Lecture 3： Model-free Prediction

8th 15th Mar. 2022

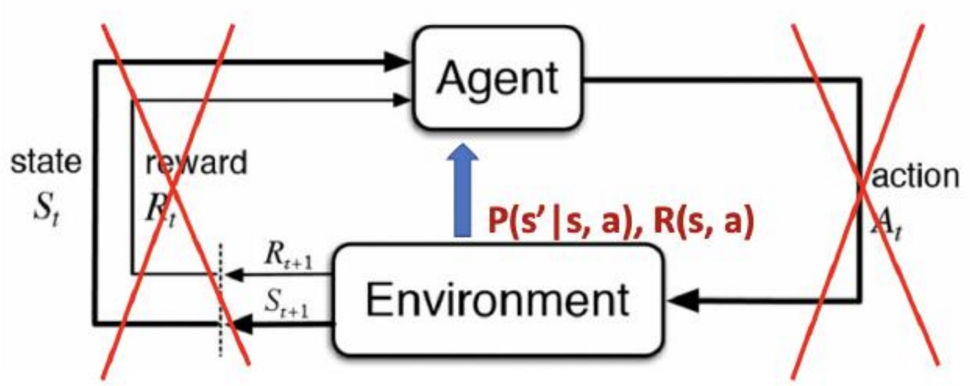
Recap

- Last lecture:
  - MDP
  - policy evaluation
  - policy iteration and value iteration for solving a known MDP

- The following two lectures:
  - Model-free prediction: Estimate value function of an unknown MDP
  - Model-free control: Optimize value function of an unknown MDP

RL with knowing how the world works

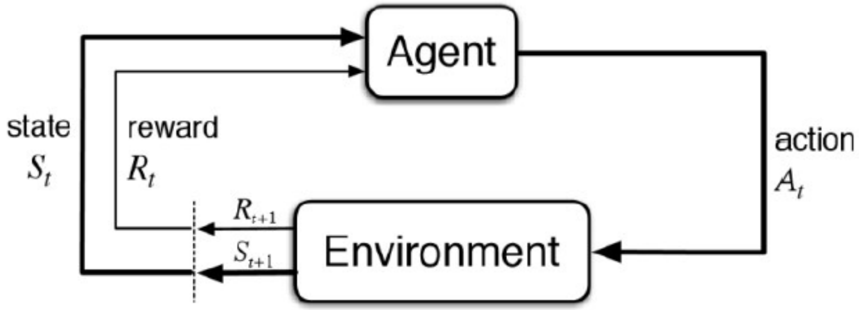
- Both of the policy iteration and value iteration assume the direct access to the dynamics and rewards of the environment



- In a lot of real-world problems, MDP model is either unknown or known by too big or too complex to use
  - Atari Game, Game of Go, Helicopter, Portfolio management, etc

Model-free RL: Learning by interaction

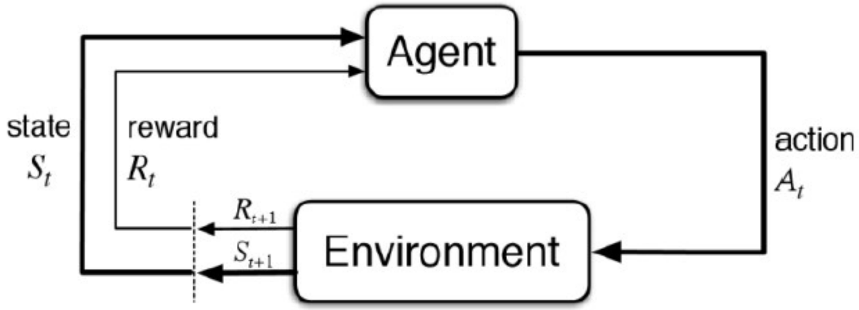
- Model-free RL can solve the problems through interaction with the environment



- No more direct access to the known transition dynamics and reward function
- Trajectories/episodes are collected by the agent's interaction with the environment
- Each trajectory/episode contains $\{S_1, A_1, R_1, S_2, A_2, R_2, ..., S_T, A_T, R_T\}$

Model-free RL: Learning by interaction

- Model-free RL can solve the problems through interaction with the environment



- No more direct access to the known transition dynamics and reward function
- Trajectories/episodes are collected by the agent's interaction with the environment
- Each trajectory/episode contains $\{S_1, A_1, R_1, S_2, A_2, R_2, ..., S_T, A_T, R_T\}$

Model-free prediction

- Model-free prediction: policy evaluation without the access to the model
- Estimating the expected return of a particular policy if we don't have access to the MDP models
  - Monte Carlo policy evaluation
  - Temporal Difference (TD) learning

Monte-Carlo Policy Evaluation

- Return: $G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ...$
- $v^\pi(s) = \mathbb{E}_{\tau \sim \pi}[G_t | s_t = s]$ thus expectation over trajectories τ generated by following π
- MC simulation: we can simply sample a lot of trajectories, compute the actual returns for all the trajectories, then average them
- MC policy evaluation uses empirical mean return instead of expected return
- MC does not require MDP dynamics/rewards, no bootstrapping, and does not assume state is Markov.
- Only applied to episodic MDPs (each episode terminates)

