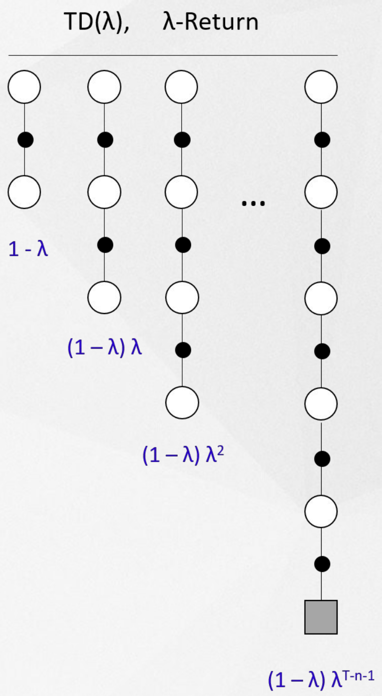
Chapter 06. 스스로 전략을 짜는 강화학습 (Reinforcement Learning)

Temporal Difference



Monte Carlo Method

Monte Carlo ES (Exploring Starts), for estimating $\pi \approx \pi_*$

Initialize:
$\quad \pi(s) \in \mathcal{A}(s)$ (arbitrarily), for all $s \in \mathcal{S}$
$\quad Q(s,a) \in \mathbb{R}$ (arbitrarily), for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$
$\quad Returns(s,a) \leftarrow$ empty list, for all $s \in \mathcal{S}$, $a \in \mathcal{A}(s)$

Loop forever (for each episode):
$\quad$ Choose $S_0 \in \mathcal{S}$, $A_0 \in \mathcal{A}(S_0)$ randomly such that all pairs have probability > 0
$\quad$ Generate an episode from S_0, A_0, following π: $S_0, A_0, R_1, \ldots, S_{T-1}, A_{T-1}, R_T$
$\quad G \leftarrow 0$
$\quad$ Loop for each step of episode, $t = T-1, T-2, \ldots, 0$:
$\quad\quad G \leftarrow \gamma G + R_{t+1}$
$\quad\quad$ Unless the pair S_t, A_t appears in $S_0, A_0, S_1, A_1 \ldots, S_{t-1}, A_{t-1}$:
$\quad\quad\quad$ Append G to $Returns(S_t, A_t)$
$\quad\quad\quad Q(S_t, A_t) \leftarrow \text{average}(Returns(S_t, A_t))$
$\quad\quad\quad \pi(S_t) \leftarrow \text{argmax}_a Q(S_t, a)$

Temporal Difference

https://sumniya.tistory.com/14?category=781573

에피소드 마다 가 아니라 매 타임스텝 마다 가치함수를 업데이트

MC : $V(S_t) \leftarrow V(S_t) + \alpha(G_t - V(S_t))$

TD(0) : $V(S_t) \leftarrow V(S_t) + \alpha(R_{t+1} + \gamma V(S_{t+1}) - V(S_t))$

Input : the policy π to be evaluated
Initialize $V(s)$ arbitrarily (e.g., $V(s) = 0, \forall s \in S^+$)
Repeat (for each episode):
    Initialize S
    Repeat (for each step of episode):
        $A \leftarrow$ action given by π for S
        Take action A; observe reward, R, and next state, S'
        $V(S) \leftarrow V(S) + \alpha[R + \gamma V(S') - V(S)]$
        $S \leftarrow S'$
    until S is terminal

Bellman Equation

The value function can be decomposed into two parts:

- immediate reward R_{t+1}
- discounted value of successor state $\gamma v(S_{t+1})$

$$
\begin{aligned}
v(s) &= \mathbb{E}\left[G_t \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \dots \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma\left(R_{t+2} + \gamma R_{t+3} + \dots\right) \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma G_{t+1} \mid S_t = s\right] \\
&= \mathbb{E}\left[R_{t+1} + \gamma v(S_{t+1}) \mid S_t = s\right]
\end{aligned}
$$

