Tree-backup and Q(σ)

Sutton & Barto - 7.5, 7.6, 12.10

KRIS DE ASIS

Overview

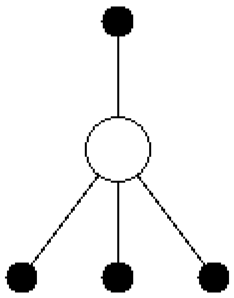
- n-step Tree-backup
- n-step Q(σ)
- n-step to Traces
- TB(λ) and Q(σ, λ)

One-step Expected Sarsa

Computes expectation under target policy directly, can do off-policy learning **without** importance sampling

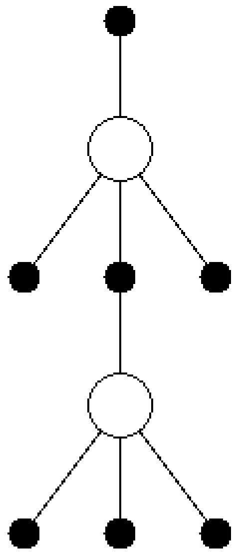
TD Target:

$$\hat{G}_t = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a)$$

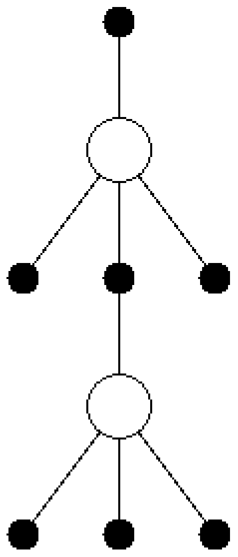


n-step Tree-backup

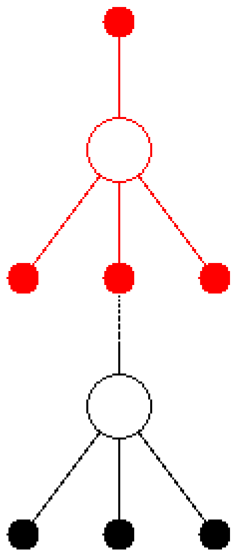
Tree-backup is a multi-step generalization of Expected Sarsa which computes this expectation under the target policy at **every step**



n-step Tree-backup’s Target



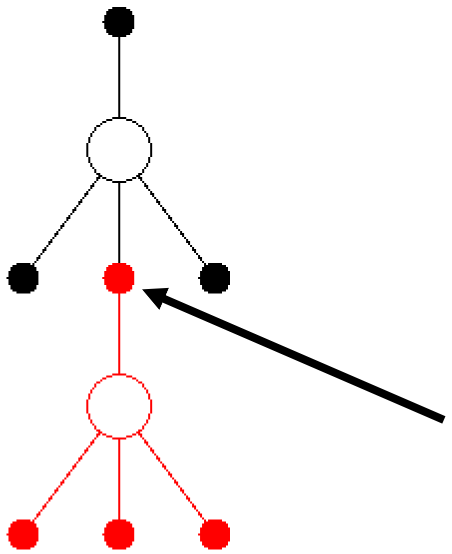
n-step Tree-backup's Target



$$\hat{G}_t = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1})Q(S_{t+1}, a)$$

Expected Sarsa

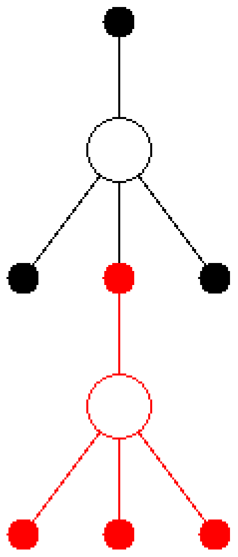
n-step Tree-backup's Target



$$\hat{G}_t = R_{t+1} + \gamma \sum_a \pi(a|S_{t+1}) Q(S_{t+1}, a)$$

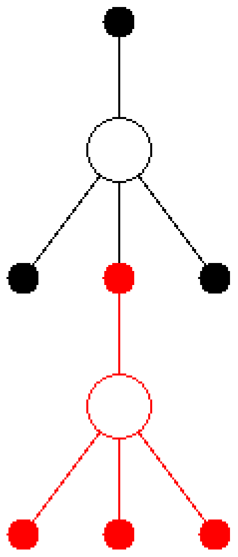
Except we know the outcome of the action we took!

n-step Tree-backup's Target



$$\hat{G}_t = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) Q(S_{t+1}, A_{t+1})$$

n-step Tree-backup's Target

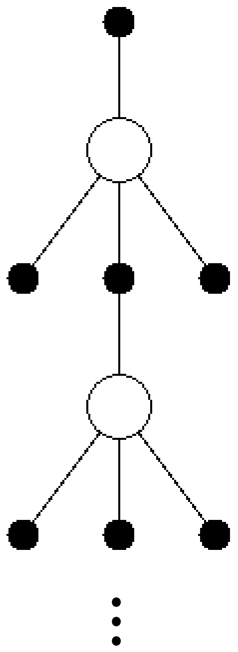


$$\hat{G}_t = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) \hat{Q}(S_{t+1}, A_{t+1})$$

$$\hat{Q}(S_{t+1}, A_{t+1}) = R_{t+2} + \gamma \sum_{a'} \pi(a'|S_{t+2}) Q(S_{t+2}, a')$$

2-step Tree-backup

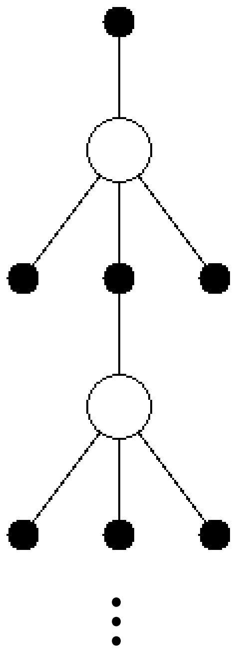
n-step Tree-backup's Target



$$\hat{G}_{t:h} = R_{t+1} + \gamma \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q(S_{t+1}, a) + \gamma \pi(A_{t+1}|S_{t+1}) \hat{G}_{t+1:h}$$

