# REINFORCEMENT LEARNING: THEORY AND PRACTICE

## Ch. 7: n-step Bootstrapping

Profs. Amy Zhang and Peter Stone

## Previously

Chapter 6 Temporal-Difference methods: Introducing bootstrapping to Monte Carlo methods

Focus on the 1-step problem: I have 1 data transition, how do I update my value function?

# Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

## Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

1-step TD update:

## Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

1-step TD update:

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

## Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

1-step TD update:

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

2-step TD update:

## Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

1-step TD update:

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

2-step TD update:

$$G_{t:t+2} \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 V_{t+1}(S_{t+2})$$

## Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

1-step TD update:

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

n-step TD update:

## Previously: MC and TD

Monte Carlo update:

$$G_t \doteq R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots + \gamma^{T-t-1} R_T$$

1-step TD update:

$$G_{t:t+1} \doteq R_{t+1} + \gamma V_t(S_{t+1})$$

n-step TD update:

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$

## N-step TD prediction

The space of methods between Monte Carlo and TD. This gives us the following state-value learning algorithm:

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\big[G_{t:t+n} - V_{t+n-1}(S_t)\big], \qquad 0 \le t < T$$

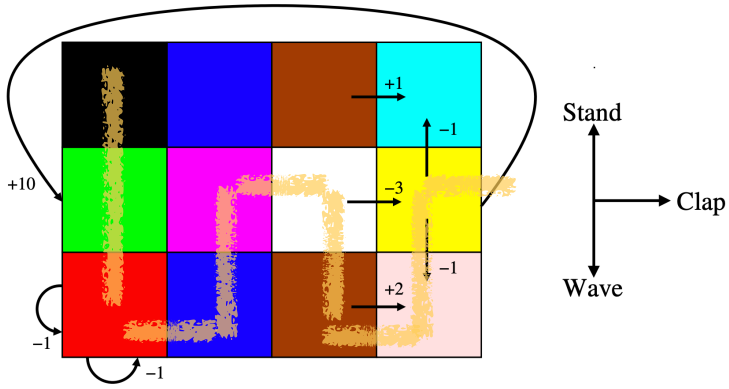
while the values of all other states remain unchanged:

$$V_{t+n}(s) = V_{t+n-1}(s), \text{ for all } s \neq S_t$$

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

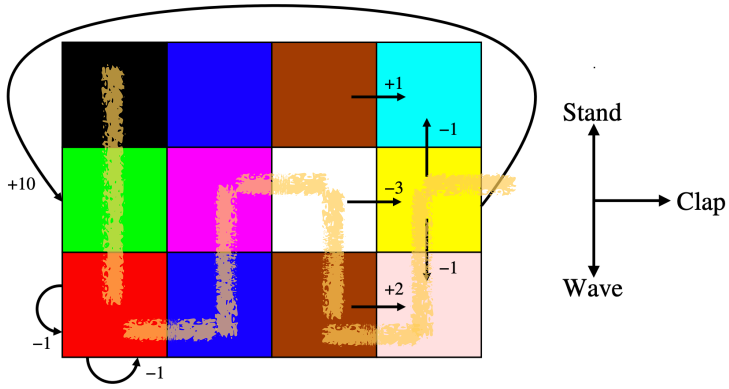
1-step:

| | | | |
|---|---|---|---|
| 0 0 | 1 0 | 2 0 | 3 0 |
| 4 0 | 5 0 | 6 0 | 7 0 |
| 8 0 | 9 0 | 10 0 | 11 0 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

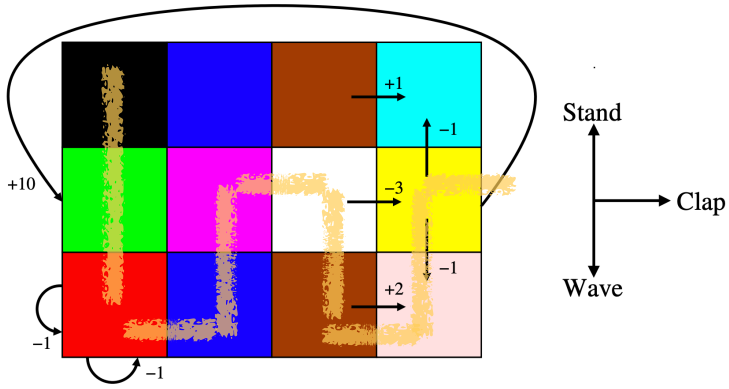
1-step:

| | | | |
|---|---|---|---|
| 0 0 | 1 0 | 2 0 | 3 0 |
| 4 0 | 5 0 | 6 0 | 7 5 |
| 8 0 | 9 0 | 10 1 | 11 0 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

