CS188 Spring 2010 Section 5: MDPs

1 Warm-up: High-Low as an MDP

The game High-Low is a card game played with an infinite deck containing three types of cards: 2, 3, 4. You start with a 3 showing, and say either higher or lower. Then, a new card is flipped; if you say higher and the new card is higher in value than your current card, you win the points shown on the new card. Similarly, if you say lower and the new card is lower in value than your current card, you win the points shown on the new card. If the new card is the same value as your current card, you don’t get any points. Otherwise, the game ends. Your current card is then discarded and the new card becomes your current card. An example of a game is [3, high, 4, low, 2, high, 3, low, 4, end]

The deck contains different proportions of 2, 3, and 4 cards, \( p_2, p_3, p_4 \) respectively (where \( p_2 + p_3 + p_4 = 1 \)), which you may or may not know.

(a) Assuming you know \( p_2, p_3, p_4 \), formulate High-Low as an MDP:

**States:** \( \{2, 3, 4, \text{end}\} \)

**Actions:** \{high, low\}

**Rewards:**

\[
\begin{array}{c|ccc}
(s,a,s') & s' & 2 & 3 & 4 \\
\hline
(s,a) & 2 & 3 & 4 \\
(2,low) & 0 & 0 & 0 \\
(2,high) & 0 & 3 & 4 \\
(3,low) & 2 & 0 & 0 \\
(3,high) & 0 & 0 & 4 \\
(4,low) & 2 & 3 & 0 \\
(4,high) & 0 & 0 & 0 \\
\end{array}
\]

**Transitions:**

\[
\begin{array}{c|cccc}
T(s,a,s') & s' & 2 & 3 & 4 & \text{end} \\
\hline
(s,a) & 2 & 3 & 4 & \text{end} \\
(2,low) & p_2 & 0 & 0 & p_3 + p_4 \\
(2,high) & p_2 & p_3 & p_4 & 0 \\
(3,low) & p_2 & p_3 & 0 & p_4 \\
(3,high) & 0 & p_3 & p_4 & p_2 \\
(4,low) & p_2 & p_3 & p_4 & 0 \\
(4,high) & 0 & 0 & p_4 & p_2 + p_3 \\
\end{array}
\]

(b) Write down non-trivial Bellman equations for \( Q^*(3, \text{high}) \), the utility of saying high after seeing 3, \( Q^*(3, \text{low}) \), the utility of saying low after seeing 3, and \( V^*(3) \), the value of seeing 3 under the optimal policy. Assume a discount factor \( \gamma = 1 \). You may use the variables \( p_2, p_3, p_4, Q^*(2, a), Q^*(3, a), Q^*(4, a) \) in your answer for \( Q^*(3, a) \) and \( p_2, p_3, p_4, V^*(2), V^*(3), V^*(4) \) in your answer for \( V^*(3) \).

There are many possible answers to this problem, because we can repeatedly apply the identity \( V^*(s) = \max_{a \in \{\text{high, low}\}} Q^*(s, a) \).

One possible answer is given below.
\[ Q^*(3, \text{high}) = \sum_{s'} T(3, \text{high}, s')(R(3, \text{high}, s') + \gamma \max_{a'} Q^*(s', a')) \]
\[ = p_3(0 + \gamma \max_{a'} Q^*(3, a')) + p_4(4 + \gamma \max_{a'} Q^*(4, a')) \]
\[ = p_3(V^*(3)) + p_4(4 + V^*(4)) \]

\[ Q^*(3, \text{low}) = \sum_{s'} T(3, \text{low}, s')(R(3, \text{low}, s') + \gamma \max_{a'} Q^*(s', a')) \]
\[ = p_2(2 + \gamma \max_{a'} Q^*(2, a')) + p_3(0 + \gamma \max_{a'} Q^*(3, a')) \]
\[ = p_3(V^*(3)) + p_2(2 + V^*(2)) \]

