Reinforcement Learning - Instrumental conditioning

Marc Vöhringer Carrera

December 13, 2023

1 Tree task

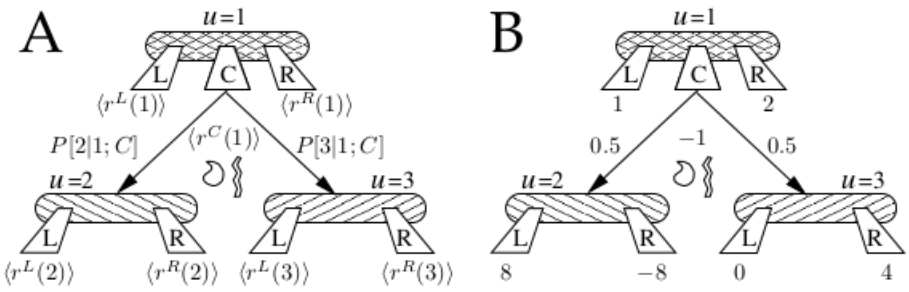

Figure 1: Tree task [1].

Fig. 1 shows a tree task, with three distinct states. In the initial state $u = 0$, three actions can be chosen, two of which deliver an instant reward. The third action (C) delivers a punishment, while taking the agent to the next state, either state $u = 2$ or $u = 3$, each with a probability of 50% . In these states, two actions are available which deliver a reward. This task constitutes a sequential action choice, which can be learned with the actor-critic algorithm.

1.1 Policy evaluation

The actor critic algorithm consists of two parts: the critic (policy evaluation) and the actor (policy improvement). First, the implementation of the policy evaluation is investigated. In order to do so, the policy is kept fixed by choosing completely random actions, uniformly distributed. The predictions

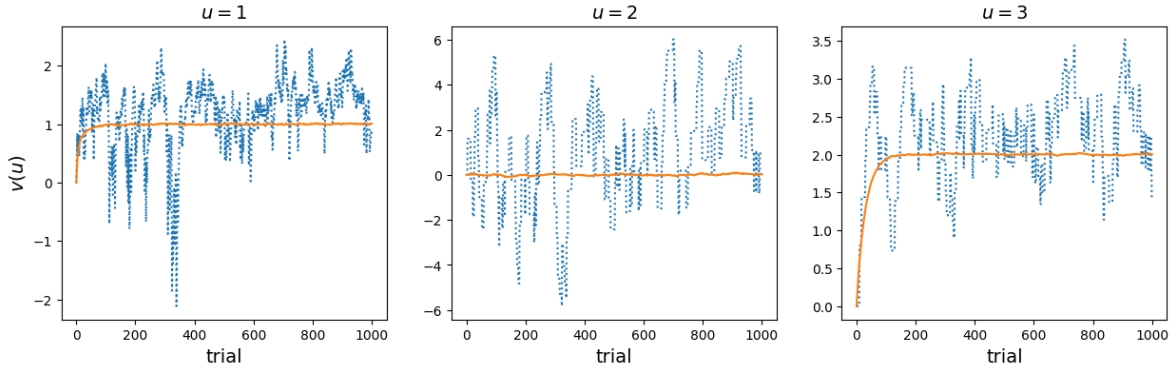

Figure 2: Predictions of the critic learning rule for fully stochastic policy. Dashed blue line represents the results for a single learning process, while the solid orange line is the result averaged over 10000 repetitions.

$v(u)$ for states $u = \{1, 2, 3\}$ are then updated over trials by the critic learning rule

$$v(u) \to v(u) + \epsilon\delta \quad \text{with} \quad \delta = r^a(u) + v(u') - v(u), \tag{1}$$

where ϵ is the learning rate, δ is the prediction error, $r^a(u)$ is the reward delivered at state u, when choosing action a and u' is the state into which the agent is transitioned. Applying the critic learning rule to the tree task from Fig. 1 with a learning rate of $\epsilon = 0.2$, 1000 trials and 10000 repetitions, yields the results displayed in Fig. 2.

