Lecture 7: Q-Learning

Course: Reinforcement Learning Theory
Instructor: Lei Ying
Department of EECS
University of Michigan, Ann Arbor

Q-learning (Watkins '89)

- Learning is needed when the system model is unknown.
- Define Q-function:

$$Q(i,u) = \boxed{\bar{c}(i,u)} + \boxed{\sum_j P_{ij}(u) J^*(j)}$$

cost of taking action u at state i

average value after taking action u at state i (assuming all actions after u are optimal)

- 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)

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

Note that $J^*(j) = \max_v Q(j, v)$. Thus,

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

$$J^*(i) = \max_u \bar{r}(i,u) + \alpha \sum_j P_{ij}(u) J^*(j)$$

Further expand,

$$Q(i,u) = \bar{r}(i,u) + \alpha E[\max_v Q(x(t+1), v) | x(t) = i, u(t) = u]$$

$$J^*(i) = \max_u E[r(i,u) + \alpha J^*(x(t+1)) | x(t) = i]$$

Q-learning (Watkins '89)

Define $T(Q)$ such that

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

Claim: $T(Q)$ is a contraction mapping, i.e.

$$\|T(Q_1) - T(Q_2)\|_\infty \leq \alpha \|Q_1 - Q_2\|_\infty$$

Proof:

$$\begin{aligned}(T(Q_1) - T(Q_2))(i,u) &= \alpha \left(\sum_j P_{ij}(u)(\max_v Q_1(j,v) - \max_v Q_2(j,v)) \right) \\ &\leq \alpha \sum_j P_{ij}(u) \max_v |Q_1(j,v) - Q_2(j,v)|\end{aligned}$$

Q-learning (Watkins '89)

Claim

$$\max_v Q_1(j,v) - \max_v Q_2(j,v)) \leq |\max_v (Q_1(j,v) - Q_2(j,v))|$$

Assume (WLOG) $\max_v Q_1(j,v) - \max_w Q_2(j,w) \geq 0$, then

$$\begin{aligned} \max_v Q_1(j,v) - \max_w Q_2(j,w) &= Q_1(j,v^*) - \max_w Q_2(j,w) \\ &\leq Q_1(j,v^*) - Q_2(j,v^*) \\ &\leq \max_v |Q_1(j,v) - Q_2(j,v)| \end{aligned} \qquad \blacksquare$$

Q-learning (Watkins '89)

Claim

$$\max_v Q_1(j,v) - \max_v Q_2(j,v)) \le |\max_v (Q_1(j,v) - Q_2(j,v))|$$

Assume (WLOG) $\max_v Q_1(j,v) - \max_w Q_2(j,w) \ge 0$, then

$$\begin{aligned}\max_v Q_1(j,v) - \max_w Q_2(j,w) &= Q_1(j,v^*) - \max_w Q_2(j,w) \\ &\le Q_1(j,v^*) - Q_2(j,v^*) \\ &\le \max_v |Q_1(j,v) - Q_2(j,v)| \end{aligned} \qquad \blacksquare$$

$$\begin{aligned}(T(Q_1) - T(Q_2))(i,u) &\le \alpha \left(\sum_j P_{ij}(u)\right) \max_{j,v} |Q_1(j,v) - Q_2(j,v)| \\ &= \alpha \|Q_1 - Q_2\|_\infty.\end{aligned}$$

Q-learning (Watkins '89)

Define $T(Q)$ such that

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

<u>Claim:</u> $T(Q)$ is a contraction mapping, i.e.

$$\|T(Q_1) - T(Q_2)\|_\infty \leq \alpha \|Q_1 - Q_2\|_\infty$$

- Thus T is a contraction mapping. Knowing $P_{ij}(u)$ and $r(i,u)$, we can use value iteration to obtain $Q(i,u)$.
- When models are unknown, we use the following ϵ-greedy algorithm, called Q-learning.

Q-learning (Watkins '89)

Q-learning

Let Q_k be the estimate of Q at time step k and let the current state be $x_k = i$, current action $a_t = u$, and next state $x_{k+1} = j$.

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

For any other state $l, (l \neq i), Q_{k+1}(l,a) = Q_k(l,a)$

Assume at state i, each action is taken with probability at least ϵ.

SARSA algorithm

SARSA

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

$$u_k \in \arg\max_v Q_k(x_k, v)$$

  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

$$Q_{k+1}(x_{k-1}, u_{k-1}) =$$
$$(1 - \beta_k)Q_k(x_{k-1}, u_{k-1}) + \beta_k(r(x_{k-1}, u_{k-1}) + \alpha Q_k(x_k, u_k))$$

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

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

Exploration in SARSA

(The convergence of Q-learning and SARSA are deferred to a later lecture)

Example: 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

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

Reference

- This lecture is based on R. Srikant's lecture notes on *Q-Learning* available at `https://sites.google.com/illinois.edu/mdps-and-rl/lectures?authuser=1`

Acknowledgements: I would like to thank Alex Zhao for helping prepare the slides, and Honghao Wei and Zixian Yang for correcting typos/mistakes.