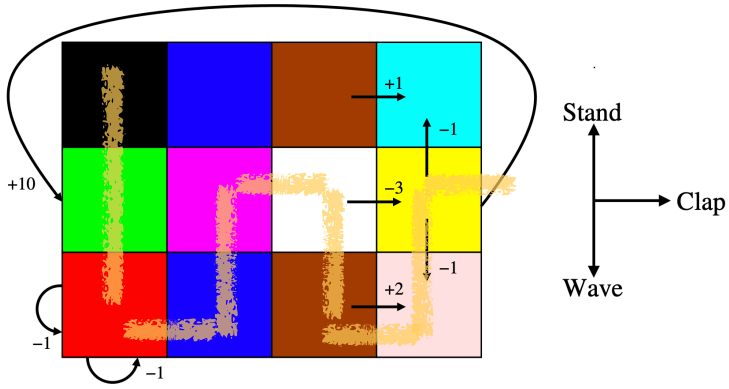
2-step:

| | | | |
|---|---|---|---|
| 0 0 | 1 0 | 2 0 | 3 0 |
| 4 0 | 5 0 | 6 0 | 7 0 |
| 8 0 | 9 0 | 10 0 | 11 0 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

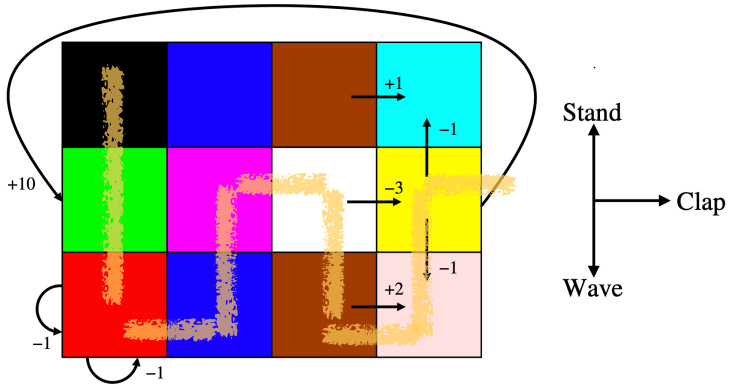
2-step:

| | | | |
|---|---|---|---|
| 0<br>0 | 1<br>0 | 2<br>0 | 3<br>0 |
| 4<br>0 | 5<br>0 | 6<br>1 | 7<br>5 |
| 8<br>0 | 9<br>0 | 10<br>1 | 11<br>5 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

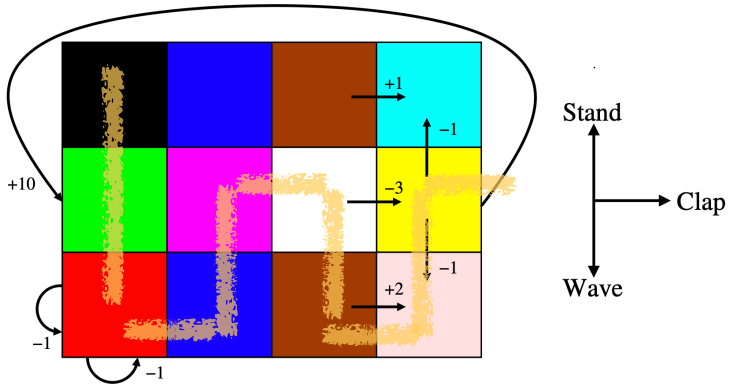
3-step:

| | | | |
|---|---|---|---|
| 0 0 | 1 0 | 2 0 | 3 0 |
| 4 0 | 5 0 | 6 0 | 7 0 |
| 8 0 | 9 0 | 10 0 | 11 0 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

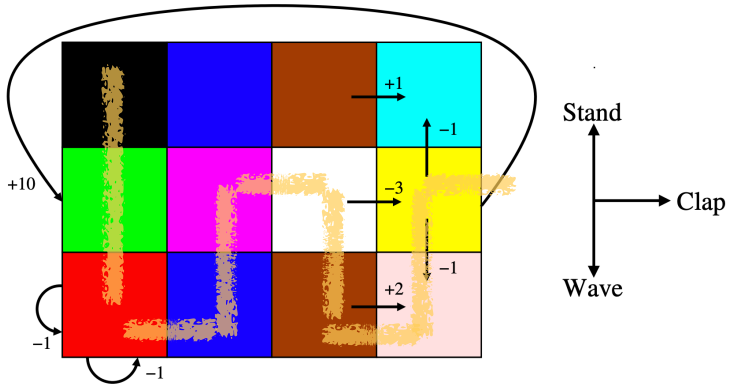
3-step:

| | | | |
|---|---|---|---|
| 0<br>0 | 1<br>0 | 2<br>0 | 3<br>0 |
| 4<br>0 | 5<br>1 | 6<br>1 | 7<br>5 |
| 8<br>0 | 9<br>0 | 10<br>6 | 11<br>5 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

