Retrace(λ)

Temporal Credit Assignment in Off-Policy Reinforcement Learning

Matteo Papini

28th November 2017

Before we start...

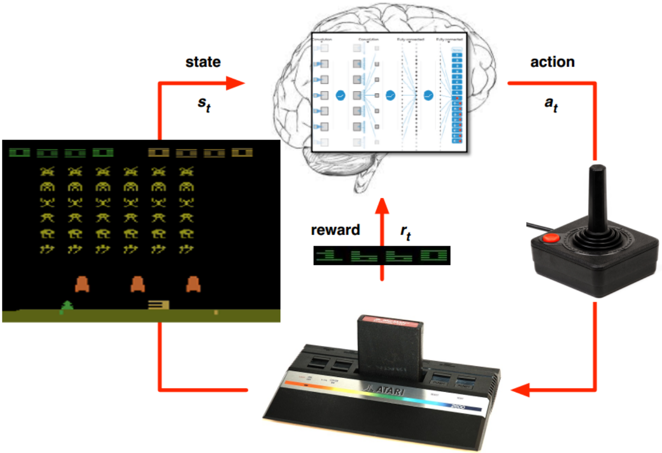
ACAI Summer School on Reinforcement Learning
Nieuwpoort, Belgium, 7-14 October 2017

Before we start...

Munos et al., "Safe and efficient off-policy reinforcement learning", NIPS 2016 [4]

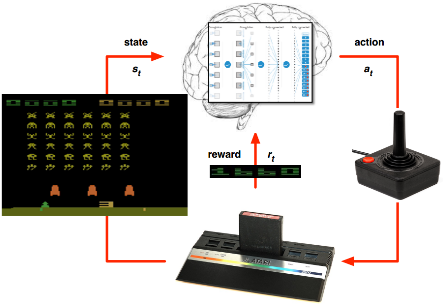
Syllabus

Reinforcement Learning

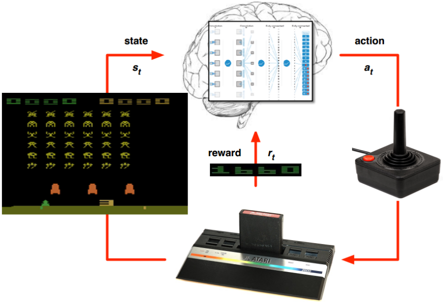


The Task

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho \rangle$$



- $\mathcal{S}$: state space
- $\mathcal{A}$: action space
- $\mathcal{P}$: transition probabilities
- $\mathcal{R}$: reward function
- γ: discount factor
- ρ: initial state distribution

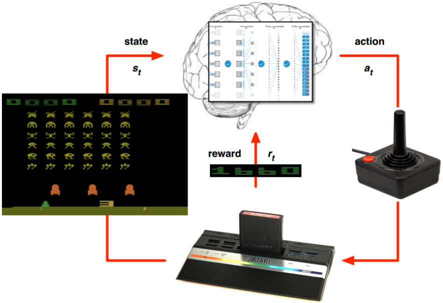


The Task

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho \rangle$$

The agent's policy

$$\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$$



The Task

$$\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma, \rho \rangle$$

The agent's policy

$$\pi : \mathcal{S} \mapsto \Delta(\mathcal{A})$$

The goal

$$\max \sum_{t \geq 0} \gamma^t r_t$$

Prediction and Control

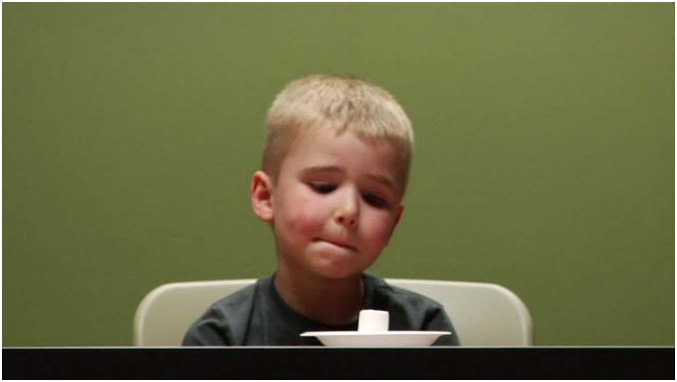
- **Prediction**: measure the performance of a given policy π
- **Control**: find the optimal policy π^*

Prediction and Control

- **Prediction**: measure the performance of a given policy π
- **Control**: find the optimal policy π^*

