The Eberhard Karl University of Tübingen

Ahmed H. Abdelrazik

Neural Modeling Course

Instrumental Learning

Dec 2023

Course Instructor Prof. Peter Dayan

Introduction:

The objective of this report is to investigate the outcomes of instrumental conditioning, as outlined in Section 9.3 on Instrumental Conditioning. In order to achieve this, I have reproduced Figures 9.4, 9.6, 9.8, and 9.9. Although not all of these results were necessary for the replication, they were easier to comprehend the effect of changing model parameters such as the learning rate and β on the overall performance of the agent before implementing a more complex environment. The reproduction process involved replicating the Indirect Actor results from the textbook as the first step, followed by reproducing the Direct Actor results, Policy Evaluation results, and finally the Actor-Critic Policy Improvement results. Our analysis and discussion will primarily focus on examining the outcomes of both policy evaluation and actor-critic approaches.

Results and Discussion

Indirect Actor

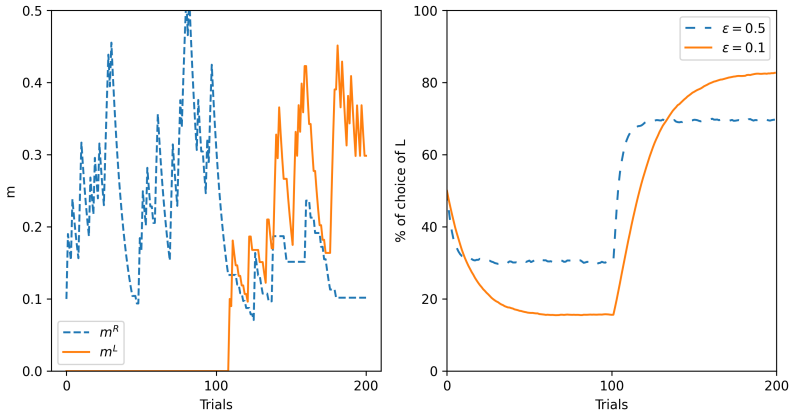

Figure 1: Indirect Actor

The indirect actor utilizes the delta rule to learn the appropriate action based on the policy probability in the form of a soft-max function. By adjusting the action variable, denoted as "m," using the delta rule, we were able to increment it with an error term, delta, reflecting the discrepancy between the received reward and the chosen action. Consequently, if no reward was obtained despite taking an action, the value of delta became negative, thus the action value m^a decreases. Conversely, at the beginning of the training, when a reward was received and an action was taken, we incrementally adjusted m^a towards the desired direction using a step size, epsilon, and this action becomes more probable to be sampled from the policy distribution.

Figure 1 shows the results of the indirect actor, in the experiment the reward distribution was $r^R = 0.25$ and $r^L = 0.05$ for the right and left actions, respectively. This reward were switched at trial = 100, which caused a shift in the response of the action value in the left panel. The value of β is 10, the learning rate is 0.1.

The ability of an agent to learn how to switch between lever mainly depends on two parameters, the learning rate and the β which is the exponent coefficient of the soft-max function. If β is so large, it amplifies the difference between mL and mR which makes the policy probability either zero or one for negative or positive difference, hence the policy is almost deterministic and the agent behavior is mostly exploitative. If β is relatively small, it allows the agent the opportunity to explore more.

Effect of changing the learning rate

In Figure 1, the right panel shows the percentage of choosing the left action for two different learning rates $\epsilon = 0.5$ and $\epsilon = 0.1$. We can see that for $\epsilon = 0.5$, the learning process becomes faster leading to faster convergence but accompanied by a steady-state error. On the other hand, for $\epsilon = 0.1$, is gradually converging, but the percentage of choosing the left action is more accurate on the expense of learning

time. Also, using small value of the learning rate becomes effective when the reward distribution changes or alters slowly relative to the learning rate. In such cases, the learning rate allows for faster adaptation to track the evolving reward distribution.

Effect of changing the β

In the previous results, the value of β was 10. We expect that if we increase the β the exploitation will be more dominant than the exploration. Figure 2 shows the results when we increase β to 100. We see the lag of adaptation once we switch the reward in both the action values on the left panel and the percentage of choosing the left action on the right panel.

