Lecture 5.

model-free → MDP X

policy evaluation(O)

다음 state를 알지 X → greedy policy X

Action-Value : Q-function

Valuefunction → evaluation X

Q-function → evaluation O

↳ state에서 할 수 있는 action 중 Q값이 가장 높은 것을 선택.

$\underbrace{\text{가장 높은 것을 선택}}_{\text{Improvement}}$

greedy하게 만 선택하면 문제가 발생

- ϵ-Greedy Exploration

  ϵ의 확률로 random하게 다른 선택을 해봄.

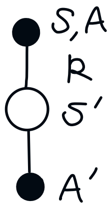
- ϵ-Greedy Improvement

  π'이 π에 대해서 improvement된 policy → $V_{\pi'}(S) \geq V_\pi(S)$

$$q_\pi(s, \pi'(s)) = \sum_{a \in A} \pi'(a|s) q_\pi(s,a)$$

$$= \epsilon/m \sum_{a \in A} q_\pi(s,a) + (1-\epsilon) \max_{a \in A} q_\pi(s,a)$$

- Sarsa



$$Q(S,A) \leftarrow Q(S,A) + \alpha(R + r Q(S',A') - Q(S,A))$$

→ policy evaluation에 사용

↳ 한 step으로 Q를 업데이트 좀 극단적!!

- GLIE : 모든 state-action pair를 방문, 결국 Greedy oplicy로 수렴
- Robbins-Monro :

∘ $\sum_{t=1}^{\infty} \alpha_t = \infty$ ∘ $\sum_{t=1} \alpha_t^2 < \infty$

- n-step sarsa

n=1 (Sarsa) $q_t^{(1)} = R_{t+1} + \gamma Q(S_{t+1})$

n=2 $q_t^{(2)} = R_{t+1} + R_{t+2} + \gamma Q(S_{t+2})$

⋮

n=∞ (MC) $q_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{T-1} R_T$

- Define the *n*-step Q-return

$$q_t^{(n)} = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{n-1} R_{t+n} + \gamma^n Q(S_{t+n})$$

- *n*-step Sarsa updates $Q(s, a)$ towards the *n*-step Q-return

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha \left(q_t^{(n)} - Q(S_t, A_t) \right)$$

- 1-step sarsa

- Sarsa (λ): 지난 모든 경로를 update 함

- Importance Sampling

$$E_{X\sim P}[f(X)] = \sum P(X)f(X)$$

$$= \sum Q(X)\frac{P(X)}{Q(X)}f(X)$$

$$= E_{X\sim Q}\left[\frac{P(X)}{Q(X)}f(X)\right]$$

- Important Sampling for Off-Policy Monte-Carlo

  - Use returns generated from μ to evaluate π
  - Weight return G_t according to similarity between policies
  - Multiply importance sampling corrections along whole episode

$$G_t^{\pi/\mu} = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}\frac{\pi(A_{t+1}|S_{t+1})}{\mu(A_{t+1}|S_{t+1})}\cdots\frac{\pi(A_T|S_T)}{\mu(A_T|S_T)}G_t$$

  - Update value towards *corrected* return

$$V(S_t) \leftarrow V(S_t) + \alpha\left(G_t^{\pi/\mu} - V(S_t)\right)$$

  - Cannot use if μ is zero when π is non-zero
  - Importance sampling can dramatically increase variance

- Important Sampling for Off-Policy TD

  - Use TD targets generated from μ to evaluate π
  - Weight TD target $R + \gamma V(S')$ by importance sampling
  - Only need a single importance sampling correction

$$V(S_t) \leftarrow V(S_t) + \alpha\left(\frac{\pi(A_t|S_t)}{\mu(A_t|S_t)}(R_{t+1} + \gamma V(S_{t+1})) - V(S_t)\right)$$

  - Much lower variance than Monte-Carlo importance sampling
  - Policies only need to be similar over a single step

- Q-Learning

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(\underbrace{R_{t+1} + \gamma Q(S_{t+1}, A')}_{\text{추측치}} - Q(S_t, A_t))$$

- Off-policy Control with Q-Learning

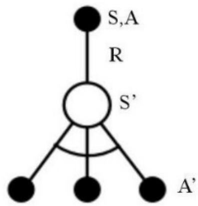
  - We now allow both behaviour and target policies to improve
  - The target policy π is greedy w.r.t. $Q(s, a)$

$$\pi(S_{t+1}) = \underset{a'}{\operatorname{argmax}}\ Q(S_{t+1}, a')$$

  - The behaviour policy μ is e.g. ϵ-greedy w.r.t. $Q(s, a)$
  - The Q-learning target then simplifies:

$$\begin{aligned}
&R_{t+1} + \gamma Q(S_{t+1}, A') \\
=&R_{t+1} + \gamma Q(S_{t+1}, \underset{a'}{\operatorname{argmax}}\ Q(S_{t+1}, a')) \\
=&R_{t+1} + \max_{a'} \gamma Q(S_{t+1}, a')
\end{aligned}$$

- Q-Learning Control Algorithm

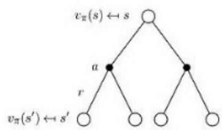
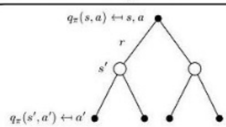
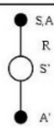
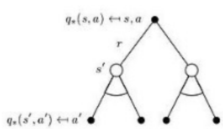
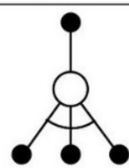


$$Q(S,A) \leftarrow Q(S,A) + \alpha \left(R + \gamma \max_{a'} Q(S',a') - Q(S,A) \right)$$

Theorem

Q-learning control converges to the optimal action-value function, $Q(s,a) \rightarrow q_*(s,a)$

Relationship Between DP and TD

	Full Backup (DP)	*Sample Backup (TD)*
Bellman Expectation Equation for $v_\pi(s)$	 Iterative Policy Evaluation	 TD Learning
Bellman Expectation Equation for $q_\pi(s,a)$	 Q-Policy Iteration	 Sarsa
Bellman Optimality Equation for $q_*(s,a)$	 Q-Value Iteration	 Q-Learning