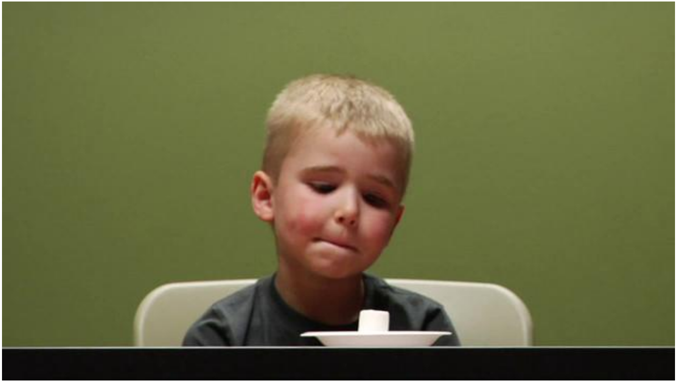
Temporal Credit Assignment

The effects of a choice may not be immediate



Temporal Credit Assignment

The effects of a choice may not be immediate
$\Longrightarrow$ delayed reward



Off-Policy Credit Assignment

- **Temporal Credit Assignment**: determine which actions, among a sequence of actions, are responsible for certain rewards
- **Off-Policy Learning**: evaluate/improve policy π while following policy μ
- **Off-Policy Credit Assignment**: how can I give credit to choices that are not actually made?

- **Temporal Credit Assignment**: determine which actions, among a sequence of actions, are responsible for certain rewards
- **Off-Policy Learning**: evaluate/improve policy π while following policy μ
- **Off-Policy Credit Assignment**: how can I give credit to choices that are not actually made?

Off-Policy Credit Assignment

- **Temporal Credit Assignment**: determine which actions, among a sequence of actions, are responsible for certain rewards
- **Off-Policy Learning**: evaluate/improve policy π while following policy μ
- **Off-Policy Credit Assignment**: how can I give credit to choices that are not actually made?

Value Function

Bellman expectation equation:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q^{\pi}(s', a') \right]$$

Bellman expectation **operator**:

$$\mathcal{T}^{\pi} Q = R(s,a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q(s', a') \right]$$

- Q^{π} is the unique fixed point of $\mathcal{T}^{\pi}$
- Contraction property: $\left\| \mathcal{T}^{\pi} Q - Q^{\pi} \right\|_{\infty} \leq \gamma \left\| Q - Q^{\pi} \right\|_{\infty}$

Value Function

Bellman expectation equation:

$$Q^{\pi}(s,a) = R(s,a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q^{\pi}(s',a') \right]$$

Bellman expectation **operator**:

$$\mathcal{T}^{\pi} Q = R(s,a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q(s',a') \right]$$

- Q^{π} is the unique fixed point of $\mathcal{T}^{\pi}$
- Contraction property: $\|\mathcal{T}^{\pi} Q - Q^{\pi}\|_{\infty} \leq \gamma \|Q - Q^{\pi}\|_{\infty}$

Value Function

Bellman expectation equation:

$$Q^\pi(s, a) = R(s, a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q^\pi(s', a') \right]$$

Bellman expectation **operator**:

$$\mathcal{T}^\pi Q = R(s, a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q(s', a') \right]$$

- Q^π is the unique fixed point of $\mathcal{T}^\pi$
- Contraction property: $\|\mathcal{T}^\pi Q - Q^\pi\|_\infty \leq \gamma \|Q - Q^\pi\|_\infty$

Value Function

Bellman expectation equation:

$$Q^\pi(s,a) = R(s,a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q^\pi(s',a') \right]$$

Bellman expectation **operator**:

$$\mathcal{T}^\pi Q = R(s,a) + \gamma \underset{\substack{s' \sim \mathcal{P} \\ a' \sim \pi}}{E} \left[Q(s',a') \right]$$

- Q^π is the unique fixed point of $\mathcal{T}^\pi$
- Contraction property: $\|\mathcal{T}^\pi Q - Q^\pi\|_\infty \leq \gamma \|Q - Q^\pi\|_\infty$

Optimal Value Function

Bellman optimality equation:

$$Q^*(s,a) = R(s,a) + \gamma \underset{s' \sim \mathcal{P}}{E}\left[\max_{a' \in \mathcal{A}} Q^*(s',a')\right]$$

Bellman optimality **operator**:

$$\mathcal{T}^* Q = R(s,a) + \gamma \underset{s' \sim \mathcal{P}}{E}\left[\max_{a' \in \mathcal{A}} Q^*(s',a')\right]$$

