From model-based to model-free

Policy evaluation without knowing how the world works

Cathy Wu

6.7920 Reinforcement Learning: Foundations and Methods

Readings

1. NDP §5.1-5.3
2. Sutton & Barto (2018), §12.1-12.2

Outline

1. RL vs DP
2. Model-free policy evaluation

Outline

1. **RL vs DP**
   a. Model-based vs model-free
   b. Why learning from samples?
   c. Types of approximation
2. Model-free policy evaluation

Model-free vs model-based methods

- Model-free: **No** direct access to model *P, r*
- Model-based: **Yes** direct access to model *P, r*
- Recall: value iteration

$$V_{i+1}(s) = \mathcal{T}V_i(s) = \max_{a \in A} r(s,a) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,a)}[V_i(s')] \quad \text{for all } s$$

So far (Part 1, Lectures 1-6), our discussion has been model-based.

Recall (L1): Key challenge of huge decision spaces

Arcade Learning Environment (ALE)

$a_t = \text{left}$

t

Game of Go

Possible game states: $4^{200} \approx 10^{120}$

Possible game states: $3^{19x19} \approx 10^{172}$

For reference:
There are between 10^{78} to 10^{82} atoms in the observable universe.

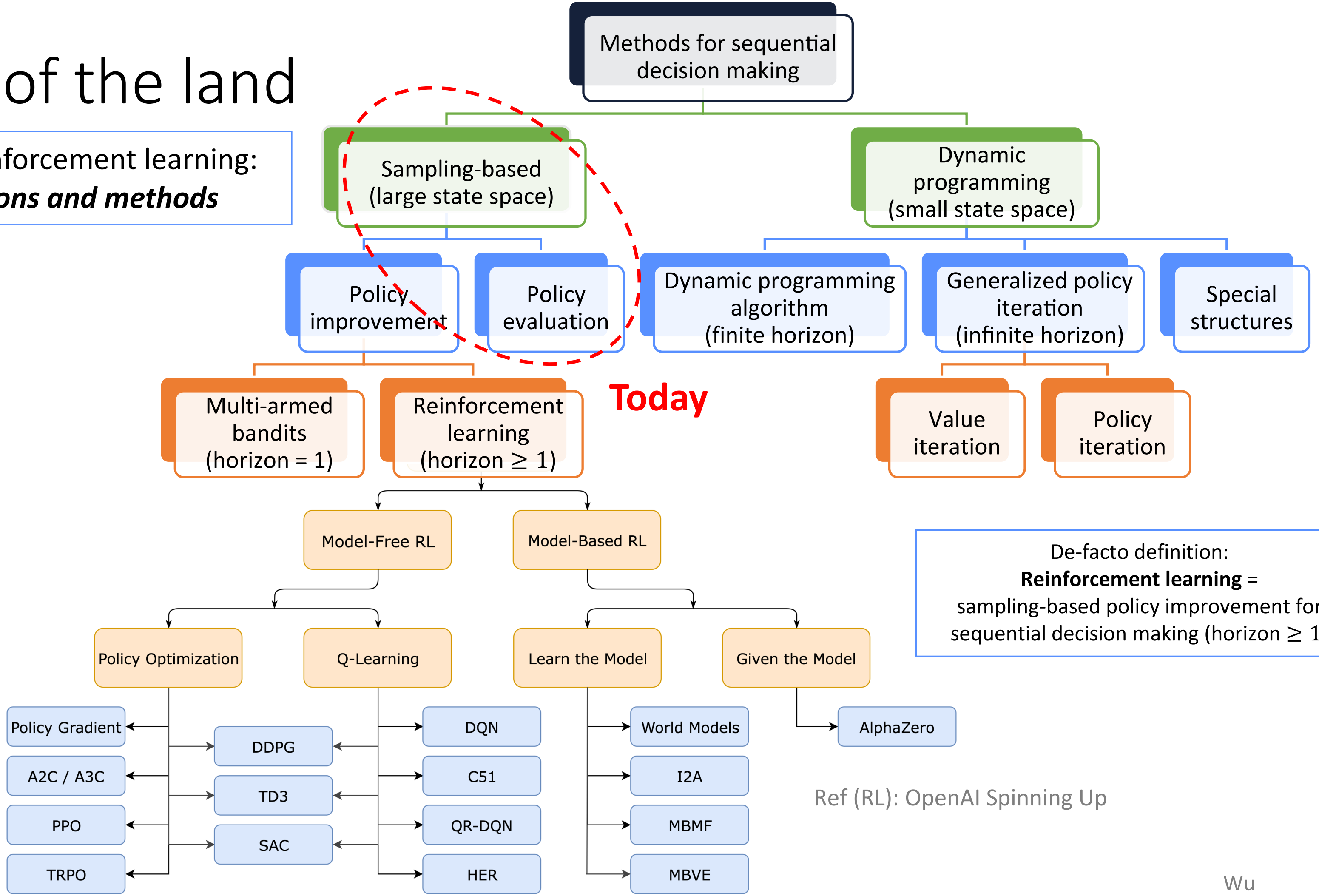
Cannot only explore. Cannot only exploit.
Must trade off exploration and exploitation.