Monte-Carlo Policy Evaluation

- To evaluate state $v(s)$

1. Every time-step t that state s is visited in an episode,
2. Increment counter $N(s) \leftarrow N(s) + 1$
3. Increment total return $S(s) \leftarrow S(s) + G_t$
4. Value is estimated by mean return $v(s) = S(s)/N(s)$

- By law of large numbers, $v(s) \rightarrow v^{\pi}(s)$ as $N(s) \rightarrow \infty$

Incremental MC Updates

- Mean from the average of samples $x_1, x_2, \ldots$

$$\begin{aligned}\mu_t &= \frac{1}{t}\sum_{j=1}^{t} x_j \\ &= \frac{1}{t}\left(x_t + \sum_{j=1}^{t-1} x_j\right) \\ &= \frac{1}{t}\left(x_t + (t-1)\mu_{t-1}\right) \\ &= \mu_{t-1} + \frac{1}{t}\left(x_t - \mu_{t-1}\right)\end{aligned}$$

- Collect one episode $(S_1, A_1, R_1, \ldots, S_t)$

- For each state s_t with computed return G_t

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

$$v(S_t) \leftarrow v(S_t) + \frac{1}{N(S_t)}\left(G_t - v(S_t)\right)$$

- Or use a running mean (old episodes are forgotten). Good for non-stationary problems.

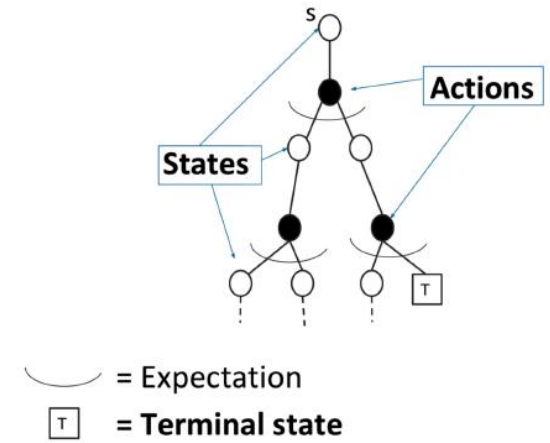
$$v(S_t) \leftarrow v(S_t) + \alpha\left(G_t - v(S_t)\right)$$

Difference between DP and MC

- Dynamic Programming (DP) computes v_i by bootstrapping the rest of the expected return by the value estimate v_{i-1}

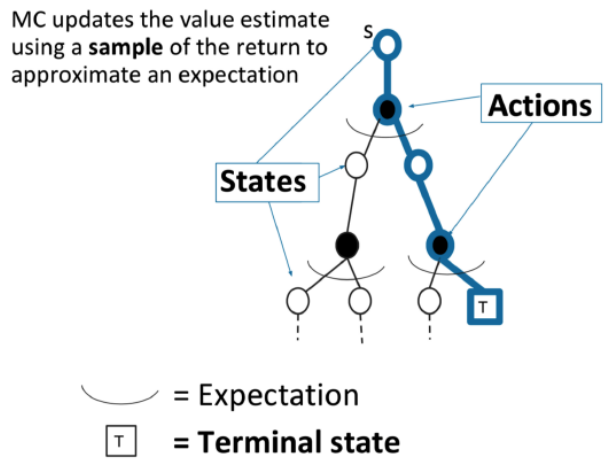
- Iteration on Bellman expectation backup:

$$v_i(s) \leftarrow \sum_{a \in \mathcal{A}} \pi(a|s)\Big(R(s,a) + \gamma \sum_{s' \in \mathcal{S}} P(s'|s,a) v_{i-1}(s')\Big)$$



- MC updates the empirical mean return with one sampled episode

$$v(S_t) \leftarrow v(S_t) + \alpha(G_{i,t} - v(S_t))$$



Advantages of MC over DP

- MC works when the environment is unknown
- Working with sample episodes has a huge advantage, even when one has complete knowledge of the environment's dynamics, for example, transition probability is complex to compute
- Cost of estimating a single state's value is independent of the total number of states. So you can sample episodes starting from the states of interest then average returns

Temporal-Difference (TD) Learning

- TD methods learn directly from episodes of experience
- TD is model-free: no knowledge of MDP transitions/rewards
- TD learns from incomplete episodes, by bootstrapping

- Objective: learn v_π online from experience under policy π
- Simplest TD algorithm: TD(0)