1.2 Policy improvement

Instead of keeping a fixed policy, the predictions for the individual states provided by the critic can be used to improve the policy according to the actor learning rule

$$m^b(u) \to (1 - \epsilon_{\mathrm{D}})m^b(u) + \epsilon_{\mathrm{A}}\delta_{ab}\delta \quad \forall b, \text{ when a is chosen}, \tag{2}$$

where $m^k(u)$ is the action value corresponding to action k in state u, ϵ_{D} is a decay that is added to the action values in order to prevent them from increasing too drastically, which can improve adaptation to changing environments, ϵ_{A} is the actor learning rate and δ_{ab} is the Kronecker-Delta.

The policy is the determined by the action values according to the softmax function

$$P[a|u] = \frac{\exp[\beta m^a(u)]}{\sum_{b=1}^{N(u)} \exp[\beta m^b(u)]}, \tag{3}$$

where $N(u)$ is the total number of actions in the current state and β is the inverse temperature, determining the trade-off between exploration and exploitation.

Fig. 3 shows the results of the actor-critic algorithm applied to the tree task from Fig. 1, with the parameters $\epsilon = 0.2$, $\epsilon_{\mathrm{D}} = 0$, $\epsilon_{\mathrm{A}} = 0.075$, $\beta = 1$, 1000 trials and 1000 repetitions.

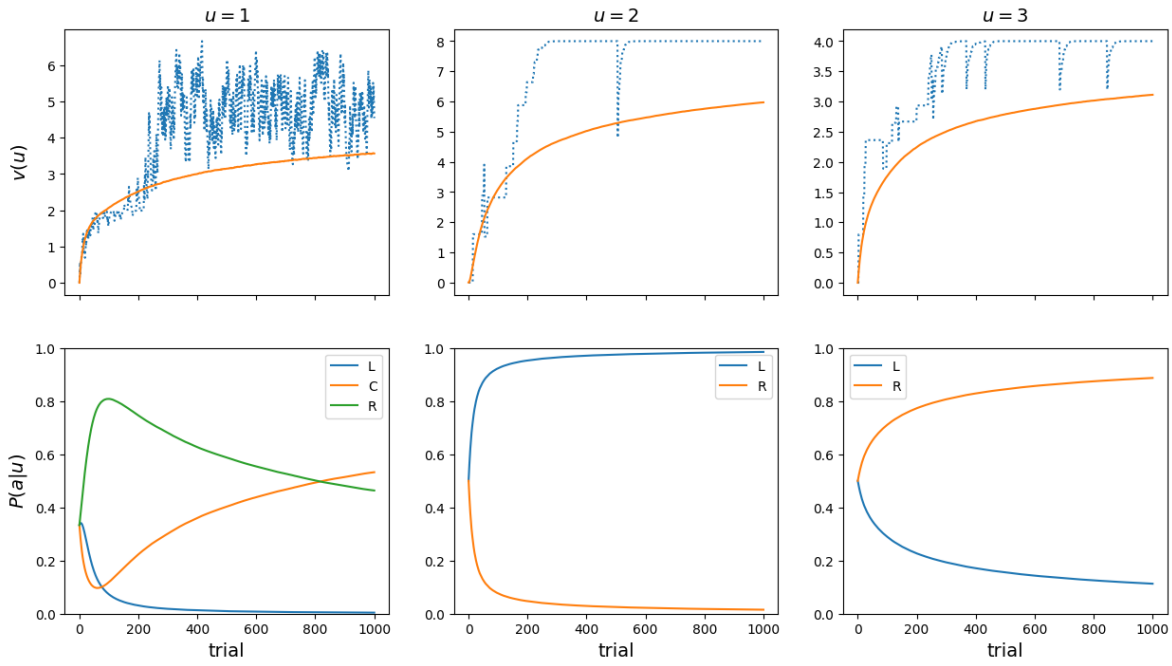

Figure 3: Predictions (upper plots) and probabilities of choosing a given action (lower plots) in states $u = \{1, 2, 3\}$. The blue dashed lines in the upper plots show the results for a single learning process, while the solid lines are an average over 1000 repetitions.

1.3 Discussion

To investigate the influence of the model parameters on the outcome of the learning, the average reward of a trained agent is computed according to

$$\langle r \rangle = P(L|1) + 2P(R|1) + (-1 + 4P(L|2) - 4P(R|2) + 2P(R|3))\, P(C|1), \tag{4}$$

which should correspond to the prediction $v(u)$ of the critic for state $u = 1$. This is done for various learning rates ϵ and actor learning rates ϵ_A. The results are displayed in Fig. 4.

