Exercise or relax, for $\gamma = 0.9$

Recall (probability, reward)-matrices for exercise, relax

<table>
<thead>
<tr>
<th>exercise</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.99, 8</td>
<td>.01, 8</td>
</tr>
<tr>
<td>unfit</td>
<td>.2, 0</td>
<td>.8, 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relax</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.7, 10</td>
<td>.3, 10</td>
</tr>
<tr>
<td>unfit</td>
<td>0, 5</td>
<td>1, 5</td>
</tr>
</tbody>
</table>
Exercise or relax, for $\gamma = 0.9$

Recall (probability, reward)-matrices for exercise, relax

<table>
<thead>
<tr>
<th>exercise</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.99, 8</td>
<td>.01, 8</td>
</tr>
<tr>
<td>unfit</td>
<td>.2, 0</td>
<td>.8, 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relax</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.7, 10</td>
<td>.3, 10</td>
</tr>
<tr>
<td>unfit</td>
<td>0, 5</td>
<td>1, 5</td>
</tr>
</tbody>
</table>

$q_0(s, a) := p(s, a, \text{fit})r(s, a, \text{fit}) + p(s, a, \text{unfit})r(s, a, \text{unfit})$

$V_n(s) := \max(q_n(s, \text{exercise}), q_n(s, \text{relax}))$

$q_{n+1}(s, a) := q_0(s, a) + .9(p(s, a, \text{fit})V_n(\text{fit}) + p(s, a, \text{unfit})V_n(\text{unfit}))$
Exercise or relax, for $\gamma = 0.9$

Recall (probability, reward)-matrices for exercise, relax

<table>
<thead>
<tr>
<th></th>
<th>exercise</th>
<th></th>
<th>relax</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>fit</td>
<td>unfit</td>
<td>fit</td>
<td>unfit</td>
</tr>
<tr>
<td>fit</td>
<td>.99, 8</td>
<td>.01, 8</td>
<td>.7, 10</td>
<td>.3, 10</td>
</tr>
<tr>
<td>unfit</td>
<td>.2, 0</td>
<td>.8, 0</td>
<td>0, 5</td>
<td>1, 5</td>
</tr>
</tbody>
</table>

$q_0(s, a) := p(s, a, \text{fit})r(s, a, \text{fit}) + p(s, a, \text{unfit})r(s, a, \text{unfit})$

$V_n(s) := \max(q_n(s, \text{exercise}), q_n(s, \text{relax}))$

$q_{n+1}(s, a) := q_0(s, a) + .9(p(s, a, \text{fit})V_n(\text{fit}) + p(s, a, \text{unfit})V_n(\text{unfit}))$

<table>
<thead>
<tr>
<th></th>
<th>exercise</th>
<th>relax</th>
<th>$\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>8</td>
<td>10</td>
<td>relax</td>
</tr>
<tr>
<td>unfit</td>
<td>0</td>
<td>5</td>
<td>relax</td>
</tr>
</tbody>
</table>
Exercise or relax, for $\gamma = 0.9$

Recall (probability, reward)-matrices for exercise, relax

<table>
<thead>
<tr>
<th></th>
<th>exercise</th>
<th></th>
<th>unfit</th>
<th>relax</th>
<th></th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.99, 8</td>
<td>01, 8</td>
<td></td>
<td>.7, 10</td>
<td>3, 10</td>
<td></td>
</tr>
<tr>
<td>unfit</td>
<td>.2, 0</td>
<td>.8, 0</td>
<td></td>
<td>0, 5</td>
<td>1, 5</td>
<td></td>
</tr>
</tbody>
</table>

$q_0(s, a) := p(s, a, \text{fit})r(s, a, \text{fit}) + p(s, a, \text{unfit})r(s, a, \text{unfit})$

$V_n(s) := \max(q_n(s, \text{exercise}), q_n(s, \text{relax}))$

$q_{n+1}(s, a) := q_0(s, a) + .9(p(s, a, \text{fit})V_n(\text{fit}) + p(s, a, \text{unfit})V_n(\text{unfit}))$

<table>
<thead>
<tr>
<th></th>
<th>exercise</th>
<th></th>
<th>unfit</th>
<th>relax</th>
<th></th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>8, 16.955</td>
<td></td>
<td></td>
<td>10, 17.65</td>
<td></td>
<td></td>
</tr>
<tr>
<td>unfit</td>
<td>0, 5.4</td>
<td></td>
<td></td>
<td>5, 9.5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

$\pi_{\text{relax, relax}}$
Exercise or relax, for $\gamma = 0.9$

Recall (probability, reward)-matrices for exercise, relax

<table>
<thead>
<tr>
<th>exercise</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.99, 8</td>
<td>.01, 8</td>
</tr>
<tr>
<td>unfit</td>
<td>.2, 0</td>
<td>.8, 0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>relax</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.7, 10</td>
<td>.3, 10</td>
</tr>
<tr>
<td>unfit</td>
<td>0, 5</td>
<td>1, 5</td>
</tr>
</tbody>
</table>