Learning from Samples

- Dynamic programming algorithms require an **explicit definition** of:
  - transition probabilities $p(\cdot|s, a)$
  - reward function $r(s, a)$
- State spaces may be too **large** to compute.
- This knowledge is often **unavailable** (i.e., wind intensity, human-computer-interaction) or expensive.
- Can we relax this assumption?
- Can we solve a DP problem ***incrementally***, as more knowledge about $p(\cdot|s, a)$ and $r(s, a)$ is uncovered?

Lay of the land

6.7920: Reinforcement learning:
foundations and methods



De-facto definition:
Reinforcement learning =
sampling-based policy improvement for
sequential decision making (horizon ≥ 1)

Ref (RL): OpenAI Spinning Up

From exact DP to approximate DP

Note: Different types of approximation!

- Model-free updates for policy evaluation (today)
  - Techniques: Monte Carlo approximation, temporal differencing
- Model-free updates for optimal value functions ["RL"]
  - e.g., Q-learning; technique: stochastic approximation
- Approximating value functions
  - E.g., Approximate VI / PI
- Finite sample approximation ["RL"]
  - E.g., Fitted Q iteration, DQN
- Approximating policies ["RL"]
  - E.g., Policy gradient methods

Tabular methods

Function approximation

Sampling settings

- Learning with generative model. A black-box simulator f of the environment is available. Given (s, a),
$$f(s, a) = \{s', r\} \text{ with } s' \sim p(\cdot \mid s, a), r = r(s, a)$$
- Episodic learning. Multiple trajectories can be repeatedly generated from some initial states and terminating when a reset condition is achieved:
$$\left(s_{0,i}, s_{1,i}, \ldots, s_{T_i,i}\right)_{i=1}^{n}$$
- Online learning. At each time t the agent is at state s_t, it takes action a_t, it observes a transition to state s_{t+1}, and it receives a reward r_t. We assume that $s_{t+1} \sim p(\cdot \mid s_t, a_t)$ and $r_t = r(s_t, a_t)$ (i.e., MDP assumption). No reset.

Notice

From now on we typically work in the
episodic discounted setting.

Most results smoothly extend to other settings.

Assume: The value functions can be represented exactly (e.g. tabular setting).

