The game High-Low, which we saw in lecture this week, is a card game played with an infinite deck containing three types of cards: 2, 3, and 4. You start with a 3 showing, and say either high or low. Then, a new card is flipped and if the you were right, you win the points shown on the new card, if it’s a tie, you don’t get any points, and if you’re wrong the game ends. An example of a game is: \([3, \text{high}, 4, \text{low}, 2, \text{high}, 3, \text{low}, 4, \text{done}]\). The deck contains different proportions of 2, 3, and 4 cards, \(p_2, p_3, \text{and } p_4\) (where \(p_2 + p_3 + p_4 = 1\)), which you may or may not know.

1 High-Low as an MDP

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

**States:**
The current card, or done. \(s \in \{2, 3, 4, \text{done}\}\)

**Actions:**
\(\text{high or low}\)

**Rewards:**
- \((2, \text{high}, i): i, i \in \{3, 4\}\)
- \((2, \text{low}, 2): 2\)
- \((3, \text{high}, 4): 4\)
- \((3, \text{high}, 3): 0\)
- \((3, \text{low}, 3): 0\)
- \((3, \text{low}, 2): 2\)
- \((4, \text{high}, 4): 0\)
- \((4, \text{low}, i)i, i \in \{2, 3\}\)

**Transitions:**

<table>
<thead>
<tr>
<th></th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>done</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2,low)</td>
<td>(p_2)</td>
<td>0</td>
<td>0</td>
<td>(p_3 + p_4)</td>
</tr>
<tr>
<td>(2, hi)</td>
<td>(p_2)</td>
<td>(p_3)</td>
<td>(p_4)</td>
<td>0</td>
</tr>
<tr>
<td>(3, low)</td>
<td>(p_2)</td>
<td>(p_3)</td>
<td>0</td>
<td>(p_4)</td>
</tr>
<tr>
<td>(3, hi)</td>
<td>0</td>
<td>(p_3)</td>
<td>(p_4)</td>
<td>(p_2)</td>
</tr>
<tr>
<td>(4, low)</td>
<td>(p_2)</td>
<td>(p_3)</td>
<td>(p_4)</td>
<td>0</td>
</tr>
<tr>
<td>(4, hi)</td>
<td>0</td>
<td>0</td>
<td>(p_4)</td>
<td>(p_2 + p_4)</td>
</tr>
</tbody>
</table>

(b) Write down Bellman equations for \(V^*(3)\), the value of seeing 3 under the optimal policy, and \(Q^*(3, \text{high})\), the utility of saying high after seeing 3 in terms of the \(p_i\)’s and other \(V^*\)’s and \(Q\)’s.

\[V^*(3) = \max \left( p_3(3 + \gamma V^*(3)) + p_4(4 + \gamma V^*(4)) , \quad p_2(2 + V^*(2)) + p_3(3 + V^*(3)) \right)\]
\[ Q(3, \text{high}) = \]

\[ Q^*(3, \text{high}) = \sum_{s'} T(3, \text{high}, s') \left( R(3, \text{high}, s') + \gamma \max_{a'} Q(s', a') \right) \]

\[ = p_3 \left( 3 + \max_{a' \in \{\text{high, low}\}} Q^*(3, a') \right) + p_4 \left( 4 + \max_{a' \in \{\text{high, low}\}} Q^*(4, a') \right) + p_2 \cdot 0 \]

\[ = 2.0 \]

\[ (2, \text{high}) = 2.0 \]
\[ (4, \text{low}) = 1.0 \]
\[ (3, \text{high}) = 1.0 \]

\[ (2, \text{low}) = 1.0 \]

2 Q-learning

Suppose we’re playing with a deck for which we don’t know the \( p_i \)’s. We still need to figure out an optimal policy, so we use Q-learning, using a learning rate of \( \alpha = 0.5 \) and a discount of \( \gamma = 0.5 \).

(a) Our first episode is \([3, \text{high}, 4, \text{low}, 2, \text{high}, 3, \text{high}, 2, \text{done}]\).

(i) How many \( Q(s, a) \) updates have we done?

(ii) Give the new \( Q(s, a) \) values for after this iteration

\[ (2, \text{high}) = 2.0 \]
\[ (4, \text{low}) = 1.0 \]
\[ (3, \text{high}) = 1.0 \]

(b) Our second episode is: \([3, \text{low}, 2, \text{low}, 3, \text{done}]\), and our third is \([3, \text{high}, 4, \text{low}, 3, \text{low}, 3, \text{low}, 4, \text{done}]\). Give the Q values after the episodes have been processed

\[ (2, \text{low}) = 0 \]
\[ (2, \text{high}) = 2.0 \]
\[ (4, \text{low}) = 2.6875 \]
\[ (3, \text{high}) = 2.75 \]
\[ (3, \text{low}) = 1.46875 \]