Temporal Difference

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

- Return $G_t = R_{t+1} + \gamma R_{t+2} + ... + \gamma^{T-1} R_T$ is *unbiased* estimate of $v_\pi(S_t)$
- True TD target $R_{t+1} + \gamma v_\pi(S_{t+1})$ is *unbiased* estimate of $v_\pi(S_t)$
- TD target $R_{t+1} + \gamma V(S_{t+1})$ is *biased* estimate of $v_\pi(S_t)$
- TD target is much lower variance than the return:
  - Return depends on *many* random actions, transitions, rewards
  - TD target depends on *one* random action, transition, reward

•MC = unbiased, high variance
•TD = biased, small variance

TD vs MC

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

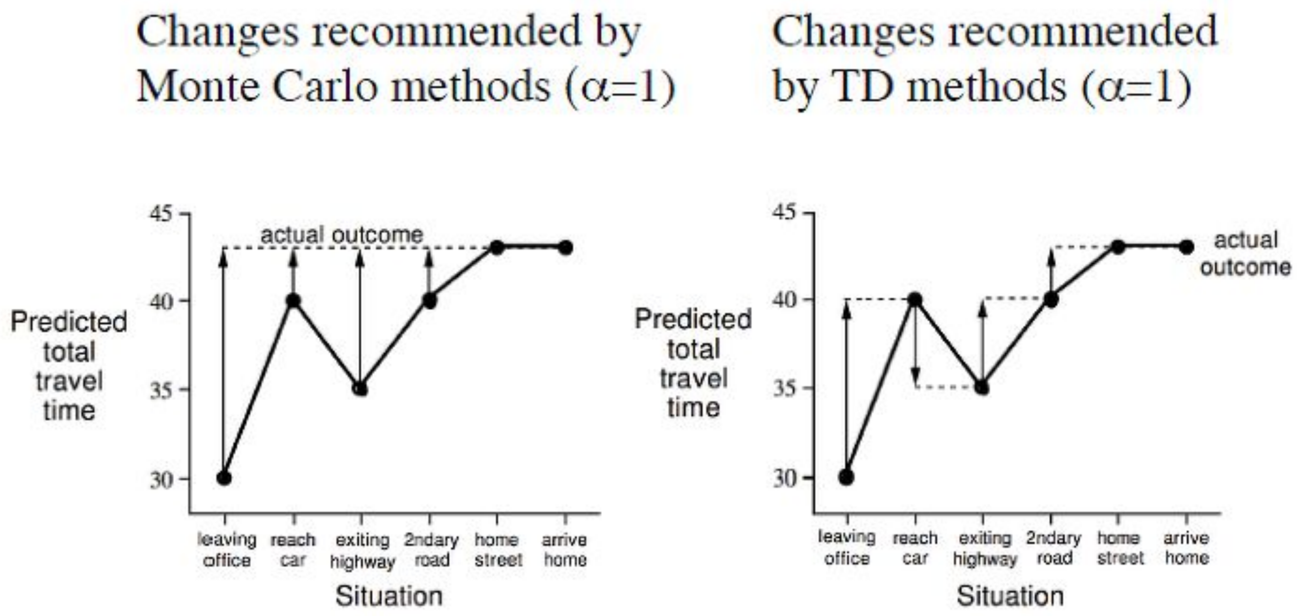
Driving Home Example

State	Elapsed Time (minutes)	Predicted Time to Go	Predicted Total Time
leaving office	0	30	30
reach car, raining	5	35	40
exit highway	20	15	35
behind truck	30	10	40
home street	40	3	43
arrive home	43	0	43