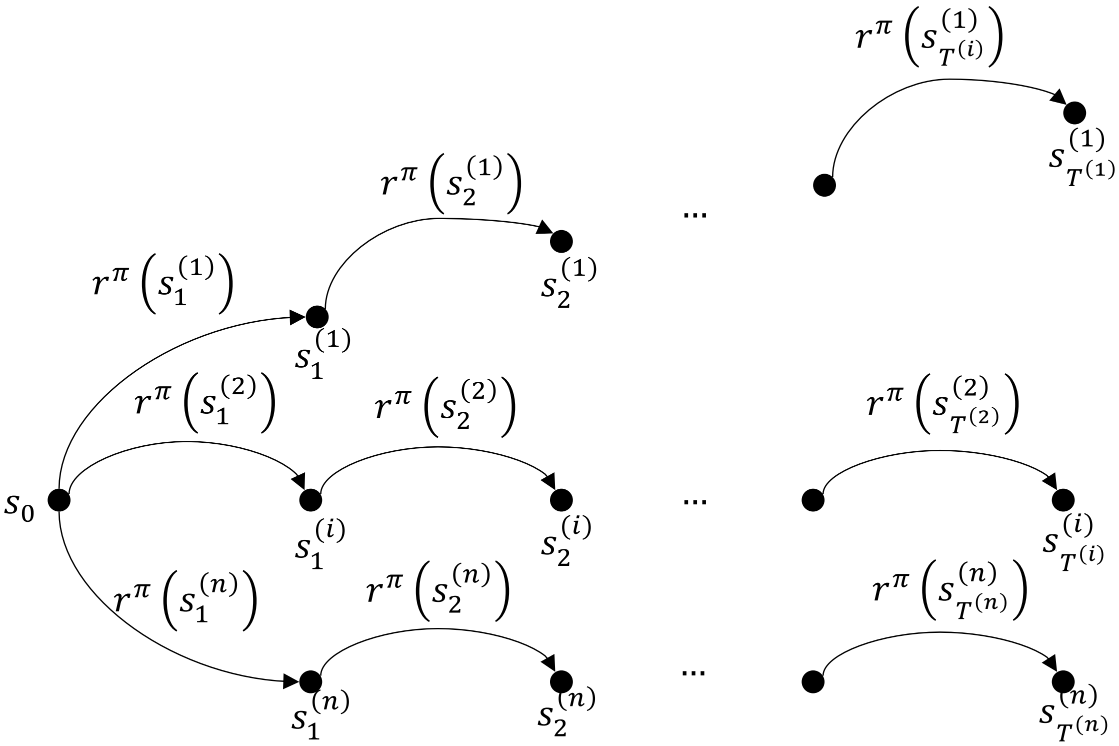
Outline

1. RL vs DP
2. **Model-free policy evaluation**
   a. Monte Carlo approximation
   b. Convergence of random variables
   c. Incremental Monte Carlo
   d. Stochastic approximation of a mean
   e. Temporal difference TD(0)
   f. TD(λ), eligibility traces

Warm-up: recall policy evaluation

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r\big(s_t, \pi(s_t)\big) | s_0 = s; \pi\right]$$

The RL Interaction Protocol



Policy Evaluation

Fixed policy π

For $i = 1, \ldots, n$

1. Set $t = 0$
2. Set initial state s_0
3. **While** ($s_{t,i}$ not terminal) [execute one trajectory]
   1. Take action $a_{t,i} = \pi(s_{t,i})$
   2. Observe next state $s_{t+1,i}$ and reward $r_{t+1,i} = r(s_{t,i}, a_{t,i})$
   3. Set $t = t + 1$

   EndWhile

Endfor

Return: Estimate of the value function $\hat{V}^{\pi}(\cdot)$

Policy Evaluation

Approach #1: Utilize the definition of **State Value Function**

Cumulative sum of rewards

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r\big(s_t, \pi(s_t)\big) | s_0 = s; \pi\right]$$

- Return of trajectory i starting from s_0

$$\hat{R}_i(s_0) = \sum_{t=0}^{T} \gamma^t r_{t,i}$$

- Estimated value function

$$\hat{V}_n^{\pi}(s_0) = \frac{1}{n}\sum_{i=1}^{n} \hat{R}_i(s_0)$$

Monte-Carlo Approximation of a Mean

Definition

Let X be a random variable with mean $\mu = \mathbb{E}[X]$ and variance $\sigma^2 = \mathbb{V}(X)$ and $x_n \sim X$ be n *i.i.d.* realizations of X. The Monte-Carlo approximation of the mean (i.e., the empirical mean) built on n i.i.d. realizations is defined as:

$$\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i$$

Monte-Carlo Approximation: Properties

Theorem

The returns used in the Monte-Carlo estimation starting from an initial state s_0 are unbiased estimators of V^π

$$\mathbb{E}\left[\hat{R}_i(s_0)\right] = \mathbb{E}\left[r_0 + \gamma r_{1,i} + \cdots + \gamma^{T_i} r_{T_{i},i}\right] = V^\pi(s_0)$$

Furthermore, the Monte-Carlo estimator converges to the value function

$$\hat{V}_n^\pi(s_0) \xrightarrow{a.s.} V^\pi(s_0)$$

- It applies to any state s used as the beginning of a trajectory (sub-trajectories could be used in practice)
- Finite-sample guarantees are possible (after n trajectories)

Convergence of Random Variables

Let X be a random variable and $\{X_n\}_{n\in\mathbb{N}}$ a sequence of random variables.

- $\{X_n\}$ converges to X almost surely, $X_n \xrightarrow{a.s.} X$, if:
$$\mathbb{P}\left(\lim_{n\to\infty} X_n = X\right) = 1$$
- $\{X_n\}$ converges to X in probability, $X_n \xrightarrow{P} X$, if for any $\epsilon > 0$:
$$\lim_{n\to\infty} \mathbb{P}[|X_n - X| > \epsilon] = 0$$
- $\{X_n\}$ converges to X in law, $X_n \xrightarrow{D} X$, if for any bounded continuous function f:
$$\lim_{n\to\infty} \mathbb{E}[f(X_n)] = \mathbb{E}[f(X)]$$
- $\{X_n\}$ converges to X in expectation, $X_n \xrightarrow{L^1} X$, if:
$$\lim_{n\to\infty} \mathbb{E}[X_n] = \mathbb{E}[X]$$

See HW0 for examples & counterexamples

Remark: $X_n \xrightarrow{a.s.} X \implies X_n \xrightarrow{P} X \implies X_n \xrightarrow{D} X$