- Q^* is the unique fixed point of $\mathcal{T}^*$
- Contraction property: $\|\mathcal{T}^* Q - Q^*\|_\infty \leq \gamma \|Q - Q^*\|_\infty$

Prediction and Control

- **Prediction**: given policy π, compute Q^π
- **Control**: find π^* such that $Q^{\pi^*} = Q^*$

- Optimal **deterministic** policy:

$$\pi^*(s) = \pi_{\mathsf{greedy}(Q^*)}(s) \doteq \arg\max_{a' \in \mathcal{A}} Q^*(s, a')$$

- Maximization as a special case of expectation:

$$\max_{a' \in \mathcal{A}} Q(s, a') = \underset{a' \sim}{E} [Q(s, a')]$$

- Optimal **deterministic** policy:

$$\pi^*(s) = \pi_{\mathsf{greedy}(Q^*)}(s) \doteq \arg\max_{a' \in \mathcal{A}} Q^*(s, a')$$

- Maximization as a special case of expectation:

$$\max_{a' \in \mathcal{A}} Q^*(s, a') = \underset{a' \sim \pi^*}{E} \left[Q^*(s, a') \right]$$

- Optimal **deterministic** policy:

$$\pi^*(s) = \pi_{\mathsf{greedy}(Q^*)}(s) \doteq \arg\max_{a' \in \mathcal{A}} Q^*(s, a')$$

- Maximization as a special case of expectation:

$$\max_{a' \in \mathcal{A}} Q(s, a') = \underset{a' \sim \pi_{\mathsf{greedy}(Q)}}{E} \left[Q(s, a') \right]$$

- Idea: repeatedly apply $\mathcal{T}$ to Q
- Approximation: update Q towards a target
- Prediction: fixed π, keep updating Q
- Control: alternate value updates and policy updates

$$\pi_{k+1} = \pi_{\text{greedy}(Q_k)}$$

Temporal Difference Learning

- Idea: repeatedly apply $\mathcal{T}$ to Q
- Approximation: update Q towards a target

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \underbrace{\left[G_t - Q(s_t, a_t)\right]}_{\text{TD error } \delta_t}$$

- Prediction: fixed π, keep updating Q
- Control: alternate value updates and policy updates

$$\pi_{k+1} = \pi_{\text{greedy}(Q_k)}$$

Temporal Difference Learning

- Idea: repeatedly apply $\mathcal{T}$ to Q
- Approximation: update Q towards a target

$$\Delta Q(s_t, a_t) = \alpha \underbrace{\left[G_t - Q(s_t, a_t) \right]}_{\text{TD error } \delta_t}$$

- Prediction: fixed π, keep updating Q
- Control: alternate value updates and policy updates

$$\pi_{k+1} = \pi_{\text{greedy}(Q_k)}$$

Temporal Difference Learning

- Idea: repeatedly apply $\mathcal{T}$ to Q
- Approximation: update Q towards a target

$$\Delta Q(s_t, a_t) = \alpha \underbrace{\left[G_t - Q(s_t, a_t)\right]}_{\text{TD error } \delta_t}$$

- Prediction: fixed π, keep updating Q
- Control: alternate value updates and policy updates

$$\pi_{k+1} = \pi_{\text{greedy}(Q_k)}$$

- Idea: repeatedly apply $\mathcal{T}$ to Q
- Approximation: update Q towards a target

$$\Delta Q(s_t, a_t) = \alpha \underbrace{\left[G_t - Q(s_t, a_t) \right]}_{\text{TD error } \delta_t}$$

- Prediction: fixed π, keep updating Q
- Control: alternate value updates and policy updates

$$\pi_{k+1} = \pi_{\text{greedy}(Q_k)}$$

Temporal Difference Learning

- Idea: repeatedly apply $\mathcal{T}$ to Q
- Approximation: update Q towards a target

$$\Delta Q(s_t, a_t) = \alpha \underbrace{\left[G_t - Q(s_t, a_t)\right]}_{\text{TD error } \delta_t}$$

- Prediction: fixed π, keep updating Q
- Control: alternate value updates and policy updates

$$\pi_{k+1} = (1 - \epsilon)\pi_{\text{greedy}(Q_k)} + \epsilon\pi_{\text{any}}$$