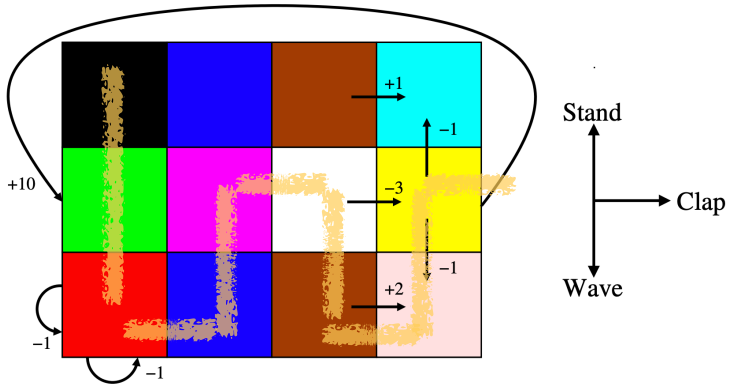
Monte Carlo:

| | | | |
|---|---|---|---|
| 0 0 | 1 0 | 2 0 | 3 0 |
| 4 0 | 5 0 | 6 0 | 7 0 |
| 8 0 | 9 0 | 10 0 | 11 0 |

# Exercise

$$V_{t+n}(S_t) \doteq V_{t+n-1}(S_t) + \alpha\left[G_{t:t+n} - V_{t+n-1}(S_t)\right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n V_{t+n-1}(S_{t+n})$$




Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

Gamma = 1, alpha = 0.5

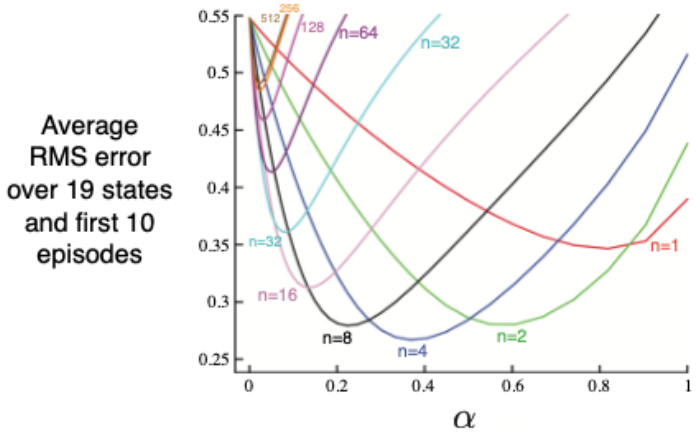
Monte Carlo:

| | | | |
|---|---|---|---|
| 0 12 | 1 0 | 2 0 | 3 0 |
| 4 12 | 5 12 | 6 12 | 7 10 |
| 8 12 | 9 12 | 10 12 | 11 10 |

## Reading responses

(Arko Banerjee) Are there any bounds one can get on the variance of n-step methods versus the variance of 1-step methods? It seems intuitively that the former would be much more stable in its updates than the latter, since it incorporates so much more information.

(Ayush Bhattacharya) I was slightly confused about why exactly n-step TD prediction was so much more useful than MC methods or the single-step TD method. I assume that it depends on the situation, but the textbook skimmed over what exactly makes this method much more useful than the others. I understand how it works but some further explanation would clear up understanding issues.




Random walk example from Sutton & Barto (Fig 7.2 on p145).

A way to unify Monte Carlo and TD - but have to wait n steps to do an n-step update. Smaller n update faster, more effective data collection.

Bias-Variance Error Bounds for Temporal Difference Updates. Kearns & Singh, 2000

## Reading responses

Why is the n-step return guaranteed to have a better estimate of Vpi than Vt+n-1? (Yihan Bai, Sumaya Al-Bedaiwi, Dewayne Benson)

Where is the error reduction property of n-step methods derived from? I didn't fully understand where the intuition for this came from.(Laith Altarabishi, Daniil Kirsanov)

## Error Reduction Property

$$\max_s \left| \mathbb{E}_\pi[G_{t:t+n} | S_t = s] - v_\pi(s) \right| \;\leq\; \gamma^n \max_s \left| V_{t+n-1}(s) - v_\pi(s) \right|$$

Relies on Jensen's inequality which gives us the property:

$$|\mathbb{E}[X]| \leq \mathbb{E}[|X|]$$

https://ai.stackexchange.com/questions/9396/how-do-we-prove-the-n-step-return-error-reduction-property

## The control problem: n-step Sarsa

Let's construct an on-policy TD control method.