$$\hat{G}_{h:h} = Q(S_h, A_h)$$

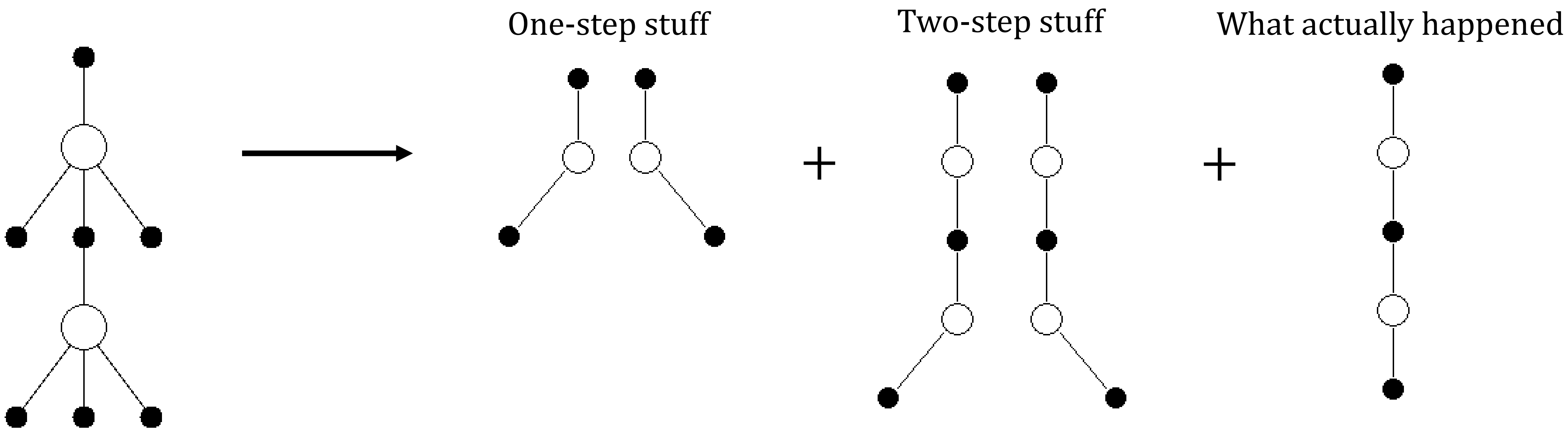
n-step Tree-backup



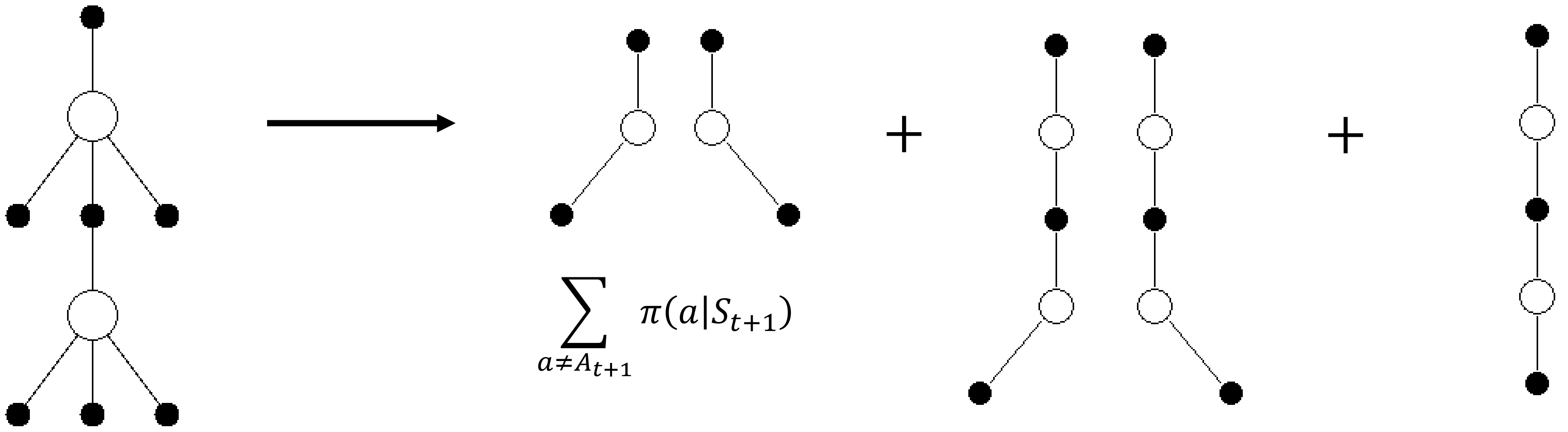
Allows for **multi-step** off-policy learning **without** importance sampling

Can be seen as bootstrapping off of what **didn't happen** to reduce the variance of sampled information

n-step Tree-backup



n-step Tree-backup

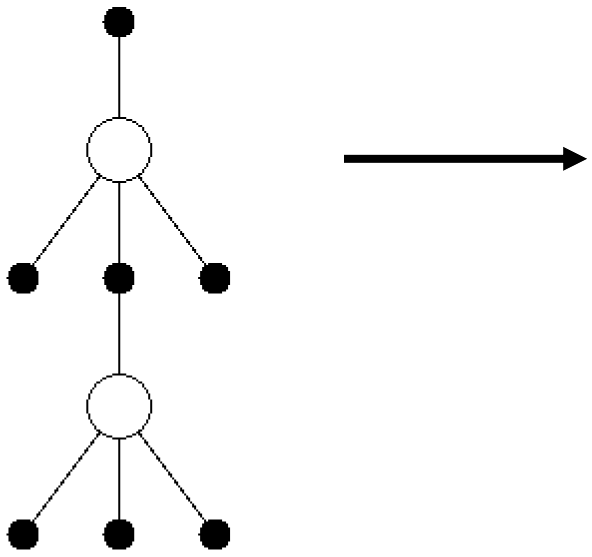


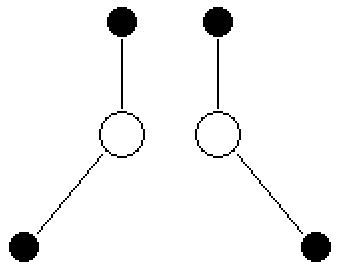
$$\sum_{a \neq A_{t+1}} \pi(a|S_{t+1})$$