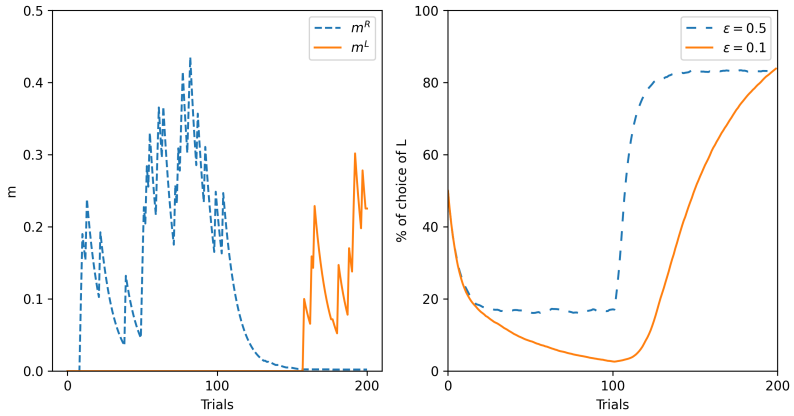

Figure 2: Indirect Actor with $\beta = 100$

Policy Evaluation

Figure 3 shows the result of the policy evaluation. As expected, the value function converges to the value of expected rewards. That is $\nu(u_3) = 0.5(0+4)$, $\nu(u_2) = 0.5(8-8) = 0$, and $\nu(u_1) = (1/3)(1+2+(-1+0.5(\nu(u_2)+\nu(u_3)))) = 1$.

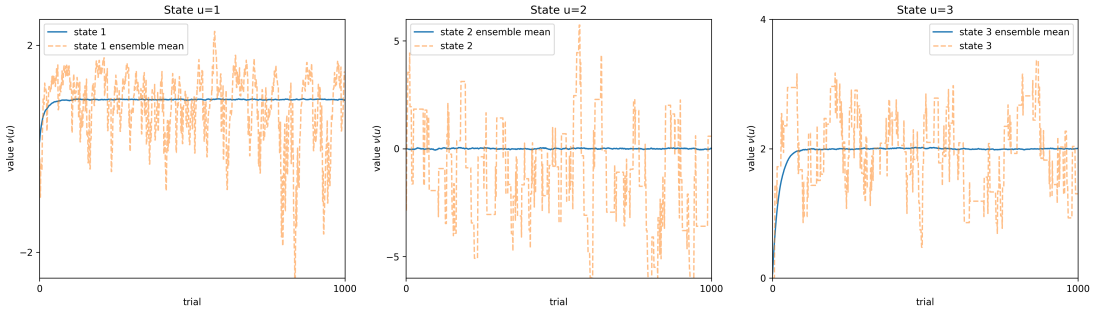

Figure 3: Policy Evaluation

In our experimental setup, once the monkey presses lever C at the first state, after receiving a shock, it takes the monkey with a %50 percent probability to either state 2 or 3. If we change this transition probability, we expect the average expected reward to also change which directly changes the value function steady-state values.

Fig 4 shows this results with Policy Evaluation, with probability of switching between u2 and u3 of (0.2,0.8), as predicted, both state 2 and state 3 doesn't change, but state 1 increases from value of 1 to 1.2. Conversely, if we changed with probability of switching between u2 and u3 of (0.2,0.8), state 1 decreases from value of 1 to 0.8, which is shown in figure 5. The only remaining parameter that has yet to be explored is the learning rate, which we have previously investigated. While it is possible to manipulate the stochasticity of receiving rewards at each state, we have already gained insights by altering the switching probabilities, its effects are primarily observed in state 1 and not in the other states.

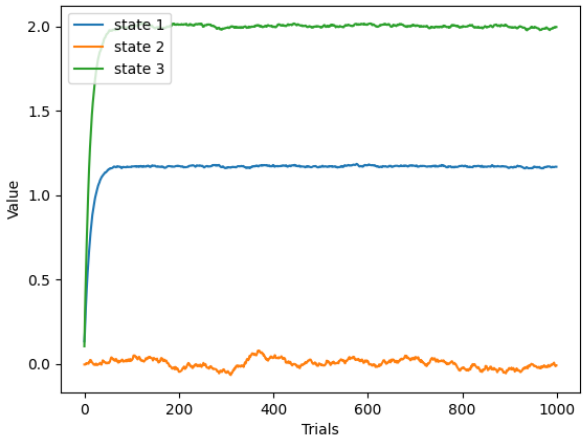

Figure 4: Policy Evaluation, with probability of switching between u2 and u3 of (0.2,0.8), as predicted, both state 2 and state 3 doesn't change, but state 1 increases from value of 1 to 1.2