Monte-Carlo Approximation of a Mean

- Unbiased estimator: Then $\mathbb{E}[\mu_n] = \mu$ $\left(\text{and } \mathbb{V}(\mu_n) = \frac{\mathbb{V}(X)}{n}\right)$
- Weak law of large numbers: $\mu_n \xrightarrow{P} \mu$
- Strong law of large numbers: $\mu_n \xrightarrow{a.s.} \mu$
- Central limit theorem (CLT): $\sqrt{n}\,(\mu_n - \mu) \xrightarrow{D} \mathcal{N}(0, \mathbb{V}(X))$
- Finite sample guarantee:

$$\mathbb{P}\Bigg[\underbrace{\left|\frac{1}{n}\sum_{i=1}^{n} X_t - \mathbb{E}[X_1]\right|}_{\text{deviation}} > \underbrace{\epsilon}_{\text{accuracy}}\Bigg] \le \underbrace{2\exp\left(-\frac{2n\epsilon^2}{(b-a)^2}\right)}_{\text{confidence}}$$

$$\mathbb{P}\left[\left|\frac{1}{n}\sum_{i=1}^{n} X_t - \mathbb{E}[X_1]\right| > \epsilon\right] \le \delta$$

If $n \ge \frac{(b-a)^2 \log\left(\frac{2}{\delta}\right)}{2\epsilon^2}$

Monte-Carlo Approximation: Extensions

Non-episodic problems:

- Interrupt trajectories after H steps:

$$\hat{R}_i(s_0) = \sum_{t=0}^{H} \gamma^t r_{t,i}$$

- Every return is ignoring a term:

$$\sum_{t=H+1}^{\infty} \gamma^t r_{t,i}$$

Monte-Carlo Approximation: Properties

Theorem

The Monte-Carlo estimator computed over H steps converges to a **biased** value function

$$\hat{V}_n^{\pi}(s_0) \xrightarrow{a.s.} \bar{V}_H^{\pi}(s_0)$$

Such that

$$|\bar{V}_H^{\pi}(s_0) - V^{\pi}(s_0)| \leq \gamma^H \frac{r_{\max}}{1-\gamma}$$

- Proof: by geometric series.

Monte-Carlo: an Incremental Implementation

- Approach #2: **Incremental version** of state value function definition
- Return of trajectory i starting from s_0

$$\hat{R}_i(s_0) = \sum_{t=0}^{T_i} \gamma^t r_{t,i}$$

- Estimated value function

$$\hat{V}_n^\pi(s_0) = \frac{1}{n}\sum_{i=1}^{n} \hat{R}_i(s_0) = \frac{n-1}{n}\hat{V}_{n-1}^\pi(s_0) + \frac{1}{n}\hat{R}_n(s_0)$$

$$\approx \big(1-\eta(n)\big)\hat{V}_{n-1}^\pi(s_0) + \eta(n)\hat{R}_n(s_0)$$

Incremental Monte-Carlo Policy Evaluation

Fixed policy π

For $i = 1, \dots, n$

1. Set $t = 0$
2. Set initial state s_0
3. **While** (s_t not terminal) [execute one trajectory]
   1. Take action $a_t = \pi(s_t)$
   2. Observe next state x_{t+1} and reward $r_t = r^{\pi}(s_t)$
   3. Set $t = t + 1$

   EndWhile
4. Update $\hat{V}_i^{\pi}(s_0)$ using $TD(1)$ approximation

TD(λ) = temporal differences with parameter λ

Endfor

~~Collect trajectories and compute $\hat{V}_n^{\pi}(s_0)$ using Monte-Carlo approximation~~

Incremental Monte-Carlo: Properties

Theorem

Let the incremental Monte-Carlo estimator be computed using a learning rate $\{\eta(n)\}_n$ such that

$$\sum_{i=0}^{\infty} \eta(i) = \infty \quad \sum_{i=0}^{\infty} \eta(i)^2 < \infty \quad \text{[Robbins Monro's condition]}$$

Then

$$\hat{V}_n^{\pi}(s_0) \xrightarrow{a.s.} V^{\pi}(s_0)$$

- Need some new mathematical tools
- Incremental Monte-Carlo estimation converges to V^{π} for a wide range of choices of learning rate schemes.
- This scheme is often referred to as $TD(1)$, for reasons that will be clear shortly.

Stochastic Approximation of a Mean

Definition

Let X be a random variable bounded in $[0,1]$ with mean $\mu = \mathbb{E}[X]$ and $x_n \sim X$ be n *i.i.d.* realizations of X. The stochastic approximation of the mean is,

$$\mu_n = (1 - \eta_n)\mu_{n-1} + \eta_n x_n$$

With $\mu_1 = x_1$ and where (η_n) is a sequence of learning steps.

Stochastic Approximation of a Mean

Proposition

If for any $n, \eta_n \geq 0$ are such that

$$\sum_{n\geq 0} \eta_n = \infty \qquad \sum_{n\geq 0} \eta_n^2 < \infty$$

Then

$$\mu_n \xrightarrow{a.s.} \mu$$

And we say that μ_n is a consistent estimator.

Remark: When $\eta_n = \frac{1}{n}$, this is the recursive (incremental) definition of the empirical mean.