$$\pi(A_{t+1}|S_{t+1}) \sum_{a \neq A_{t+2}} \pi(a|S_{t+2})$$

$$\pi(A_{t+1}|S_{t+1})\pi(A_{t+2}|S_{t+2})$$

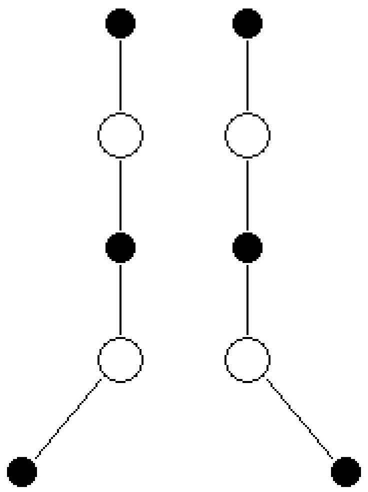
n-step Tree-backup



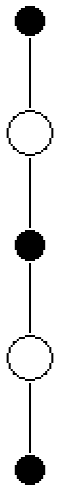


$\left(1 - \pi(A_{t+1}|S_{t+1})\right)$

+



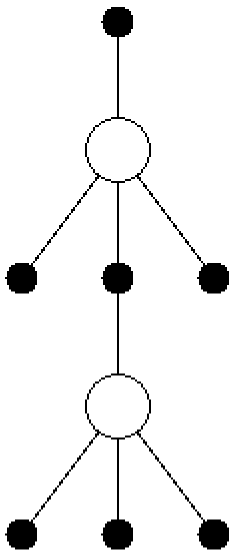
+

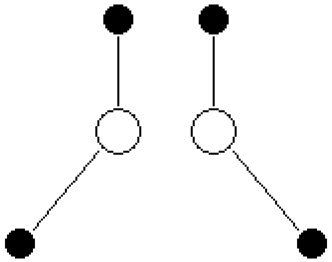


$\pi(A_{t+1}|S_{t+1})\left(1 - \pi(A_{t+2}|S_{t+2})\right)$

$\pi(A_{t+1}|S_{t+1})\pi(A_{t+2}|S_{t+2})$

n-step Tree-backup

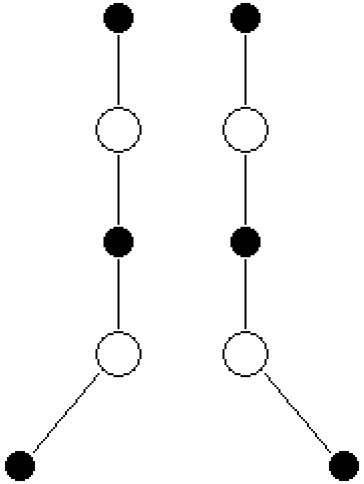




$\big(1-\pi(A_{t+1}|S_{t+1})\big)$

$(1-\lambda)$

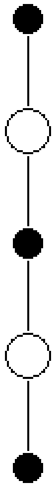
$+$



$\pi(A_{t+1}|S_{t+1})\big(1-\pi(A_{t+2}|S_{t+2})\big)$

$\lambda(1-\lambda)$

$+$



$\pi(A_{t+1}|S_{t+1})\pi(A_{t+2}|S_{t+2})$

λ^2

n-step Truncated λ-return!

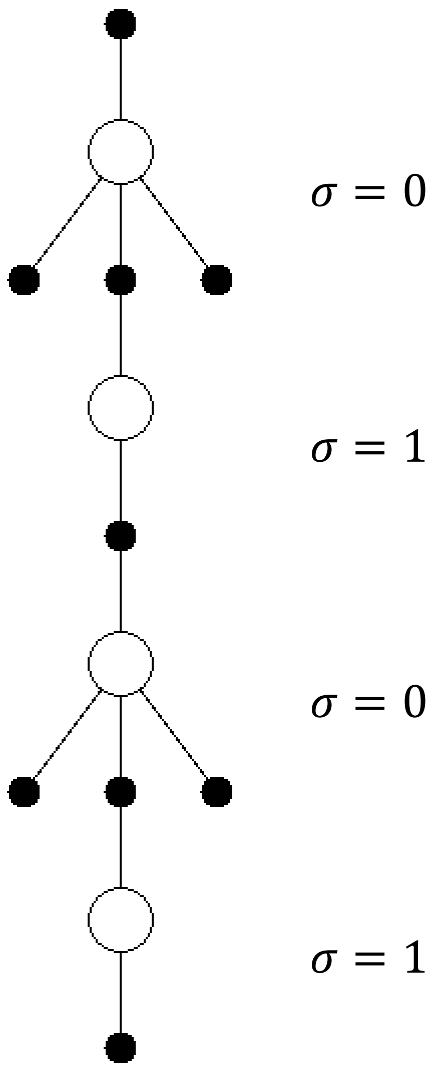
n-step Q(σ)

Decide on a per-decision basis whether to:

- Use sampled information ($\sigma = 1$)
- Compute an expectation ($\sigma = 0$)
- A mix of the two ($0 < \sigma < 1$)

Unifies several existing action-value multi-step TD methods:

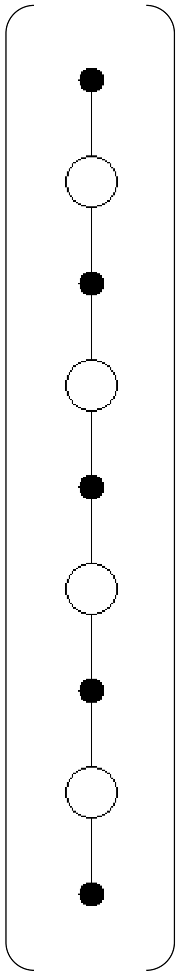
- n-step Sarsa
- n-step Expected Sarsa
- n-step Tree-backup



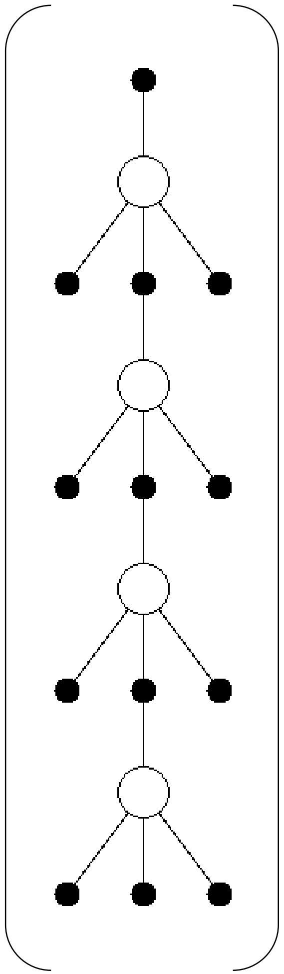
n-step Q(σ)

