Q-Learning: A Mode-Free Algorithm[1]

[1] Sections 3.1-3.4: Yang&Ying

Q-learning (Watkins '89)

$$V^*(i) = \max_{\mu} E\left[\sum_{k=0}^{\infty} \alpha^k r(X_k, \mu(X_k)) \mid X_0 = i\right]$$

$$\bar{r}(i,u) = E[r(i,u)]$$

Define Q-function (also called action value function):

$$\begin{aligned} Q(i,u) &= E\left[r(i,u) + \alpha V^*(j)\right] \\ &= \bar{r}(i,u) + \alpha \sum_j P_{ij}(u) V^*(j) \end{aligned}$$

Q-learning (Watkins '89)

Define Q-function (also called action value function):

$$\begin{aligned} Q(i,u) =& E\left[r(i,u) + \alpha V^*(j)\right] \\ =& \bar{r}(i,u) + \alpha \sum_j P_{ij}(u) V^*(j) \end{aligned}$$

Given Q, we can find the optimal policy by taking

$$\max_u Q(i,u)$$

(Note: does not require the model $P_{ij}(u)$)

Q-learning (Watkins '89)

Define Q-function (also called action value function):

$$\begin{aligned} Q(i,u) =& E\left[r(i,u) + \alpha V^*(j)\right] \\ =& \bar{r}(i,u) + \alpha \sum_j P_{ij}(u) V^*(j) \end{aligned}$$

Given Q, we can find the optimal policy by taking

$$\max_u Q(i,u)$$

(Note: does not require the model $P_{ij}(u)$)

Q-learning: A learning algorithm to learn the Q-function.

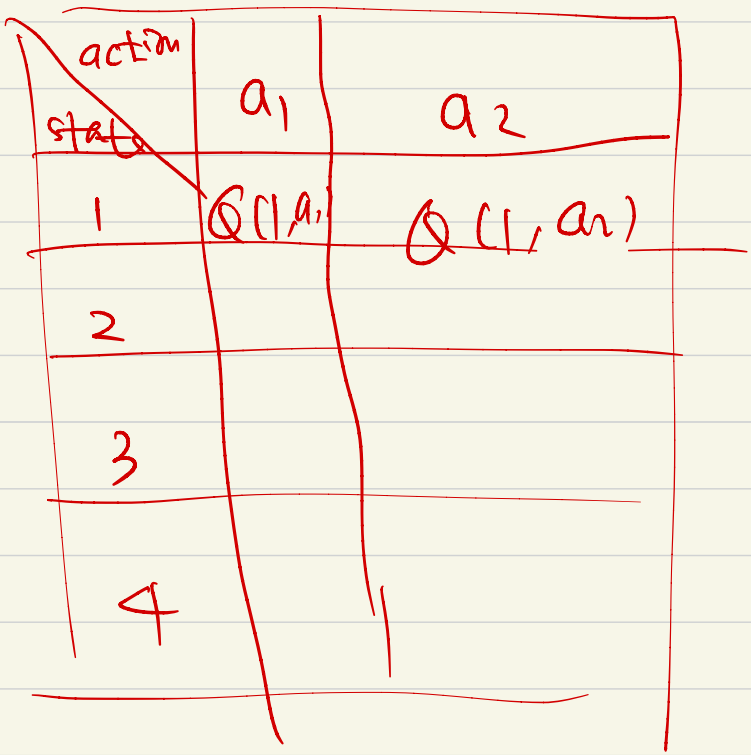
Q-learning (Watkins '89)

Q-learning (off-policy)

Let $\tilde{Q}$ be the estimate of Q. Given an experience (current state $x_k = i$, current action $a_k = u$, received reward $r(i,u)$, and next state $x_{k+1} = j$), Q-learning updates $\tilde{Q}$ as follows:

$$\begin{aligned}\tilde{Q}(i,u) \longleftarrow & (1-\beta_k)\tilde{Q}(i,u) + \beta_k\left(r(i,u) + \alpha \max_v \tilde{Q}(j,v)\right) \\ = & \tilde{Q}(i,u) + \beta_k\left(r(i,u) + \alpha \max_v \tilde{Q}(j,v) - \tilde{Q}(i,u)\right)\end{aligned}$$

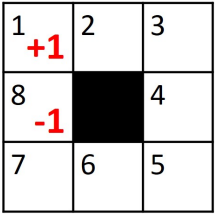
$$Q(i,u) = \bar{r}(i,u) + \alpha \sum_j P_{ij}(u) V^*(j) = \bar{r}(i,u) + \alpha \sum_j P_{ij}(u) \max_v Q(j,v).$$


action
state
a_1
a_2
1
$Q(1, a_1)$
$Q(1, a_2)$
2
3
4

Q-Learning: Grid world example

Assume the current estimate is

(1,c)	(2,c)	(3,c)	(4,c)	(5,c)	(6,c)	(7,c)	(8, c)
1	0.4	0.38	0.14	0.2	-0.8	-0.9	-1.2
(1,cc)	(2,cc)	(3,cc)	(4,cc)	(5,cc)	(6,cc)	(7,cc)	(8,cc)
0.9	0.85	0.78	0.34	0.3	-0.5	-0.34	-1



Q-Learning: Grid world example

Assume the current estimate is

(1,c) 1	(2,c) 0.4	(3,c) 0.38	(4,c) 0.14	(5,c) 0.2	(6,c) -0.8	(7,c) -0.9	(8, c) -1.2
(1,cc) 0.9	(2,cc) 0.85	(3,cc) 0.78	(4,cc) 0.34	(5,cc) 0.3	(6,cc) -0.5	(7,cc) -0.34	(8,cc) -1