\[ V^*(3) = \max_a Q^*(3, a) \]
\[ = \max_a \sum_{s'} T(3, a, s')(R(3, a, s') + \gamma V^*(s')) \]
\[ = \max(p_2(0 + \gamma V^*(\text{done})) + p_3(0 + \gamma V^*(3)) + p_4(4 + \gamma V^*(4)), p_2(2 + \gamma V^*(2)) + p_3(0 + \gamma V^*(3)) + p_4(0 + \gamma V^*(\text{done}))) \]
\[ = p_3 V^*(3) + \max(p_4(4 + V^*(4)), p_2(2 + V^*(2))) \]
Consider the above MDP, representing a robot on a balance beam. Each grid square is a state and the available actions are Right and Left. The agent starts in state S2. Transitioning to S1 or S7 gives +3 and +8 reward, respectively, and transitioning to Ground gives a -2 reward. Moving left or right results in a move left or right with probability \( p \). With probability \( 1 - p \) the robot falls off the beam and transitions to Ground. Falling off, or reaching either endpoint results in the end of the episode (terminal states). Note that terminal states receive no future reward.

(a) Consider the policy \( \pi_R \). Under \( \pi_R \), the robot always chooses action R. Perform five Bellman backups to compute the value of this policy (What is the correct value to use for terminal states?) Assume \( \gamma = 1, p = 0.8 \). The following equation may be useful.

\[
V_{i+1}^\pi(s) = \sum_{s'} T(s, \pi(s), s')(R(s, \pi(s), s') + \gamma V_i^\pi(s'))
\]

A calculator will be helpful for filling in this table.

As an example, consider computing \( V_2^\pi(R)(S5) \)

\[
V_2^\pi(R)(S5) = \sum_{s' \in \{S6, G\}} T(S5, R(S5), s')(R(S5, R(S5), S6) + \gamma V_1^\pi(S6)) + T(S5, R, G)(R(S5, R(S5), G) + \gamma V_1^\pi(G))
\]

\[
= 0.8(0 + (1)(6)) + 0.2(-2 + (1)(0))
\]

\[
= 4.4
\]
(b) Given initial value estimates of zero, show the results of two rounds of value iteration. Assume $\gamma = 1, p = 0.8$. The following equations may be useful.

$$V^*_i(s) = \max_a Q^*_i(s,a)$$

$$Q^*_i(s,a) = \sum_{s'} T(s,a,s') (R(s,a,s') + \gamma V^*_{i-1}(s'))$$

As an example, consider computing $Q^*_2(S4, L)$

$$Q^*_i(s,a) = \sum_{s'} T(s,a,s') (R(s,a,s') + \gamma V^*_{i-1}(s'))$$

$$Q^*_2(S4, L) = \sum_{s' \in \{S4, G\}} T(S4, L, s') (R(S4, L, s') + \gamma V^*_1(s'))$$

$$= T(S4, L, S3) (R(S4, L, S3) + \gamma V^*_1(S3)) + T(S4, L, G) (R(S4, L, G) + \gamma V^*_1(G))$$

$$= 0.8(0 + (1)(-0.4)) + 0.2(-2 + (0)(0))$$

$$= -0.72$$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$Q^*_i(S2, L)$</th>
<th>$Q^*_i(S3, L)$</th>
<th>$Q^*_i(S4, L)$</th>
<th>$Q^*_i(S5, L)$</th>
<th>$Q^*_i(S6, L)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.2</td>
<td>-0.72</td>
<td>-0.72</td>
<td>-0.72</td>
</tr>
</tbody>
</table>

As an example, consider computing $Q^*_2(S5, R)$

$$Q^*_i(s,a) = \sum_{s'} T(s,a,s') (R(s,a,s') + \gamma V^*_{i-1}(s'))$$

$$Q^*_2(S5, R) = \sum_{s' \in \{S6, G\}} T(S5, R, s') (R(S5, R, s') + \gamma V^*_1(s'))$$

$$= T(S5, R, S6) (R(S5, R, S6) + \gamma V^*_1(S6)) + T(S5, R, G) (R(S5, R, G) + \gamma V^*_1(G))$$

$$= 0.8(0 + (1)(6)) + 0.2(-2 + (0)(0))$$

$$= 4.4$$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$Q^*_i(S2)$</th>
<th>$Q^*_i(S3)$</th>
<th>$Q^*_i(S4)$</th>
<th>$Q^*_i(S5)$</th>
<th>$Q^*_i(S6)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>-0.72</td>
<td>-0.72</td>
<td>-0.72</td>
<td>4.4</td>
<td>6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$i$</th>
<th>$V^*_i(S2)$</th>
<th>$V^*_i(S3)$</th>
<th>$V^*_i(S4)$</th>
<th>$V^*_i(S5)$</th>
<th>$V^*_i(S6)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>-0.4</td>
<td>-0.4</td>
<td>-0.4</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1.2</td>
<td>-0.72</td>
<td>4.4</td>
<td>6</td>
</tr>
</tbody>
</table>

In the table below, we report the state-action value function $Q^*(s,a)$ for several different choices of $p$, computed by running value iteration to convergence. Assume $\gamma = 1$.