Intuition: Incremental updates

$\eta_n = \frac{1}{n}$ i.e., $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \ldots$

- Consider a simple setting: **mean of a sequence of numbers**
- $x = (x_n) = (5, 2, 9, 10, 1, 3)$
- Mean: $\bar{x} = \frac{5+2+9+10+1+3}{6} = 5$
- Incremental mean:

$$\mu_0 = 0$$

$$\mu_{n+1} = (1 - \eta_n)\underbrace{\mu_n}_{\text{Policy evaluation estimate}} + \underbrace{\eta_n x_n}_{\text{increment}}$$

$$\mu_{n+1} = \underbrace{\mu_n}_{\text{Policy evaluation estimate}} + \eta_n \underbrace{(x_n - \mu_n)}_{\text{error}}$$

$$\mu_0 = 0$$

$$\mu_1 = 0 \cdot 0 + 1 \cdot 5$$

$$\mu_2 = \frac{1}{2} 5 + \frac{1}{2} 2 = 3.5$$

$$\mu_3 = \frac{2}{3} 3.5 + \frac{1}{3} 9 = 5.333$$

$$\mu_4 = \frac{3}{4} 5.333 + \frac{1}{4} 10 = 6.5$$

$$\mu_5 = \frac{4}{5} 6.5 + \frac{1}{5} 1 = 5.4$$

$$\mu_6 = \frac{5}{6} 5.4 + \frac{1}{6} 3 = 5$$

← Success!

Stochastic Approximation of a Mean

If $\eta_n = \frac{1}{n}$, then $\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i$.

Proof: Base case $(n = 1)$**:** $\mu_1 = x_1$ (given).

Induction step. Assume $\mu_n = \frac{1}{n}\sum_{i=1}^{n} x_i$.

$$\mu_{n+1} = \left(1 - \frac{1}{n+1}\right)\mu_n + \frac{1}{n+1}x_{n+1}$$

$$= \left(\frac{n}{n+1}\right)\mu_n + \frac{1}{n+1}x_{n+1}$$

$$= \left(\frac{n}{n+1}\right)\frac{1}{n}\sum_{i=1}^{n} x_i + \frac{1}{n+1}x_{n+1}$$

$$= \left(\frac{1}{n+1}\right)\sum_{i=1}^{n} x_i + \frac{1}{n+1}x_{n+1}$$

$$= \left(\frac{1}{n+1}\right)\sum_{i=1}^{n+1} x_i$$

Intuition: Incremental updates

$\eta_n = \frac{1}{n}$ i.e., $1, \frac{1}{2}, \frac{1}{3}, \frac{1}{4} \ldots$

- Consider a simple setting: **mean of a sequence of numbers**
- $x = (x_n) = (5, 2, 9, 10, 1, 3)$
- Mean: $\bar{x} = \frac{5+2+9+10+1+3}{6} = 5$
- Incremental mean:

$$\mu_0 = 0$$

$$\mu_{n+1} = (1 - \eta_n)\underbrace{\mu_n}_{\text{(Optimal) value function estimate}} + \eta_n \underbrace{x_n}_{\text{increment}}$$

$$\mu_{n+1} = \overbrace{\mu_n}^{\text{(Optimal) value function estimate}} + \eta_n \underbrace{(x_n - \mu_n)}_{\text{error}}$$

Preview of upcoming lectures:
Also works for Bellman operators!
i.e., (optimal) value functions

Incremental update of a **fixed point**

Same basic idea
Analysis is more involved

Temporal Difference $TD(1)$: Extensions

- Non-episodic problems: Truncated trajectories

- Multiple sub-trajectories
  - Updates of all the states using sub-trajectories
  - State-dependent learning rate $\eta_i(x)$
  - i is the index of the number of updates in that specific state

Incremental Monte-Carlo Policy Evaluation

Fixed policy π

For $i = 1, \dots, n$

1. Set $t = 0$
2. Set initial state s_0
3. **While** (s_t not terminal) [execute one trajectory]
   1. Take action $a_t = \pi(s_t)$
   2. Observe next state x_{t+1} and reward $r_t = r^\pi(s_t)$
   3. Set $t = t + 1$

   EndWhile
4. Update $\hat{V}_i^\pi(s_0)$ using $TD(1)$ approximation

TD(λ) = temporal differences with parameter λ

Endfor

~~Collect trajectories and compute $\hat{V}_n^\pi(s_0)$ using Monte-Carlo approximation~~

Temporal-Difference $TD(0)$ Estimation

- Approach #3: Conduct **incremental updates** *<u>within trajectories</u>*, leveraging the **Bellman equation**
- Recall: The Bellman equation

$$V^\pi(s) = r\big(s, \pi(s)\big) + \gamma \mathbb{E}_{s' \sim P(\cdot|s,\pi(s))}[V^\pi(s')]$$