$q_0(s, a) := p(s, a, \text{fit})r(s, a, \text{fit}) + p(s, a, \text{unfit})r(s, a, \text{unfit})$

$V_n(s) := \max(q_n(s, \text{exercise}), q_n(s, \text{relax}))$

$q_{n+1}(s, a) := q_0(s, a) + .9(p(s, a, \text{fit})V_n(\text{fit}) + p(s, a, \text{unfit})V_n(\text{unfit}))$

<table>
<thead>
<tr>
<th></th>
<th>exercise</th>
<th>relax</th>
<th>$\pi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>8, 16.955, 23.812</td>
<td>10, 17.65, 23.685</td>
<td>relax, relax, exercise</td>
</tr>
<tr>
<td>unfit</td>
<td>0, 5.4, 10.017</td>
<td>5, 9.5, 13.55</td>
<td>relax, relax, relax</td>
</tr>
</tbody>
</table>
Temporal difference (TD)

A sequence of values

\[ v_1, v_2, v_3, \ldots \]

averages at time \( k \) to

\[ A_k := \frac{v_1 + \cdots + v_k}{k} \]
Temporal difference (TD)

A sequence of values

\[ v_1, v_2, v_3, \ldots \]

averages at time \( k \) to

\[ A_k := \frac{v_1 + \cdots + v_k}{k} \]

which learning \( v_{k+1} \) updates to

\[
A_{k+1} = \frac{v_1 + \cdots + v_k + v_{k+1}}{k + 1} \\
= \frac{k}{k + 1} A_k + \frac{1}{k + 1} v_{k+1}
\]
Temporal difference (TD)

A sequence of values

\[ v_1, v_2, v_3, \ldots \]

averages at time \( k \) to

\[ A_k := \frac{v_1 + \cdots + v_k}{k} \]

which learning \( v_{k+1} \) updates to

\[
A_{k+1} = \frac{v_1 + \cdots + v_k + v_{k+1}}{k + 1} \\
= \frac{k}{k + 1} A_k + \frac{1}{k + 1} v_{k+1}
\]

so if \( \alpha_k = \frac{1}{k} \),

\[
A_{k+1} = (1 - \alpha_{k+1}) A_k + \alpha_{k+1} v_{k+1}
\]

\[
= \underbrace{A_k}_{\text{old}} + \alpha_{k+1} (v_{k+1} - A_k)_{\text{temp diff: new-old}}
\]
Q-Learning

Assume $v_{k+1}$ is derived from $r_{k+1}, s_{k+1}$, observed sequentially

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \cdots \quad s_k \xrightarrow{a_k} r_{k+1}, s_{k+1} \xrightarrow{a_{k+1}} \cdots$$

experience from which we learn

$$v_{k+1} := r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a)$$
Q-Learning

Assume $v_{k+1}$ is derived from $r_{k+1}, s_{k+1}$, observed sequentially

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \cdots \quad s_k \xrightarrow{a_k} r_{k+1}, s_{k+1} \xrightarrow{a_{k+1}} \cdots$$

experience from which we learn

$$v_{k+1} := r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a)$$

given $0 \leq \gamma < 1$, $Q_1 : (S \times A) \rightarrow \mathbb{R}$ and $v_1 \in \mathbb{R}$, with

$$Q_{k+1}(s_k, a_k) := (1 - \alpha)Q_k(s_k, a_k) + \alpha v_{k+1}$$

for $0 \leq \alpha \leq 1$, 
Q-Learning

Assume $v_{k+1}$ is derived from $r_{k+1}, s_{k+1}$, observed sequentially

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \ldots s_k \xrightarrow{a_k} r_{k+1}, s_{k+1} \xrightarrow{a_{k+1}} \ldots$$

experience from which we learn

$$v_{k+1} := r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a)$$

given $0 \leq \gamma < 1$, $Q_1 : (S \times A) \rightarrow \mathbb{R}$ and $v_1 \in \mathbb{R}$, with

$$Q_{k+1}(s_k, a_k) := (1 - \alpha)Q_k(s_k, a_k) + \alpha v_{k+1}$$

for $0 \leq \alpha \leq 1$, smelling like

$$A_{k+1} = (1 - \alpha_{k+1})A_k + \alpha_{k+1} v_{k+1} \quad \text{for } \alpha_{k+1} = \frac{1}{k + 1}$$

from previous slide (on TD).
Averaging?

