Recitation 8 March 22

1 MDPs: Racing

Consider a modification of the racing robot car example seen in lecture. In this game, the car repeatedly moves a random number of spaces that is equally likely to be 2, 3, or 4. The car can either Move or Stop if the total number of spaces moved is less than 6.

If the total spaces moved is 6 or higher, the game automatically ends, and the car receives a reward of 0. When the car Stops, the reward is equal to the total spaces moved (up to 5), and the game ends. There is no reward for the Move action.

Let's formulate this problem as an MDP with the states $\{0, 2, 3, 4, 5, Done\}$.

(a) What is the transition function for this MDP? (You should specify discrete values for specific state/action inputs.)

(b) What is the reward function for this MDP?

(c) Recall the value iteration update equation:

$$V_{k+1}(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma V_k(s')]$$

Perform value iteration for 4 iterations with $\gamma = 1$.

States	0	2	3	4	5
V_0					
V_1					
V_2					
V_3					
V_4					

(d) You should have noticed that value iteration converged above. What is the optimal policy?

States	0	2	3	4	5
π^*					

(e) How would our results change with $\gamma = 0.1$?

(f) Now recall the policy evaluation and policy improvement equations, which together make up policy iteration:

$$V^{\pi}_{k+1}(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^{\pi}_k(s')]$$

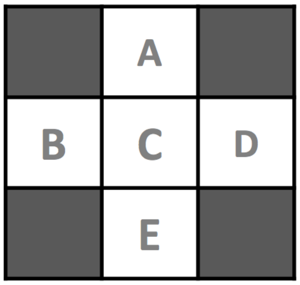
$$\pi_{new}(s) = \underset{a}{\text{argmax}} \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^{\pi_{old}}(s')]$$

Perform two iterations of policy iteration for one step of this MDP, starting from the fixed policy below. Use the initial $\gamma = 1$.

States	0	2	3	4	5
π_0	Move	Stop	Move	Stop	Move
V^{π_0}					
π_1					
V^{π_1}					
π_2					

2 Reinforcement Learning

Consider the Gridworld example that we looked at in lecture. We would like to use TD learning to find the values of these states.



Suppose we observe the following transitions:

(B, East, C, 2), (C, South, E, 4), (C, East, A, 6), (B, East, C, 2)

The initial value of each state is 0. Let $\gamma = 1$ and $\alpha = 0.5$.

(a) What are the learned values for each state from TD learning after all four observations?

(b) In class, we presented the following two formulations for TD-learning:

$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + (\alpha)sample \tag{1}$$

$$V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s)) \tag{2}$$

Mathematically, these two equations are equivalent. However, they represent two conceptually different ways of understanding TD value updates. How might we explain each of these equations?

3 Discussion-Based Questions

(a) In class, we learned that the Bellman Equations can be used to characterize optimal utility in MDPs. For reference, recall that this equation is given as:

$$V^*(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma V^*(s')]$$

What do we call γ in this equation? Why is it necessary? What happens as γ grows larger? As it grows smaller?

(b) What are the key distinctions between the value iteration and policy iteration algorithms, and when might you prefer one to the other?

(c) When does policy iteration end? Immediately after policy iteration ends (without performing additional computation), do we have the values of the optimal policy?

(d) What changes if during policy iteration, you only run one iteration of policy evaluation instead of running it until convergence? Do you still get an optimal policy?

(e) **(Bonus)** Think of a problem that seems like it can be modeled using an MDP, and formulate it (what are its states, reward function, and transition function)? Once you're complete, swap with a partner and verify your formulation. Try to get creative! There's quite a lot that can be modeled as an MDP.