Previously: Sarsa -> one-step Sarsa or Sarsa(0)

## The control problem: n-step Sarsa

Let's construct an on-policy TD control method.

Previously: Sarsa -> one-step Sarsa or Sarsa(0)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big]$$

We redefine n-step returns in terms of estimated action-values:

## The control problem: n-step Sarsa

Let's construct an on-policy TD control method.

Previously: Sarsa -> one-step Sarsa or Sarsa(0)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big]$$

We redefine n-step returns in terms of estimated action-values:

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}), \quad n \geq 1, 0 \leq t < T - n$$

New update:

## The control problem: n-step Sarsa

Let's construct an on-policy TD control method.

Previously: Sarsa -> one-step Sarsa or Sarsa(0)

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t)\Big]$$

We redefine n-step returns in terms of estimated action-values:

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}), \quad n \geq 1, 0 \leq t < T - n$$

New update:

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \left[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)\right]$$

# Exercise

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \left[ G_{t:t+n} - Q_{t+n-1}(S_t, A_t) \right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}),$$

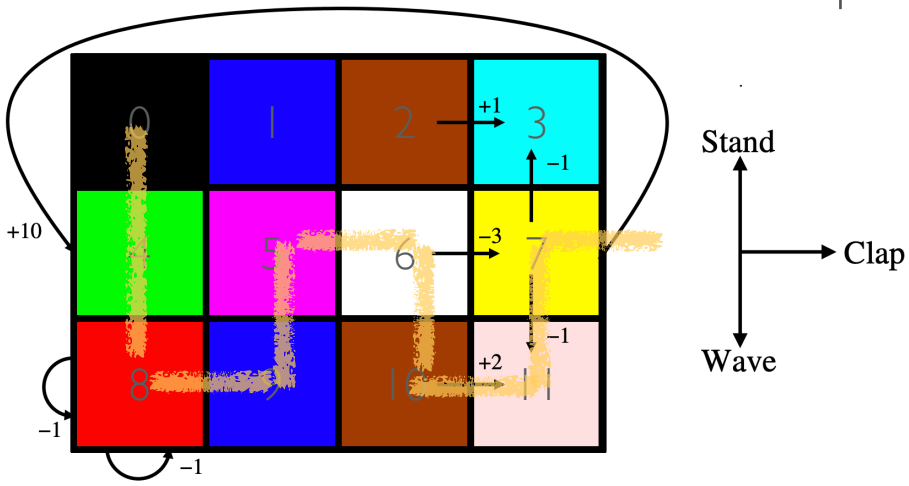
Gamma = 1, alpha = 0.5

Episode:

State, Action, Reward

0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

2-step:




| | Stand | Clap | Wave |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 6 | 0 | 0 | 0 |
| 7 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 |
| 10 | 0 | 0 | 0 |
| 11 | 0 | 0 | 0 |

# Exercise

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \left[ G_{t:t+n} - Q_{t+n-1}(S_t, A_t) \right]$$

$$G_{t:t+n} \doteq R_{t+1} + \gamma R_{t+2} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n Q_{t+n-1}(S_{t+n}, A_{t+n}),$$

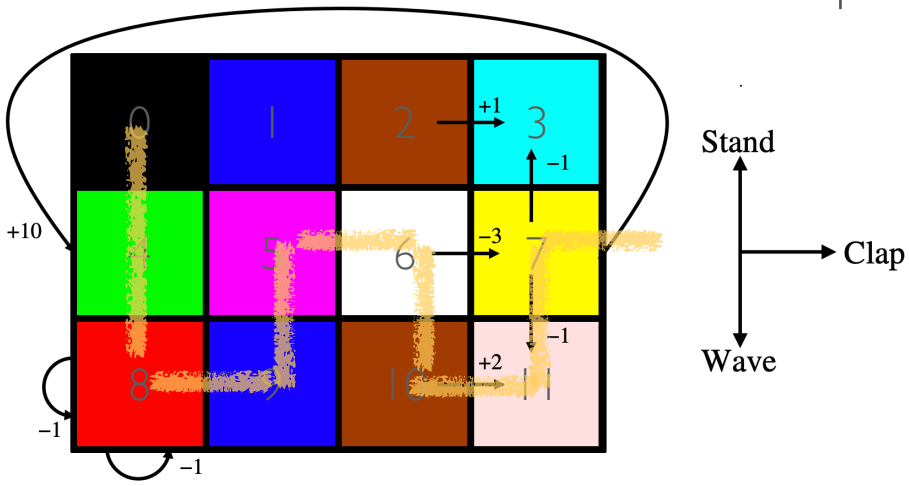
Gamma = 1, alpha = 0.5

Episode:

