Exercise or relax, for $\gamma = 0.9$

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

exercise	fit	unfit
fit	.99, 8	.01, 8
unfit	.2, 0	.8, 0

relax	fit	unfit
fit	.7, 10	.3, 10
unfit	0, 5	1, 5

Exercise or relax, for $\gamma = 0.9$

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

exercise	fit	unfit
fit	.99, 8	.01, 8
unfit	.2, 0	.8, 0

relax	fit	unfit
fit	.7, 10	.3, 10
unfit	0, 5	1, 5

$$\begin{aligned} q_0(s,a) &:= p(s,a,\text{fit})r(s,a,\text{fit}) + p(s,a,\text{unfit})r(s,a,\text{unfit}) \\ V_n(s) &:= max(q_n(s,\text{exercise}), q_n(s,\text{relax})) \\ q_{n+1}(s,a) &:= q_0(s,a) + .9\big(p(s,a,\text{fit})V_n(\text{fit}) + p(s,a,\text{unfit})V_n(\text{unfit})\big) \end{aligned}$$

Exercise or relax, for $\gamma = 0.9$

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

exercise	fit	unfit
fit	.99, 8	.01, 8
unfit	.2, 0	.8, 0

relax	fit	unfit
fit	.7, 10	.3, 10
unfit	0, 5	1, 5

$$\begin{aligned}
q_0(s,a) &:= p(s,a,\mathrm{fit})r(s,a,\mathrm{fit}) + p(s,a,\mathrm{unfit})r(s,a,\mathrm{unfit}) \\
V_n(s) &:= max(q_n(s,\mathrm{exercise}), q_n(s,\mathrm{relax})) \\
q_{n+1}(s,a) &:= q_0(s,a) + .9\big(p(s,a,\mathrm{fit})V_n(\mathrm{fit}) + p(s,a,\mathrm{unfit})V_n(\mathrm{unfit})\big)
\end{aligned}$$

	exercise	relax	π
fit	8	10	relax
unfit	0	5	relax

Exercise or relax, for $\gamma = 0.9$

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

exercise	fit	unfit
fit	.99, 8	.01, 8
unfit	.2, 0	.8, 0

relax	fit	unfit
fit	.7, 10	.3, 10
unfit	0, 5	1, 5

$$\begin{aligned} q_0(s,a) &:= p(s,a,\text{fit})r(s,a,\text{fit}) + p(s,a,\text{unfit})r(s,a,\text{unfit}) \\ V_n(s) &:= max(q_n(s,\text{exercise}), q_n(s,\text{relax})) \\ q_{n+1}(s,a) &:= q_0(s,a) + .9\big(p(s,a,\text{fit})V_n(\text{fit}) + p(s,a,\text{unfit})V_n(\text{unfit})\big) \end{aligned}$$

	exercise	relax	π
fit	8, 16.955	10, 17.65	relax, relax
unfit	0, 5.4	5, 9.5	relax, relax

Exercise or relax, for $\gamma = 0.9$

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

exercise	fit	unfit
fit	.99, 8	.01, 8
unfit	.2, 0	.8, 0

relax	fit	unfit
fit	.7, 10	.3, 10
unfit	0, 5	1, 5

$$\begin{aligned} q_0(s,a) &:= p(s,a,\text{fit})r(s,a,\text{fit}) + p(s,a,\text{unfit})r(s,a,\text{unfit}) \\ V_n(s) &:= max(q_n(s,\text{exercise}), q_n(s,\text{relax})) \\ q_{n+1}(s,a) &:= q_0(s,a) + .9\big(p(s,a,\text{fit})V_n(\text{fit}) + p(s,a,\text{unfit})V_n(\text{unfit})\big) \end{aligned}$$

	exercise	relax	π
fit	8, 16.955, 23.812	10, 17.65, 23.685	relax, relax, exercise
unfit	0, 5.4, 10.017	5, 9.5, 13.55	relax, relax, relax

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

$$\begin{aligned} A_{k+1} \ &= \ \frac{v_1 + \cdots + v_k + v_{k+1}}{k+1} \\ &= \ \frac{k}{k+1} A_k + \frac{1}{k+1} v_{k+1} \end{aligned}$$

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

$$\begin{aligned} A_{k+1} &= \frac{v_1 + \cdots + v_k + v_{k+1}}{k+1} \\ &= \frac{k}{k+1} A_k + \frac{1}{k+1} v_{k+1} \end{aligned}$$

so if $\alpha_k = \frac{1}{k}$,

$$\begin{aligned} A_{k+1} &= (1 - \alpha_{k+1}) A_k + \alpha_{k+1} \overbrace{v_{k+1}}^{\text{new}} \\ &= \underbrace{A_k}_{\text{old}} + \alpha_{k+1} \underbrace{(v_{k+1} - A_k)}_{\text{temp diff: new}-\text{old}} \end{aligned}$$

Q-Learning

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

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \cdots \underbrace{s_k \xrightarrow{a_k} r_{k+1}, s_{k+1}}_{\text{\textit{experience} from which we learn}} \xrightarrow{a_{k+1}} \cdots$$

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

Q-Learning

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

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \cdots \underbrace{s_k \xrightarrow{a_k} r_{k+1}, s_{k+1}}_{\textit{experience} \text{ from which we learn}} \xrightarrow{a_{k+1}} \cdots$$

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

given $0 \leq \gamma < 1$, $Q_1 : (S \times A) \to \mathbb{R}$ 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

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \cdots \underbrace{s_k \xrightarrow{a_k} r_{k+1}, s_{k+1}}_{\text{\textit{experience} from which we learn}} \xrightarrow{a_{k+1}} \cdots$$

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

given $0 \leq \gamma < 1$, $Q_1 : (S \times A) \to \mathbb{R}$ 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

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

from previous slide (on TD).