<table>
<thead>
<tr>
<th>$p$</th>
<th>$Q^*(S2, R)$</th>
<th>$Q^*(S3, R)$</th>
<th>$Q^*(S4, R)$</th>
<th>$Q^*(S5, R)$</th>
<th>$Q^*(S6, R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>-0.285</td>
<td>0.4010</td>
<td>1.43</td>
<td>2.9</td>
<td>5</td>
</tr>
<tr>
<td>0.8</td>
<td>1.2768</td>
<td>2.0960</td>
<td>3.12</td>
<td>4.4</td>
<td>6</td>
</tr>
<tr>
<td>0.9</td>
<td>3.905</td>
<td>4.561</td>
<td>5.29</td>
<td>6.1</td>
<td>7</td>
</tr>
<tr>
<td>1.0</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>
(c) What is the minimum value of $p \in \{0.7, 0.8, 0.9, 1.0\}$ for which the optimal action from $S2$ is to move right? Assume an infinite horizon and a discount $\gamma$ of 1.

We need $p$ s.t. $Q^*(S2, R) \geq Q^*(S2, L)$. This occurs somewhere between 0.8 and 0.9. 0.9 will guarantee that the optimal action is to move right.

To compute exactly which value, note that if the optimal action from $S2$ is to move right then the optimal policy must always be to move right (any policy that goes back and forth just increases the probability of ending up in ground).

Then the payoff from moving right is either 8 if we reach $S7$ or -2 otherwise.

$$E[R] = 8p^5 + (-2)(1 - p^5) = 10p^5 - 2$$
$$E[L] = 3p + (-2)(1 - p) = 5p - 2$$

We need the value from moving right to exceed the value from moving left.

$$10p^5 - 2 > 5p - 2$$
$$p^5 > 0.5$$
$$p > 0.84$$

In the table below, we report the state-action value function $Q^*(s, a)$ for several different choices of $\gamma$ computed by running value iteration to convergence. Assume $p = 1$.

<table>
<thead>
<tr>
<th>$\gamma$</th>
<th>$Q^*(S2, R)$</th>
<th>$Q^*(S3, R)$</th>
<th>$Q^*(S4, R)$</th>
<th>$Q^*(S5, R)$</th>
<th>$Q^*(S6, R)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.7</td>
<td>1.921</td>
<td>2.744</td>
<td>3.92</td>
<td>5.6</td>
<td>8</td>
</tr>
<tr>
<td>0.8</td>
<td>3.277</td>
<td>4.096</td>
<td>5.12</td>
<td>6.4</td>
<td>8</td>
</tr>
<tr>
<td>0.9</td>
<td>5.2488</td>
<td>5.832</td>
<td>6.48</td>
<td>7.2</td>
<td>8</td>
</tr>
<tr>
<td>1.0</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
</tbody>
</table>

(d) What is the minimum value of $\gamma \in \{0.7, 0.8, 0.9, 1.0\}$ for which the optimal action from $S2$ is to move right? Assume an infinite horizon and $p = 1$.

We need $Q(S2, R) \geq Q(S2, L)$. This happens between 0.7 and 0.8, so 0.8 is the first answer where $S2$ should move right.

