Given a specification $R$ of immediate rewards after particular actions calculate the return $Q$ of particular actions over time via

$$Q = \lim_{n \to \infty} Q_n$$
A generalisation

Given a specification $R$ of immediate rewards after particular actions calculate the return $Q$ of particular actions over time via

$$Q = \lim_{n \to \infty} Q_n$$

$$Q_{n+1}(s, s') := R(s, s') + \frac{1}{2} \max\{Q_n(s', s'') \mid arc=(s', s'')\} \quad (1)$$

$$Q_{n+1}(s, a) \approx \alpha [R(s, a) + \gamma \max\{Q_n(s', a') \mid a' \in A\}]$$
$$+ \ (1 - \alpha)Q_n(s, a) \quad (2)$$
A generalisation

Given a specification $R$ of immediate rewards after particular actions calculate the return $Q$ of particular actions over time via

$$Q = \lim_{n \to \infty} Q_n$$

$$Q_{n+1}(s, s') := R(s, s') + \frac{1}{2} \max\{Q_n(s', s'') \mid \text{arc}= (s', s'')\} \quad (1)$$

$$\Rightarrow$$

$$Q_{n+1}(s, a) \approx \alpha [R(s, a) + \gamma \max\{Q_n(s', a') \mid a' \in A\}]$$

$$+ (1 - \alpha) Q_n(s, a) \quad (2)$$

(1) is (2) with action $a$ resulting in $s'$ deterministically for $\alpha = 1$, with $\gamma = \frac{1}{2}$

$s'$ is learned from experience (environment)
Markov decision process (MDP)

a 5-tuple $\langle S, A, p, r, \gamma \rangle$ consisting of

- a finite set $S$ of states $s, s', \ldots$
- a finite set $A$ of actions $a, \ldots$
- a function $p : S \times A \times S \rightarrow [0, 1]$

$$p(s, a, s') = \text{prob}(s'|s, a) = \text{how probable is } s' \text{ after doing } a \text{ at } s$$

$$\sum_{s'} p(s, a, s') = 1 \text{ for all } a \in A, \ s \in S$$
Markov decision process (MDP)

is a 5-tuple \( \langle S, A, p, r, \gamma \rangle \) consisting of

- a finite set \( S \) of states \( s, s', \ldots \)
- a finite set \( A \) of actions \( a, \ldots \)
- a function \( p : S \times A \times S \to [0, 1] \)

\[
p(s, a, s') = \text{prob}(s'|s, a) = \text{how probable is } s' \text{ after doing } a \text{ at } s
\]

\[
\sum_{s'} p(s, a, s') = 1 \text{ for all } a \in A, \ s \in S
\]

- a function \( r : S \times A \times S \to \mathbb{R} \)

\[
r(s, a, s') = \text{immediate reward at } s' \text{ after } a \text{ is done at } s
\]
Markov decision process (MDP)

a 5-tuple \( \langle S, A, p, r, \gamma \rangle \) consisting of

- a finite set \( S \) of states \( s, s', \ldots \)
- a finite set \( A \) of actions \( a, \ldots \)
- a function \( p: S \times A \times S \rightarrow [0, 1] \)
  
  \[ p(s, a, s') = \text{prob}(s'|s, a) = \text{how probable is } s' \text{ after doing } a \text{ at } s \]

  \[ \sum_{s'} p(s, a, s') = 1 \text{ for all } a \in A, \ s \in S \]

- a function \( r: S \times A \times S \rightarrow \mathbb{R} \)
  
  \[ r(s, a, s') = \text{immediate reward at } s' \text{ after } a \text{ is done at } s \]

- a discount factor \( \gamma \in [0, 1] \)
Markov decision process (MDP)

a 5-tuple $\langle S, A, p, r, \gamma \rangle$ consisting of

- a finite set $S$ of states $s, s', \ldots$
- a finite set $A$ of actions $a, \ldots$
- a function $p : S \times A \times S \rightarrow [0, 1]$

\[ p(s, a, s') = \text{prob}(s'|s, a) = \text{how probable is } s' \text{ after doing } a \text{ at } s \]

\[ \sum_{s'} p(s, a, s') = 1 \text{ for all } a \in A, \ s \in S \]

- a function $r : S \times A \times S \rightarrow \mathbb{R}$

\[ r(s, a, s') = \text{immediate reward at } s' \text{ after } a \text{ is done at } s \]

