Assignment 4: Reinforcement learning
BMI/COMPSCI/PSYCH 841: Computational Cognitive Science Prof. Austerweil
Due 04/23 at 8:59pm

In this problem set, you will implement a reinforcement learning algorithm (Q-learning) and explore how it learns depending on different patterns of feedback. We will be exploring the "GardenPath" domain (Ho et al., 2015). Remember from the paper and class that the goal of the domain is to teach an agent to get to the top-right square (the "goal") using only the left-most and top-most squares (the "path") while avoiding going in the bottom right 2×2 square (the "garden"). Figure 1 depicts the GardenPath domain. Although a person gets the image shown in Figure 1, a reinforcement learning algorithm only knows that there is a 3×3 world that it can move from location to location in and receive feedback from the world. To help you implement the Q-learning on this task and visualize its solution, you are being provided with some Python or R code that you should download from Canvas named. The files provide you with helper functions and stencil code for implementing Q-learning. There are also instructions on the Canvas site for which lines to edit to complete the implementation and change the reward functions given to the Q-learning algorithm.

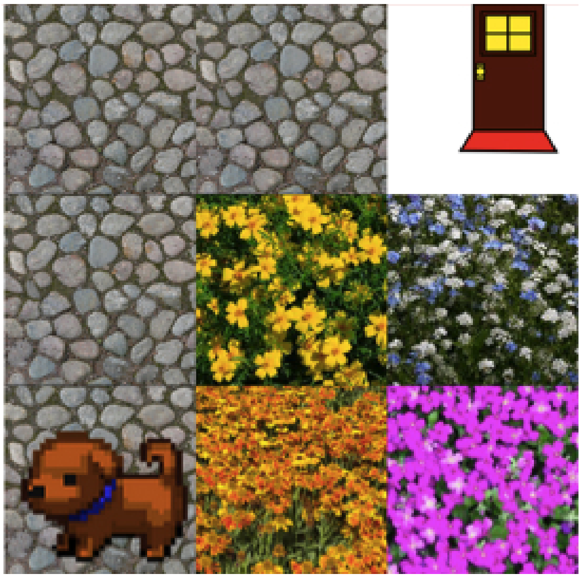


Figure 1: The GardenPath domain. Agents learn to get to the goal by moving along the path and avoiding going into the garden.

Remember that the goal of reinforcement learning is to learn a *policy* $\pi : S \rightarrow A$ (that is a function telling the agent which action a out of the set of possible actions A to take in each state s out of the set of possible states S) that results in the agent maximizing its reward. It does so by exploring states, taking actions, getting feedback, and adjusting its policy accordingly.

You will implement one particular reinforcement learning algorithm Q-learning, which is guaranteed to find the optimal policy for simple worlds.[1] Q-learning solves for a matrix

[1] Technically, Q-learning is guaranteed to find the optimal policy in Markov decision processes, which have the added assumption that the transition probabilities and expected immediate feedback only depend on the current state, the action taken, and for the latter, also the next state that the agent ended up in.

Q which stores the expected future reward of taking an action in a state. This happens according to the following algorithm:

1. Initialize the elements of a $S \times A$ dimensional Q-matrix to some value (usually zero).

2. For training episode $e = 1, 2, \ldots,$ we start a new training episode at the initial state and set $t = 1$.

   2a. Select an action from the set of possible actions. We will be using an ϵ-greedy rule, which means we pick a best possible action in our current state according to the current values of the Q matrix with probability $1 - \epsilon$ (there may be more than one, and then we choose randomly between them), and with probability ϵ we pick a random possible action.

   2b. We take that action, move to a new state, and receive feedback f_{t+1}.

   2c. Let α be a learning rate (usually 0.1) and γ be a discount factor (how much we penalize expected future rewards). We update our Q matrix according to the following rule:

$$Q_{t+1} = Q_t(s_t, a_t) + \alpha \left(f_{t+1} + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t) \right) \qquad (1)$$

Intuitively, this equation says “update my q-value based on the difference between my current q-value (last term) and the sum of the feedback I received (first term in parentheses) and the feedback I expect to get if I take the best actions going forward (second term in parentheses).

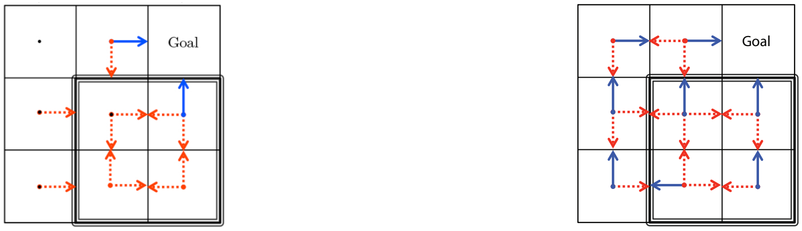
And now your task...

1. (Implement Q-learning). Using the code stencil of your choosing, please follow the instructions in Canvas for the lines to fill in to complete the Q-learning implementation. Above that line there is some explanation of the different variables you might need to use.

2. (Reward-Maximizing Feedback). Follow the instructions on Canvas for which lines to change to provide reward maximizing feedback. Figure 2a shows the pattern of feedback given by this function. Run Q-learning and then visualize the learned policy by running this file (make sure you close the figure or start a new one if you run it multiple times as they might overwrite each other otherwise). If you are using python, it will produce a number of number of plots. One of them will plot the Q-value of each action for each state, with blue being positive and red being negative. The size of the arrow is proportional to its Q-value. Save the plots and include it in your report. (If you are using R, you will see a single plot with the optimal policy plotted with black arrows). Did it do what you expected? Does the learned policy make sense? In a few sentences, explain why or why not.

3. (Action-Feedback... Feedback).

   Follow the instructions on Canvas for which lines to change to provide action-feedback feedback. Figure 2b shows the pattern of feedback given by this function. Run Q-learning and then visualize the learned policy by running this file (make sure you close



(a) Reward-Maximizing Feedback

(b) Action-Feedback Feedback

Figure 2: Solid blue arrows indicate actions given positive feedback, dashed red arrows indicate actions given negative feedback, and no arrow indicates a feedback value of zero.

the figure or start a new one if you run it multiple times as they might overwrite each other otherwise). If you are using python, it will produce a number of number of plots. One of them will plot the Q-value of each action for each state, with blue being positive and red being negative. The size of the arrow is proportional to its Q-value. Save the plots and include it in your report. (If you are using R, you will see a single plot with the optimal policy plotted with black arrows) Did it do what you expected? Does the learned policy make sense? In a few sentences, explain why or why not.

References

Ho, M. H., Littman, M. L., Cushman, F., and Austerweil, J. L. (2015). Teaching with rewards and punishments: Reinforcement or communication? In Noelle, D. C., Dale, R., Warlaumont, A. S., Yoshimi, J., Matlock, T., Jennings, C. D., and Maglio, P. P., editors, *Proceedings of the 37th Annual Meeting of the Cognitive Science Society*, pages 920–925, Austin, TX. Cognitive Science Society.