# CS188 Fall 2018 Section 5: MDP + RL

# 1 MDPs: Micro-Blackjack

In micro-blackjack, you repeatedly draw a card (with replacement) that is equally likely to be a 2, 3, or 4. You can either Draw or Stop if the total score of the cards you have drawn is less than 6. If your total score is 6 or higher, the game ends, and you receive a utility of 0. When you Stop, your utility is equal to your total score (up to 5), and the game ends. When you Draw, you receive no utility. There is no discount ($\gamma = 1$). Let's formulate this problem as an MDP with the following states: $0, 2, 3, 4, 5$ and a *Done* state, for when the game ends.

1. What is the transition function and the reward function for this MDP?

   The transition function is

$$
\begin{aligned}
T(s, \mathit{Stop}, \mathit{Done}) &= 1 \\
T(0, \mathit{Draw}, s') &= 1/3 \text{ for } s' \in \{2, 3, 4\} \\
T(2, \mathit{Draw}, s') &= 1/3 \text{ for } s' \in \{4, 5, \mathit{Done}\} \\
T(3, \mathit{Draw}, s') &= \begin{cases} 1/3 \text{ if } s' = 5 \\ 2/3 \text{ if } s' = \mathit{Done} \end{cases} \\
T(4, \mathit{Draw}, \mathit{Done}) &= 1 \\
T(5, \mathit{Draw}, \mathit{Done}) &= 1 \\
T(s, a, s') &= 0 \text{ otherwise}
\end{aligned}
$$

   The reward function is

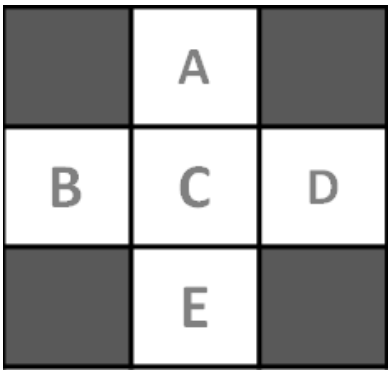
$$
\begin{aligned}
R(s, \mathit{Stop}, \mathit{Done}) &= s, s \leq 5 \\
R(s, a, s') &= 0 \text{ otherwise}
\end{aligned}
$$

2. Perform one iteration of policy iteration for one step of this MDP, starting from the fixed policy below:

| States | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| $\pi_i$ | Draw | Stop | Draw | Stop | Draw |
| $V^{\pi_i}$ | 2 | 2 | 0 | 4 | 0 |
| $\pi_{i+1}$ | Draw | Stop | Stop | Stop | Stop |

# 2 Learning in Gridworld

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




Suppose that we have the following observed 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. Assume that $\gamma = 1$ and $\alpha = 0.5$.

1. What are the learned values from TD learning after all four observations?

   $V(B) = 3.5$
   $V(C) = 4$
   All other states have a value of 0.

2. What are the learned Q-values from Q-learning after all four observations?

   $Q(B, East) = 3$
   $Q(C, South) = 2$
   $Q(C, East) = 3$
   All other q-states have a value of 0.

# 3 Pacman with Feature-Based Q-Learning

We would like to use a Q-learning agent for Pacman, but the state size for a large grid is too massive to hold in memory. To solve this, we will switch to feature-based representation of Pacman's state.

1. Say our two minimal features are the number of ghosts within 1 step of Pacman ($F_g$) and the number of food pellets within 1 step of Pacman ($F_p$). You'll notice that these features depend only on the state, not the actions you take. Keep that in mind as you answer the next couple of questions. For this pacman board:

   Extract the two features (calculate their values).
   $f_g = 2, f_p = 1$

2. With Q Learning, we train off of a few episodes, so our weights begin to take on values. Right now $w_g = 100$ and $w_p = -10$. Calculate the Q value for the state above.
   First of all, the Q value will not depend on what action is taken, because the features we extract do not depend on the action, only the state.

$$Q(s,a) = w_g * f_g + w_p * f_p = 100 * 2 + -10 * 1 = 190$$

3. We receive an episode, so now we need to update our values. An episode consists of a start state $s$, an action $a$, an end state $s'$, and a reward $r$. The start state of the episode is the state above (where you already calculated the feature values and the expected Q value). The next state has feature values $F_g = 0$ and $F_p = 2$ and the reward is 50. Assuming a discount of $\gamma = 0.5$, calculate the new estimate of the Q value for $s$ based on this episode.

$$\begin{aligned} Q_{new}(s,a) &= R(s,a,s') + \gamma * \max_{a'} Q(s',a') \\ &= 50 + 0.5 * (100 * 0 + -10 * 2) \\ &= 40 \end{aligned}$$

4. With this new estimate and a learning rate ($\alpha$) of 0.5, update the weights for each feature.

$$w_g = w_g + \alpha * (Q_{new}(s,a) - Q(s,a)) * f_g(s,a) = 100 + 0.5 * (40 - 190) * 2 = -50$$
$$w_p = w_p + \alpha * (Q_{new}(s,a) - Q(s,a)) * f_p(s,a) = -10 + 0.5 * (40 - 190) * 1 = -85$$

   Note that now the weight on ghosts is negative, which makes sense (ghosts should indeed be avoided). Although the weight on food pellets is now also negative, the difference between the two weights is now much lower.