Constant values of σ:

σ



$+ (1 - \sigma)$



But σ doesn't have to be constant!

n-step Q(σ)'s Target

$$\hat{G}_{t:h} = R_{t+1} + \gamma(\sigma_{t+1}(\text{Sarsa}) + (1 - \sigma_{t+1})(\text{TB}))$$

$\rho_{t+1} G_{t+1:h}$

$\pi(A_{t+1}|S_{t+1}) G_{t+1:h} + \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q(S_{t+1}, a)$

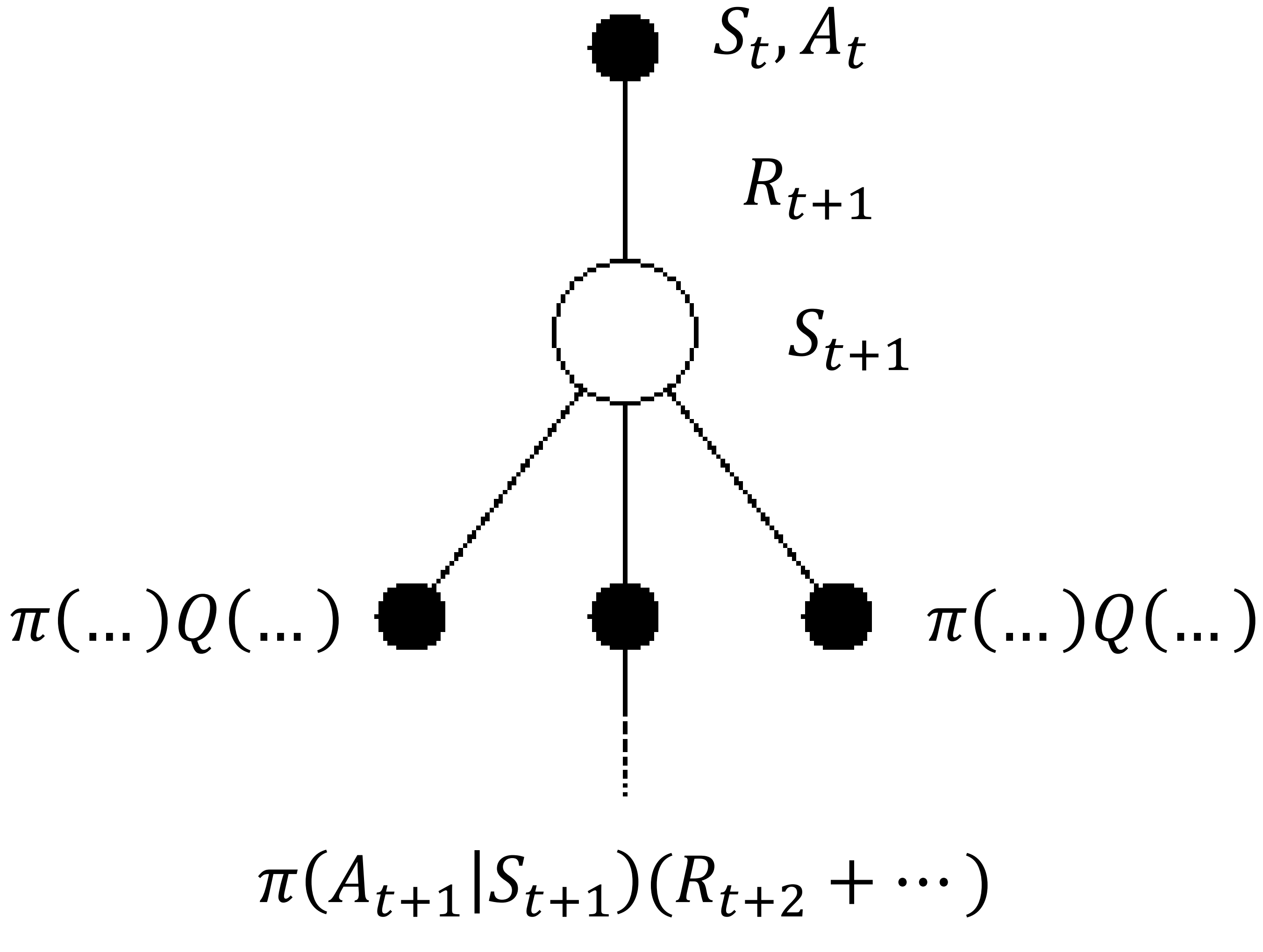
$$\hat{G}_{h:h} = Q(S_h, A_h)$$

Why unify?