Since $p = 1$, we never end up in ground. Our optimal policy is still to always move left / move right; any policy that goes back and forth just extends the time before reaching a terminal state, which decreases the payoff based on the discount factor. Moving right earns 8 after 5 time steps, moving left earns 3 after 1 time step.
\[8\gamma^5 > 3\gamma\]
\[\gamma > 0.78\]

(e) We can write down two-step lookahead variants of the Bellman equations which define state values in terms of the next \(s'\) and the following state \(s''\). Write down the utility \(V^\pi(s)\) of a state \(s\) under policy \(\pi\) in terms of the next two states, given that

\[
V^\pi(s) = \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V^\pi(s')]
\]

(f) Write a two-step look-ahead value iteration update that involves \(V(s)\) and \(V(s'')\), where \(s''\) is the state two time steps later. You should make sure to include the base case where you are in the 0th or 1st time steps. Why would this update not be used in practice?

\[
V_{i+2}(s) = \max_a \sum_{s'} T(s, a, s')[R(s, a, s') + \gamma \max_{a'} \sum_{s''} T(s', a', s'')[R(s', a', s'') + \gamma V_i(s'')]}
\]

In practice, this update would not be used because it wastes time implicitly computing \(V_{i+1}\).
3 Soccer

A soccer robot $A$ is on a fast break toward the goal, starting in position 1. From positions 1 through 3, it can either shoot (S) or dribble the ball forward (D). From 4 it can only shoot. If it shoots, it either scores a goal (state G) or misses (state M). If it dribbles, it either advances a square or loses the ball, ending up in $M$. When shooting, the robot is more likely to score a goal from states closer to the goal; when dribbling, the likelihood of missing is independent of the current state.

In this MDP, the states are 1,2,3,4,G and M, where G and M are terminal states. The transition model depends on the parameter $y$, which is the probability of dribbling success. Assume a discount of $\gamma = 1$.

$$T(k, S, G) = \frac{k}{6}$$
$$T(k, S, M) = 1 - \frac{k}{6}$$
$$T(k, D, k + 1) = y \text{ for } k \in \{1, 2, 3\}$$
$$T(k, D, M) = 1 - y \text{ for } k \in \{1, 2, 3\}$$
$$R(k, S, G) = 1$$

Rewards are 0 for all other transitions.

(a) What is $V^\pi(1)$ for the policy $\pi$ that always shoots?

$$V^\pi(1) = T(1, S, G)R(1, S, G) + T(1, S, M)R(1, S, M) = \left(\frac{1}{6}\right)(1) + \left(\frac{5}{6}\right)(0) = \frac{1}{6}$$

(b) What is $Q^*(3, D)$ in terms of $y$?

$$Q^*(3, D) = T(3, D, 4)(R(3, D, 4) + V^*(4)) + T(3, D, M)R(3, D, M)$$
$$= T(3, D, 4)V^*(4)$$
$$= T(3, D, 4)Q^*(4, S)$$
$$= T(3, D, 4)(T(4, S, G)R(4, S, G) + T(4, S, M)R(4, S, M))$$
$$= T(3, D, 4)T(4, S, G)R(4, S, G)$$
$$= y(4/6)(1) = 2/3y$$
(c) Using \( y = \frac{3}{4} \), complete the first two iterations of value iteration.

As an example, consider computing \( Q_2(1, D) \)

\[
Q_i^*(s, a) = \sum_{s'} T(s, a, s')(R(s, a, s') + \gamma V^*_{i-1}(s'))
\]

\[
Q_2(1, D) = \sum_{s' \in \{2, M\}} T(1, D, s')(R(1, D, s') + \gamma V^*_1(s'))
\]

\[
= T(1, D, 2)(R(1, D, 2) + \gamma V^*_1(2)) + T(1, D, M)(R(1, D, M) + \gamma V^*_1(M))
\]

\[
= y(0 + (1)(1/3)) + (1 - y)(0 + (0)(0))
\]

\[
= 1/4
\]

\[
\begin{array}{c|cccc}
 i & Q_i(1, S) & Q_i(2, S) & Q_i(3, S) & Q_i(4, S) \\
0 & 0 & 0 & 0 & 0 \\
1 & 1/6 & 1/3 & 1/2 & 2/3 \\
2 & 1/6 & 1/3 & 1/2 & 2/3 \\
\end{array}
\]

\[
\begin{array}{c|ccc}
 i & Q_i(1, D) & Q_i(2, D) & Q_i(3, D) \\
0 & 0 & 0 & 0 \\
1 & 0 & 0 & 0 \\
2 & 1/4 & 3/8 & 1/2 \\
\end{array}
\]

\[
\begin{array}{c|cccc}
 i & V_i^*(1) & V_i^*(2) & V_i^*(3) & V_i^*(4) \\
0 & 0 & 0 & 0 & 0 \\
1 & 1/6 & 1/3 & 1/2 & 2/3 \\
2 & 1/4 & 3/8 & 1/2 & 2/3 \\
\end{array}
\]

(d) After how many iterations will value iteration compute the optimal values for all states?