\[ \nu_{k+1} = r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a) \]

\[ Q_{k+1}(s_k, a_k) = (1 - \alpha) Q_k(s_k, a_k) + \alpha \nu_{k+1} \]

\[ A_{k+1} \neq Q_k(s_{k-1}, a_{k-1}) = A_k \]
Averaging?

\[
\begin{align*}
\nu_{k+1} &= r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a) \\
Q_{k+1}(s_k, a_k) &= (1 - \alpha) Q_k(s_k, a_k) + \alpha \nu_{k+1} \\
A_{k+1} &\neq Q_k(s_{k-1}, a_{k-1}) = A_k
\end{align*}
\]

for a deterministic MDP

i.e., \( p(s, a, s') \in \{0, 1\} \) for all \( s, a, s' \)

let \( \alpha = 1 \) as \( \nu_{k+1} \) may look ahead further than \( Q_k \) for the experience \( s_k, a_k, r_{k+1}, s_{k+1} \) (determined by \( s_k, a_k \))
Averaging?

\[ v_{k+1} = r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a) \]

\[
Q_{k+1}(s_k, a_k) = (1 - \alpha) Q_k(s_k, a_k) + \alpha v_{k+1}
\]

\[ A_{k+1} \neq Q_k(s_{k-1}, a_{k-1}) = A_k \]

for a deterministic MDP

i.e., \( p(s, a, s') \in \{0, 1\} \) for all \( s, a, s' \)

let \( \alpha = 1 \) as \( v_{k+1} \) may look ahead further than \( Q_k \) for the experience \( s_k, a_k, r_{k+1}, s_{k+1} \) (determined by \( s_k, a_k \))

for \( 0 < p(s, a, s') < 1 \), sample \( s' \) at frequency \( \propto p(s, a, s') \) to average \( Q \) as a whole (not just \( Q(s, a) \) at a particular \( (s, a) \)), converging to optimal \( Q \) under certain assumptions, including

\[ \sum \alpha_k = \infty \quad \text{and} \quad \sum \alpha_k^2 < \infty \quad (\text{e.g. } \alpha_k = \frac{1}{k}) \]
Exploration-exploitation tradeoff

\[ s \xrightarrow{a} r', s' \quad r', s' \text{ from environment, but } a? \]

\[ Q_{n+1}(s, a) := \alpha [r' + \gamma \max_{a'} Q_n(s', a')] + (1 - \alpha) Q_n(s, a) \]

from functional policy \( \pi : S \to A \) [e.g. \( \pi_Q(s) = \arg \max_a Q(s, a) \)]
Exploration-exploitation tradeoff

\[ s \xrightarrow{a} r', s' \quad \text{from environment, but } a? \]

\[ Q_{n+1}(s, a) := \alpha [r' + \gamma \max_{a'} Q_n(s', a')] + (1 - \alpha) Q_n(s, a) \]

from functional policy \( \pi : S \to A \) [e.g. \( \pi_Q(s) = \arg \max_a Q(s, a) \)]

to \( \pi : (S \times A) \to [0, 1] \) s.t. \( \sum_{a \in A} \pi(s, a) = 1 \) for each \( s \in S \)

e.g.

\[ \pi^\epsilon_Q(s, a) = \begin{cases} 
1 - \frac{n-1}{n} \epsilon & \text{if } a = \arg \max_{a'} Q(s, a') \\
\frac{\epsilon}{n} & \text{otherwise, where } |A| = n \end{cases} \]

(\( \dagger \)) says exploit: use what we know

(\( \ddagger \)) says explore: try something new (for the future)
Exploration-exploitation tradeoff

\[ s \xrightarrow{a} r', s' \xrightarrow{a'} \cdots \quad r', s' \text{ from environment, but } a? \]

\[ Q_{n+1}(s, a) := \alpha[r' + \gamma \max_{a'} Q_n(s', a')] + (1 - \alpha) Q_n(s, a) \]

from functional policy \( \pi : S \rightarrow A \) [e.g. \( \pi_Q(s) = \arg\max_a Q(s, a) \)]

to \( \pi : (S \times A) \rightarrow [0, 1] \) s.t. \( \sum_{a \in A} \pi(s, a) = 1 \) for each \( s \in S \)

e.g.

\[ \pi^\epsilon_Q(s, a) = \begin{cases} 
1 - \frac{n-1}{n} \epsilon & \text{if } a = \arg\max_{a'} Q(s, a') \quad (\dagger) \\
\frac{\epsilon}{n} & \text{otherwise, where } |A| = n \quad (\ddagger) 
\end{cases} \]

(\dagger) says exploit: use what we know

(\ddagger) says explore: try something new (for the future)

SARSA: replace arg max by policy in use

\[ Q_{n+1}(s, a) := \alpha[r' + \gamma Q_n(s', a')] + (1 - \alpha) Q_n(s, a) \]