Exercise Sheet: RL (Model Based)

Exercise 1. Assume we are in the 3×2 gridworld as a model-based (R-Max) agent.

We start at the bottom left state 1 and aim to reach the top right state 6. In each state, we have four possible actions: up, down, left and right. For each action we move in the specific direction on the grid. However, except in state 6, there is always a 10% probability that we *slip*, which causes us to actually stay at the same location and not move at all. The same thing occurs when actions that will bring us outside the grid are taken. When we are in state 6, any action will deterministically lead to a transition to the terminal state, at which point we will receive a reward of $+10$ and return to the start for a new episode. Every time we take an action that does not lead to the terminal state, the reward is -1. The discount factor γ is set to be 0.95.

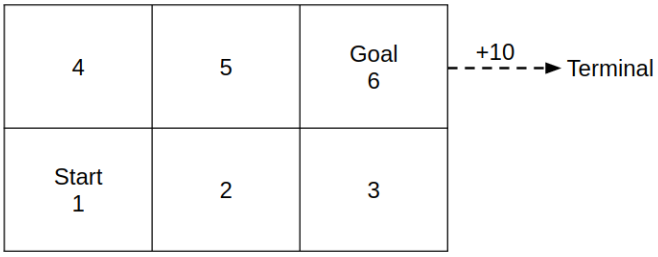


Figure 1: 3×2 gridworld problem.

a) As a recap, different from model-free methods, R-Max learns a model rather than a policy from real experience and then plans a policy within the learned model as the output of the algorithm.

To enable systematic exploration for learning the model, R-Max uses the "optimism in the face of uncertainty" principle. The key idea is that while interacting with the environment to collect real experience, actions should be chosen based on the assumption that state-action pairs that have not been visited sufficiently will yield the largest immediate reward possible R_{max}.

What is R_{max} in the 3×2 gridworld problem shown in Figure 1?

b) Why does this action selection method encourage systematic exploration for learning the model?

c) Specifically, R-Max learns a model from the observed transtisions $\{(s, a, r, s')\}$ by maintaining a list of observed rewards $R_{list}(s, a)$ for every state-action pair (s, a) and a count of transitions $N(s, a, s')$ for every tuple (s, a, s'). To choose actions, R-Max utilizes an optimistic model with transition function P_{opt} and reward function R_{opt}.

On initialization, for every state-action pair (s, a),

- $P_{opt}(s'|s, a) = I(s = s')$ (i.e., the state s transitions to s with probability 1 after action a)
- $R_{opt}(s, a) = R_{max}$

What does the optimistic model look like in our problem on initialization, i.e., before any learning starts? Assume we already know that there is a terminal state that terminates the episode upon arrival. You can either draw the model or specify R_{opt} and P_{opt}.

d) R-Max marks a state-action pair (s, a) as *known* when it has been observed at least m times, i.e., $N(s, a) = \sum_{s'} N(s, a, s') \geq m$ where $m \geq 1$ is a hyperparameter of R-Max. When $N(s, a)$ increases to m because of a newly observed transition from (s, a), $P_{opt}(s, a)$ and $R_{opt}(s, a)$ will be updated using $R_{list}(s, a)$ and $N(s, a, s')$.

Suppose now a new transition (s, a, s', r) is experienced and $N(s, a) < m$, specify the update rules for N, R_{list}, P_{opt} and R_{opt}.

e) Suppose after running R-Max with $m = 10$ in the environment for a few episodes, we have the following statistics for state 1:

N(s=1, a, s')	$s' = 1$	$s' = 2$	$s' = 3$	$s' = 4$	$s' = 5$	$s' = 6$
UP	1	0	0	3	0	0
DOWN	10	0	0	0	0	0
LEFT	10	0	0	0	0	0
RIGHT	1	9	0	0	0	0

Specify $P_{opt}(s = 1, a, s')$ and $R_{opt}(s = 1, a)$.

f) R-Max takes actions that maximize the return in the optimistic model for exploration. What methods introduced in previous weeks can be used?

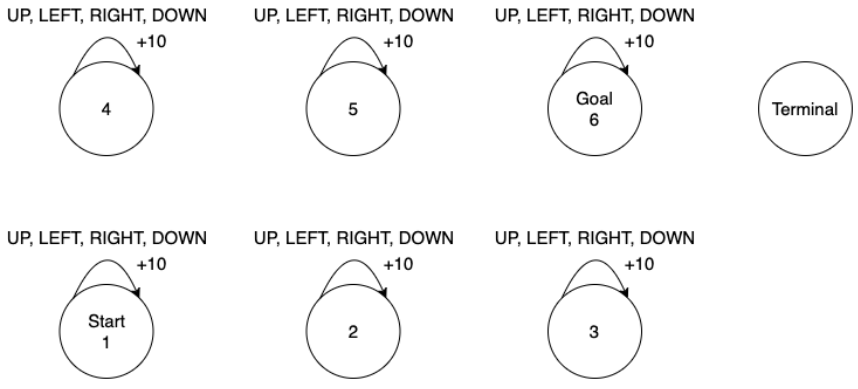
g) (optional) The R-Max algorithm presented above requires to save all the observed rewards for every state-action pair (s, a) in a list $R_{list}(s, a)$. We are now interested in a more space efficient implementation of R-Max where the list $R_{list}(s, a)$ is replaced by an average reward $R_{avg}(s, a)$ that is incrementally updated. Specify the incremental update rule for $R_{avg}(s, a)$ on the observation of a transition (s, a, s', r).

Solution:

a) 10.

b) By assuming unknown state-action pairs (s, a) yield the largest immediate reward possible R_{max}, the R-Max agent will be encouraged to visit those state-action pairs that have not been sufficiently explored, which results in systematic exploration.

c) See below.



d) $N(s, a, s') \leftarrow N(s, a, s') + 1$
append r to $R_{list}(s, a)$
if $N(s, a) = m$ **then**
$R_{opt}(s, a) = mean(R_{list}(s, a))$
for every s_{next}, $P_{opt}(s, a, s_{next}) = N(s, a, s_{next})/N(s, a)$
end if

e) See below.

$P_{opt}(s=1,a,s')$	$s'=1$	$s'=2$	$s'=3$	$s'=4$	$s'=5$	$s'=6$
UP	1.0	0	0	0	0	0
DOWN	1.0	0	0	0	0	0
LEFT	1.0	0	0	0	0	0
RIGHT	0.1	0.9	0	0	0	0

$R_{opt}(s=1,\text{UP})=10$ and $R_{opt}(s=1,\text{DOWN})=R_{opt}(s=1,\text{LEFT})=R_{opt}(s=1,\text{RIGHT})=-1$.

f) Since we have the full specification of the optimistic model, given by P_{opt} and R_{opt}, this is a classic planning problem in MDPs. As we have learned before, many methods can be applied, including value iteration, policy iteration and reinforcement learning methods as well.

g) $R_{avg}(s,a) \leftarrow \frac{r+R_{avg}(s,a)*N(s,a)}{N(s,a)+1} = R_{avg}(s,a) + \frac{r-R_{avg}(s,a)}{N(s,a)+1}$. Note that similarly you can use R_{sum}.

▲