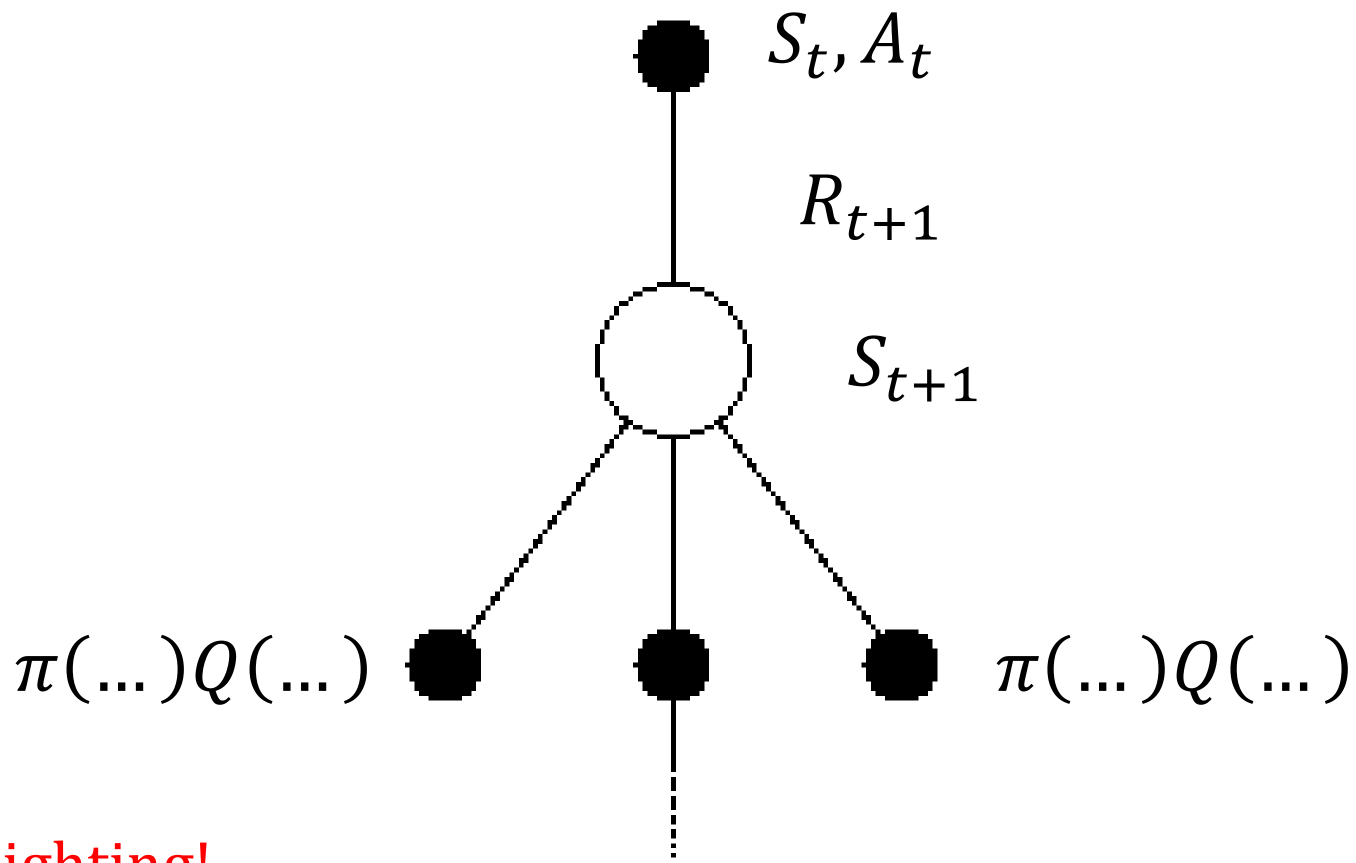
In the one-step case, Expected Sarsa generally outperforms Sarsa

How does n-step Tree-backup compare to n-step Sarsa?

Why unify?



Why unify?



R_{t+2} has a lower weighting!

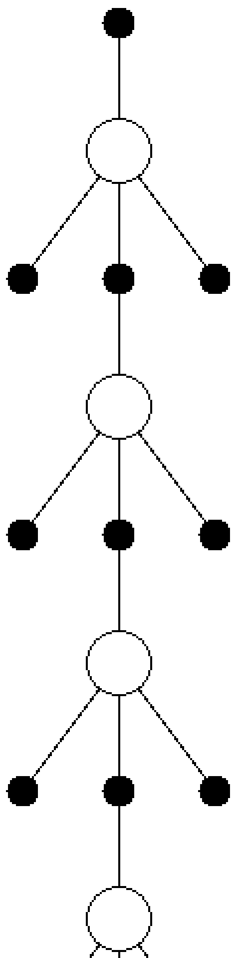
$$\pi(A_{t+1}|S_{t+1})(R_{t+2} + \cdots)$$

Why unify?

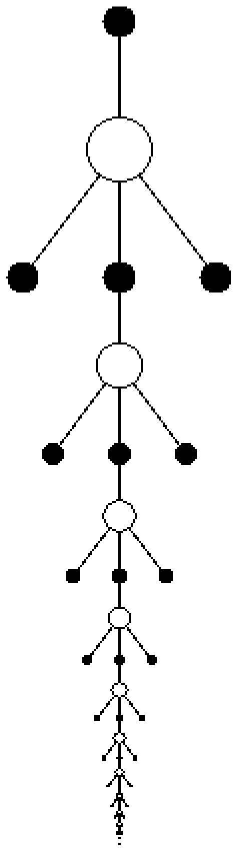
Reward	Weight
R_{t+1}	1
R_{t+2}	$\pi(A_{t+1} \vert S_{t+1})$
R_{t+3}	$\pi(A_{t+1} \vert S_{t+1})\pi(A_{t+2} \vert S_{t+2})$
R_{t+4}	$\pi(A_{t+1} \vert S_{t+1})\pi(A_{t+2} \vert S_{t+2})\pi(A_{t+3} \vert S_{t+3})$
$\vdots$	$\vdots$

Why unify?

Expectations



Reality



Why unify?

n-step Tree-backup has considerably low variance, but is more biased than n-step Sarsa (for the same n)

It gives less weight to the sampled rewards, and compensates by giving additional weight to bootstrapping (in the direction of **actions not taken**)

n-step Q(σ) can be viewed as a way to address this bias by adjusting the weighting given to sampled information

Dynamic σ

When value estimates are poor, it's better to use more sampled information

As estimates get better, can leverage bootstrapping to get targets with lower variance

Simple heuristic- start with $\sigma = 1$ (full sampling) and decay towards $\sigma = 0$ (no sampling- increased bootstrapping through computing expectations)