State, Action, Reward

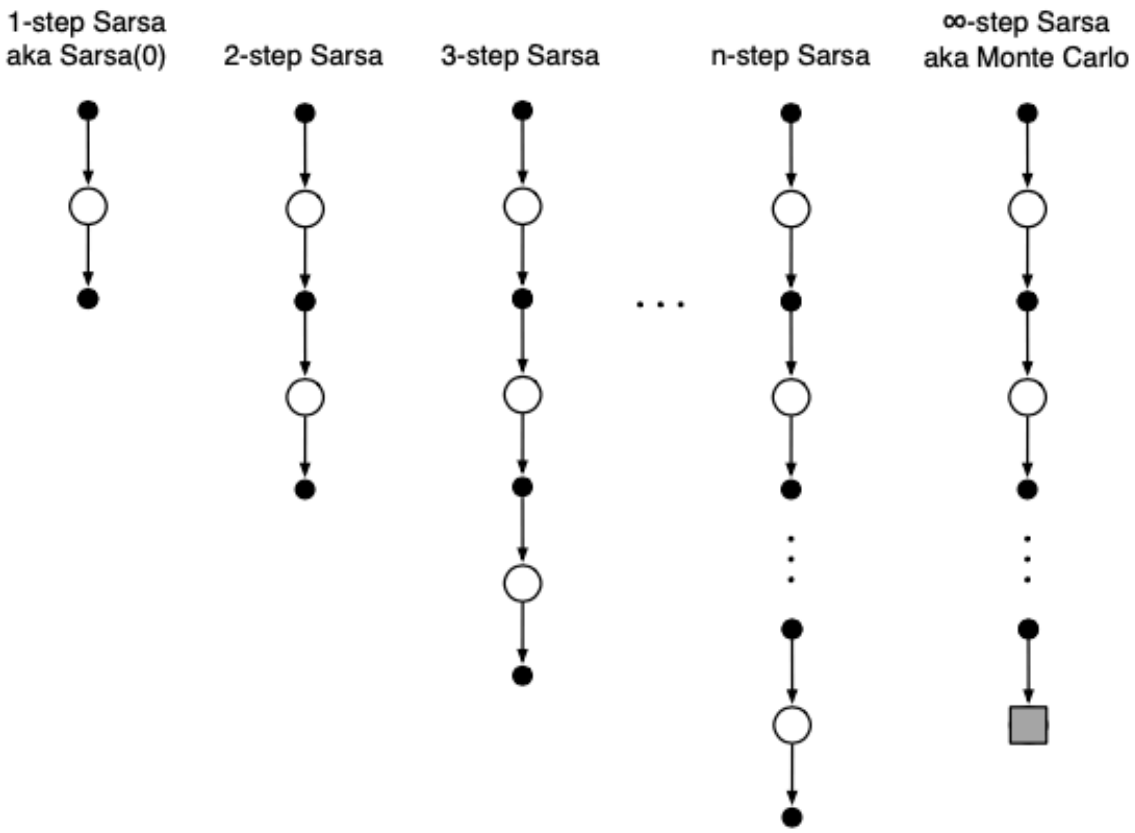
0, Wave, 0
4, Wave, 0
8, Clap, 0
9, Stand, 0
5, Clap, 0
6, Wave, 0
10, Clap, 2
11, Stand, 0
7, Clap, 10

2-step:




| | Stand | Clap | Wave |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 |
| 3 | 0 | 0 | 0 |
| 4 | 0 | 0 | 0 |
| 5 | 0 | 0 | 0 |
| 6 | 0 | 0 | 1 |
| 7 | 0 | 5 | 0 |
| 8 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 |
| 10 | 0 | 1 | 0 |
| 11 | 5 | 0 | 0 |

# Backup diagrams for n-step Sarsa



# What about expected Sarsa?

Expected Sarsa update:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big]$$
$$\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

N-step Expected Sarsa update:

## What about expected Sarsa?

Expected Sarsa update:

$$Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \mathbb{E}[Q(S_{t+1}, A_{t+1}) \mid S_{t+1}] - Q(S_t, A_t)\Big]$$
$$\leftarrow Q(S_t, A_t) + \alpha\Big[R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a) - Q(S_t, A_t)\Big]$$

