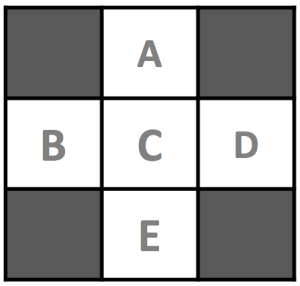
1 Temporal Difference and Q-Learning

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 $(s, a, s', R(s, a, s'))$* transitions and rewards:

$$(B, \text{East}, C, 2), (C, \text{South}, E, 4), (C, \text{East}, A, 6), (B, \text{East}, C, 2)$$

**Note that the $R(s, a, s')$ in this notation refers to observed reward, not a reward value computed from a reward function.*

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 could we intuitively explain each of these equations?

(c) What are the learned Q-values from Q-learning after all four observations? Use the same $\alpha = 0.5, \gamma = 1$ as before.

2 RL: Conceptual Questions

Recall that in Q-learning, we continually update the values of each Q-state by learning through a series of episodes, ultimately converging upon the optimal policy.

(a) What's the main shortcoming of TD learning that Q-learning resolves?

(b) We are given a pre-existing table of current estimate of Q-values (and its corresponding policy), and asked to perform ϵ-greedy Q-learning. Individually, what effect does setting each of the following constants to 0 have on this process?

(i) α:

(ii) γ:

(iii) ϵ:

(c) Consider a variant of the ϵ-greedy Q-learning algorithm that is changed such that instead of using the policy extracted from our current Q-values, we use a fixed policy instead. We still perform exploration with probability ϵ. If this fixed policy happens to be optimal, how does the performance of this algorithm compare to normal ϵ-greedy Q-learning?

(d) Let's revisit the CandyGrab code from recitation 1 (https://www.cs.cmu.edu/~15281/recitations/rec1/candygrab.zip). What RL strategies does `AgentRL` employ? Does it evaluate states or Q-states?

(e) Contrast the following pairs of reinforcement learning terms:

(i) Off-policy vs. on-policy learning

(ii) Model-based vs. model-free

(iii) Passive vs. active