TD vs MC

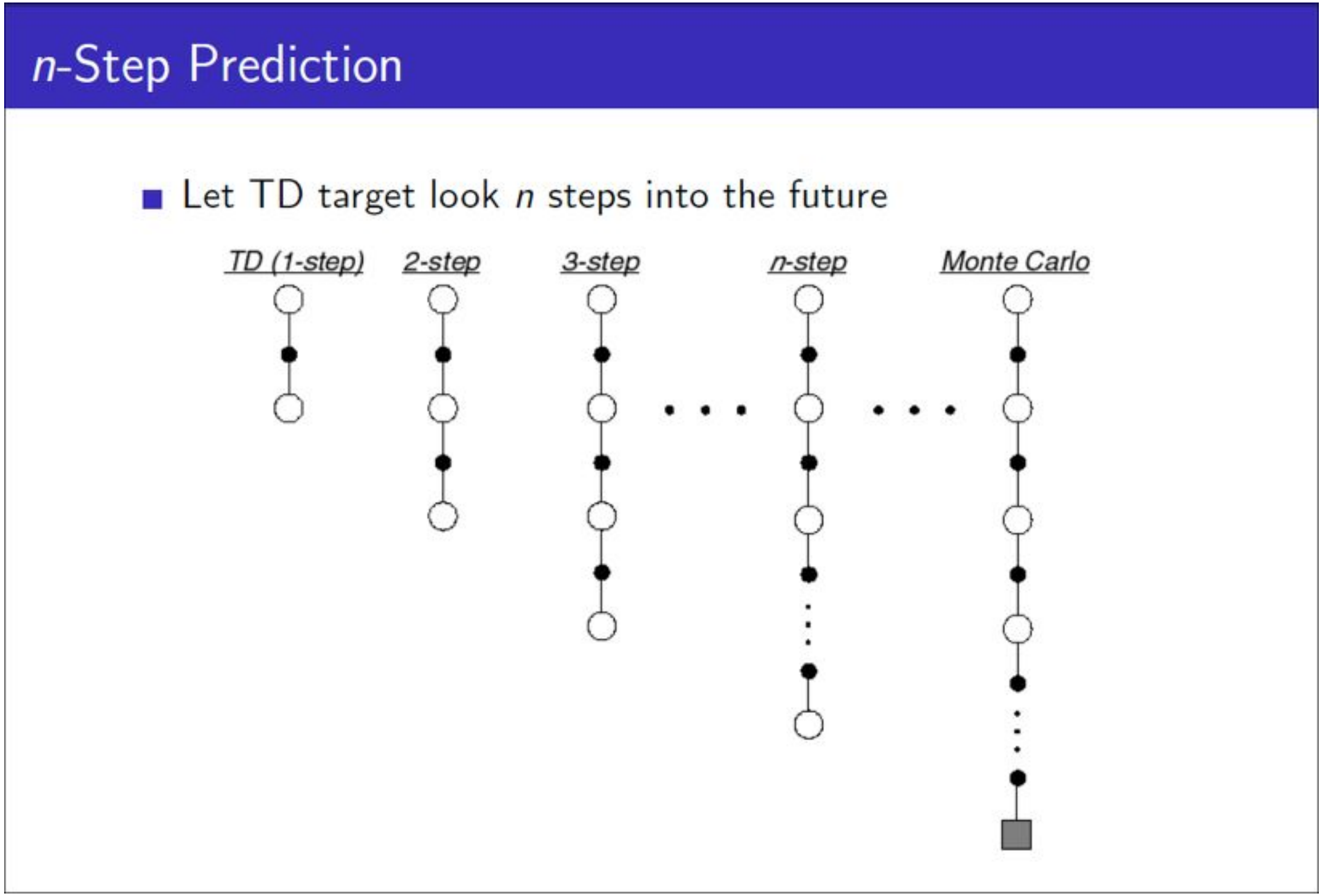
http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

Driving Home Example: MC vs. TD



N- Step TD

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html



N- Step TD

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

n-Step Return

- Consider the following n-step returns for $n = 1, 2, \infty$:

$$\begin{array}{lll} n = 1 & (TD) & G_t^{(1)} = R_{t+1} + \gamma V(S_{t+1}) \\ n = 2 & & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2}) \\ \vdots & & \vdots \\ n = \infty & (MC) & G_t^{(\infty)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{T-1} R_T \end{array}$$

- Define the n-step return

$$G_t^{(n)} = R_{t+1} + \gamma R_{t+2} + \ldots + \gamma^{n-1} R_{t+n} + \gamma^n V(S_{t+n})$$