19-state Random Walk

A 1×19 grid world with deterministic 2-directional movement

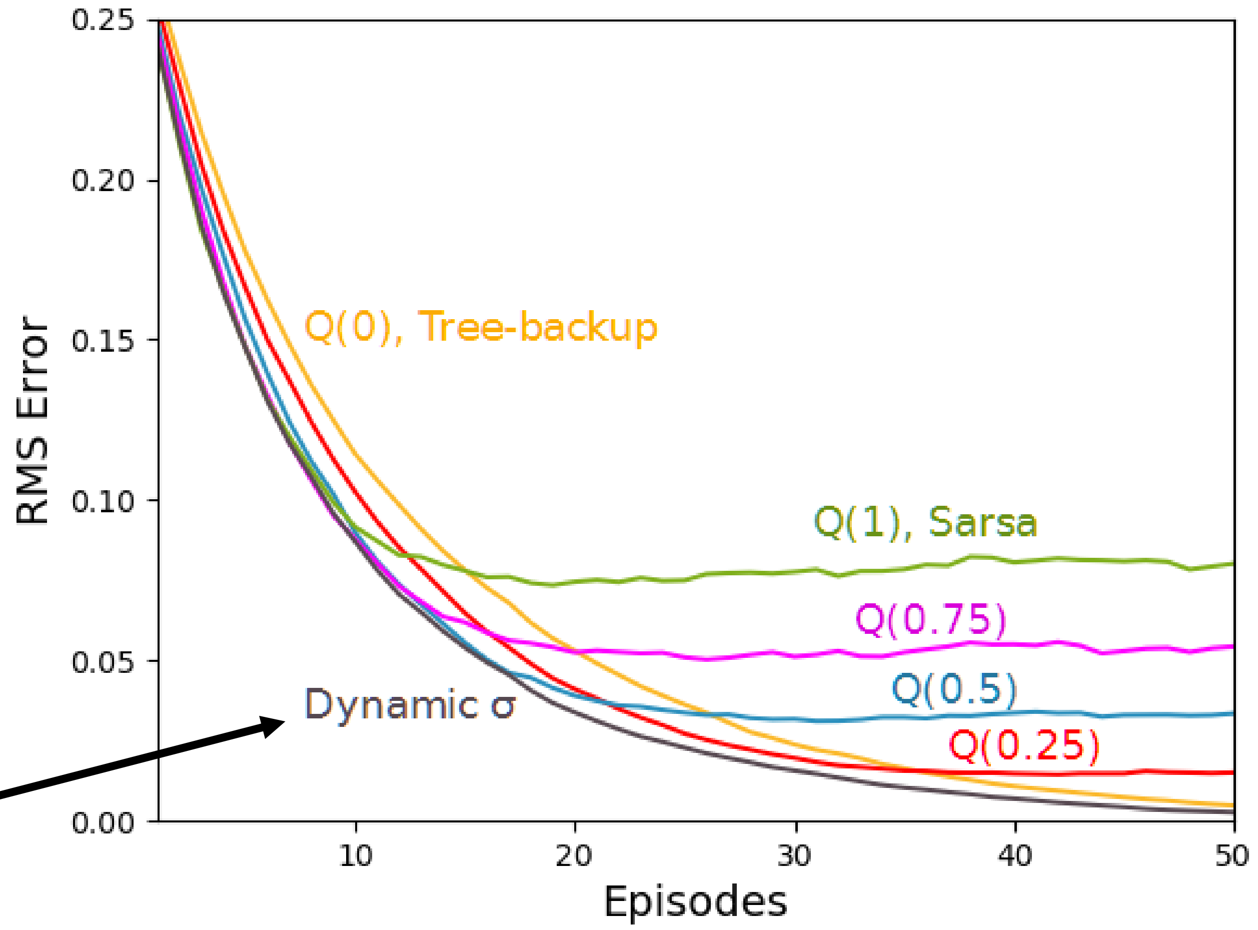
Terminal transitions on each end with rewards -1 and +1, all other transitions have a reward of 0.

On-policy policy evaluation with an equiprobable random policy

19-state Random Walk

3-step Q(σ)

Showing best constant step size in terms of RMS error after 50 episodes



Decaying σ after each episode from 1 to 0

n-step Q(σ)

σ controls a trade-off between initial and final performance- more sampling results in quicker learning but larger variance, and vice versa

n (or λ) tunes the degree of bootstrapping in the direction of **actions taken**, σ tunes the degree of bootstrapping in the direction of **actions not taken**

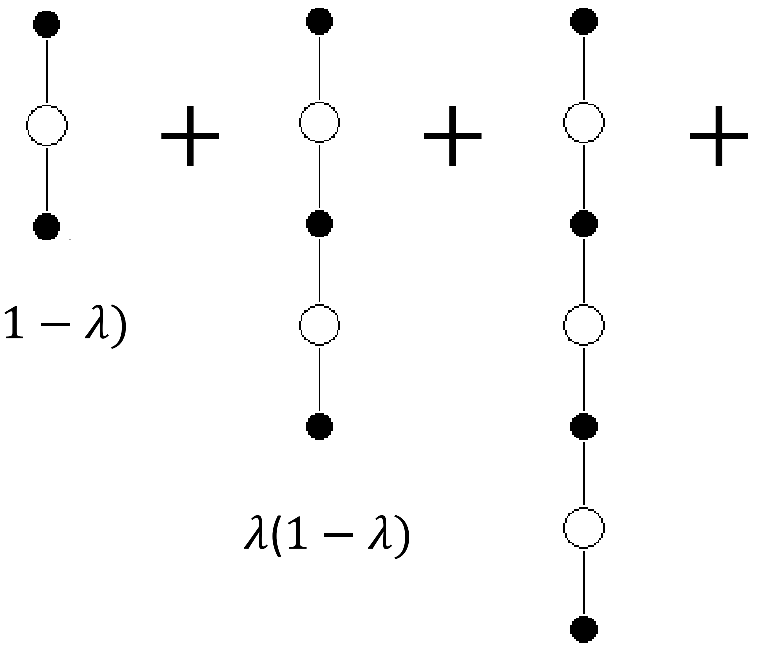
From n-step to traces

λ is often referred to as a **degree of bootstrapping**, but can also be viewed as the **probability of (not) bootstrapping** in a stochastic process

With probability $1 - \lambda$, we bootstrap off of current estimates. With probability λ, we continue to the next reward in the return

From n-step to traces

On-policy Sarsa(λ):



$(1-\lambda)$ $\quad \lambda(1-\lambda)$ $\quad \lambda^2(1-\lambda)$ $\quad \ldots$

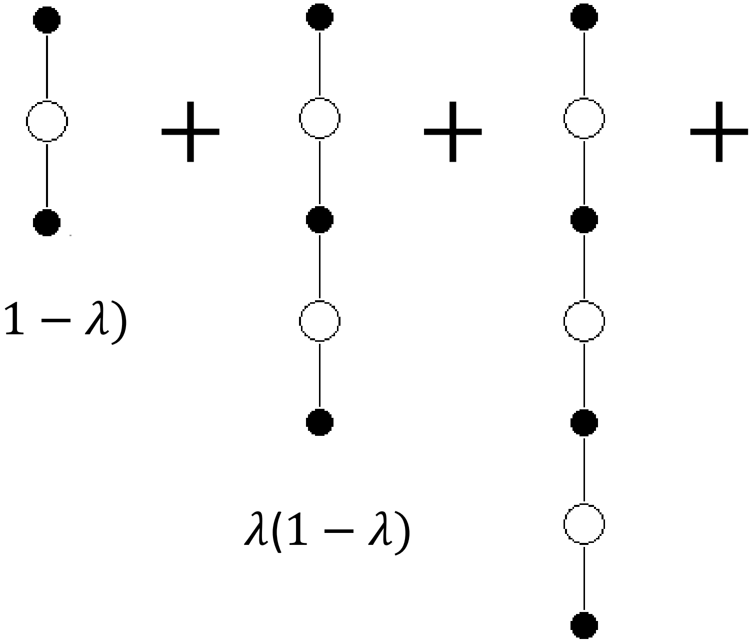
$$\hat{G}_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty} \lambda^{n-1}\hat{G}_{t:t+n}$$

$$\Downarrow$$

$$\hat{G}_t^\lambda =$$
$$(1-\lambda)\big(R_{t+1} + \gamma Q(S_{t+1}, A_{t+1})\big)$$
$$\lambda\big(R_{t+1} + \gamma G_{t+1}^\lambda\big)$$