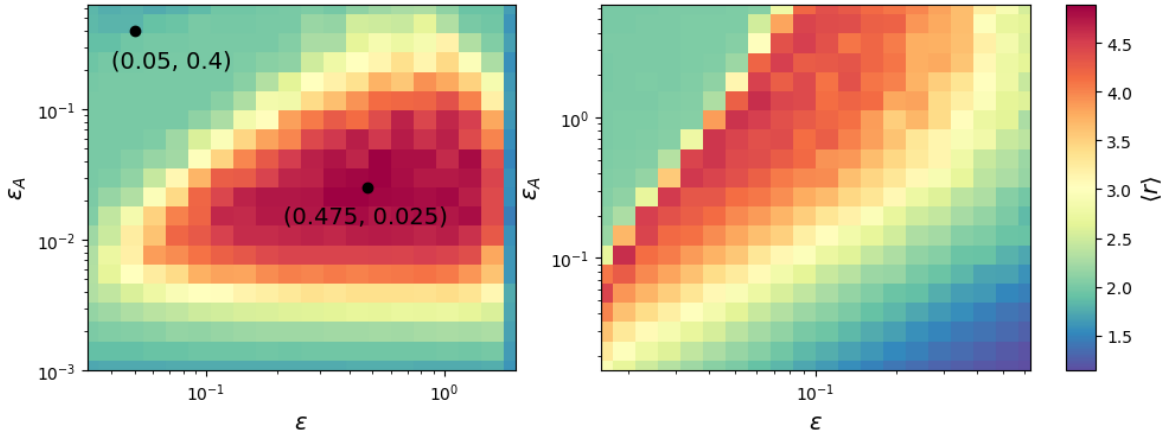

Figure 4: Average reward of the trained model for different learning rates. (Left) $\epsilon_D = 0$ (Right) $\epsilon_D = \epsilon$. The inverse temperature is $\beta = 1$ in both plots.

In the left plot the resulting mean reward is displayed for $\epsilon_D = 0$, while in the right plot the decay is set to $\epsilon_D = \epsilon$. One can see that in both cases there is a region where the mean reward $\langle r \rangle$ is close to the ideal reward of 5. The regions lie at different combinations of ϵ and ϵ_A for the different decay cases. For this tree task, if the environment is not changing, it is not necessary to use a decay ϵ_D. The influence of using decaying action values is going to be discussed in Sec. 2, where a changing

