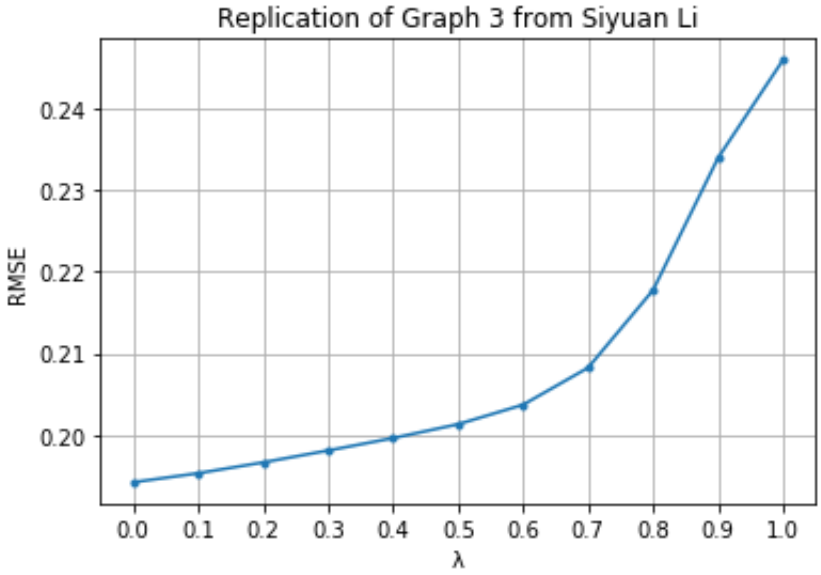
Experiment

Simulate the random walk specified in Figure 1 and use TD(λ) to calculate predicted probability. Compare the result generated by different λ with the theoretical value and use Root Mean Square(RMS) to evaluate the performance of different λ.

Steps

1) Simulate the random walk in Figure 1 and record each step. This will generate a 100*10 matrix with each component record a complete walk.
2) For each component, use the equation (4) given in Sutton 1988 paper to calculate Δw. Accumulate it and get the final weight, which is considered as our predicted value.
3) Compare the predicted value with theoretical value and generate the RMS.
4) Plot the RMS and do the evaluation between different λ.

Replicated Graph

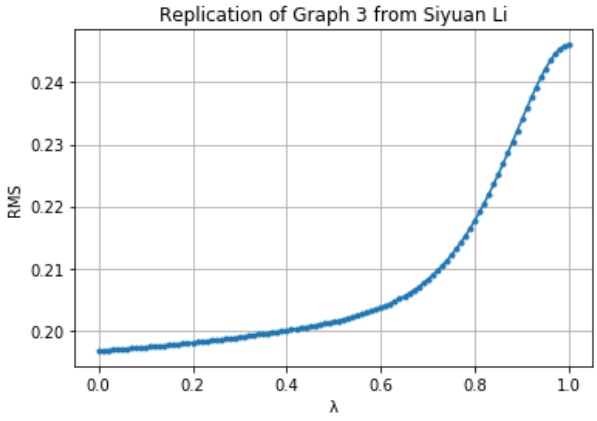


Result Discussion

From the replicated Graph 3, we can see that with the reduction of λ from 1 to 0, the error of the prediction is decreasing. In my replication, I used λ=0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 and 1.0 and the result is still similar to the one in Sutton's paper. I also did a 0.01 stepwise λ selection to see the trend and the result is also similar, as shown in the graph below. Please note, code for this part was not saved in pr1.py file.