Given experience $(x_k = 5, u_k = c, r_k = 0, x_{k+1} = 6)$, Q-learning updates $\tilde{Q}(5, c)$

Q-Learning: Grid world example

Assume the current estimate is

(1,c) 1	(2,c) 0.4	(3,c) 0.38	(4,c) 0.14	(5,c) 0.2	(6,c) -0.8	(7,c) -0.9	(8, c) -1.2
(1,cc) 0.9	(2,cc) 0.85	(3,cc) 0.78	(4,cc) 0.34	(5,cc) 0.3	(6,cc) -0.5	(7,cc) -0.34	(8,cc) -1

Given experience $(x_k = 5, u_k = c, r_k = 0, x_{k+1} = 6)$, Q-learning updates $\tilde{Q}(5, c)$

$$\tilde{Q}(5,c) \longleftarrow 0.2 + \beta_k \left(0 + \alpha \max\{-0.8, -0.5\} - 0.2\right)$$

$$\tilde{Q}(i,u) \longleftarrow \tilde{Q}(i,u) + \beta_k \left(r(i,u) + \alpha \max_v \tilde{Q}(j,v) - \tilde{Q}(i,u)\right)$$

State–action–reward–state–action (SARSA) (Rummery& Niranjan'94)

SARSA (on-policy)

- At step k, with probability $1 - \epsilon_k$, choose action u_k such that

$$u_k \in \arg\max_{v} \tilde{Q}(x_k, v)$$

and with probability ϵ_k, choose an action u_k uniformly at random. Observe x_{k+1} and $r(x_k, u_k)$.

- With data $(x_{k-1}, u_{k-1}, x_k, u_k)$, update Q such that

$$\tilde{Q}(x_{k-1}, u_{k-1}) \longleftarrow (1 - \beta_k)\tilde{Q}(x_{k-1}, u_{k-1}) + \beta_k(r(x_{k-1}, u_{k-1}) + \alpha\tilde{Q}(x_k, u_k))$$

- Choose $\{\epsilon_k\}$ such that $\epsilon_k \to 0$ as $k \to \infty$

SARSA: Grid world example

Assume at time k, the estimate is

(1,c) 1	(2,c) 0.4	(3,c) 0.38	(4,c) 0.14	(5,c) 0.2	(6,c) -0.8	(7,c) -0.9	(8, c) -1.2
(1,cc) 0.9	(2,cc) 0.85	(3,cc) 0.78	(4,cc) 0.34	(5,cc) 0.3	(6,cc) -0.5	(7,cc) -0.34	(8,cc) -1

and assume $x_{k-1} = 4$, $u_{k-1} = cc$, $r_{k-1} = 0$, and $x_k = 5$.

SARSA: Grid world example

Assume at time k, the estimate is

(1,c) 1	(2,c) 0.4	(3,c) 0.38	(4,c) 0.14	(5,c) 0.2	(6,c) -0.8	(7,c) -0.9	(8, c) -1.2
(1,cc) 0.9	(2,cc) 0.85	(3,cc) 0.78	(4,cc) 0.34	(5,cc) 0.3	(6,cc) -0.5	(7,cc) -0.34	(8,cc) -1

and assume $x_{k-1} = 4$, $u_{k-1} = cc$, $r_{k-1} = 0$, and $x_k = 5$.
SARSA chooses action:

$$u_k = \arg\max_{c,cc} \{\tilde{Q}(5, c) = 0.2, \tilde{Q}(5, cc) = 0.3\} = cc.$$

SARSA: Grid world example

Assume at time k, the estimate is

(1,c) 1	(2,c) 0.4	(3,c) 0.38	(4,c) 0.14	(5,c) 0.2	(6,c) -0.8	(7,c) -0.9	(8, c) -1.2
(1,cc) 0.9	(2,cc) 0.85	(3,cc) 0.78	(4,cc) 0.34	(5,cc) 0.3	(6,cc) -0.5	(7,cc) -0.34	(8,cc) -1

and assume $x_{k-1} = 4$, $u_{k-1} = cc$, $r_{k-1} = 0$, and $x_k = 5$. $u_k = cc$

SARSA updates $\tilde{Q}(4, cc)$:

discount

$$\tilde{Q}(4, cc) \longleftarrow 0.34 + \beta_k (0 + \alpha \times 0.3 - 0.34) .$$

$$\tilde{Q}(x_{k-1}, u_{k-1}) \longleftarrow \tilde{Q}(x_{k-1}, u_{k-1}) + \beta_k (r(x_{k-1}, u_{k-1}) + \alpha \tilde{Q}(x_k, u_k) - \tilde{Q}(x_{k-1}, u_{k-1}))$$

Exploration in SARSA

Boltzman Exploration

Choose $\mu_k(x_k) = u$ with probability

$$\frac{\exp\left(\frac{Q_k(x_k,u)}{T}\right)}{\sum_v \exp\left(\frac{Q_k(x_k,v)}{T}\right)} = \frac{1}{1+\sum_{v\neq u}\exp\left(\frac{Q_k(x_k,v)-Q_k(x_k,u)}{T}\right)}$$

Note that as $T \to 0$, the policy chooses u^* such that

$$u^* \in \arg\max_u Q_k(x_k, u)$$

Off-policy vs. on-policy reinforcement learning

Target policy: The policy to be learned

Behavior policy: The policy used to generate samples

- Q-learning: target policy - optimal policy
  behavior policy - any policy under which each action is taken infinitely often
- SARSA: target policy - ϵ-greedy
  behavior policy - ϵ-greedy