Averaging?

$$
\begin{aligned}
v_{k+1} &= r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a) \\
\underbrace{Q_{k+1}(s_k, a_k)}_{A_{k+1}} &= (1-\alpha) \underbrace{Q_k(s_k, a_k)}_{\neq Q_k(s_{k-1}, a_{k-1}) = A_k} + \alpha v_{k+1}
\end{aligned}
$$

Averaging?

$$\begin{aligned} v_{k+1} &= r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a) \\ \underbrace{Q_{k+1}(s_k, a_k)}_{A_{k+1}} &= (1-\alpha) \underbrace{Q_k(s_k, a_k)}_{\neq Q_k(s_{k-1}, a_{k-1}) = A_k} + \alpha v_{k+1} \end{aligned}$$

for a deterministic MDP

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

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)

Averaging?

$$
\begin{aligned}
v_{k+1} &= r_{k+1} + \gamma \max_a Q_k(s_{k+1}, a) \\
\underbrace{Q_{k+1}(s_k, a_k)}_{A_{k+1}} &= (1-\alpha) \underbrace{Q_k(s_k, a_k)}_{\neq Q_k(s_{k-1}, a_{k-1}) = A_k} + \alpha v_{k+1}
\end{aligned}
$$

for a deterministic MDP

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

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

$$\sum \alpha_k = \infty \quad \text{and} \quad \sum \alpha_k^2 < \infty \quad (\text{e.g. } \alpha_k = \frac{1}{k})$$

MDP, one experience at a time

Update $q : (S \times A) \to \mathbb{R}$ via p, r for

$$q'(s,a) \;:=\; \sum_{s'} p(s,a,s')(r(s,a,s') + \gamma \max_{a'} q(s',a'))$$

or pointwise via experience $s_1 \xrightarrow{a_1} r_2, s_2$ for

$$q'(s,a) \;:=\; \begin{cases} \alpha(r_2 + \gamma \max_{a'} q(s_2,a')) \\ \qquad\qquad + (1-\alpha)q(s,a) & \text{if } s = s_1 \text{ and } a = a_1 \\ q(s,a) & \text{otherwise.} \end{cases}$$

MDP, one experience at a time

Update $q : (S \times A) \to \mathbb{R}$ via p, r for

$$q'(s,a) \ := \ \sum_{s'} p(s,a,s')(r(s,a,s') + \gamma \max_{a'} q(s',a'))$$

or pointwise via experience $s_1 \xrightarrow{a_1} r_2, s_2$ for

$$q'(s,a) \ := \ \begin{cases} \alpha(r_2 + \gamma \max_{a'} q(s_2,a')) \\ \qquad\qquad + (1-\alpha)q(s,a) & \text{if } s = s_1 \text{ and } a = a_1 \\ q(s,a) & \text{otherwise.} \end{cases}$$

To converge to MDP's optimal Q-value, visit every state-action pair (s,a) repeatedly (for $s \xrightarrow{a} r', s'$ with diff s', r' under p, r).

MDP, one experience at a time

Update $q : (S \times A) \to \mathbb{R}$ via p, r for

$$q'(s,a) \;:=\; \sum_{s'} p(s,a,s')(r(s,a,s') + \gamma \max_{a'} q(s',a'))$$

or pointwise via experience $s_1 \xrightarrow{a_1} r_2, s_2$ for

$$q'(s,a) \;:=\; \begin{cases} \alpha(r_2 + \gamma \max_{a'} q(s_2,a')) \\ \qquad\qquad + (1-\alpha)q(s,a) & \text{if } s = s_1 \text{ and } a = a_1 \\ q(s,a) & \text{otherwise.} \end{cases}$$

To converge to MDP's optimal Q-value, visit every state-action pair (s,a) repeatedly (for $s \xrightarrow{a} r', s'$ with diff s', r' under p, r).

End **episode**

$$s_1 \xrightarrow{a_1} r_2, s_2 \xrightarrow{a_2} r_3, s_3 \xrightarrow{a_3} \cdots \xrightarrow{a_{n-1}} r_n, s_n$$

at an absorbing state s_n with $r(s_n, a, s_n) = 0$ for every action a.

Exploration-exploitation tradeoff

$$s \xrightarrow{a} r', s' \qquad\qquad r', s' \text{ from environment, but } a?$$

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

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

Exploration-exploitation tradeoff

$$s \xrightarrow{a} r', s' \qquad\qquad r', s' \text{ from environment, but } a?$$

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

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

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. for n actions, m having max $Q(s,\cdot)$

$$\pi_Q^{\epsilon}(s,a) \;=\; \begin{cases} \frac{1-\epsilon}{m} + \frac{\epsilon}{n} & \text{if } Q(s,a) \text{ is max} \quad (\dagger) \\ \frac{\epsilon}{n} & \text{otherwise} \quad (\ddagger) \end{cases}$$

($\dagger$) says exploit: use what we know
($\ddagger$) says explore: try something new (for the future)

Exploration-exploitation tradeoff

$$s \xrightarrow{a} r', s' \xrightarrow{a'} \cdots \qquad r', s' \text{ from environment, but } a?$$

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

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

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. for n actions, m having max $Q(s,\cdot)$

$$\pi_Q^{\epsilon}(s,a) \ = \begin{cases} \frac{1-\epsilon}{m} + \frac{\epsilon}{n} & \text{if } Q(s,a) \text{ is max} \quad (\dagger) \\ \frac{\epsilon}{n} & \text{otherwise} \quad (\ddagger) \end{cases}$$

($\dagger$) says exploit: use what we know
($\ddagger$) says explore: try something new (for the future)

SARSA: replace arg max by policy in use

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