Problems

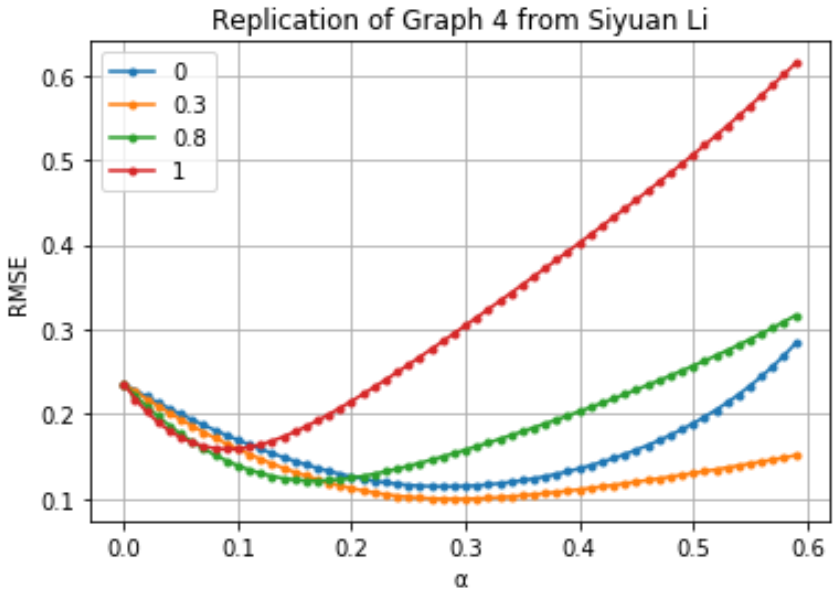
For Graph 3, I did not encounter very tough problems. It took me some time to understand the steps of the experiment but it went through after careful reading.



Graph 4

Target

To find the relationship between the predicted error and the learning rate(α)



Experiment

Keep the simulation result the same and use different learning rate to do the prediction. Compare the predicted result with the theoretical value and evaluate the performance.

Step

1) Simulate the random walk and anchor it
2) Feed the random walk result to different learning rate and predict the value
3) Use different λ to show the relationship under different λ.

Replicated Graph

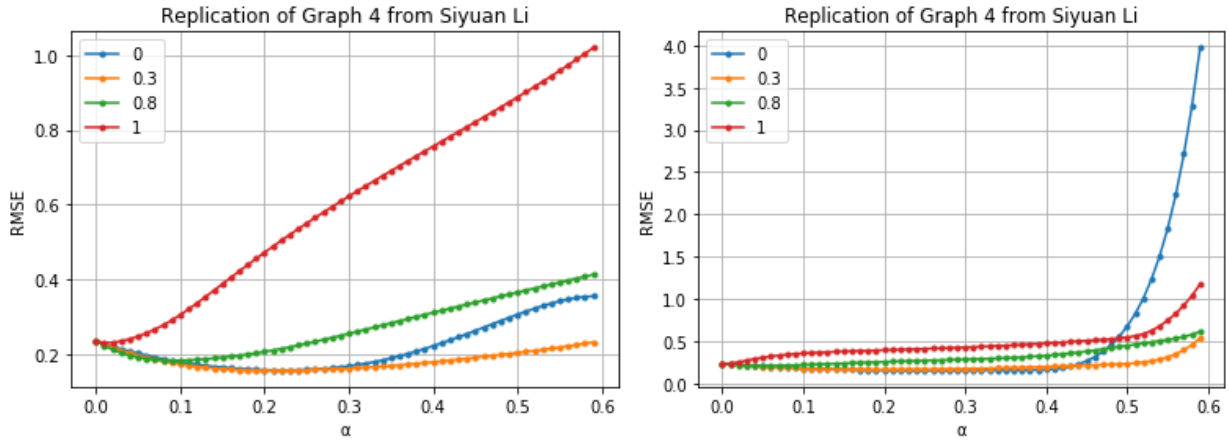
Shown as above.

Result Discussion

The result shown above has very similar performance with the one shown in Sutton's paper. The learning rate has significant impact on the learning performance. For most λ, the relationship between α and RMS is like a U-curve, with best performance at the intermediate value.

Problem Encountered

When I was trying to replicate this graph, I saw some different graphs in different run, like the following two graphs.



I think this is caused by the randomness of the simulation. For further improvement, running the experiment with different simulation result and use the averaged RMS may solve this problem. However, due to time limit, I don't have enough time to improve this experiment. I will follow up this point afterwards.

Graph 5

Target

To find the best error level under each λ level

Experiment

Keep the simulation result the same and use different λ to do TD(λ) prediction. Under each λ level, use multiple learning rate(α) to predict and record the smallest error level.