- Incremental update: At each step t, observe s_t, r_t, s_{t+1} and update estimate $\hat{V}^\pi$ as

$$\hat{V}^\pi(s_t) = (1-\eta)\hat{V}^\pi(s_t) + \eta\left(r_t + \gamma \hat{V}^\pi(s_{t+1})\right)$$

Temporal-Difference $TD(0)$: Estimation

- At each step t, observe s_t, r_t, s_{t+1} and update estimate $\hat{V}^\pi$ as

$$\hat{V}^\pi(s_t) = (1-\eta)\hat{V}^\pi(s_t) + \eta\left(r_t + \gamma \hat{V}^\pi(s_{t+1})\right)$$

- Interpretation: moving average
  - Mix between old and new estimate of $V^\pi(s_t)$:

    old estimate $\hat{V}^\pi(s_t)$ new estimate $r_t + \gamma \hat{V}^\pi(s_{t+1})$
  - Weighted average:

$$\hat{V}^\pi(s_t) = (1-\eta)\hat{V}^\pi(s_t) + \eta\left(r_t + \gamma \hat{V}^\pi(s_{t+1})\right)$$

Temporal-Difference $TD(0)$: Estimation

- Equivalently

temporal difference (TD) error

$$\hat{V}^{\pi}(s_t) = \hat{V}^{\pi}(s_t) + \eta\left(\underbrace{r_t + \gamma\hat{V}^{\pi}(s_{t+1}) - \hat{V}^{\pi}(s_t)}_{\text{temporal difference (TD) error}}\right)$$

- Interpretation: temporal-difference error
  - Temporal difference error of estimate $\hat{V}^{\pi}$ w.r.t. transition (s_t, r_t, s_{t+1}):

$$\delta_t = r_t + \gamma\hat{V}^{\pi}(s_{t+1}) - \hat{V}^{\pi}(s_t)$$

  - Bellman error for function $\hat{V}$ at state s:

$$\mathcal{B}^{\pi}\left(\hat{V}; s\right) = \mathcal{T}^{\pi}\hat{V}(s) - \hat{V}(s)$$
$$= r^{\pi}(s) + \gamma\mathbb{E}_{s'|s}\left[\hat{V}(s')\right] - \hat{V}(s) \qquad [\mathcal{B}^{\pi}(V^{\pi}; s) = 0]$$

  - Conditioned on s_t, δ_t is an unbiased estimate of $\mathcal{B}^{\pi}$:

$$\mathbb{E}_{r_t,s_{t+1}}[\delta_t|s_t] = r^{\pi}(s_t) + \gamma\mathbb{E}_{s_{t+1}|s_t}\left[\hat{V}^{\pi}(s_{t+1})\right] - \hat{V}^{\pi}(s_t) = \mathcal{B}^{\pi}\left(\hat{V}^{\pi}, s_t\right)$$

Temporal-Difference $TD(0)$: Properties

Theorem

Let $TD(0)$ run with learning rate $\eta(N_t(s_t))$ where $N_t(s_t)$ is the number of visits to the state s_t. If all states are visited infinitely often and the learning rate is set such that:

$$\sum_{t=0}^{\infty} \eta(t) = \infty \qquad \sum_{t=0}^{\infty} \eta(t)^2 < \infty \qquad \text{[Robbins Monro's condition]}$$

Then for any state $s \in \mathcal{S}$

$$\hat{V}^{\pi}(s) \xrightarrow{a.s.} V^{\pi}(s)$$

Temporal Difference $TD(0)$

For $i = 1, \dots, n$

1. Set $t = 0$
2. Set initial state s_0
3. **While** (s_t not terminal) [execute one trajectory]
   1. Take action $a_{t,i} = \pi(s_{t,i})$
   2. Observe next state $s_{t+1,i}$ and reward $r_{t,i} = r(s_{t,i}, a_{t,i})$
   3. Set $t = t + 1$
   4. Update $\hat{V}^{\pi}(s_{t,i})$ using $TD(0)$ estimation

   EndWhile
4. Update $\hat{V}_i^{\pi}(s_0)$ using incremental Monte-Carlo estimation

Endfor

Incremental Monte-Carlo as a "TD method"

Temporal difference $\delta_t = r_t + \gamma \hat{V}^\pi(s_{t+1}) - \hat{V}^\pi(s_t)$