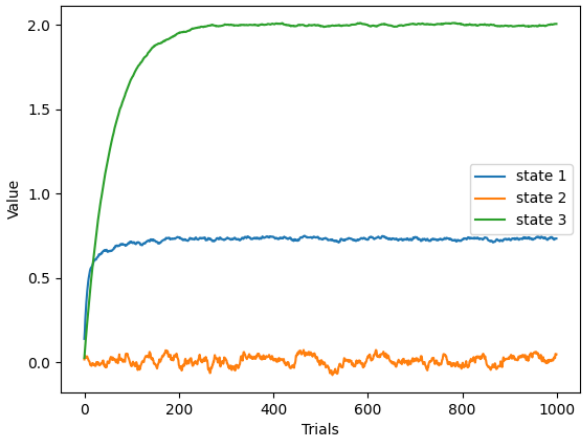

Figure 5: Policy Evaluation, with probability of switching between u2 and u3 of (0.8,0.2), as predicted, both state 2 and state 3 doesn't change, but state 1 decreases from value of 1 to 0.8

Actor-Critic

Figure 6 shows the result of the actor critic algorithm with learning rate for the value function equals to 0.2 and for the action values of 0.075. To reproduce this result we did not used any decaying factor in the update rule of the action values. the β value was set to 1. In the first row of the figure are the ensemble average of the value function for state 1, 2, and 3. where the dotted line is one realization of the function. Unlike the policy evaluation algorithm, the reward received is not only the immediate reward for the case of the actor critic, but the immediate reward plus the future expected reward. That justifies why the steady state values for the value function in figure 6 is higher than the one obtained previously in policy evaluation Figure 3. Which comes from the fact that the critic term partially uses the temporal difference kernel.

In the second row of figure 6, the policy is depicted as a function of trials. In the first column, for state 1, the agent starts learning that choosing the right lever is the best option as a reward of 2 is received, during the learning process and updating the action values according to the learning rule that $m^a(u) = (1 - \epsilon_d)m^a(u) + \epsilon_a \delta_{ab} \delta$ where $\delta = r^a(u) + \nu(\prime u) - \nu(u)$ and $\prime u$ is the next state. and $\nu(\prime u)$ is zero if we are at a leaf, thus the expected future reward for that case is zero. Thus, after some trials, the agent learns that the future reward with pressing level C is higher and thus after some trials the policy of the center lever increases to be the most probable action that maximize the total rewards for state 1. For state 2 or 3, the agent learns that the left lever or right lever are the best, respectively.

For the case of Actor-Critic, the range of parameter space that we can explore consists of the learning rate of the action values ϵ_A, the learning rate of the value function denoted by ϵ_v, the decaying parameter ϵ_d, the inverse temperature β, the switching probability between state 2 and state 3, and other parameters such as the stochasticity of the rewards. It's worth noting that when we change one parameter we keep the other fixed unless otherwise is mentioned.