From n-step to traces

On-policy Sarsa(λ):



$(1-\lambda)$ $\lambda(1-\lambda)$ $\lambda^2(1-\lambda)$ ...

$$\hat{G}_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\hat{G}_{t:t+n}$$

$$\hat{G}_t^\lambda = R_{t+1} + \gamma(\underbrace{\ldots}_{(1-\lambda)Q(S_{t+1},A_{t+1}) \; + \; \lambda G_{t+1}^\lambda})$$

Tree-backup(λ)

$$\hat{G}_{t:h} = R_{t+1} + \gamma \left(\pi(A_{t+1}|S_{t+1}) \hat{G}_{t+1:h} + \sum_{a \neq A_{t+1}} \pi(a|S_{t+1}) Q(S_{t+1}, a) \right)$$

$$\hat{G}_{h:h} = Q(S_h, A_h)$$

Tree-backup(λ)

$$\hat{G}_{t:h} = R_{t+1} + \gamma\left(\pi(A_{t+1}|S_{t+1})\hat{G}_{t+1:h} + \mathbb{E}_\pi[Q(S_{t+1},\cdot)] - \pi(A_{t+1}|S_{t+1})Q(S_{t+1},A_{t+1})\right)$$
$$\hat{G}_{h:h} = Q(S_h, A_h)$$

Tree-backup(λ)

$$\hat{G}_{t:h} = R_{t+1} + \gamma\big(\pi_{t+1}\hat{G}_{t+1:h} + \underbrace{\mathbb{E}_\pi[Q_{t+1}] - \pi_{t+1}Q_{t+1}}_{\text{Expectation correction- like a control variate!}}\big)$$
$$\hat{G}_{h:h} = Q_h$$

$$Q_t = Q(S_t, A_t)$$
$$\pi_t = \pi(A_t|S_t)$$

$\hat{G}_{t:h}$ with $\hat{G}_{t+1:h}$ outlines the continuation condition in the recursive λ-return

$\hat{G}_{t:h}$ with $\hat{G}_{h:h}$ substituted into $\hat{G}_{t+1:h}$ outlines the bootstrapping condition

Tree-backup(λ)

$$\hat{G}_{t:h} = R_{t+1} + \gamma\big(\pi_{t+1}\hat{G}_{t+1:h} + \mathbb{E}_\pi[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)$$
$$\hat{G}_{h:h} = Q_h$$

$$\hat{G}_t^\lambda =$$

$$(1-\lambda)\big(R_{t+1} + \gamma(\pi_{t+1}Q_{t+1} + \mathbb{E}[Q_{t+1}] - \pi_{t+1}Q_{t+1})\big) +$$
$$\lambda\left(R_{t+1} + \gamma\big(\pi_{t+1}\hat{G}_{t+1}^\lambda + \mathbb{E}[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)\right)$$

Easier to work with than:

$$\hat{G}_t^\lambda = (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}\hat{G}_{t:t+n}$$

Tree-backup(λ)

$$\hat{G}_t^{\lambda} = R_{t+1} + \gamma\left((1-\lambda)\mathbb{E}_{\pi}[Q_{t+1}] + \lambda\big(\pi_{t+1}G_{t+1}^{\lambda} + \mathbb{E}_{\pi}[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)\right)$$

$$\hat{G}_t^{\lambda} = \sum_{k=t}^{\infty} \underbrace{(R_{k+1} + \gamma\mathbb{E}_{\pi}[Q_{k+1}] - Q_k)}_{\text{One-step TD error}} \prod_{i=t+1}^{\mathrm{k}} \gamma\lambda\pi_i$$

Trace-decay rate

Tree-backup(λ)

$$\hat{G}_t^{\lambda} = R_{t+1} + \gamma\left((1-\lambda)\mathbb{E}_\pi[Q_{t+1}] + \lambda\big(\pi_{t+1}G_{t+1}^{\lambda} + \mathbb{E}_\pi[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)\right)$$

$$\hat{G}_t^{\lambda} = \sum_{k=t}^{\infty}(R_{k+1} + \gamma\mathbb{E}_\pi[Q_{k+1}] - Q_k)\prod_{i=t+1}^{\mathrm{k}}\gamma\lambda\pi_i$$

$$\pi_t \le \rho_t = \frac{\pi_t}{\mu_t}$$

For the same λ, Tree-backup's traces decay quicker- performs "shorter" backups

$$\hat{G}_t^{\lambda} = \sum_{k=t}^{\infty}(R_{k+1} + \gamma\rho_{k+1}Q_{k+1} - Q_k)\prod_{i=t+1}^{\mathrm{k}}\gamma\lambda\rho_i$$

Sarsa(λ)

$\mathbf{Q(\sigma, \lambda)}$

$$\hat{G}_t^\lambda = R_{t+1} + \gamma\big(\sigma_{t+1}(\text{Sarsa}) + (1-\sigma_{t+1})(\text{TB})\big)$$

$$(1-\lambda)\rho_{t+1}Q_{t+1} + \lambda\rho_{t+1}G_{t+1}^\lambda$$

$$(1-\lambda)\mathbb{E}_\pi[Q_{t+1}] + \lambda\big(\pi_{t+1}G_{t+1}^\lambda + \mathbb{E}_\pi[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)$$

$$\hat{G}_t^\lambda = \sum_{k=t}^{\infty} \underbrace{\left(R_{k+1} + \gamma(\sigma_{k+1}\rho_{k+1}Q_{k+1} + (1-\sigma_{k+1})\mathbb{E}_\pi[Q_{k+1}]) - Q_k\right)}_{\text{One-step TD error}} \prod_{i=t+1}^{k} \gamma\lambda(\sigma_i\rho_i + (1-\sigma_i)\pi_i)$$