Incremental Monte-Carlo

$$\hat{V}^\pi_{n+1}(s_0) = (1-\eta_{n+1})\hat{V}^\pi_n(s_0) + \eta_{n+1}\hat{R}_{n+1}(s_0)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(\hat{R}_{n+1}(s_0) - \hat{V}^\pi_n(s_0)\right)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(r_{0,n} + \gamma r_{1,n} + \gamma^2 r_{2,n} + \gamma^3 r_{3,n} + \cdots - \hat{V}^\pi_n(s_0)\right)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(r_{0,n} + \gamma\hat{V}^\pi_n(s_{1,n}) - \hat{V}^\pi_n(s_0) - \gamma\hat{V}^\pi_n(s_{1,n}) + \gamma r_{1,n} + \gamma^2 r_{2,n} + \gamma^3 r_{3,n} + \cdots\right)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(\delta_{0,n} - \gamma\hat{V}^\pi_n(s_{1,n}) + \gamma r_{1,n} + \gamma^2 r_{2,n} + \gamma^3 r_{3,n} + \cdots\right)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(\delta_{0,n} + \gamma r_{1,n} + \gamma^2\hat{V}^\pi_n(s_{2,n}) - \gamma\hat{V}^\pi_n(s_{1,n}) - \gamma^2\hat{V}^\pi_n(s_{2,n}) + \gamma^2 r_{2,n} + \gamma^3 r_{3,n} + \cdots\right)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(\delta_{0,n} + \gamma\delta_{1,n} - \gamma^2\hat{V}^\pi_n(s_{2,n}) + \gamma^2 r_{2,n} + \gamma^3 r_{3,n} + \cdots\right)$$
$$= \hat{V}^\pi_n(s_0) + \eta_{n+1}\left(\delta_{0,n} + \gamma\delta_{1,n} + \gamma^2\delta_{2,n} + \ \ldots + \gamma^{T_n - 1}\delta_{T_n,n}\right)$$

Compare: TD(0)

$$\hat{V}^\pi(s_t) = \hat{V}^\pi(s_t) + \eta\left(r_t + \gamma\hat{V}^\pi(s_{t+1}) - \hat{V}^\pi(s_t)\right) = \hat{V}^\pi(s_t) + \eta\delta_t$$

Temporal Difference $TD(\lambda)$

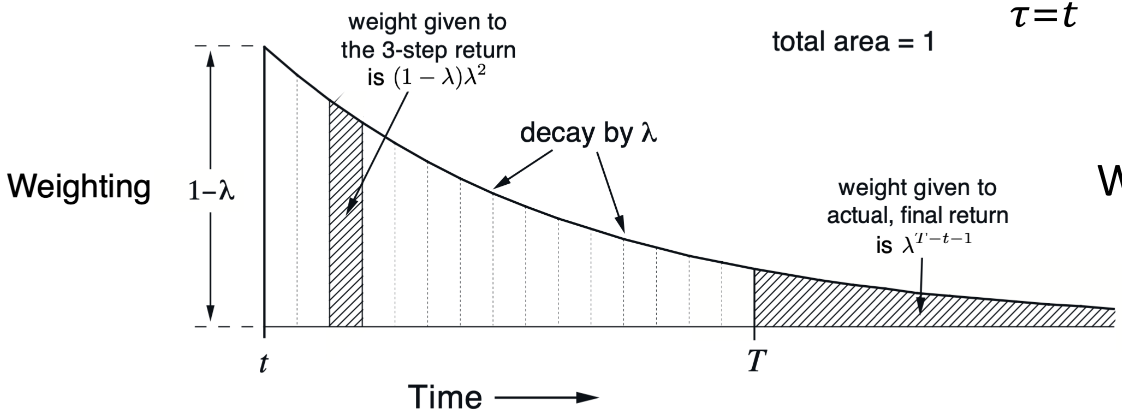
Idea: Use the whole series of temporal differences to update $\hat{V}^{\pi}$

- Temporal difference of a function $\hat{V}^{\pi}$ for a transition $\langle s_t, r_t, s_{t+1} \rangle$

$$\delta_t = r_t + \gamma \hat{V}^{\pi}(s_{t+1}) - \hat{V}^{\pi}(s_t)$$

- Estimated value function

$$\hat{V}^{\pi}(s_t) = \hat{V}^{\pi}(s_t) + \eta(s_t) \sum_{\tau=t}^{T} (\gamma\lambda)^{\tau-t} \delta_\tau$$



Weighting given in the λ-return to each of the n-step returns

Comparison of $TD(1)$ [Incremental MC] and $TD(0)$

Temporal difference $\delta_t = r_t + \gamma \hat{V}^\pi(s_{t+1}) - \hat{V}^\pi(s_t)$

- Incremental Monte-Carlo, i.e. $TD(1)$:

$$\hat{V}^\pi(s_0) = \hat{V}^\pi(s_0) + \eta[\delta_0 + \gamma\delta_1 + \ \ldots + \gamma^{T-1}\delta_T]$$

$\Longrightarrow$ No bias, large variance [long trajectory]

- $TD(0)$:

$$\hat{V}^\pi(s_0) = \hat{V}^\pi(s_0) + \eta\delta_0$$

$\Longrightarrow$ Large bias [“bootstrapping” on wrong values], small variance

The $\mathcal{T}_\lambda^\pi$ Bellman Operator

Definition

Given $\lambda < 1$, then the Bellman operator $\mathcal{T}_\lambda^\pi$ is:

$$\mathcal{T}_\lambda^\pi = (1-\lambda) \sum_{m \geq 0} \lambda^m (\mathcal{T}^\pi)^{m+1}$$

Remark: Convex combination of the m-step Bellman operators $(\mathcal{T}^\pi)^m$ weighted by a sequence of coefficients defined as a function of a λ.