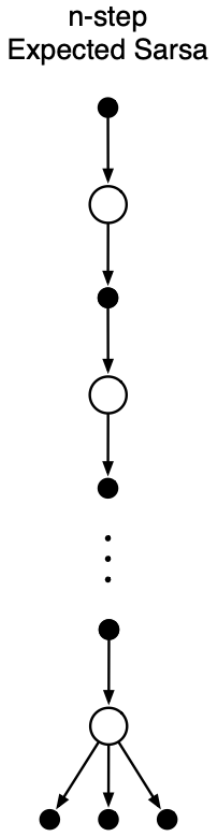
N-step Expected Sarsa update:

$$G_{t:t+n} \doteq R_{t+1} + \cdots + \gamma^{n-1} R_{t+n} + \gamma^n \sum \pi(a|S_{t+n}) Q_{t+n-1}(S_{t+n}, a)$$

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \left[G_{t:t+n} - Q_{t+n-1}(S_t, A_t)\right]$$

# Backup diagram for n-step expected Sarsa?

# Backup diagram for n-step expected Sarsa?



## Combining n-step and off-policy: n-step off-policy learning by importance sampling

Use our old friend, the importance sampling ratio.

How does computing this ratio change from the Monte Carlo version?

## Combining n-step and off-policy: n-step off-policy learning by importance sampling

Use our old friend, the importance sampling ratio.

How does computing this ratio change from the Monte Carlo version?

$$\rho_{t:h} \doteq \prod_{k=t}^{\min(h,T-1)} \frac{\pi(A_k|S_k)}{b(A_k|S_k)}$$

We only care about computing the probability of the next n steps, not all the way to the end of the episode.

## Combining n-step and off-policy: n-step off-policy learning by importance sampling

What's our new off-policy, n-step TD update?

## Combining n-step and off-policy: n-step off-policy learning by importance sampling

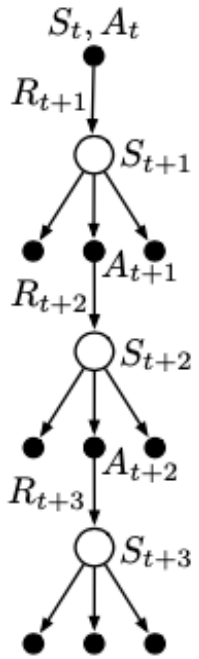
What's our new off-policy, n-step TD update?

$$Q_{t+n}(S_t, A_t) \doteq Q_{t+n-1}(S_t, A_t) + \alpha \rho_{t+1:t+n-1} \left[ G_{t:t+n} - Q_{t+n-1}(S_t, A_t) \right]$$

## Combining n-step and off-policy: n-step off-policy learning *without* importance sampling

1-step (Expected Sarsa)

$$G_{t:t+1} \doteq R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q_t(S_{t+1}, a), \qquad t < T-1$$



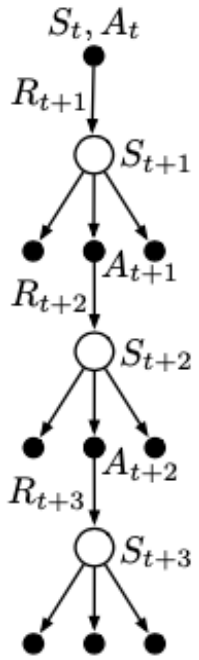

the 3-step tree-backup update

## Combining n-step and off-policy: n-step off-policy learning *without* importance sampling

1-step (Expected Sarsa)

$$G_{t:t+1} \doteq R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q_t(S_{t+1}, a), \qquad t < T-1$$

2nd-step tree-backup return

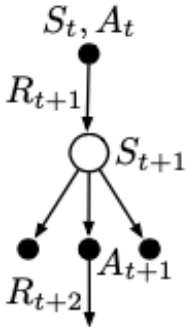



the 3-step tree-backup update

## Combining n-step and off-policy: n-step off-policy learning *without* importance sampling

1-step (Expected Sarsa)

$$G_{t:t+1} \doteq R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q_t(S_{t+1}, a), \qquad t < T-1$$




2nd-step tree-backup return

$$G_{t:t+2} \doteq R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) \Big( R_{t+2} + \gamma \sum_a \pi(a|S_{t+2}) Q_{t+1}(S_{t+2}, a) \Big)$$

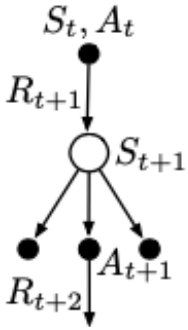
$$= R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) \; + \; \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+2}, \qquad t < T-2.$$

the 3-step tree-backup update

## Combining n-step and off-policy: n-step off-policy learning *without* importance sampling

1-step (Expected Sarsa)

$$G_{t:t+1} \doteq R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q_t(S_{t+1}, a), \qquad t < T-1$$




