CNeuro 2024, Problems
Soft RL and Maximum Occupancy Principle
Rubén Moreno Bote

In these exercises, we will study, simulate and compare the behaviors of reward- and entropy-seeking (maximally occupying) agents (R- vs MOP- agents). The theory described below for the MOP-agent is based on [Ramírez-Ruiz et al.(2024)Ramírez-Ruiz, Grytskyy, and Moreno-Bote].

1 Problem setting

For both the R- and MOP-agent we consider the following problem:

Environment The arena is a single room having size 5×5 locations where the agent can be in. There are food sources located at the corners $(1, 1)$ and $(5, 5)$ in Cartesian coordinates, where $(1, 1)$ is the position of the bottom-left corner of the arena. The discount factor is $\gamma = 0.9$.

States The (discrete) states are $s = (x, y, E)$, the Cartesian product between location (x, y) and internal energy state E. The energy takes discrete values in units of 1 between a minimum of 0 and a maximum capacity of E_{max}. All states such that $(x, y, E = 0)$ are terminal (death) states, independently of the location (x, y).

Actions The agent has a maximum of 5 actions, consisting in moving right, left, up or down one unit or staying in its current position. At the boundaries of the arena, actions that bring the agent outside the arena are not available. At the terminal states ($E = 0$), only the "action" of staying in the current position is available.

Transitions At every time step, the energy decreases one time step, regardless of the action made. If the agent is currently at one of the food locations, then the energy increases in the next time step by $E_{gain} = 5$, topped by the maximum reservoir energy $E_{max} = 10$. (Q: Would you use an E_{max} smaller than 5? Why?) An action causes the agent to move to the desired nearby location (or with some probability to other nearby locations, see below).

2 R-agent

We first consider a reward-maximizing agent in the usual RL sense. In this case, the reward is $r(x, y, E) = 1$ when at a food source, and zero otherwise, for any internal energy value. To allow the reward-maximizing agent to display some stochasticity, we used an ϵ-greedy policy: at any given state, the best action a is chosen with probability $1 - \epsilon$ with $0 \leq \epsilon < 1$, and with probability ϵ any action (including the optimal one) is randomly chosen. The state-value function for the ϵ-greedy R-agent satisfies the optimality Bellman equation

$$V_\epsilon(s) = (1-\epsilon) \max_a \sum_{s'} p(s'|s,a) \left(r(s) + \gamma V_\epsilon(s')\right) + \frac{\epsilon}{|\mathcal{A}(s)|} \sum_{a,s'} p(s'|s,a) \left(r(s) + \gamma V_\epsilon(s')\right), \quad (1)$$

where $|\mathcal{A}(s)|$ is the number of admissible actions at state $s = (x, y, E)$, and $p(s'|s, a)$ is the probability of transitioning from state s to state s' after performing action a. Note that in our case the transition matrix is deterministic: e.g, after performing the action "Right" at state $s = (x, y, E) = (2, 2, 1)$, the agent moves at the terminal state $s' = (x', y', E') = (3, 2, 0)$, with zero energy. From this state there is only action "Stay", and therefore it cannot move anymore. Note also that $|\mathcal{A}(s)| = 5$ in every non-terminal state s except at the boundaries (where $|\mathcal{A}(s)| < 5$), and that $|\mathcal{A}(s)| = 1$ at terminal states.

1. To solve for the optimal value in this Bellman equation, perform value iteration. That is, start from an initial condition $V_0(s)$ and iterate the estimate of the state-value function $V_t(s)$ at iteration t based on its estimate at the previous iteration $t - 1$ as

$$V_t(s) = (1-\epsilon) \max_a \sum_{s'} p(s'|s, a) \left(r(s) + \gamma V_{t-1}(s')\right) + \frac{\epsilon}{|\mathcal{A}(s)|} \sum_{a,s'} p(s'|s, a) \left(r(s) + \gamma V_{t-1}(s')\right), \tag{2}$$

   Stop after convergence (in practice, stop when changes are very small, e.g. $< 10^{-5}$). Enforce that $V(s) = 0$ for all terminal states, as the cumulative future reward is zero in those states.