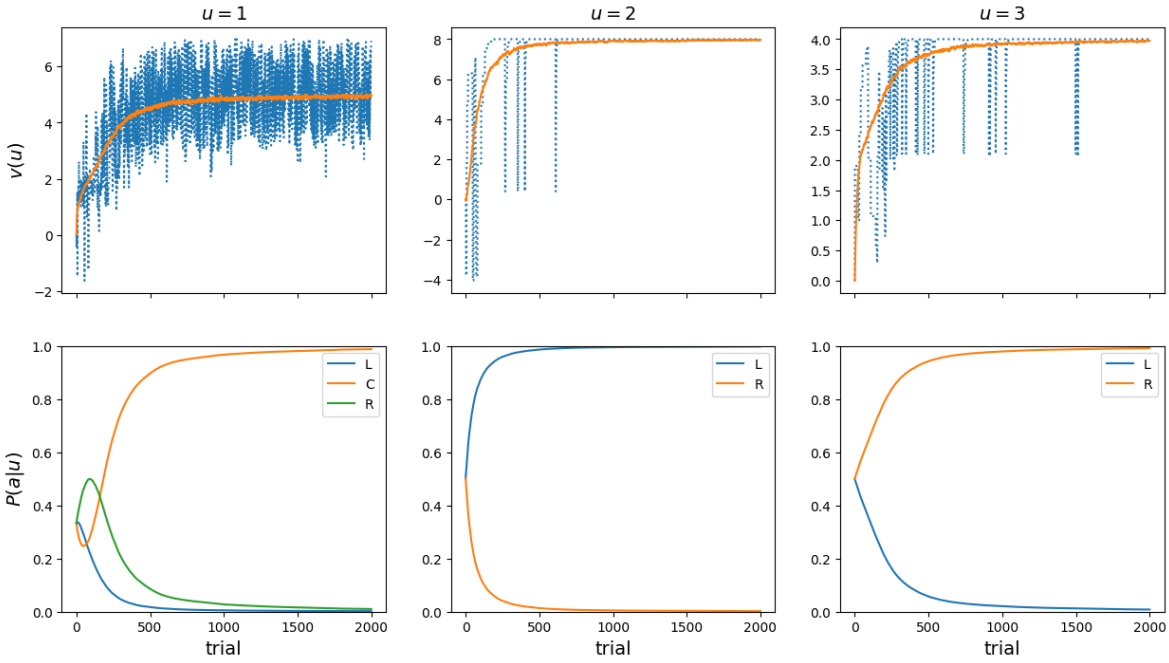

Figure 5: Results of the actor-critic algorithm for the tree task with ideal parameters.

environment will be investigated. For the case without decaying action values the ideal combination of (ϵ, ϵ_A) lies approximately at $(0.475, 0.025)$. The predictions of the critic and the policy for that particular parameter set is plotted in Fig. 5. As one can see the probability of picking action Cin state $u = 1$ initially decreases, since it is associated with a negative reward of -1. Over time, the actor-critic agent is able to learn through exploration, that the subsequent states are able to provide larger rewards. After 2000 trials the average reward takes a value of $\langle r \rangle = 4.96$.

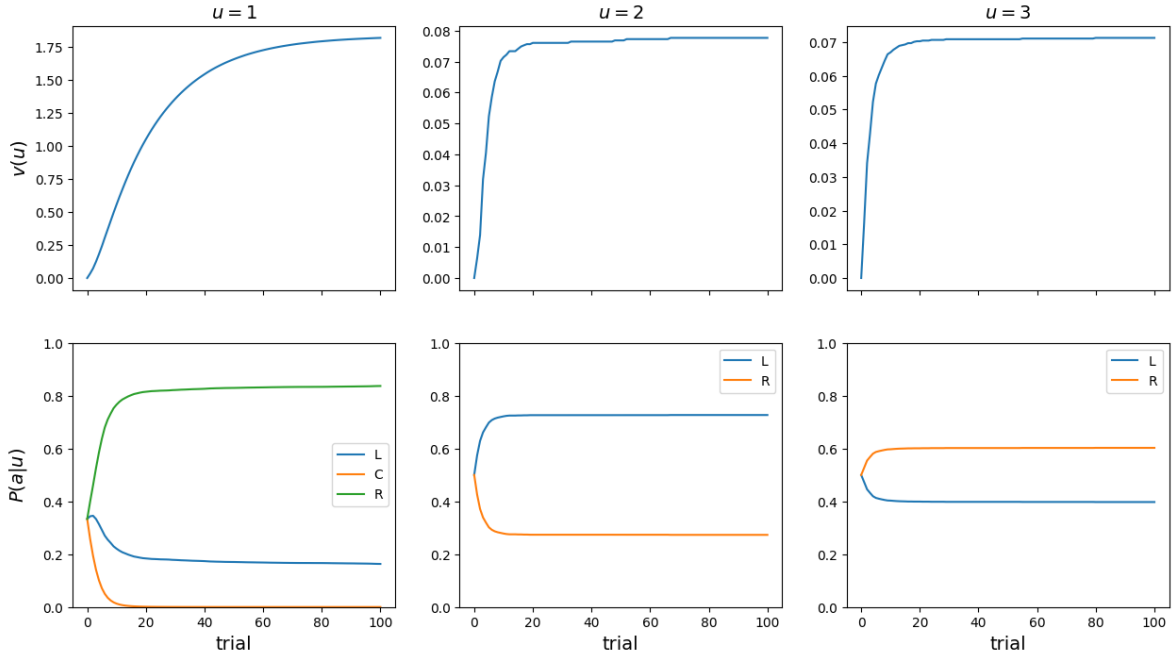

Figure 6: Results of the actor-critic algorithm for the tree task with suboptimal parameters.