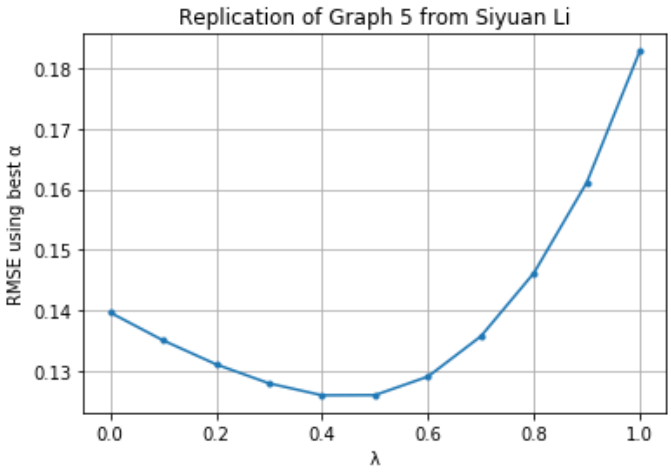
In my implementation, I used 0.01 as step length to go over each α from 0-0.6 (corresponding to the range specified in graph 4)

Step

1) Generate simulation result matrix (same as Graph 3 and 4)
2) For each λ in [0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,1.0], loop through α from 0-0.6 with step length as 0.01
3) Record the smallest error for each λ

4) Plot

Replicated Graph

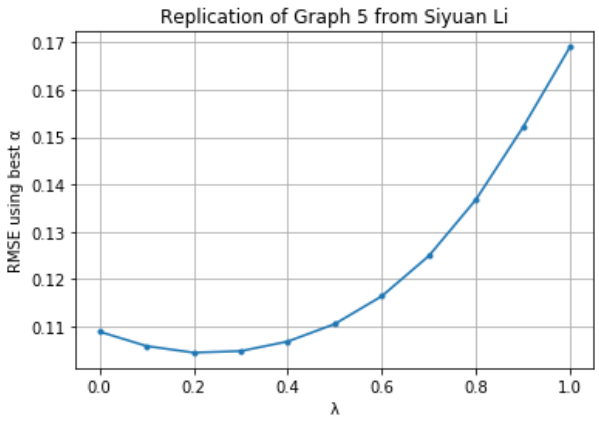


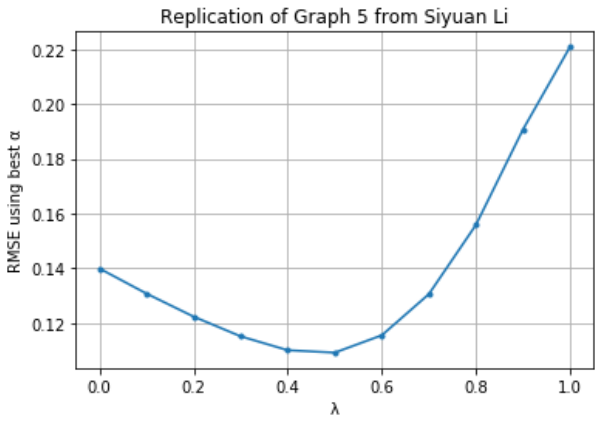
Result Discussion

The graph shown as above has very similar appearance as the one in Sutton's paper. Both the shape of the curve and the scale of the value are very similar. This is as expected. Although some randomness exists in Graph 3 and 4, Graph 5 captures the lowest level of each λ by calculating the prediction value 60 times. This implementation can highly lower the level of randomness.

Problem Encountered

When I tried to implement this Graph 5, the most concern is how to balance the step length selection of α and the running time. I tried 0.001-0.005 but finally select 0.01 in the final version due to time limit. This actually resulted some randomness (see 2 graph below) but it is acceptable considering their similar shape and value range.





Reference

[1] Learning to Predict by the Methods of Temporal Differences. Richard S. Sutton 1988
[2] Github: https://github.com/vjp23/TD_Lambda by Vince Petaccio
[3] Github: https://github.com/conge/RLFinalProject by conge