1. Update $v(S_t)$ toward estimated return $R_{t+1} + \gamma v(S_{t+1})$

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

$$\delta_t = R_{t+1} + \gamma v(S_{t+1}) - v(S_t)$$ ⇒ TD error

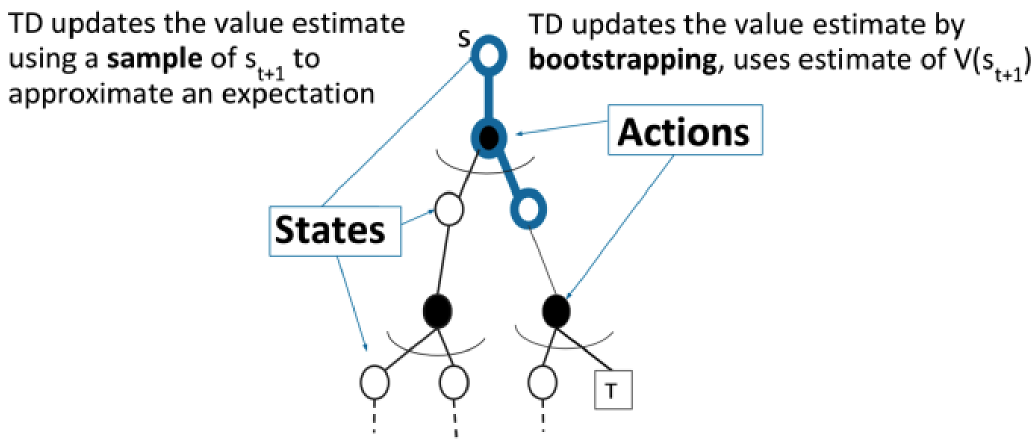
TD target

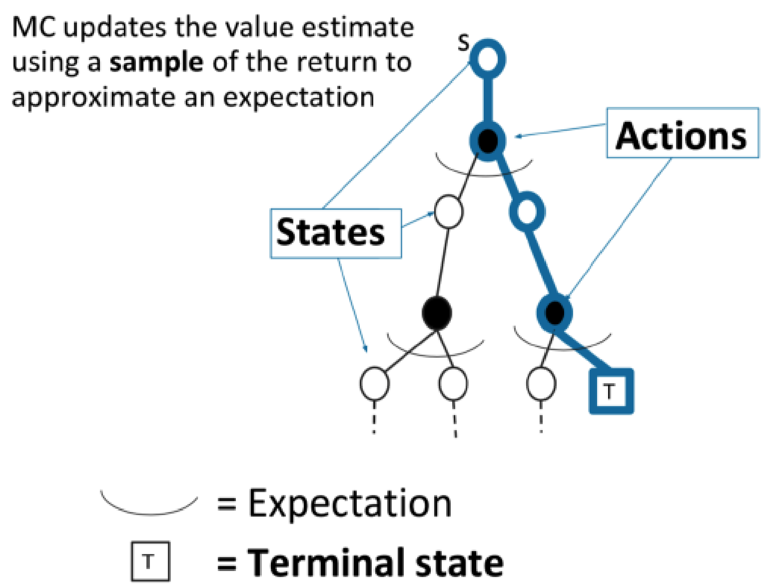
- Comparison: Incremental Monte-Carlo

1. Update $v(S_t)$ toward actual return G_t given an episode i

$$v(S_t) \leftarrow v(S_t) + \alpha(G_{i,t} - v(S_t))$$

Advantages of TD over MC





Comparison of TD and MC

- TD can learn online after every step
- MC must wait until end of episode before return is known

- TD can learn from incomplete sequences
- MC can only learn from complete sequences

- TD works in continuing (non-terminating) environments
- MC only works for episodic (terminating) environments

- TD exploits Markov property, more efficient in Markov environments
- MC does not exploit Markov property, more effective in non-Markov environments

Bias/Variance Trade-Off

- 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

Comparison of TD and MC

- MC has high variance, zero bias
  - Good convergence properties
  - (even with function approximation)
  - Not very sensitive to initial value
  - Very simple to understand and use

- TD has low variance, some bias
  - Usually more efficient than MC
  - TD(0) converges to $V_{\pi}(s)$
  - (but not always with function approximation)
  - More sensitive to initial value

Batch MC and TD

- MC and TD converge: $V(s) \to v_\pi(s)$ as experience $\to \infty$
- But what about batch solution for finite experience?

$$s_1^1, a_1^1, r_2^1, \ldots, s_{T_1}^1$$
$$\vdots$$
$$s_1^K, a_1^K, r_2^K, \ldots, s_{T_K}^K$$

  - e.g. Repeatedly sample episode $k \in [1, K]$
  - Apply MC or TD(0) to episode k

Certainty Equivalence

- MC converges to solution with minimum mean-squared error
  - Best fit to the observed returns

$$\sum_{k=1}^{K} \sum_{t=1}^{T_k} \left(G_t^k - V(s_t^k) \right)^2$$

- TD(0) converges to solution of max likelihood Markov model
  - Solution to the MDP $\langle \mathcal{S}, \mathcal{A}, \hat{\mathcal{P}}, \hat{\mathcal{R}}, \gamma \rangle$ that best fits the data

$$\hat{\mathcal{P}}_{s,s'}^a = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_k} \mathbf{1}(s_t^k, a_t^k, s_{t+1}^k = s, a, s')$$

$$\hat{\mathcal{R}}_s^a = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{t=1}^{T_k} \mathbf{1}(s_t^k, a_t^k = s, a) r_t^k$$

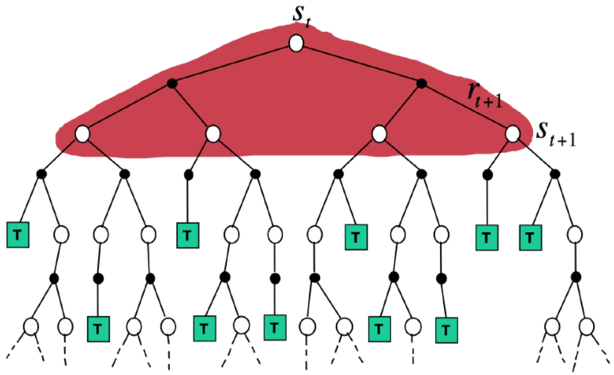
Bootstrapping and Sampling for DP, MC, and TD