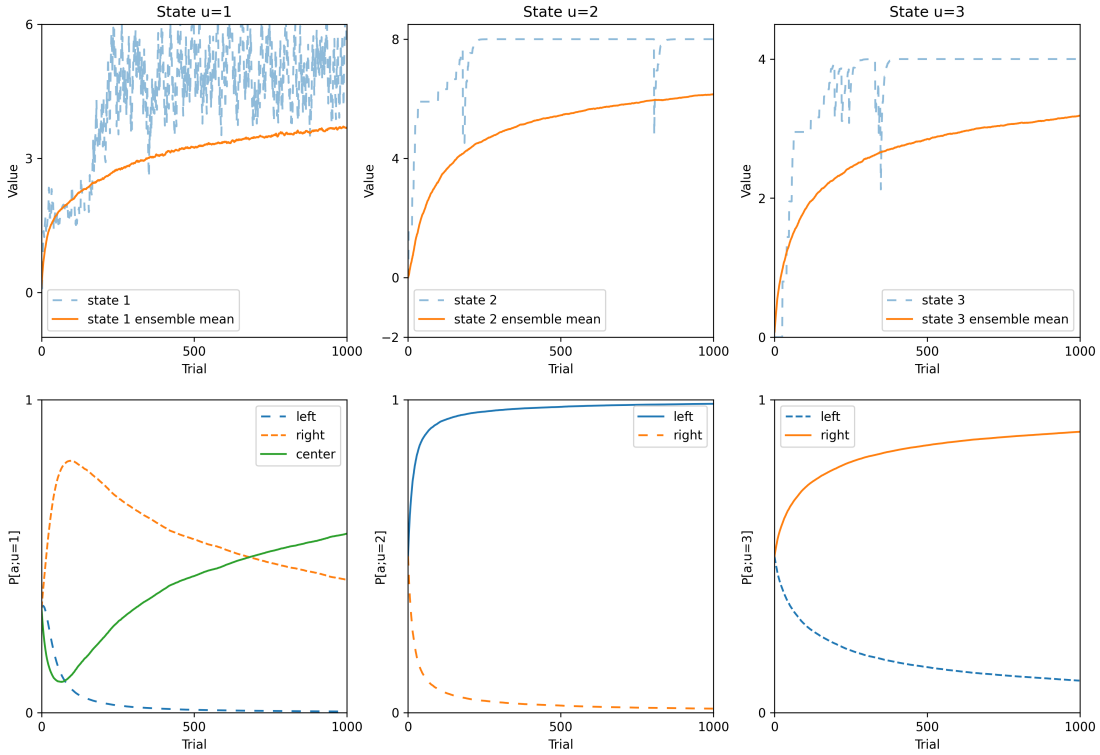

Figure 6: Actor-Critic Algorithm

Remarks on the relation between the critic and actor

For the actor to learn from the critic, it requires that the critic to lead the actor, i.e., for the learning rate of the critic to be relatively larger than the actor. In the main example, the learning rate of the critic part was 0.2, while for the actor part was 0.075.

To investigate that relationship, figure 7 shows the effect when we relatively increase the action values learning rate from 0.075 to 0.1 which is relatively close to the learning rate of the critic. If we made both are equal to 0.2, it is as if we are loosing the the significance of the critic part as shown in 8, this is shown in the policy of state one, that is the optimum action is always choosing the right level at state 1, and we observe that the value of state 2 and 3 decreases at trial=1000. The reason for that is, the action values are rushing to rewards but is not accurately being updated with the value function, which is in return chooses a suboptimal action when samples from the action policy. So no matter how long the trials are, it's very very slowly improving after 1000 trials.