2. Plot the state value function over the space of the arena (as heatmap) for $E = 1$, $E = 5$ and $E = 10$. Use $\epsilon = 0.1$. How do you interpret the results?

3. Bonus exercise: Use the resulting optimal policy $\pi^*(a|s)$ to generate some sample trajectories. Note that the optimal policy is defined based on the state-value function as

$$\pi^*(a|s) = \begin{cases} 1 - \epsilon + \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{if } a = \arg\max_{a'} \sum_{s'} p(s'|s, a') \left(r + \gamma V^*_\epsilon(s')\right) \\ \frac{\epsilon}{|\mathcal{A}(s)|}, & \text{otherwise} \end{cases}$$

   where ties in arg max are broken randomly. Note that if $\epsilon = 0$, we obtain the usual greedy optimal policy that maximizes reward.

   What is the behavior of the R-agent when $\epsilon = 0$? How does it compare to the case $\epsilon > 0$?

3 MOP-agent

Now we move to model an entropy-seeking agent. In this agent, there is no external reward $r(x, y)$ (or, identically, it is zero everywhere). The only goal of the agent is to maximally occupy action-state path space, that is, to optimize the policy $\pi \equiv \pi(a|s)$ to maximize the state-value function

$$V_\pi(s) \equiv \mathbb{E}_\pi \left[\sum_{t=0}^{\infty} \gamma^t \left(\alpha \mathcal{H}(A|s_t) + \beta \mathcal{H}(S'|s_t, a_t)\right) \Big| s_0 = s \right], \tag{3}$$

with action and state entropies $\mathcal{H}(A|s) = -\sum_a \pi(a|s) \ln \pi(a|s)$ and $\mathcal{H}(S'|s, a) = -\sum_{s'} p(s'|s, a) \ln p(s'|s, a)$, respectively. Note that the state-value $V_\pi(s)$ does not contain external rewards, as opposed to what it is customary in RL. The parameters $\alpha > 0$ and $\beta \geq 0$ respectively weight the action and state entropies in the objective. To find the solution to our problem, take the values $(\alpha = 1, \beta = 0)$, corresponding to maximizing action path occupancy.

The optimal value function is the only root with $V^*(s) \geq 0$ of the set of non-linear equations

$$V^*(s) = \ln Z(s) \equiv \ln \left[\sum_a \exp\left(\gamma \sum_{s'} p(s'|s, a) V^*(s') \right) \right], \tag{4}$$

where $Z(s)$ is the partition function and the sum over actions extends over the actions available at state s. The optimal policy satisfies

$$\pi^*(a|s) = \frac{1}{Z(s)} \exp\left(\gamma \sum_{s'} p(s'|s,a) V^*(s')\right). \tag{5}$$

1. To solve for the optimal value in this Bellman equation, perform generalized value iteration. That is, start from an initial condition $V_0(s) > 0$ (Q: Would it make sense to start with a negative initial condition?) and iterate the estimate of the state-value function $V_t(s)$ at iteration t based on its estimate at the previous iteration $t-1$ as

$$V_t(s) = \ln\left[\sum_a \exp\left(\gamma \sum_{s'} p(s'|s,a) V_{t-1}(s')\right)\right], \tag{6}$$

   Stop after convergence. Enforce that $V(s) = 0$ for all terminal states, as the cumulative future action entropy is zero in those states.

2. Plot the state value function over the space of the arena of $E = 1$, $E = 5$ and $E = 10$. How do you interpret the results? Compare them to the R-agent.

3. Bonus exercise: Using the resulting optimal policy, generate some sample trajectories.

   What is the behavior of the MOP-agent? How does it compare to the R-agent?

References

[Ramírez-Ruiz et al.(2024)Ramírez-Ruiz, Grytskyy, and Moreno-Bote] Jorge Ramírez-Ruiz, Dmytro Grytskyy, and Rubén Moreno-Bote. Seeking entropy: complex behavior from intrinsic motivation to occupy action-state path space. *arXiv preprint arXiv:2205.10316, Accepted in Nature Communications*, 2024.