Increasingly greedy policy: $\epsilon \to 0$ as $t \to \infty$

Expected Sarsa

- **Forward view**: look one step forward to compute the target

$$\Delta Q(s_t, a_t) = \alpha \left[r_{t+1} + \gamma \underset{a\sim\pi}{E} [Q(s_{t+1}, a)] - Q(s_t, a_t) \right]$$

- **Backward view**: wait one step to update $Q(s_t, a_t)$

$$\Delta Q(s_{t-1}, a_{t-1}) = \alpha \left[r_t + \gamma \underset{a\sim\pi}{E} [Q(s_t, a)] - Q(s_{t-1}, a_{t-1}) \right]$$

- **Forward view**: look one step forward to compute the target

$$\Delta Q(s_t, a_t) = \alpha \left[r_{t+1} + \gamma \underset{a \sim \pi}{E} \left[Q(s_{t+1}, a) \right] - Q(s_t, a_t) \right]$$

- **Backward view**: wait one step to update $Q(s_t, a_t)$

$$\Delta Q(s_{t-1}, a_{t-1}) = \alpha \left[r_t + \gamma \underset{a \sim \pi}{E} \left[Q(s_t, a) \right] - Q(s_{t-1}, a_{t-1}) \right]$$

- **Forward view**: look one step forward to compute the target

$$\Delta Q(s_t, a_t) = \alpha \left[r_{t+1} + \gamma \underset{a \sim \pi}{E} \left[Q(s_{t+1}, a) \right] - Q(s_t, a_t) \right]$$

- **Backward view**: wait one step to update $Q(s_t, a_t)$

$$\Delta Q(s_{t-1}, a_{t-1}) = \alpha \left[r_t + \gamma \underset{a \sim \pi}{E} \left[Q(s_t, a) \right] - Q(s_{t-1}, a_{t-1}) \right]$$

Convergence in tabular case [7]

Why "Expected" Sarsa?

- Short answer: useful later
- Longer answer: less variance than traditional Sarsa
- Complete answer: Van Seijen et al. [7]

Temporal Credit Assignment

How to give credit (assign value) to (s_t, a_t)?

- **Problem**: the choice of a_t in s_t could be rewarded at time $t + n$
- **Solution**: look at future reward!

Temporal Credit Assignment

How to give credit (assign value) to (s_t, a_t)?

- **Problem**: the choice of a_t in s_t could be rewarded at time $t + n$
- **Solution**: look at future reward!

Look far (n steps) in the future:

$$G_t^{(n)} \doteq r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n \underset{a \sim \pi}{E} \left[Q(s_{t+n}, a) \right]$$

$$= \sum_{k=1}^{n} \gamma^{k-1} r_{t+k} + \gamma^n \underset{a \sim \pi}{E} \left[Q(s_{t+n}, a) \right]$$

- $G_t^{(1)}$ is a TD target: high bias, low variance
- $G_t^{(T-t)}$ is a Monte Carlo target: no bias, high variance

Look far (n steps) in the future:

$$G_t^{(n)} \doteq r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n \underset{a\sim\pi}{E}[Q(s_{t+n}, a)]$$
$$= r_{t+1} + \gamma \underset{a\sim\pi}{E}[Q(s_{t+n}, a)]$$

- $G_t^{(1)}$ is a TD target: high bias, low variance
- $G_t^{(T-t)}$ is a Monte Carlo target: no bias, high variance

Look far (n steps) in the future:

$$G_t^{(n)} \doteq r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n \underset{a\sim\pi}{E}\left[Q(s_{t+n}, a)\right]$$
$$= \sum_{k=1}^{T-t} \gamma^{k-1} r_{t+k} + \gamma^n \underset{a\sim\pi}{E}\left[Q(s_{t+n}, a)\right]$$

- $G_t^{(1)}$ is a TD target: high bias, low variance
- $G_t^{(T-t)}$ is a Monte Carlo target: no bias, high variance

n-step Expected Sarsa

Look far (n steps) in the future:

$$G_t^{(n)} \doteq r_{t+1} + \gamma r_{t+2} + \cdots + \gamma^{n-1} r_{t+n} + \gamma^n \underset{a \sim \pi}{E} [Q(s_{t+n}, a)]$$
$$= \sum_{k=1}^{n} \gamma^{k-1} r_{t+k} + \gamma^n \underset{a \sim \pi}{E} [Q(s_{t+n}, a)]$$

- $G_t^{(1)}$ is a TD target: high bias, low variance
- $G_t^{(T-t)}$ is a Monte Carlo target: no bias, high variance

How can we use all future returns

- **without storing all history**
- **without delaying updates**

Forward View

Average **all** n-step targets:

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

- $\lambda = 0$ gives $G_t^{(1)}$, the TD target
- $\lambda = 1$ gives $G_t^{(T-t)}$, the Monte Carlo target

Forward View

Average **all** n-step targets:

$$G_t^\lambda = (1-\lambda)\sum_{n=1}^{T-t-1}\lambda^{n-1}G_t^{(n)} + \lambda^{T-t-1}G_t^{(T-t)}$$

- $\lambda = 0$ gives $G_t^{(1)}$, the TD target
- $\lambda = 1$ gives $G_t^{(T-t)}$, the Monte Carlo target

Forward View

Average **all** n-step targets:

$$G_t^\lambda = (1-0)0^0 G_t^{(1)} + (1-0) \sum_{n=2}^{T-t-1} 0^{n-1} G_t^{(n)} + 0^{T-t-1} G_t^{(T-t)}$$

- $\lambda = 0$ gives $G_t^{(1)}$, the TD target
- $\lambda = 1$ gives $G_t^{(T-t)}$, the Monte Carlo target

Forward View

Average **all** n-step targets:

$$G_t^\lambda = (1-1)\sum_{n=1}^{T-t-1} 1^{n-1} G_t^{(n)} + 1^{T-t-1} G_t^{(T-t)}$$

- $\lambda = 0$ gives $G_t^{(1)}$, the TD target
- $\lambda = 1$ gives $G_t^{(T-t)}$, the Monte Carlo target

Forward View

Average **all** n-step targets:

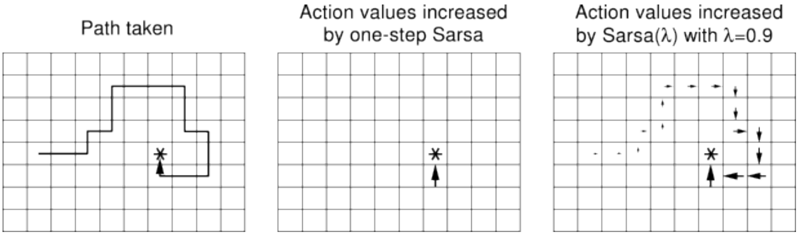
$$G_t^\lambda \doteq (1-\lambda)\sum_{n=1}^{\infty}\lambda^{n-1}G_t^{(n)}$$

- $\lambda = 0$ gives $G_t^{(1)}$, the TD target
- $\lambda = 1$ gives $G_t^{(T-t)}$, the Monte Carlo target

Another way of interpolating between TD and MC

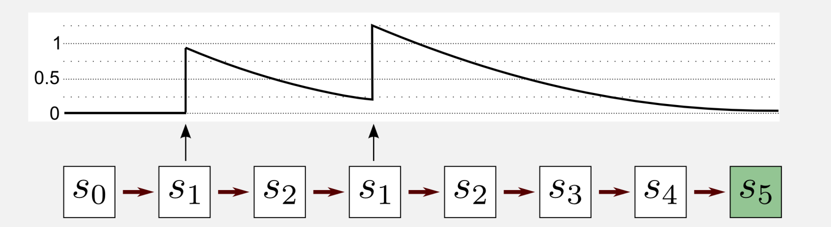
Backward View

- Propagate TD error to past states
- Fading memory



Eligibility Traces

- Assign a "trace" to each state-action pair
- Set the trace to one when visited
- All traces fade with time
- TD update proportional to trace



Accumulating Traces

Initialization:

$$e(s, a) \leftarrow 0 \text{ for all} s, a$$

Update **all states** at each step t:

$$e(s, a) \leftarrow \mathbb{1}\{s = s_t, a = a_t\} + \gamma\lambda e(s, a) \quad \Delta Q(s, a) = \alpha e(s, a)(G_t^{(1)} - Q($$

Back to the forward view

Accumulating traces

$$\Delta Q(s_t, a_t) =$$

$$\alpha_k \sum_{h \geq t} \left[(G_h^{(1)} - Q(s_h, a_h)) \sum_{j=t}^{n} (\gamma\lambda)^{h-j} \mathbb{1} \{ s_j, a_j = s_t, a_t \} \right]$$

Two policies:

- **Target** policy π: the one that is evaluated/improved
- **Behavioral** policy μ: the one that is used to interact with the environment

Potentially, both can change!