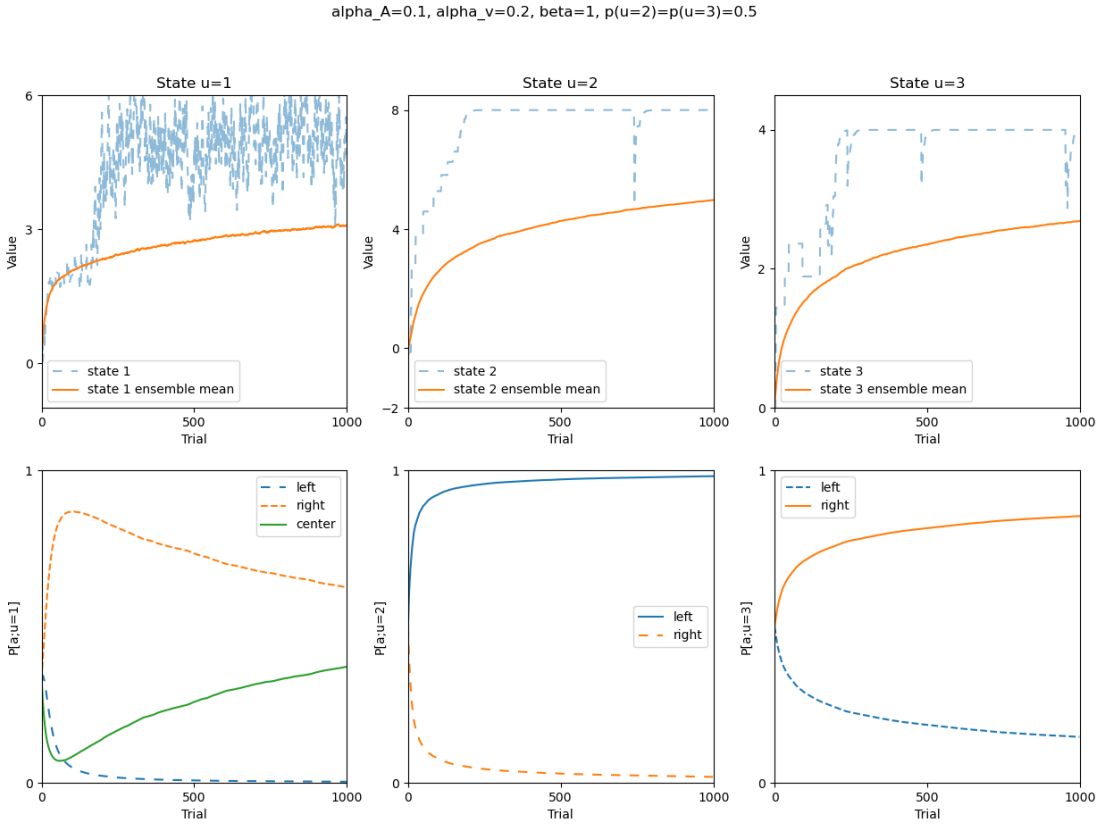

Figure 7: Actor-Critic, with α_A=0.1, α_v=0.2

To clearly elaborate on this relativity between both learning rates and not only on their absolute values, we decrease the learning rate of the value function from 0.2 to 0.1, and increase the learning rate of the action values to 0.2. We expect to observe that we're almost loosing the critic part and going back to the policy evaluation results for the value function, figure 9 shows that the value function for each state almost converges to the same value as before without having any critic part. Thus, we conclude this point by emphasizing the importance of both the relative and absolute values of the learning rates between the actor and the critic.

Remarks on including decay factor and changing the probability between state 1 and 2

In this part, we added a decaying factor, this decaying factor reduces the action values that is not choosing at a trial step. This factor is mainly useful if the environment is unstable so we allow the agent to flexibily choose and alter between different actions until converging to the optimum. However, if our case, our environment is relatively stable, that is, we are not altering the probabilities of receiving rewards from each lever and each state. the effect of adding a decay term of 0.2. is shown in figure 10.

