

HW5: Reinforcement Learning

1 TD and Q in Blockworld

Consider the gridworld figure shown on Slide 5 from the Q-learning lecture (day10). Suppose that we run two episodes that yield the following sequences of (state, action, reward) tuples:

S	A	R	S	A	R
(1,1)	up	-1	(1,1)	up	-1
(2,1)	left	-1	(1,2)	up	-1
(1,1)	up	-1	(1,3)	right	-1
(1,2)	up	-1	(2,3)	right	-1
(1,3)	up	-1	(2,3)	right	-1
(2,3)	right	-1	(3,3)	right	-1
(3,3)	right	-1	(4,3)	exit	+100
(4,3)	exit	+100	(done)		
(done)					

1. According to direct estimation, what are the values for every state in the grid?
2. According to model-based learning, what are the transition probabilities for every (state, action, state) triple. Don't bother listing all the ones that we have no information about.
3. Suppose that we run Q-learning. However, instead of initializing all our Q values to zero, we initialize them to some large positive number ("large" with respect to the maximum reward possible in the world: say, 10 times the max reward). I claim that this will cause a Q-learning agent to initially explore a lot and then eventually start exploiting. Why should this be true? Justify your answer in a short paragraph.

2 Policy Gradient

(6300 only) In order to do policy gradient, we need to be able to compute the gradient of the evaluation function ρ with respect to a parameter vector θ : $\nabla_{\theta}\rho(\theta)$. By our algebraic magic, we expressed this as:

$$\nabla_{\theta}\rho(\theta) = \sum_a \pi_{\theta}(s_0, a) R(a) \underbrace{\nabla_{\theta} \log(\pi_{\theta}(s_0, a))}_{g(s_0, a)} \quad (1)$$

If we use a linear function thrown through a soft-max as our stochastic policy, we have:

$$\pi_{\theta}(s, a) = \frac{\exp(\sum_{i=1}^n \theta_i f_i(s, a))}{\sum_{a'} \exp(\sum_{i=1}^n \theta_i f_i(s, a'))} \quad (2)$$

Compute a closed form solution for $g(s_0, a)$. Explain in a few sentences *why* this leads to a sensible update for gradient ascent (i.e., if we plug this in to Eq (1) and do gradient ascent, why is the derived form reasonable)?