Sequential Decision Process Explained
(in context of 17.4 in textbook)

Value Iteration

\[ U'(s) = R(s) + \max_a \sum_s T(s, a, s') U(s') \]

At each iteration, apply this formula to each state to update the utility of those states.

- \( s \) is a state.
- \( U'(s) \) is the updated utility of state \( s \).
- \( R(s) \) is the reward of state \( s \).
- \( \gamma \) is the discounting factor.

For all resulting state \( s' \) of doing an action \( a \) in state \( s \), we take the summation of \( T(s, a, s') U(s') \), where \( T(s, a, s') \) is the transition function that returns the probability of reaching a resulting state \( s' \) and \( U(s') \) is the utility of that resulting state.

We take the maximum of those summations, add it to the reward of state \( s \), and that is our new utility for state \( s, U'(s) \).

Policy Iteration

For policy iteration, we do alternating steps of

- Value determination

\[ U(s) = R(s) + \gamma \sum_s T(s, \pi(s), s') U(s') \]

- \( s \) is a state.
- \( U'(s) \) is the utility of state \( s \) in one iteration.
- \( R(s) \) is the reward of state \( s \).
- \( \gamma \) is the discounting factor.

For all resulting states of the action that the current policy suggests (for state \( s \)), take the summation of \( T(s, \pi(s), s') U(s') \), i.e. the probability of reaching the resulting state \( s' \) from \( s \) times the utility of the resulting state \( s' \).

- Policy update
\[ \pi'(s) = \arg \max_a \sum_s T(s,a,s')U(s') \]

In this step, we choose the optimal action that has the best expected utility, which is the summation of \( T(s,a,s')U(s') \) over all resulting state given the action and current state \( s \).

**Example**

The original problem is 17.4 in the textbook, and here is the summarized version.

<table>
<thead>
<tr>
<th>State ((s))</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reward ((r))</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
</tr>
</tbody>
</table>

Actions: \( a, b \)

Transition functions:

\[
T(1,a,1) = 0.2 \\
T(1,a,2) = 0.8 \\
T(1,b,1) = 0.9 \\
T(1,b,3) = 0.1 \\
T(2,a,1) = 0.8 \\
T(2,a,2) = 0.2 \\
T(2,b,2) = 0.9 \\
T(2,b,3) = 0.1
\]

State 3 is a terminal state, so there is no transition function for it.

Discounting factor \( \gamma \) is assumed to be 1.

**Solution**

- Value Iteration

We initialize the initial utilities of state 1, 2 and 3, to be 0, i.e. \( U_0(1) = 0, U_0(2) = 0, U_0(3) = 0 \). The subscript of \( U \) denotes the number of iteration step.

**Iteration 1:**

\[
U_1(1) = R(1) + \max \left( \sum_s T(1,a,s')U_0(s'), \sum_s T(1,b,s')U_0(s') \right) \\
= -1 + \max(0.2U_0(1) + 0.8U_0(2), 0.1U_0(3) + 0.9U_0(1)) \\
= -1 + \max(0,0)
\]
\[ U_1(2) = R(2) + \max \left( \sum_s T(1, a, s')U_0(s'), \sum_s T(1, b, s')U_0(s') \right) \]
\[ = -2 + \max(0.2U_0(2) + 0.8U_0(1), 0.1U_0(3) + 0.9U_0(2)) \]
\[ = -2 + \max(0,0) \]
\[ = -2 \]
\[ U_1(3) = 0 \text{ as state 3 is the terminal state (no transition functions for state 3 and the reward for state 3 is 0)} \]

**Iteration 2:**

\[ U_2(1) = R(1) + \max \left( \sum_s T(1, a, s')U_1(s'), \sum_s T(1, b, s')U_1(s') \right) \]
\[ = -1 + \max(0.2U_1(1) + 0.8U_1(2), 0.1U_1(3) + 0.9U_1(1)) \]
\[ = -1 + \max(0.2 \cdot -1 + 0.8 \cdot -2, 0.1 \cdot 0 + 0.9 \cdot -1) \]
\[ = -2.8 \]

\[ U_2(2) = R(2) + \max \left( \sum_s T(1, a, s')U_1(s'), \sum_s T(1, b, s')U_1(s') \right) \]
\[ = -2 + \max(0.2U_2(2) + 0.8U_1(1), 0.1U_1(3) + 0.9U_1(2)) \]
\[ = -2 + \max(0.2 \cdot -2 + 0.8 \cdot -1, 0.1 \cdot 0 + 0.9 \cdot -2) \]
\[ = -2 \]

\[ U_1(3) = 0 \]

You do the same thing for iteration 3,4... until they converge.