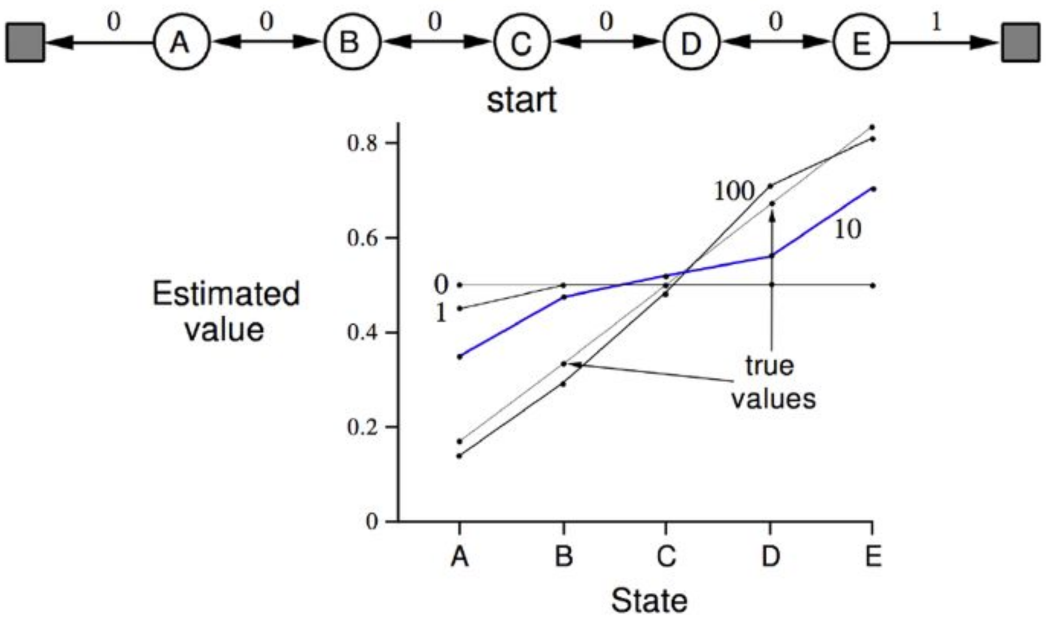
- n-step temporal-difference learning

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

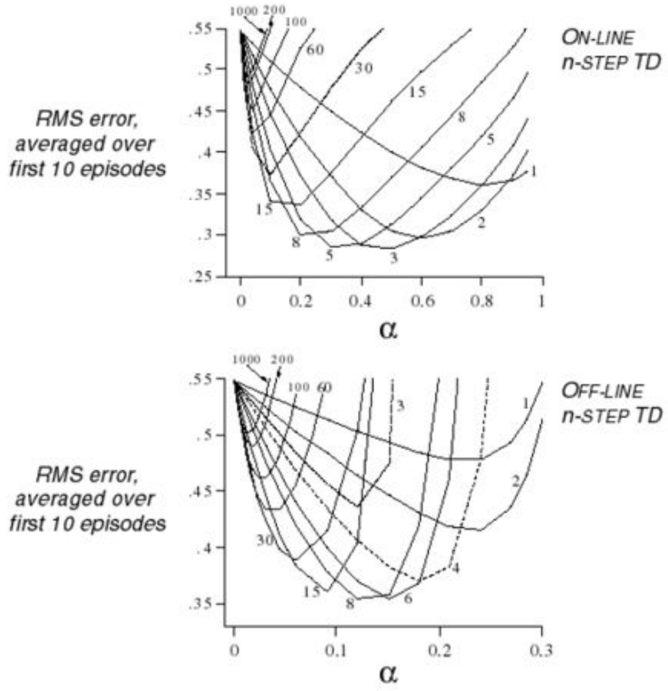
TD(λ)

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

Random Walk Example



Large Random Walk Example



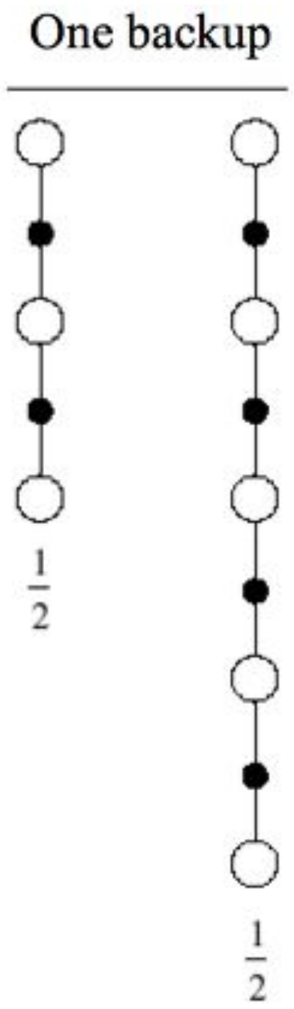
Forward view TD(λ)