- Bootstrapping: update involves an estimate
  - MC does not bootstrap
  - DP bootstraps
  - TD bootstraps

- Sampling: update samples an expectation
  - MC samples
  - DP does not sample
  - TD samples

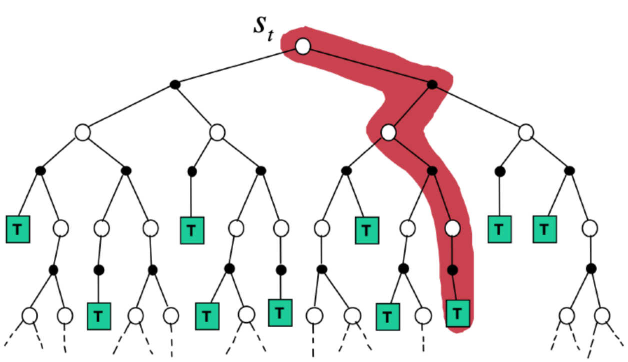
Unified View

$$v(S_t) \leftarrow \mathbb{E}_\pi[R_{t+1} + \gamma v(S_{t+1})]$$



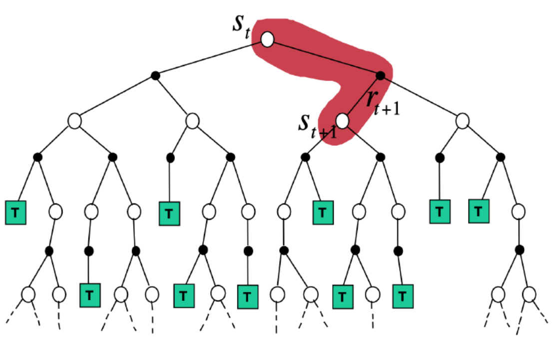
Dynamic Programming Backup

$$v(S_t) \leftarrow v(S_t) + \alpha(G_t - v(S_t))$$



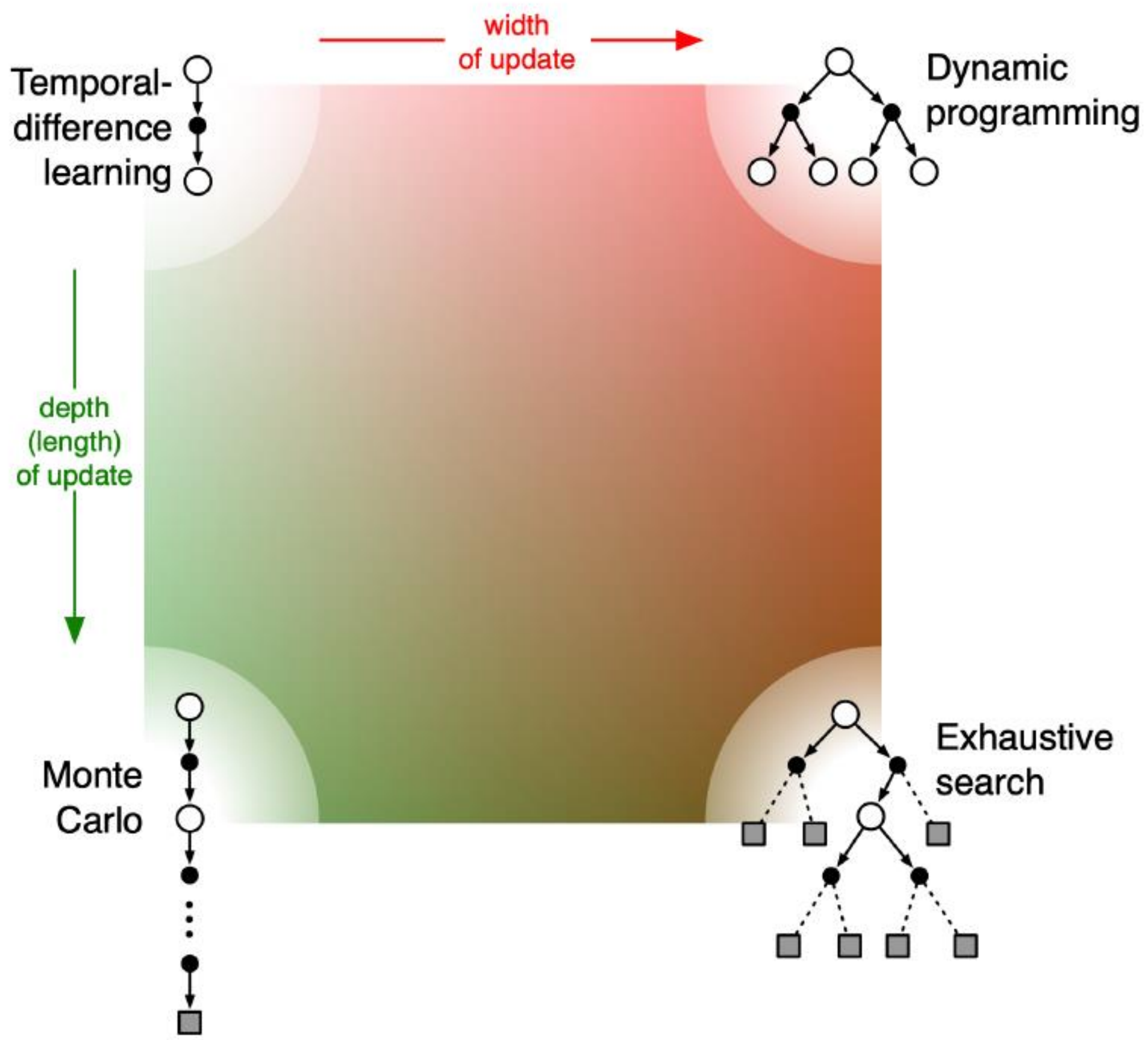
Monte-Carlo Backup

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



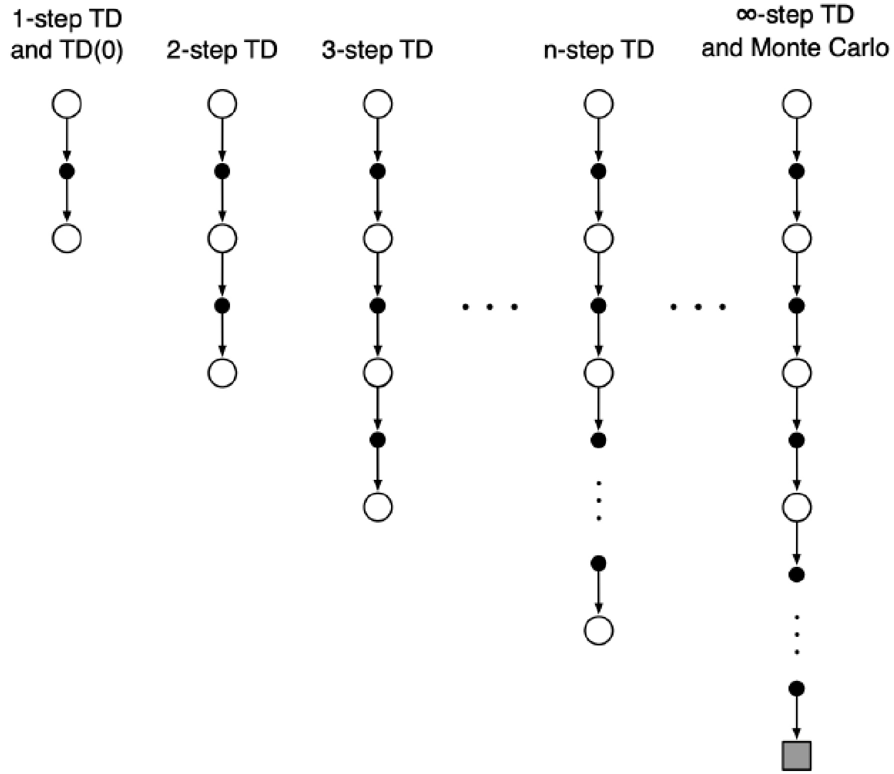
Temporal-Difference Backup

Unified View of RL



n-step TD

- n-step TD methods that generalize both one-step TD and MC.
- We can shift from one to the other smoothly as needed to meet the demands of a particular task.



n-step TD prediction

- Consider the following n-step returns for n = 1,2,∞

$$\begin{aligned} n = 1 (TD) \quad & G_t^{(1)} = R_{t+1} + \gamma v(S_{t+1}) \\ n = 2 \quad & G_t^{(2)} = R_{t+1} + \gamma R_{t+2} + \gamma^2 v(S_{t+2}) \\ & \vdots \\ n = \infty (MC) \quad & G_t^{\infty} = R_{t+1} + \gamma R_{t+2} + \dots + \gamma^{T-t-1} R_T \end{aligned}$$