2nd-step

$$G_{t:t+2} \doteq R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) \Big( R_{t+2} + \gamma \sum_a \pi(a|S_{t+2}) Q_{t+1}(S_{t+2}, a) \Big)$$

$$= R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) \; + \; \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+2}, \qquad t < T-2.$$

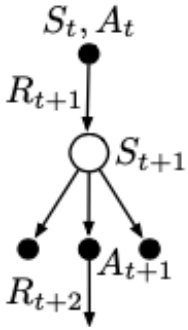
n-step

the 3-step tree-backup update

## Combining n-step and off-policy: n-step off-policy learning *without* importance sampling

1-step (Expected Sarsa)

$$G_{t:t+1} \doteq R_{t+1} + \gamma \sum \pi(a|S_{t+1}) Q_t(S_{t+1}, a), \qquad t < T-1$$

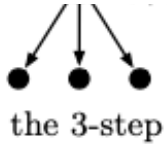



2nd-step

$$G_{t:t+2} \doteq R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) \Big( R_{t+2} + \gamma \sum_{a} \pi(a|S_{t+2}) Q_{t+1}(S_{t+2}, a) \Big)$$

$$= R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+1}(S_{t+1}, a) \; + \; \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+2}, \qquad t < T-2.$$

n-step




$$G_{t:t+n} \doteq R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q_{t+n-1}(S_{t+1}, a) \; + \; \gamma \pi(A_{t+1}|S_{t+1}) G_{t+1:t+n}, \;\; t+1 < T, n > 1$$

## Reading responses

(Cashel Fitzgerald) Why do you use only the actions not taken by the policy to estimate value in the n-step tree-backup algorithm?

(Jiaheng Hu) Why is the tree-backup algorithm not so popular in modern reinforcement learning?

## Reading responses

(Cashel Fitzgerald) Why do you use only the actions not taken by the policy to estimate value in the n-step tree-backup algorithm?
We are using the action taken to estimate the value, but we are using a "rolled out" version of its estimate.

(Jiaheng Hu) Why is the tree-backup algorithm not so popular in modern reinforcement learning?
N-step methods not common — in practice we always use 1-step Q learning because it's simplest and works best in deterministic environments.

## Reading responses

(Nicolas Hsu) Is there a formal definition for what information backup diagrams convey about an algorithm's update operation? The textbook said in chapter 3 that they diagram "relationships that form the basis of the update or backup operations that are at the heart of reinforcement learning methods", but after seeing so many backup diagrams in this book/class/chapter 7 I've realized that I don't really feel like I understand backup diagrams, so it would be nice to have the idea formalized. Or maybe I'm asking for more rigor than is necessary?

## Reading responses

(Nicolas Hsu) Is there a formal definition for what information backup diagrams convey about an algorithm's update operation? The textbook said in chapter 3 that they diagram "relationships that form the basis of the update or backup operations that are at the heart of reinforcement learning methods", but after seeing so many backup diagrams in this book/class/chapter 7 I've realized that I don't really feel like I understand backup diagrams, so it would be nice to have the idea formalized. Or maybe I'm asking for more rigor than is necessary?

I don't think it's a notion of "rigor." It's just a way to visualize all the algorithms we're learning in a single unifying format.

## Reading responses

(Harshal Bharatia) Page 152: When do you use off policy with importance sampling and without importance sampling?

(Nidhi Dubagunta) Is there a reason to want to use the tree-back up algorithm instead of importance sampling? It doesn't seem less computationally expensive than the algorithm discussed in section 7.5 and there were no performance comparisons between the algorithms--does this have performance benefits (and if so, why?)

## Reading responses

(Harshal Bharatia) Page 152: When do you use off policy with importance sampling and without importance sampling?
If you're computing the on-policy expectation over all possible trajectories you don't need importance sampling.

(Nidhi Dubagunta) Is there a reason to want to use the tree-back up algorithm instead of importance sampling? It doesn't seem less computationally expensive than the algorithm discussed in section 7.5 and there were no performance comparisons between the algorithms--does this have performance benefits (and if so, why?)
More computationally expensive but lower variance because we are computing the expectation over all possible trajectories, while with importance sampling we are computing the probability of a single trajectory.

## Reading responses

(Vaishnav Bipin) - Why doesn't Q-learning, as an off-policy algorithm, require an importance ratio to be factored into the update?