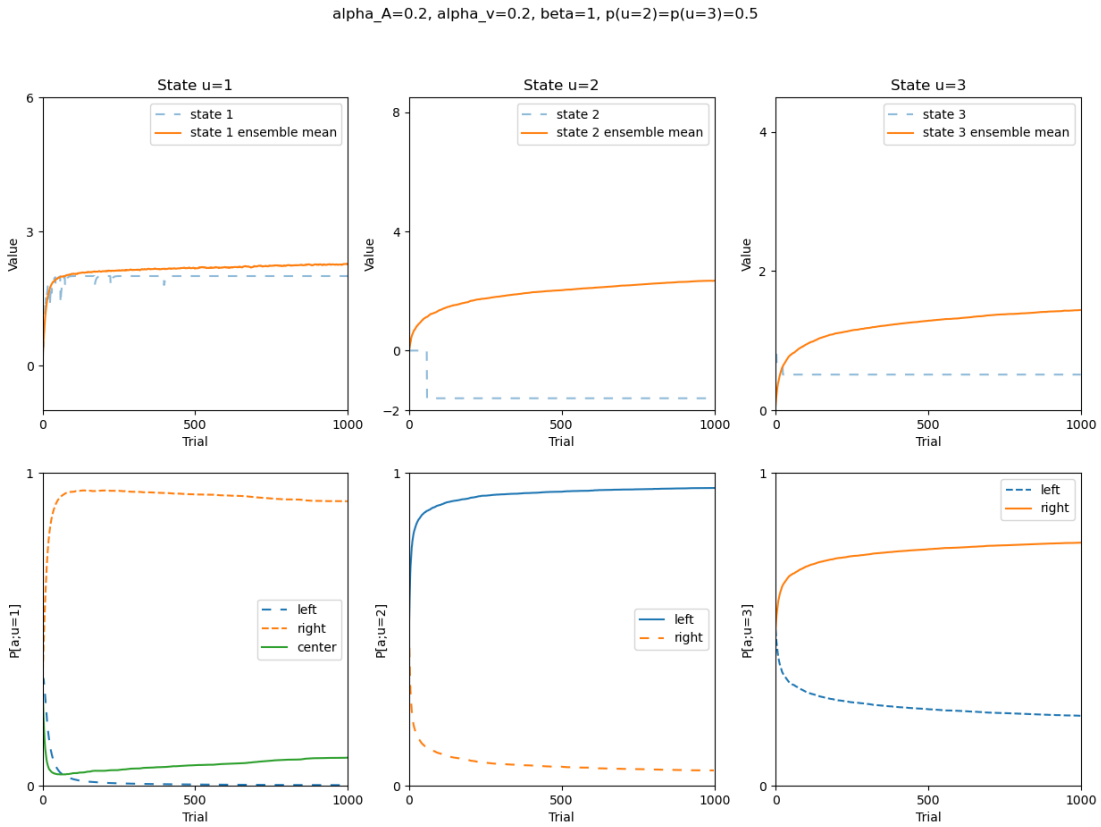

Figure 8: Actor-Critic, with α_A=0.2, α_v=0.2

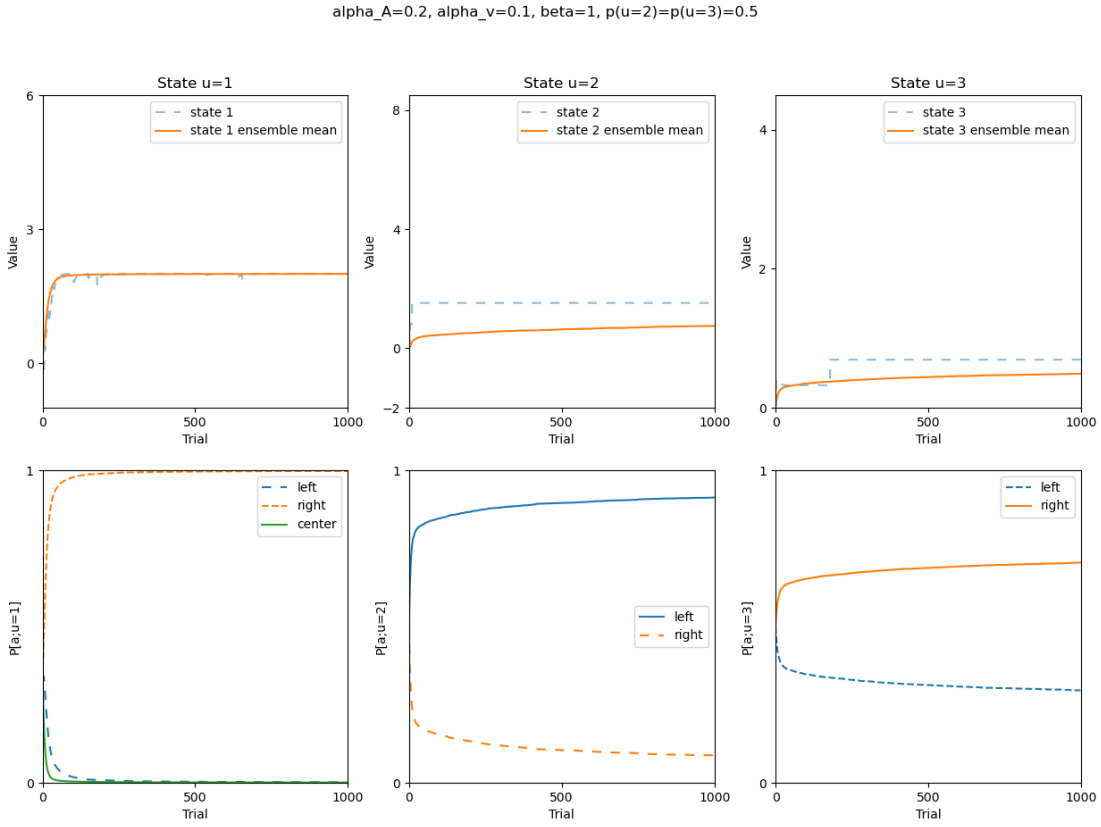

Figure 9: Actor-Critic, with α_A=0.2, α_v=0.1

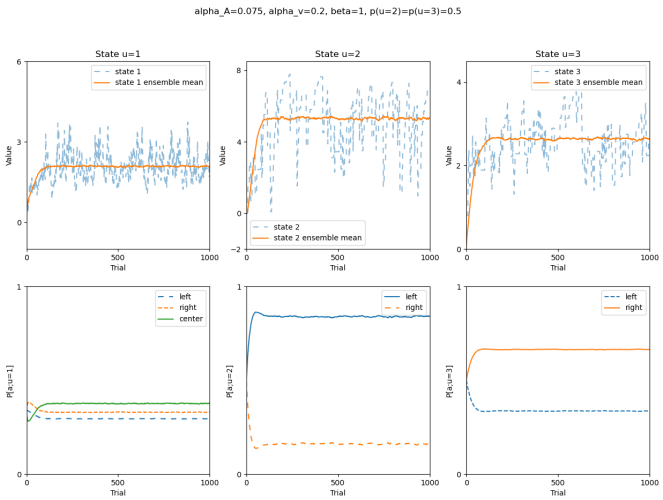

Figure 10: Policy Improvement, with decay of 0.2

Lastly, we alter the switching probability after pressing lever C from (0.5,0.5) to probabilistically to either state 2 or 3 to (0.2,0.8) and (0.8,0.2). The results are depicted in figures 11 and 12. The effects of altering these