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

- We can average n-step returns over different n
- e.g. average the 2-step and 4-step returns

$$\frac{1}{2}G^{(2)} + \frac{1}{2}G^{(4)}$$

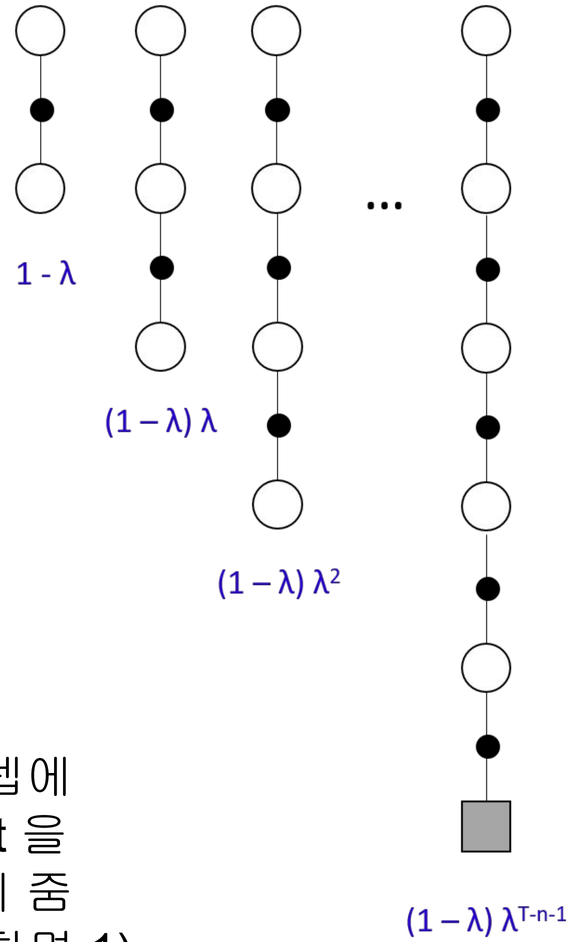
- Combines information from two different time-steps
- Can we efficiently combine information from all time-steps?



Forward view TD(λ)

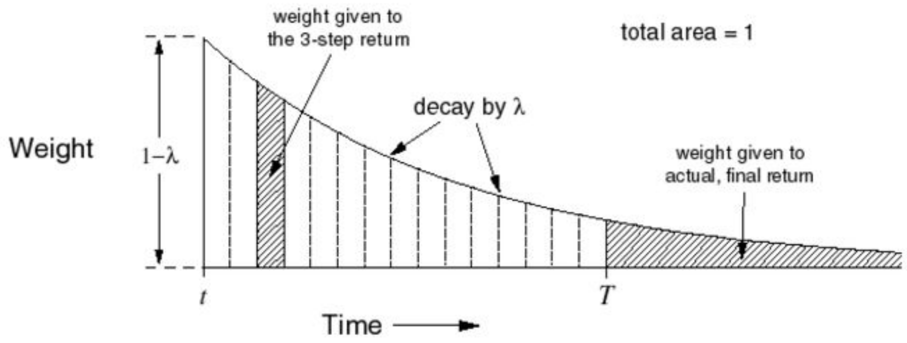
http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

TD(λ),   λ-Return



$$V(S_t) \leftarrow V(S_t) + \alpha(G_t^\lambda - V(S_t))$$

$$when\ \ \lambda\text{-}return,\ \ G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