- **Policy Iteration**
  Our randomly generated initial policy for each state is
  \[ \pi_0(1) = b, \pi_0(2) = b \]
  There is no policy for state 3 because it is terminal. The subscript of \( \pi \) denotes

**Iteration 1:**

- **Value determination**
  Since \( \pi_0(1) \) suggest action \( b \) for state 1, we have
  \[ U(1) = R(1) + \left( T(1, b, 1)U(1) + T(1, b, 3)U(3) \right) \]
  \[ = -1 + (0.9U(1) + 0.1U(3)) \]
  Since \( \pi_0(2) \) suggest action \( b \) for state 1, we have
  \[ U(2) = R(2) + \left( T(2, b, 1)U(1) + T(2, b, 3)U(3) \right) \]
  \[ = -2 + (0.9U(2) + 0.1U(3)) \]
  \[ U(3) = R(3) = 0 \]
  After solving the set of equations, we get \( U(1) = -10 \) and \( U(2) = -20 \).
- **Policy update**
\[
\pi'(1) = \arg \max_{\text{action}} \sum_s T(1, \text{action}, s') U(s')
\]

For action a, \(\sum_s T(1, a, s') U(s') = 0.2U(1) + 0.8U(2) = -18\)

For action b, \(\sum_s T(1, b, s') U(s') = 0.1U(3) + 0.9U(1) = -9\)

Since the expected utility for action b is bigger \((-9 > -18)\), \(\pi_1(1) = b\).

Similarly,
\[
\pi'(2) = \arg \max_{\text{action}} \sum_s T(2, \text{action}, s') U(s')
\]

For action a, \(\sum_s T(2, a, s') U(s') = 0.2U(2) + 0.8U(1) = -12\)

For action b, \(\sum_s T(2, b, s') U(s') = 0.1U(3) + 0.9U(2) = -18\)

Since the expected utility for action a is bigger \((-12 > -18)\), \(\pi_1(2) = a\).

**Iteration 2:**

- **Value determination**

Since \(\pi_1(1)\) suggest action \(b\) for state 1, we have
\[
U(1) = R(1) + \left( T(1, b, 1)U(1) + T(1, b, 3)U(3) \right) \\
= -1 + \left( 0.9U(1) + 0.1U(3) \right)
\]

Since \(\pi_1(2)\) suggest action \(a\) for state 1, we have
\[
U(2) = R(2) + \left( T(2, a, 1)U(1) + T(2, a, 3)U(3) \right) \\
= -2 + \left( 0.2U(2) + 0.8U(1) \right)
\]

\(U(3) = R(3) = 0\)

After solving the set of equations, we get \(U(1) = -10\) and \(U(2) = -12.5\).

- **Policy update**

The equations are the same as those in iteration 1, but \(U(1)\) and \(U(2)\) are updated.
\[
\pi'(1) = \arg \max_{\text{action}} \sum_s T(1, \text{action}, s') U(s')
\]

For action a, \(\sum_s T(1, a, s') U(s') = 0.2U(1) + 0.8U(2) = -12\)

For action b, \(\sum_s T(1, b, s') U(s') = 0.1U(3) + 0.9U(1) = -9\)
Since the expected utility for action $b$ is bigger ($-9 > -12$), $\pi_2(1) = b$.

Similarly,

$$\pi'(2) = \arg \max_{\text{action}} \sum_{s'} T(2, \text{action}, s') U(s')$$

For action $a$, $\sum_{s'} T(2, a, s') U(s') = 0.2U(2) + 0.8U(1) = -10.5$

For action $b$, $\sum_{s'} T(2, b, s') U(s') = 0.1U(3) + 0.9U(2) = -11.25$

Since the expected utility for action $a$ is bigger ($-10.5 > -11.25$), $\pi_1(2) = a$.

$\pi_1$ and $\pi_2$ are the same, so our policy is $\pi(1) = b$, $\pi(2) = a$. 