probability are as expected and explained in previous section for the policy evaluation part.

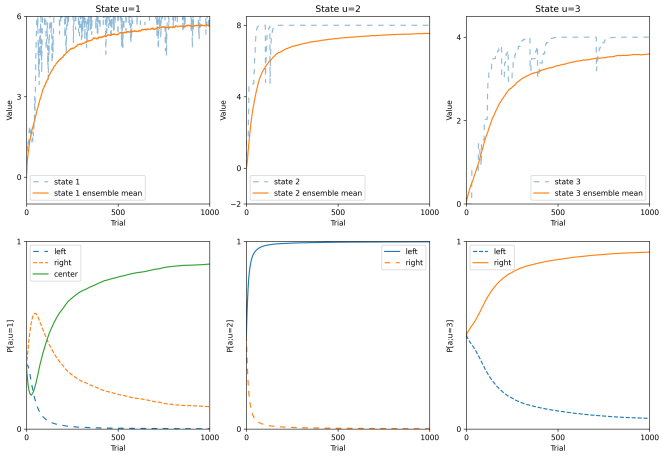

Figure 11: Policy Improvement, with probability of switching between u2 and u3 of [0.8,0.2],

Appendix

Direct Actor

Figure 13 shows the result of the direct actor.

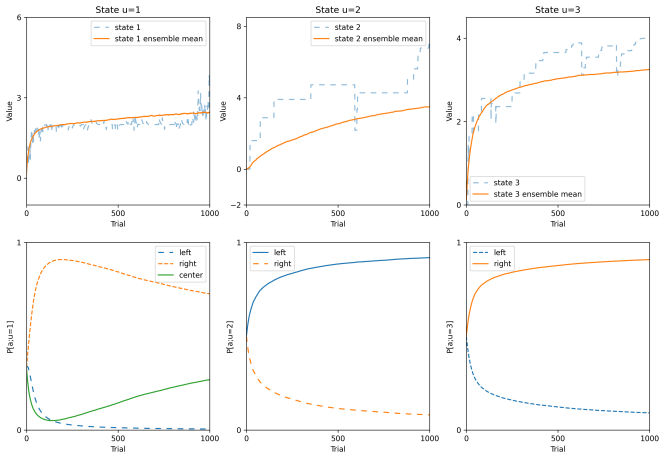

Figure 12: Policy Improvement, with probability of switching between u2 and u3 of [0.2,0.8],

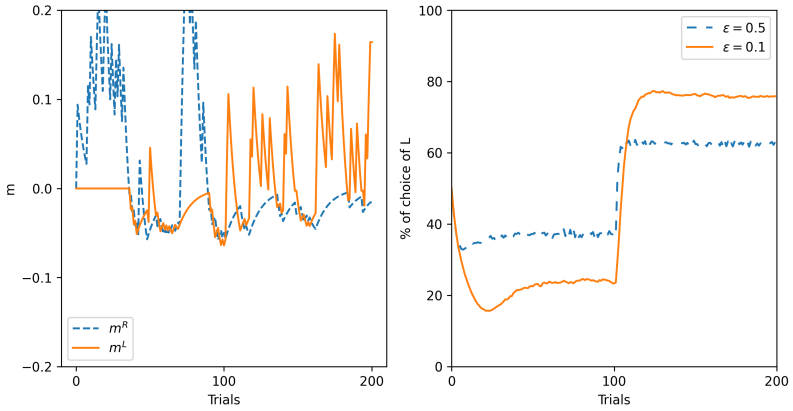

Figure 13: Indirect Actor