Temporal Difference $TD(\lambda)$

Estimated value function

$$\hat{V}^{\pi}(s_t) = \hat{V}^{\pi}(s_t) + \eta(s_t) \sum_{\tau=t}^{T} (\gamma\lambda)^{\tau-t} \delta_{\tau}$$

$\Longrightarrow$ Once again requires the whole trajectory before updating…

Temporal Difference $TD(\lambda)$: Eligibility Traces

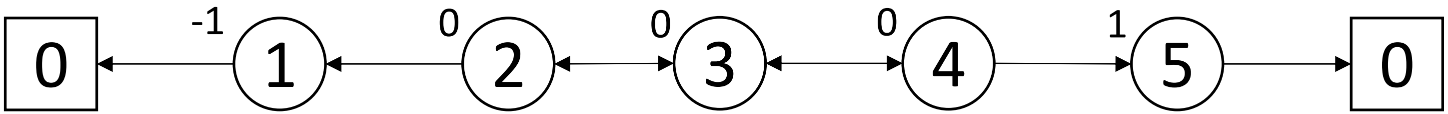
- **Eligibility** traces $z \in \mathbb{R}^S$. **Short-term memory vector**.
- At the start of the episode, reset the traces: $z = 0$
- For every transition $s_t \rightarrow s_{t+1}$
  1. Compute the temporal difference
  $$\delta_t = r_t(s_t) + \gamma \hat{V}^\pi(s_{t+1}) - \hat{V}^\pi(s_t)$$
  2. Update the eligibility traces
  $$z(s) = \begin{cases} \gamma\lambda z(s) & \text{if } s \neq s_t \\ 1 + \gamma\lambda z(s) & \text{if } s = s_t \end{cases}$$
  [decay the contribution]
  [increment the contribution]
  3. For all state $s \in S$ [all states are updated at each step]
  $$\hat{V}^\pi(s) \leftarrow \hat{V}^\pi(s) + \eta(s) z(s) \delta_t$$

Sensitivity to λ

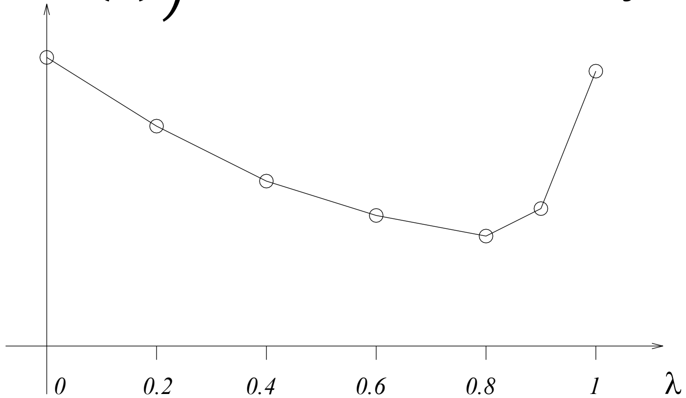
- $\lambda < 1$: smaller variance w.r.t. $\lambda = 1$ ($\approx$ incremental Monte-Carlo)
- $\lambda > 0$: faster propagation of rewards w.r.t. $\lambda = 0$

Example: Sensitivity to λ

Linear chain example



Error $\sum_{s\in S}\left(\hat{V}^{\pi}(s) - \hat{V}^{\pi}(s)\right)^2$ after $n = 100$ trajectories



Summary of methods

	Dynamic Programming	Monte Carlo	Temporal Difference
Model Free?	No	Yes	Yes
Non-episodic domains?	Yes	No	Yes
Non-Markovian domains?	No	Yes	No
Converges to true value	Yes	Yes	Yes
Unbiased Estimate	N/A	Yes	No
Variance	N/A	High	Low

Table courtesy Youkow Homma, Emma Brunskill

Summary

- Reinforcement **learning** vs dynamic programming
- **Learning = incremental updates**. Also called **bootstrapping**
- **Types of approximation** in approximate dynamic programming
- **Incremental mean**: warm-up for stochastic approximation
- Policy evaluation: **Monte-Carlo** and **Temporal Difference** (definition, methods, pros and cons)

References

1. Alessandro Lazaric. INRIA Lille. Reinforcement Learning. 2017, Lectures 2-3.

2. Sutton & Barto (2018). §12.1-12.2