- a discount factor $\gamma \in [0, 1]$

Missing: policy $\pi : S \rightarrow A$ (what to do at $s$)
Exercise (Poole & Mackworth, chap 9)

Sam is either fit or unfit

\[ S = \{ \text{fit, unfit} \} \]

and has to decide whether to exercise or relax

\[ A = \{ \text{exercise, relax} \}. \]
Exercise (Poole & Mackworth, chap 9)

Sam is either fit or unfit

\[ S = \{ \text{fit, unfit}\} \]

and has to decide whether to exercise or relax

\[ A = \{ \text{exercise, relax}\}. \]

\( p(s, a, s') \) and \( r(s, a, s') \) are a-table entries for row \( s \), col \( s' \)

<table>
<thead>
<tr>
<th>exercise</th>
<th>fit</th>
<th>unfit</th>
</tr>
</thead>
<tbody>
<tr>
<td>fit</td>
<td>.99</td>
<td>8</td>
</tr>
<tr>
<td>unfit</td>
<td>.2</td>
<td>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</td>
<td>10</td>
</tr>
<tr>
<td>unfit</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

Entries in red follow from assuming immediate rewards do not depend on the resulting state
Exercise (Poole & Mackworth, chap 9)

Sam is either fit or unfit

\[ S = \{ \text{fit, unfit} \} \]

and has to decide whether to exercise or relax

\[ A = \{ \text{exercise, relax} \}. \]

\( p(s, a, s') \) and \( r(s, a, s') \) are a-table entries for row \( s \), col \( s' \)

<table>
<thead>
<tr>
<th></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></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>

Entries in red follow from assuming immediate rewards do not depend on the resulting state, and

\[ \sum_{s'} p(s, a, s') = 1 \]
Grid World

states: 100 positions
actions: up, down, left, right
punish -1 when banging into wall & 4 reward/punish states
prob: 0.7 as directed (if possible)

Poole & Mackworth, 9.5
Policy from an MDP

Given state \( s \), pick action \( a \) that maximizes return for different outcomes \( s' \) and discounted future:

\[
Q(s, a) := \sum_{s'} p(s, a, s') \left( r(s, a, s') + \gamma V(s') \right)
\]

immediate

for \( V \) tied back to \( Q \) via policy \( \pi : S \rightarrow A \):

\[
V_\pi(s) := Q(s, \pi(s))
\]
Policy from an MDP

Given state $s$, pick action $a$ that maximizes return

$$Q(s, a) := \sum_{s'} p(s, a, s') \left( r(s, a, s') + \gamma V(s') \right)$$

immediate

for $V$ tied back to $Q$ via policy $\pi : S \rightarrow A$

$$V_\pi(s) := Q(s, \pi(s))$$

e.g., the greedy $Q$-policy above

$$\pi(s) := \arg \max_a Q(s, a)$$

for

$$Q(s, a) = \sum_{s'} p(s, a, s')(r(s, a, s') + \gamma \max_{a'} Q(s', a'))$$
Value iteration

Mutual recursion between $Q/V$ and $\pi$

value of an action/state vs what to do at a state
Value iteration

Mutual recursion between $Q/V$ and $\pi$

value of an action/state vs what to do at a state

Focus on $Q$, approached in the limit

$$\lim_{n \to \infty} q_n$$

from iterates

$$q_0(s, a) := \sum_{s'} p(s, a, s') r(s, a, s')$$

$$q_{n+1}(s, a) := \sum_{s'} p(s, a, s') \left( r(s, a, s') + \gamma \max_{a'} q_n(s', a') \right)$$
Value iteration

Mutual recursion between $Q/V$ and $\pi$

value of an action/state vs what to do at a state

Focus on $Q$, approached in the limit

$$\lim_{n \to \infty} q_n$$

from iterates

$$q_0(s, a) := \sum_{s'} p(s, a, s') r(s, a, s')$$

$$q_{n+1}(s, a) := \sum_{s'} p(s, a, s') (r(s, a, s') + \gamma \max_{a'} q_n(s', a'))$$

In case $p(s, a, s') = 1$ for some $s'$ (necessarily unique), the iterates simplify to

$$q_0(s, a) := r(s, a, s')$$

$$q_{n+1}(s, a) := r(s, a, s') + \gamma \max_{a'} q_n(s', a')$$
Deterministic actions and absorbing states (game over)

Fix an MDP with min reward $m$.
An action $a$ is $s$-deterministic if $p(s, a, s') = 1$ for some $s'$. 
Deterministic actions and absorbing states (game over)

Fix an MDP with min reward $m$.