- Thus the n-step return is defined as

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

- n-step TD: $v(S_t) \leftarrow v(S_t) + \alpha \left(G_t^n - v(S_t) \right)$

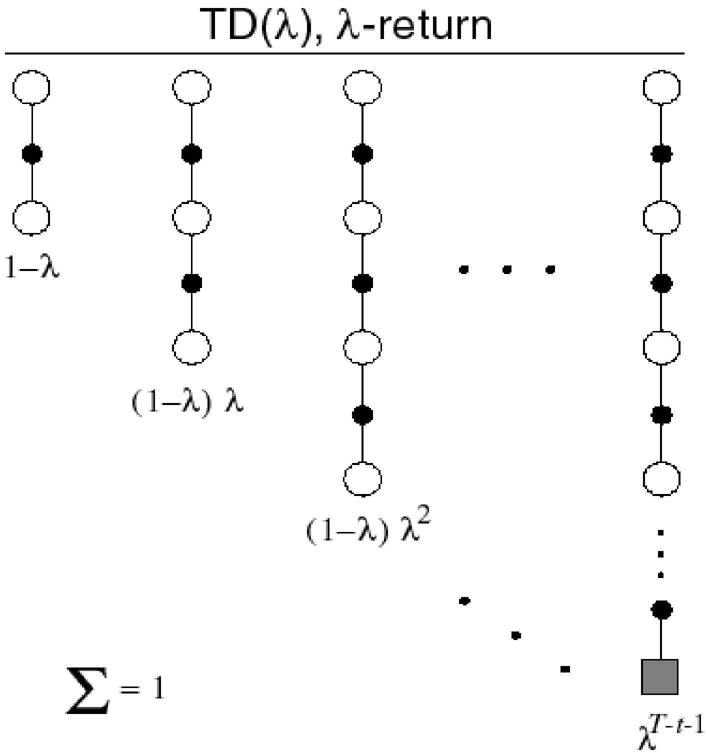
$\lambda - reutrn$

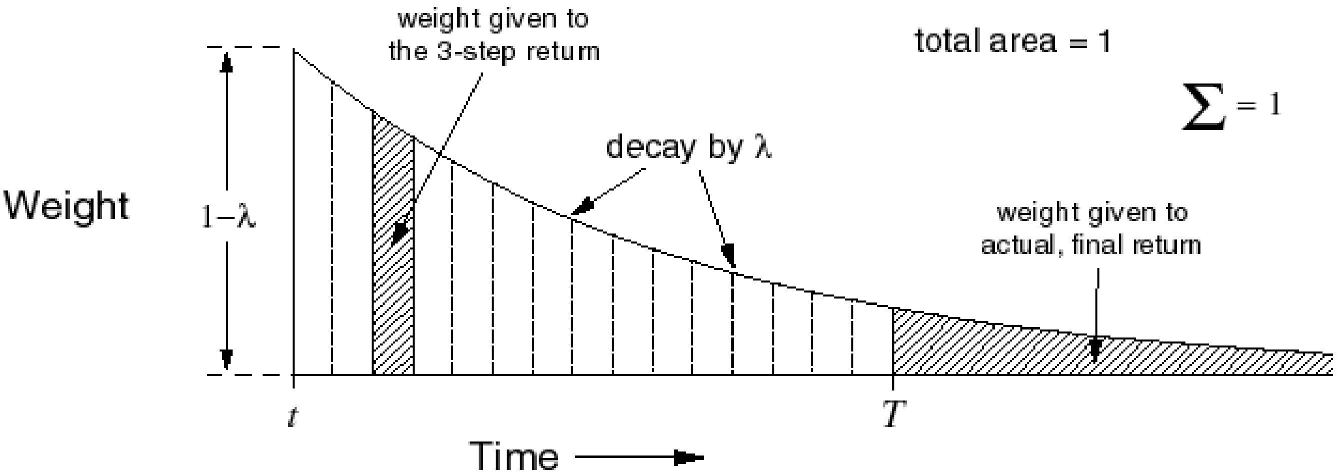
- The $\lambda - return\ G_t^\lambda$ combines all n-step returns G_t^n
- Using weight $(1-\lambda)\lambda^{n-1}$

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

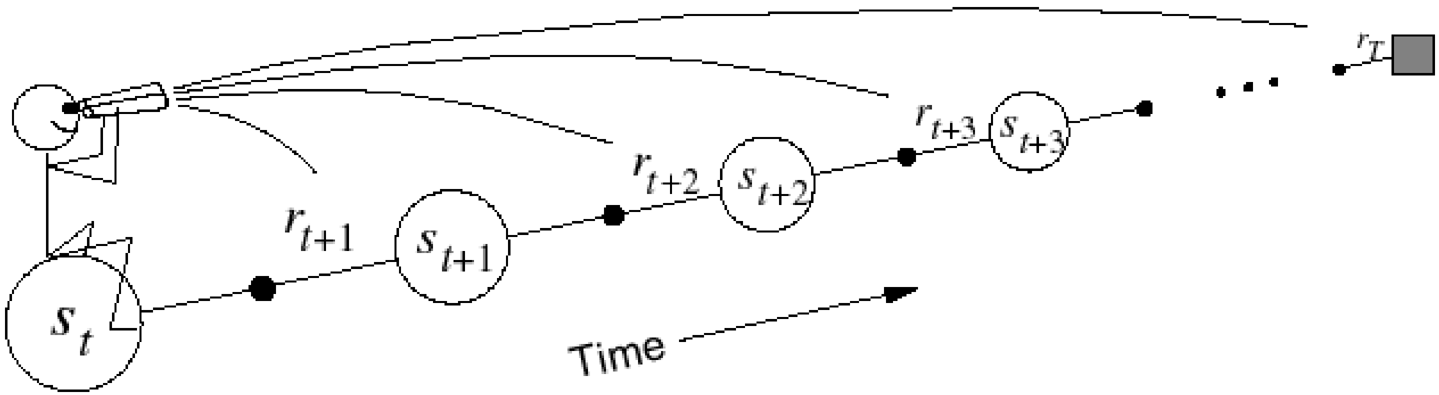
- Forward-view TD(λ)

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





Forward-view TD(λ)

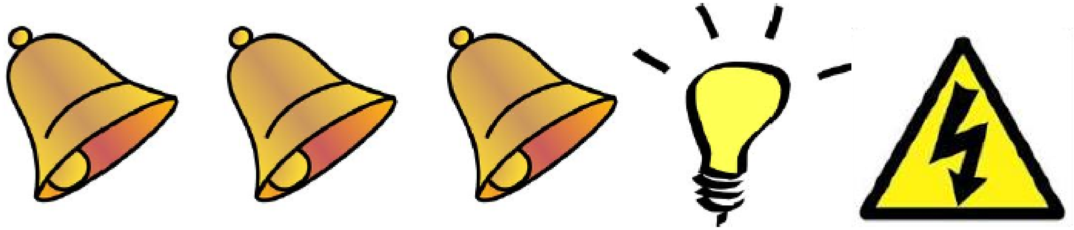


- Update value function towards the $\lambda - return$
- Forward-view looks into the future to compute G_t^{λ}
- Like MC, can only be computed from complete episodes