In contrast, if the learning rates are chosen suboptimally ($\epsilon = 0.05, \epsilon_A = 0.4$) the action values are updated too quickly, such that the policy is not being evaluated correctly. The agent is not able to learn through exploration, that the states $u = \{1, 2\}$ provide larger rewards than the instant rewards of choosing actions L and R in state $u = 1$. The average evolution of $v(u)$ and $P(a|u)$ over the course of 200 trials is shown in Fig. 6. The average reward after 200 trials takes a value of $\langle r \rangle = 1.84$ in that case.

2 Maze Task

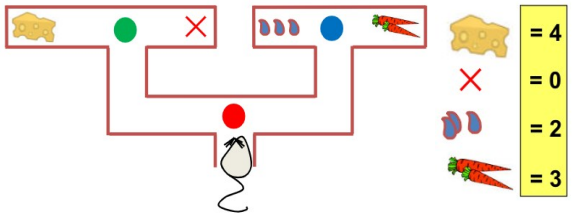

Figure 7: Maze task.

In this section a similar task to the previous tree task is investigated. This task is called maze task and is depicted in Fig. 7. The agent can first choose between two states, while no immediate reward or punishment is delivered. It afterwards enters a subsequent state, depending on its initial choice, in which it can again choose between two actions providing a reward. As before, we first investigate how

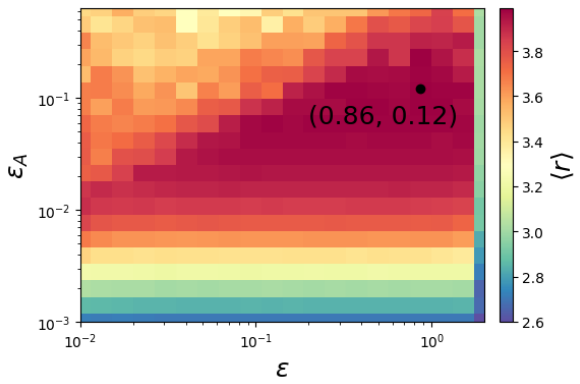

Figure 8: Result of the parameter scan for the maze task.

the average reward changes with the learning rates (without decay) to find the optimal parameters. The result is plotted in Fig. 9.

The optimal parameters are approximately $\epsilon = 0.86$ and $\epsilon_{\mathrm{A}} = 0.12$. To explore the influence of ϵ_{D}, we now flip the rewards that are being delivered upon a given action, meaning that in Fig. 7 the position of the cheese and the cross are exchanged, while the position of the water droplets and the carrots are also exchanged. This switching around of the rewards is being repeated every 500 trials, over the course of 5000 trials. The average reward over all trials is then computed and plotted as a function of ϵ_{D} in Fig. 9.

As one can see, the average reward initially increases and then slowly decreasing again after reaching its peak at approximately $\epsilon_{\mathrm{D}} = 0.017$. This shows that using decaying action values can help keeping a good exploration-exploitation ratio and therefore increases the adaptability to changes in the environment. It should be noted that changing ϵ_{D} also changes the optimal values of the critic and actor learning rates as can be seen in Fig. 4. This means that the avergae reward could even be further increased by tweaking the parameters again.

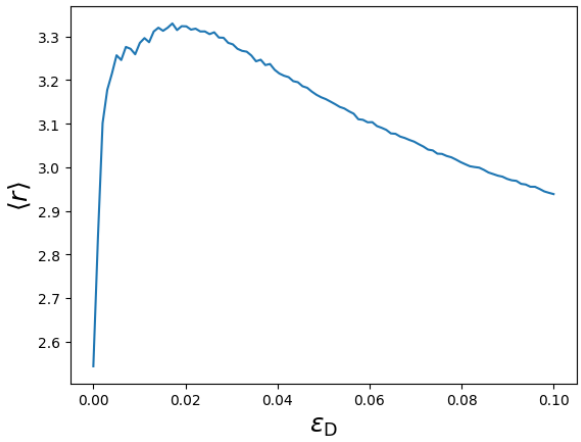

Figure 9: Average reward $\langle r \rangle$ for varying action value decay ϵ_{D}.

References

[1] Peter Dayan and Laurence Abbott. *Theoretical neuroscience*. 2001.