Trace-decay rate

$\mathbf{Q(\sigma, \lambda)}$

$$\hat{G}_t^\lambda = R_{t+1} + \gamma\big(\sigma_{t+1}(\text{Sarsa}) + (1-\sigma_{t+1})(\text{TB})\big)$$

$$(1-\lambda)\rho_{t+1}Q_{t+1} + \lambda\rho_{t+1}G_{t+1}^\lambda$$

$$(1-\lambda)\mathbb{E}_\pi[Q_{t+1}] + \lambda\big(\pi_{t+1}G_{t+1}^\lambda + \mathbb{E}_\pi[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)$$

$$\hat{G}_t^\lambda = \sum_{k=t}^{\infty}(R_{k+1} + \gamma(\sigma_{k+1}\rho_{k+1}Q_{k+1} + (1-\sigma_{k+1})\mathbb{E}_\pi[Q_{k+1}]) - Q_k)\prod_{i=t+1}^{k}\gamma\lambda(\sigma_i\rho_i + (1-\sigma_i)\pi_i)$$

Adjustable action-dependent trace decay rate through σ

CV Q(σ, λ)

$$\hat{G}_t^\lambda = R_{t+1} + \gamma\big(\sigma_{t+1}(\text{ACV Sarsa}) + (1-\sigma_{t+1})(\text{TB})\big)$$

$$(1-\lambda)\mathbb{E}_\pi[Q_{t+1}] + \lambda\big(\pi_{t+1}G_{t+1}^\lambda + \mathbb{E}_\pi[Q_{t+1}] - \pi_{t+1}Q_{t+1}\big)$$

$$(1-\lambda)\mathbb{E}_\pi[Q_{t+1}] + \lambda\big(\rho_{t+1}G_{t+1}^\lambda + \mathbb{E}_\pi[Q_{t+1}] - \rho_{t+1}Q_{t+1}\big)$$

$$\hat{G}_t^\lambda = \sum_{k=t}^{\infty} \underbrace{(R_{k+1} + \gamma\mathbb{E}_\pi[Q_{k+1}] - Q_k)}_{\text{One-step TD error}} \prod_{i=t+1}^{k} \gamma\lambda(\sigma_i\rho_i + (1-\sigma_i)\pi_i)$$

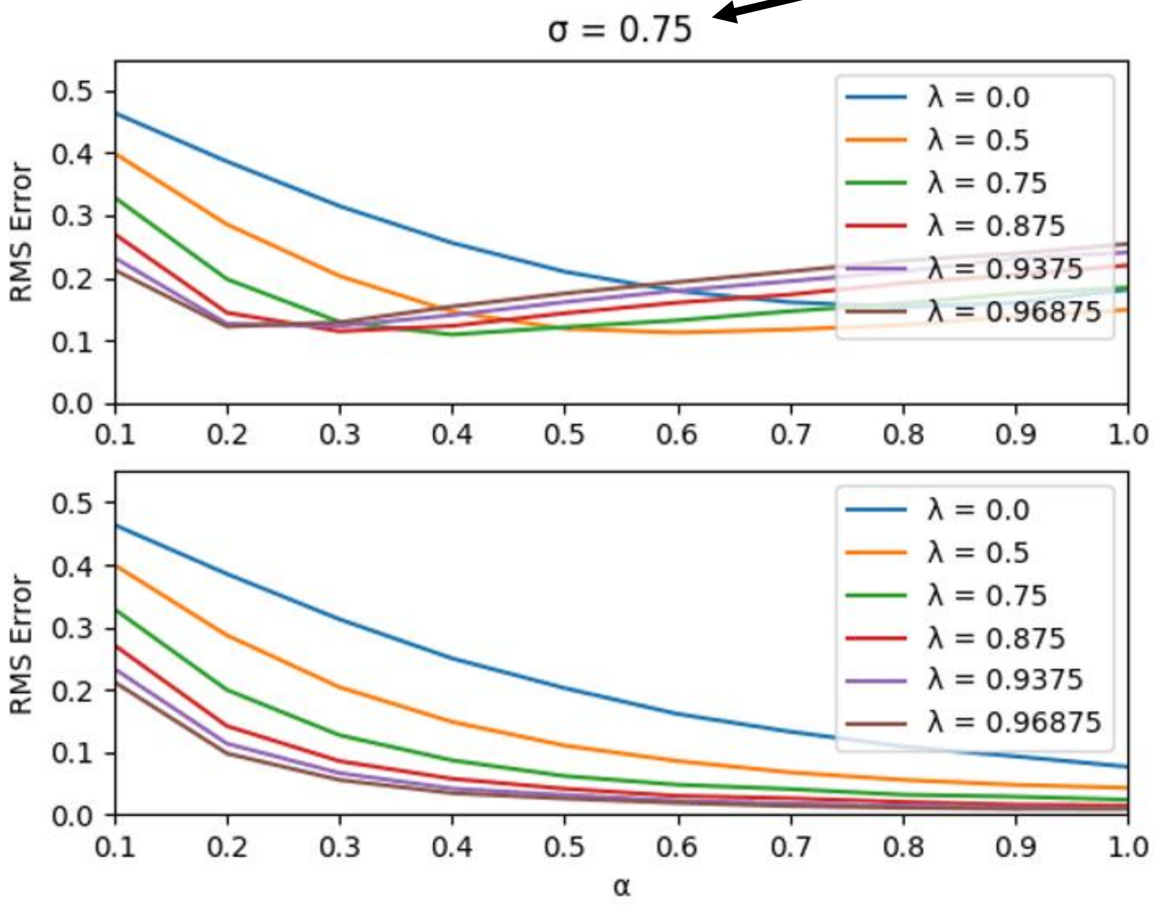
Trace-decay rate

19-State Random Walk

Lots of sampling

$Q(\sigma, \lambda)$:

CV $Q(\sigma, \lambda)$:



Questions?

- De Asis, K.*, Hernandez-Garcia, J. F.*, Holland G. Z.*, Sutton, R. S. (2018). Multi-step Reinforcement Learning: A Unifying Algorithm. *AAAI 2018*.