Backward-view TD(λ)

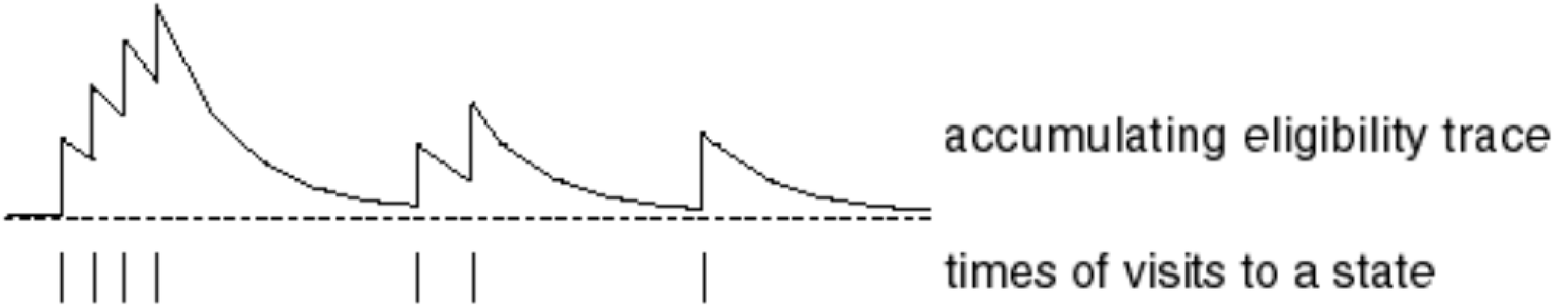
- Forward view provides theory
- Backward view provides mechanism
- Update online, every step, from incomplete sequences

Eligibility Traces



- 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

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

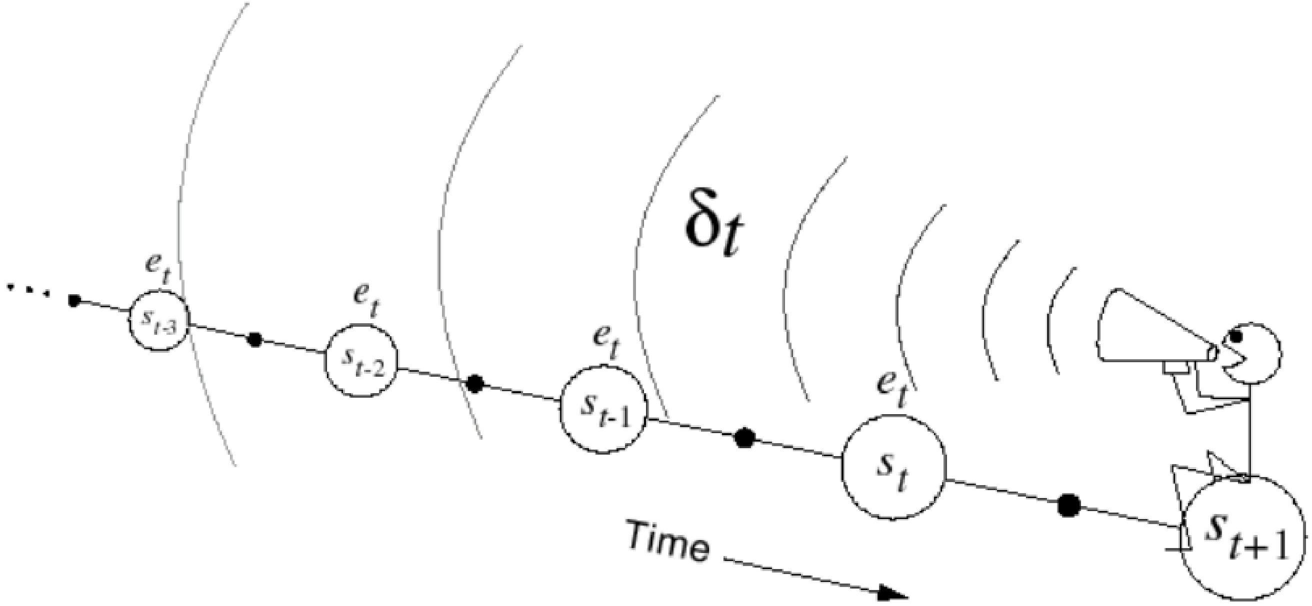


Backward-view TD(λ)

- 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 $E_t(s)$

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

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



TD(λ) and TD(0)

- When $\lambda = 0$, only current state is updated

$$E_t(s) = \mathbf{1}(S_t = s)$$
$$V(s) \leftarrow V(s) + \alpha \delta_t E_t(s)$$

- This is exactly equivalent to TD(0) update

$$V(S_t) \leftarrow V(S_t) + \alpha \delta_t$$

TD(λ) and MC

- 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)

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

TD(1) and MC

- Consider an episode where s is visited once at time-step k,
- TD(1) eligibility trace discounts time since visit,

$$\begin{aligned} E_t(s) &= \gamma E_{t-1}(s) + \mathbf{1}(S_t = s) \\ &= \begin{cases} 0 & \text{if } t < k \\ \gamma^{t-k} & \text{if } t \geq k \end{cases} \end{aligned}$$