Note that \( Q(i, S) \) converges after 1 iteration because we can only transition to terminal states. It takes 1 iteration for these values to propagate to the \( Q(i, D) \) table. Exactly 2 entries from the \( Q(i, D) \) table appear in \( V^*_2 \), each of which requires 1 iteration to converge for a total of 1+2=3 iterations.

If \( y > 3/4 \), a fourth iteration is required; 3 entries from \( Q(i, D) \) appear. This is the maximum because there are only 4 transitions.

(e) For what range of values of \( y \) is \( Q^*(3, S) \geq Q^*(3, D) \)?

\[
\begin{align*}
Q^*(3, S) & \geq Q^*(3, D) \\
T(3, S, G) & \geq T(3, D, 4)T(4, S, G) \\
1/2 & \geq 2/3y \\
3/4 & \geq y \geq 0
\end{align*}
\]

The dribble success probability \( y \) depends on the presence or absence of a defending robot \( R \). \( A \) has no way of detecting whether \( R \) is present, but does know some statistical properties of its environment. \( R \) is present \( \frac{2}{3} \) of the time. When \( R \) is absent, \( y = \frac{3}{4} \). When \( R \) is present, \( y = \frac{1}{2} \).

(f) What transition model should \( A \) use in order to correctly compute its maximum expected reward when it doesn’t know whether or not \( R \) is present?

To maximize expected total reward, the agent should model the situation as accurately as possible. After 1 successful dribble, it is less likely that a defender was present, and we can use Bayes’ rule to update the transition model to reflect the updated beliefs about the probability of a successful dribble.
Let $X_1, X_2, X_3$ denote successful first, second, and third dribbles.  
Using prior beliefs over $R,$

$$T(1, D, 2) = P(X_1) = P(R)P(X_1|R) + P(\bar{R})P(X_1|\bar{R}) = (2/3)(1/4) + (1/3)(3/4) = 5/12$$

Using Bayes rule,

$$T(2, D, 3) = P(X_2|X_1) = \sum_R P(R|X_1)P(X_2|R)$$

$$P(R|X_1) = \frac{P(X_1|R)P(R)}{P(X_1|R)P(R) + P(X_1|\bar{R})P(\bar{R})} = \frac{(1/4)(2/3)}{(1/4)(2/3) + (3/4)(1/3)} = 2/5$$

$$P(X_2|X_1) = (2/5)(1/4) + (3/5)(3/4) = 11/20$$

so $T(2, D, 3) = 11/20,$

Finally, to compute the probability of the last transition,

$$T(3, D, 4) = P(X_3|X_1, X_2) = \sum_R P(R|X_1, X_2)P(X_3|R)$$

$$P(R|X_1, X_2) = \frac{P(X_1, X_2|R)P(R)}{P(X_1, X_2|R)P(R) + P(X_1, X_2|\bar{R})P(\bar{R})} = \frac{(1/4)(1/4)(2/3)}{(1/4)(1/4)(2/3) + (3/4)(3/4)(1/3)} = 2/11$$

$$P(X_3|X_1, X_2) = (2/11)(1/4) + (9/11)(3/4) = 29/44$$

The probability of scoring a goal remains unchanged.

(g) What is the optimal policy $\pi^*$ when $A$ doesn’t know whether or not $R$ is present?

The optimal policy is to shoot from every state.

Using the transition model from part (f),

$$V^*(4) = 2/3$$
$$Q^*(3, S) = 1/2$$
$$Q^*(3, D) = 29/44V^*(4) + 15/44(0) = 58/132 \approx 0.44$$
$$V^*(3) = 1/2$$
$$\pi^*(3) = S$$
$$Q^*(2, S) = 1/3$$
$$Q^*(2, D) = 11/20V^*(3) + 9/20(0) = 11/40 \approx 0.28$$
$$V^*(2) = 1/3$$
$$\pi^*(2) = S$$
$$Q^*(1, S) = 1/6$$
$$Q^*(1, D) = 5/12V^*(3) + 7/12(0) = 5/36 < 6/36$$
$$V^*(1) = 1/6$$
$$\pi^*(2) = S$$