* Model Free Prediction

- Law of Large Number
- Incremental mean
- Monte-Carlo
- Temporal-Difference

* Model Free Prediction

DP에서는 model의 $R_s^a, P_{ss'}^a$를 알았다.

→ $V_\pi(s) = R_s^a + \gamma \cdot \sum_{s'} P_{ss'}^a \cdot V_\pi(s')$ ⟶ $V_{k+1}(s) = R_s^a + \gamma \cdot \sum_{s'} P_{ss'}^a \cdot V_k(s')$ ↙ Policy eval

→ $\pi'(s) = \underset{a}{\text{argmax}}\ q_\pi(s,a)$ ↙ Policy improvement (Greedy Policy)

But generally $R_s^a, P_{ss'}^a$? ... 몰라!

⟶ MC / TD를 써보자 ~

* Law of Large Numbers

= 큰 수의 법칙

R.V X의 실제 mean ⟶ $E[X] = \mu$ ...

$X_1, X_2, X_3, \cdots$가 iid인 X의 반복적인 측정결과 일 때

Also R.V

- Sample mean : $M_n = \frac{1}{n} \sum_{i=1}^{n} X_i$
- Strong Law of Large Numbers : $P[\lim_{n \to \infty} M_n = \mu] = 1$

→ 어떤 측정치들의 Sample mean이든 결국 100% $E[X] = \mu$로 수렴!

* Incremental mean

$\Rightarrow \mu_k = \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$

→ 기존에 저장해 둔 평균값과 새 측정값으로 평균값 갱신!

Coding에 유리!

[PF] $\mu_k = \frac{1}{k} \sum_{i=1}^{k} x_i$

$= \frac{1}{k}\left(x_k + \sum_{i=1}^{k-1} x_i\right) = \frac{1}{k}\left(x_k + (k-1)\mu_{k-1}\right)$

$\therefore \mu_k = \mu_{k-1} + \frac{1}{k}(x_k - \mu_{k-1})$

*Remind

$G_t = R_{t+1} + \gamma R_{t+2} + \cdots \gamma^{k-1} \cdot R_{t+k}$

$V_\pi(s) = E_\pi[G_t \mid S_t = s]$

* Monte-Carlo

→ Terminated!

- Episode를 통한 Policy Evaluation

→ Given π에 따른 episode들을 만들고, 각 episode들을 모두 이용해서 경험적 데이터로 E 구하기!

→ Expected return 대신 Empirical mean return! (Law of Large Numbers)

→ $E[X] = \sum_k x_k P_X(x_k)$

→ $E[X] = \frac{1}{k}\sum_{i=1}^{k} x_i$

- Monte-Carlo Policy Evaluation

Goal : learn V_π from episodes

$S_1\ A_1\ R_2 \rightarrow S_2\ A_2\ R_3 \rightarrow \cdots$

① First Visit Monte-Carlo Policy Evaluation

→ 각 Episode에서 내가 관심있는 State 'S'가 나오면,
1 시점을 기준으로 return을 모두 더해서 mean!

→ 1개의 episode에 'S'가 여러번 등장? → 첫 등장 시점에서의 G_t만!

② Every Visit Monte-Carlo Policy Evaluation

→ 각 Episode에서 내가 관심있는 State 'S'가 나오면,
1 시점을 기준으로 return을 모두 더해서 mean!

→ 1개의 episode에 'S'가 여러번 등장? → 각 시점에서의 G_t 모두 Sum!

i) Check episode 1 by 1

ii) if State 's' appeared ⋯ $N(s_t) \leftarrow N(s_t) + 1$

$S(s_t) \leftarrow S(s_t) + G_t$

iii) Through All episode!

iv) $V(s_t) = \frac{S(s_t)}{N(s_t)} \xrightarrow{\text{With LLN}} V(s_t) = v_\pi(s_t)$ as $N(s_t) \rightarrow \infty$

⊕ Incremental MC Updates

1개의 Episode로 얻은 V(s)를 Update

$N(s_t) \leftarrow N(s_t) + 1$

$V(s_t) \leftarrow V(s_t) + \frac{1}{N(s_t)}(G_t - V(s_t))$ $\left(\mu_k = \mu_{k-1} + \frac{1}{k}(x_k + \mu_{k-1})\right)$

⋮

$V(s_t) \leftarrow V(s_t) + \alpha\,(G_t - V(s_t))$

↳ 경험값을 쓰기도 한다.

* Temporal-Difference

→ Not Terminated!

- Episode를 통한 Policy Evaluation

↳ Incomplete Episode를 통한 V_π 찾아가기:

↳ By Bootstrapping

↳ 확실히 모르는 info로도 원하는 값을 추정!

- A Simple TD Algorithm (TD(0))

Goal : Learn V_π online through experience in episode

$V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma \cdot V(S_{t+1}) - V(S_t))$

$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$ → to $V(S_{t+1})$

$V(S_{t+1})$ → 추정값

TD Target : $R_{t+1} + \gamma V(S_{t+1})$

TD Error : $R_{t+1} + \gamma V(S_{t+1}) - V(S_t) = \delta_t$