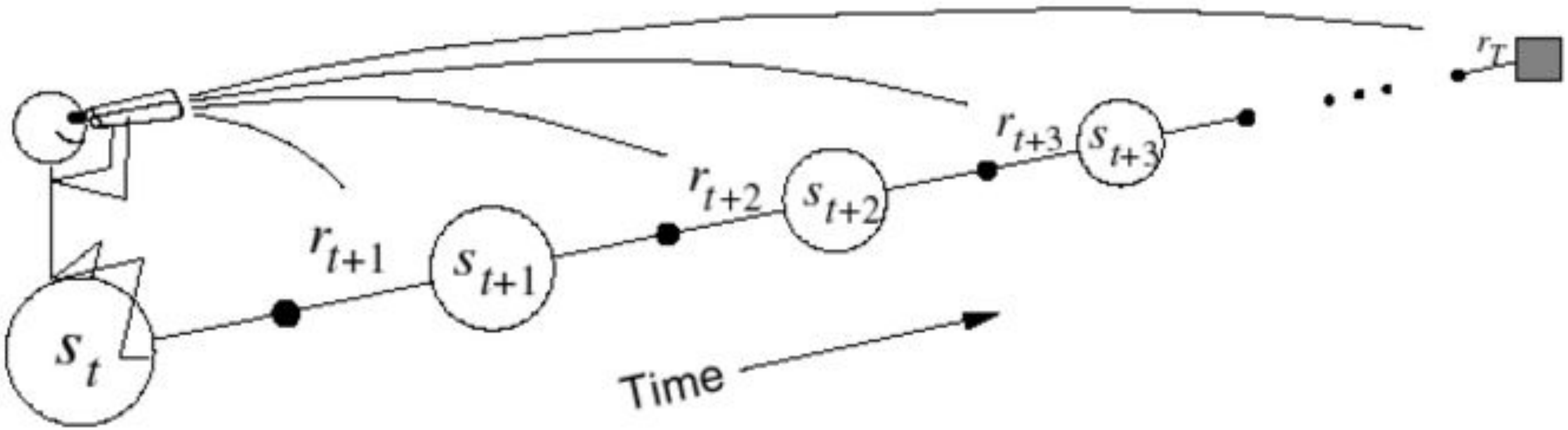


$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1} G_t^{(n)}$$

각 스텝에
weight 을
다르게 줌
(다더하면 1)

Forward view TD(λ)

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html



- Update value function towards the λ-return
- Forward-view looks into the future to compute G_t^λ
- Like MC, can only be computed from complete episodes

Backward view TD (λ)

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

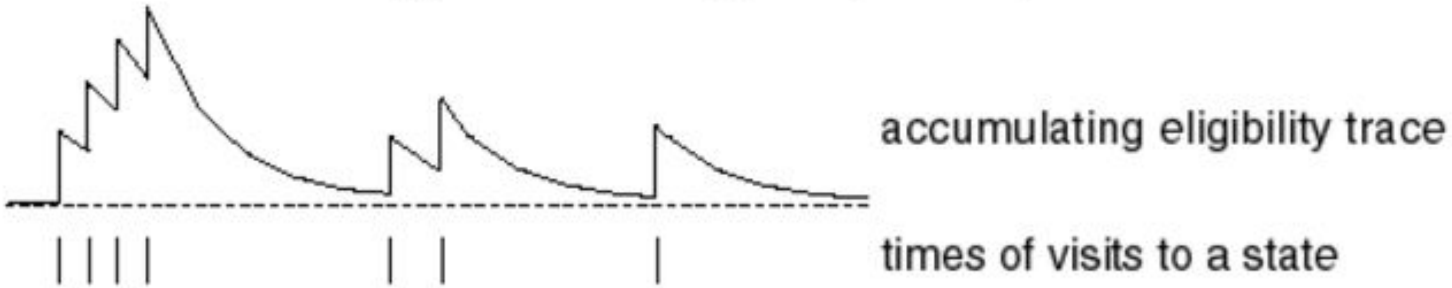
- Forward view provides theory
- Backward view provides mechanism
- Update online, every step, from incomplete sequences

Backward view TD (λ)

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

- Credit assignment problem: did bell or light cause shock?
- Frequency heuristic: assign credit to most frequent states
- Recency heuristic: assign credit to most recent states
- *Eligibility traces* combine both heuristics