- TD(1) updates accumulate error *online*

$$\sum_{t=1}^{T-1} \alpha \delta_t E_t(s) = \alpha \sum_{t=k}^{T-1} \gamma^{t-k} \delta_t = \alpha \left(G_k - V(S_k) \right)$$

- By end of episode it accumulates total error

$$\delta_k + \gamma \delta_{k+1} + \gamma^2 \delta_{k+2} + \ldots + \gamma^{T-1-k} \delta_{T-1}$$

Telescoping in TD(1)

When $\lambda = 1$, sum of TD errors telescopes into MC error,

$$\begin{aligned}
&\delta_t + \gamma\delta_{t+1} + \gamma^2\delta_{t+2} + ... + \gamma^{T-1-t}\delta_{T-1} \\
= & R_{t+1} + \gamma V(S_{t+1}) - V(S_t) \\
+ & \gamma R_{t+2} + \gamma^2 V(S_{t+2}) - \gamma V(S_{t+1}) \\
+ & \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3}) - \gamma^2 V(S_{t+2}) \\
& \quad \vdots \\
+ & \gamma^{T-1-t} R_T + \gamma^{T-t} V(S_T) - \gamma^{T-1-t} V(S_{T-1}) \\
= & R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} ... + \gamma^{T-1-t} R_T - V(S_t) \\
= & G_t - V(S_t)
\end{aligned}$$

TD(1) and TD(λ)

- TD(1) is roughly equivalent to every-visit Monte-Carlo
- Error is accumulated online, step-by-step
- If value function is only updated offline at end of episode
  Then total update is exactly the same as MC

Telescoping in TD(λ)

For general λ, TD errors also telescope to λ-error, $G_t^\lambda - V(S_t)$

$$
\begin{aligned}
G_t^\lambda - V(S_t) = -V(S_t) \quad & + \quad (1-\lambda)\lambda^0 \left(R_{t+1} + \gamma V(S_{t+1})\right) \\
& + \quad (1-\lambda)\lambda^1 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 V(S_{t+2})\right) \\
& + \quad (1-\lambda)\lambda^2 \left(R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 V(S_{t+3})\right) \\
& + \quad ... \\
= -V(S_t) \quad & + \quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma V(S_{t+1}) - \gamma\lambda V(S_{t+1})\right) \\
& + \quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma V(S_{t+2}) - \gamma\lambda V(S_{t+2})\right) \\
& + \quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma V(S_{t+3}) - \gamma\lambda V(S_{t+3})\right) \\
& + \quad ... \\
= \quad & \quad\quad (\gamma\lambda)^0 \left(R_{t+1} + \gamma V(S_{t+1}) - V(S_t)\right) \\
& + \quad (\gamma\lambda)^1 \left(R_{t+2} + \gamma V(S_{t+2}) - V(S_{t+1})\right) \\
& + \quad (\gamma\lambda)^2 \left(R_{t+3} + \gamma V(S_{t+3}) - V(S_{t+2})\right) \\
& + \quad ... \\
= \delta_t + \gamma\lambda\delta_{t+1} & + (\gamma\lambda)^2\delta_{t+2} + ...
\end{aligned}
$$

Forwards and Backwards TD(λ)

- Consider an episode where s is visited once at time-step k,
- TD(λ) eligibility trace discounts time since visit,

$$\begin{aligned} E_t(s) &= \gamma\lambda E_{t-1}(s) + \mathbf{1}(S_t = s) \\ &= \begin{cases} 0 & \text{if } t < k \\ (\gamma\lambda)^{t-k} & \text{if } t \geq k \end{cases} \end{aligned}$$

- Backward TD(λ) updates accumulate error *online*

$$\sum_{t=1}^{T} \alpha\delta_t E_t(s) = \alpha \sum_{t=k}^{T} (\gamma\lambda)^{t-k}\delta_t = \alpha\left(G_k^\lambda - V(S_k)\right)$$

- By end of episode it accumulates total error for λ-return
- For multiple visits to s, $E_t(s)$ accumulates many errors

Equivalence of Forward and Backward TD

- Offline updates
  - Updates are accumulated within episode
  - but applied in batch at the end of episode
- Online updates
  - TD(λ) updates are applied online at each step within episode
  - Forward and backward-view TD(λ) are slightly different
  - NEW: Exact online TD(λ) achieves perfect equivalence
  - By using a slightly different form of eligibility trace
  - Sutton and von Seijen, ICML 2014

Summary

Offline updates	$\lambda = 0$	$\lambda \in (0, 1)$	$\lambda = 1$
Backward view	TD(0)	TD(λ)	TD(1)
	$\parallel$	$\parallel$	$\parallel$
Forward view	TD(0)	Forward TD(λ)	MC
Online updates	$\lambda = 0$	$\lambda \in (0, 1)$	$\lambda = 1$
Backward view	TD(0)	TD(λ)	TD(1)
	$\parallel$	$\nparallel$	$\nparallel$
Forward view	TD(0)	Forward TD(λ)	MC
	$\parallel$	$\parallel$	$\parallel$
Exact Online	TD(0)	Exact Online TD(λ)	Exact Online TD(1)

$=$ here indicates equivalence in total update at end of episode.

Conclusion

- Model-free prediction
  - Evaluate the state value without knowing the MDP model, by only interacting with the environment
  - Main methods
    - Monte-Carlo Policy Evaluation
    - Temporal-Difference (TD) Learning