(Michelle Ding) [Ch 7] One thing I learned was that you can make an off-policy learning algorithm on Sarsa (as detailed by the algorithm on page 149). I thought this is was an interesting idea since Sarsa is traditionally an on-policy method. So, I am wondering what makes the algorithm different from Q-learning given that there is a behavioral policy b being applied to Sarsa.

# Reading responses

(Vaishnav Bipin) - Why doesn't Q-learning, as an off-policy algorithm, require an importance ratio to be factored into the update?
Because it is only a one-step update of a state-action pair, we don't care how likely the action we're updating is under our current policy.

(Michelle Ding) [Ch 7] One thing I learned was that you can make an off-policy learning algorithm on Sarsa (as detailed by the algorithm on page 149). I thought this is was an interesting idea since Sarsa is traditionally an on-policy method. So, I am wondering what makes the algorithm different from Q-learning given that there is a behavioral policy b being applied to Sarsa.
N-step Sarsa is still not using a max in the Q(S, A) update:

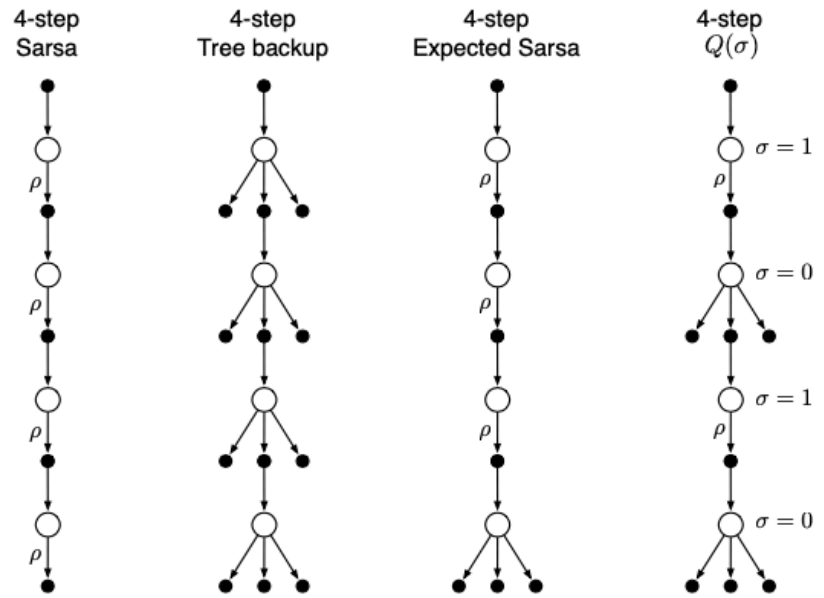
$$Q(S_\tau, A_\tau) \leftarrow Q(S_\tau, A_\tau) + \alpha\rho\left[G - Q(S_\tau, A_\tau)\right]$$

vs

$$Q(S, A) \leftarrow Q(S, A) + \alpha\left[R + \gamma \max_a Q(S', a) - Q(S, A)\right]$$

# Unifying algorithm: n-step Q

Choose which of the three methods to use at every step.
One step further: can implement a *continuous* variation between sampling and expectation




Pure expectation

Full sampling

## Reading responses

(Brian Kim) What is the intuition behind creating a unifying algorithm? In what cases would one use the 3 different algorithms over n-step Q(sigma)?

(Jorge Diaz) For the n-step Q(sigma) algorithm, how do we select at each state whether to sample or not? How is the parameter sigma selected? Why and when would we prefer to sample over simply using expectations?

# Reading responses

(Brian Kim) What is the intuition behind creating a unifying algorithm? In what cases would one use the 3 different algorithms over n-step Q(sigma)?
Simplifying - can transition across all 3 using a single hyper parameter.

(Jorge Diaz) For the n-step Q(sigma) algorithm, how do we select at each state whether to sample or not? How is the parameter sigma selected? Why and when would we prefer to sample over simply using expectations?
Not clear! So far results are mostly empirical. See paper:
Multi-Step Reinforcement Learning: A Unifying Algorithm, Asis et al., 2018.
https://arxiv.org/pdf/1703.01327.pdf

# Final Logistics

Next lecture:
Chapter 8: Planning and Learning with Tabular Methods

Chapter 7+8 Homework on edX **due Friday 11:59PM CST**
Project proposal due 11:59PM Thursday March 7th (tentative)

Office hours:
**Mon:** Michael 1-2PM GDC Basement TA Station #5, Caroline 5-6PM
**Tues:** Peter 1:10-2PM GDC 3.508
**Wed:** Amy 2-3PM EER 6.878
**Thurs:** Haoran 11-12PM; Siddhant 5-6PM
**Fri:** Shuozhe 4-5PM