$$E_0(s) = 0$$
$$E_t(s) = \gamma \lambda E_{t-1}(s) + \mathbf{1}(S_t = s)$$

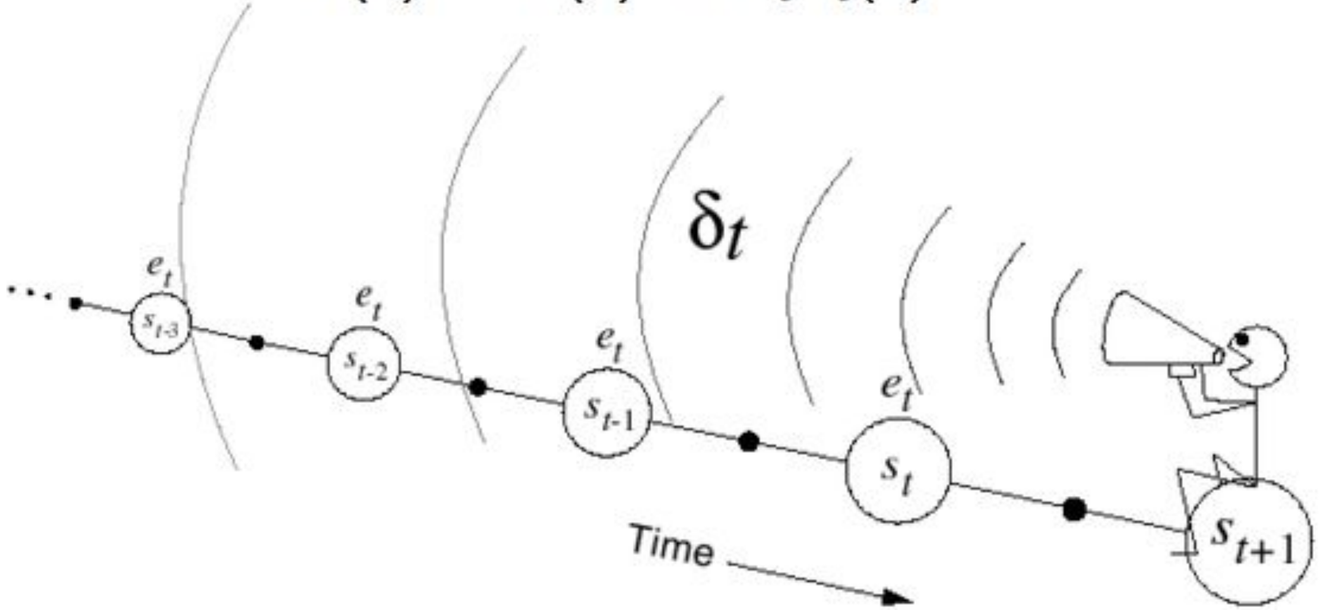


Backward view TD (λ)

- Keep an eligibility trace for every state s
- Update value $V(s)$ for every state s
- In proportion to TD-error δ_t and eligibility trace $E_t(s)$

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

$$V(s) \leftarrow V(s) + \alpha \delta_t E_t(s)$$



Backward view TD (λ)

http://www0.cs.ucl.ac.uk/staff/d.silver/web/Home.html

- When $\lambda = 1$, credit is deferred until end of episode
- Consider episodic environments with offline updates
- Over the course of an episode, total update for TD(1) is the same as total update for MC

Theorem

The sum of offline updates is identical for forward-view and backward-view TD(λ)

$$\sum_{t=1}^{T} \alpha \delta_t E_t(s) = \sum_{t=1}^{T} \alpha \left(G_t^{\lambda} - V(S_t) \right) \mathbf{1}(S_t = s)$$

fast campus

For/Backward view TD (λ)

https://sumniya.tistory.com/14?category=781573

forward-view TD(λ)는 MC와 TD의 장점을 모두 취하기 위해 n-step을 쓰려했으나, n에 따라 각기 다른 장점이 있기에 MC의 update식에 있는 target(return)을 λ-return으로 사용하여 각 n-step의 장점을 모두 취하는 것에 의의를 두었다면, backward-view TD(λ)는 TD의 update식에서 eligibility trace라는 방법을 이용해서 새롭게 weight을 주는 것에 의의를 두었다고 생각할 수 있습니다. 개인적으로는 forward-view TD(λ)는 MC의 high variance 문제를 episode의 수를 끝날 때까지가 아닌 특정 n으로 줄여서 해결하려했더니 α와 n에 따라 optimal한 정도가 다르기 때문에 이를 아우르기 위한 방법, 그리고 backward-view TD(λ)는 TD의 high bias문제를 그 동안 지나왔던 state에 heuristic으로 기준을 주어 바로 다음 step뿐만 아니라 이 전의 event도 영향을 주게끔 해결하려는 시도로 이해했습니다.

- *Thank you*