An action $a$ is $s$-deterministic if $p(s, a, s') = 1$ for some $s'$.

A state $s$ is absorbing if $p(s, a, s) = 1$ for every action $a$, whence

$$Q(s, a) = r(s, a, s) + \gamma V(s)$$

$$V(s) = \frac{r_s}{1 - \gamma} \text{ where } r_s = \max_a r(s, a, s)$$

A state $s$ is a sink if it is absorbing and $r(s, a, s) = m$ for all $a$. 
Deterministic actions and absorbing states (game over)

Fix an MDP with min reward $m$.

An action $a$ is $s$-deterministic if $p(s, a, s') = 1$ for some $s'$.

A state $s$ is absorbing if $p(s, a, s) = 1$ for every action $a$, whence

$$Q(s, a) = r(s, a, s) + \gamma V(s)$$

$$V(s) = \frac{r_s}{1 - \gamma} \quad \text{where } r_s = \max_a r(s, a, s)$$

A state $s$ is a sink if it is absorbing and $r(s, a, s) = m$ for all $a$.

An action $a$ is an $s$-drain if for some sink $s'$,

$$p(s, a, s') = 1 \text{ and } r(s, a, s') = m$$
Deterministic actions and absorbing states (game over)

Fix an MDP with min reward $m$.

An action $a$ is \emph{s-deterministic} if $p(s, a, s') = 1$ for some $s'$.

A state $s$ is \emph{absorbing} if $p(s, a, s) = 1$ for every action $a$, whence

$$Q(s, a) = r(s, a, s) + \gamma V(s)$$

$$V(s) = \frac{r_s}{1 - \gamma} \quad \text{where } r_s = \max_a r(s, a, s)$$

A state $s$ is a \emph{sink} if it is absorbing and $r(s, a, s) = m$ for all $a$.

An action $a$ is an \emph{s-drain} if for some sink $s'$,

$$p(s, a, s') = 1 \text{ and } r(s, a, s') = m$$

Let

$$A(s) := \{ a \in A \mid a \text{ is not an s-drain} \}$$

so if $A(s) \neq \emptyset$,

$$V(s) = \max\{ Q(s, a) \mid a \in A \} = \max\{ Q(s, a) \mid a \in A(s) \}$$
Arcs & goals as a deterministic MDP ($p \in \{0, 1\}$)

Given arc and goal set $G$, let

$$A = \{ s \mid (\exists s') \ arc= (s', s) \} = S$$

where for each $a \in A$,

$$p(s, a, s') = \begin{cases} 1 & \text{if } a = s' \text{ and } arc= (s, s') \\ 0 & \text{otherwise} \end{cases}$$

$$r(s, a, s') = \begin{cases} R(s, s') & \text{if } a = s' \text{ and } arc= (s, s') \\ \text{anything} & \text{otherwise} \end{cases}$$
Arcs & goals as a deterministic MDP ($p \in \{0, 1\}$)

Given arc and goal set $G$, let

$$A = \{ s \mid (\exists s') \text{ arc}=(s', s) \} = S$$

where for each $a \in A$,

$$p(s, a, s') = \begin{cases} 1 & \text{if } a = s' \text{ and } \text{arc}=(s, s') \\ 0 & \text{otherwise} \end{cases}$$

$$r(s, a, s') = \begin{cases} R(s, s') & \text{if } a = s' \text{ and } \text{arc}=(s, s') \\ \text{anything} & \text{otherwise} \end{cases}$$

Satisfy prob constraint $\sum_{s'} p(s, a, s') = 1$ via sink state $\perp \notin A$, requiring of every $a \in A$ and $s \in S$,

$$p(s, a, \perp) = \begin{cases} 1 & \text{if not } \text{arc}=(s, a) \\ 0 & \text{otherwise} \end{cases}$$

$$p(\perp, a, s) = \begin{cases} 1 & \text{if } s = \perp \\ 0 & \text{otherwise} \end{cases}$$

$$r(s, a, \perp) = \text{min reward}$$