Advantages:

- Safe
- Separate exploration from evaluation
- Reuse past experience

Importance Sampling

Correct the update with likelihood ratios

$$\Delta Q(s_t, a_t) = \alpha \prod_{h \geq t} \frac{\pi(a_h \mid s_h)}{\mu(a_h \mid s_h)} \left[G_t^{(n)} - Q(s_t, a_t) \right]$$

Importance Sampling

Correct the update with likelihood ratios

$$\Delta Q(s_t, a_t) = \alpha \prod_{h \geq t} \frac{\pi(a_h \mid s_h)}{\mu(a_h \mid s_h)} \left[G_t^{(n)} - Q(s_t, a_t) \right]$$

Issue: high variance!

Q-learning

Approximate $\mathcal{T}^*$ directly

$$\Delta Q(s_t, a_t) = \alpha \left[r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

Approximate $\mathcal{T}^*$ directly

$$\Delta Q(s_t, a_t) = \alpha \left[r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q(s_{t+1}, a) - Q(s_t, a_t) \right]$$

Implicit target policy: $\pi_{\text{greedy}(Q)}$

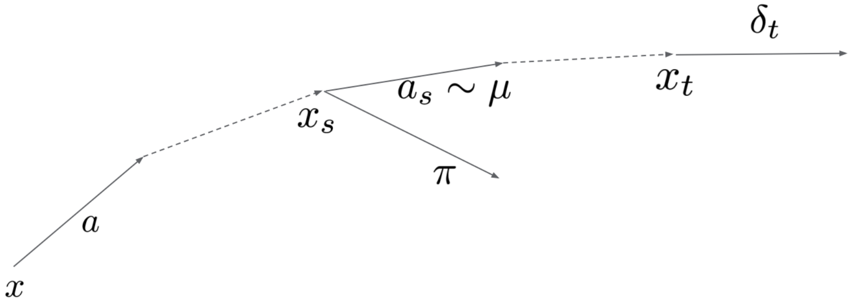
Approximate $\mathcal{T}^*$ directly

$$\Delta Q(s_t, a_t) = \alpha \left[G_t^* - Q(s_t, a_t) \right]$$

Implicit target policy: $\pi_{\text{greedy}(Q)}$

Maximization as a special case of expectation
$\implies G_t^* = G_t^{(1)}$ when π is a greedy policy

Credit Assignment



Watkins' Q(λ)

Cut the eligibility trace each time μ performs a non-greedy action

For all s, a

$$e \leftarrow \gamma\lambda e + \mathbb{1}_t$$

$$\Delta Q(s, a) = \alpha e(s, a) \left[G_t^* - Q(s, a)\right]$$

If $a_t \neq \arg\max_{a' \in \mathcal{A}} Q(s_t, a')$

$$e(s, a) \leftarrow 0$$

Credit is assigned **up to the last greedy action**

Issues:

- Traces are cut too often
- Convergence was an open problem since 1989!

General Off-policy Value Update

Back to the forward view

$$\Delta Q(s_t, a_t) =$$

$$\alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} (\gamma\lambda)^{k-j} \mathbb{1}\left\{s_j, a_j = s_t, a_t\right\} \right]$$

General Off-policy Value Update

Back to the forward view

$$\Delta Q(s_t, a_t) =$$

$$\alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \lambda^{k-j} \mathbb{1}\{s_j, a_j = s_t, a_t\} \right]$$

General Off-policy Value Update

Back to the forward view

$$\Delta Q(s_t, a_t) =$$

$$\alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} c_i \right) \mathbb{1}\left\{ s_j, a_j = s_t, a_t \right\} \right]$$

General Off-policy Value Update

Back to the forward view

$$\Delta Q(s_t, a_t) =$$
$$\alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

In this formulation we call the $\mathbf{c_i}$ "traces"

General Off-policy Prediction

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

- $c_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$
- $c_i = \lambda$
- $c_i = \lambda \pi(a_i \mid s_i)$

General Off-policy Prediction

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

- $c_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)} \implies$ Importance Sampling!
- $c_i = \lambda$
- $c_i = \lambda \pi(a_i \mid s_i)$

General Off-policy Prediction

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

- $c_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)} \implies$ Importance Sampling!
- $c_i = \lambda \implies Q^\pi(\lambda)$ [1]
- $c_i = \lambda \pi(a_i \mid s_i)$

General Off-policy Prediction

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

- $c_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)} \implies$ Importance Sampling!
- $c_i = \lambda \implies Q^{\pi}(\lambda)$ [1]
- $c_i = \lambda \pi(a_i \mid s_i) \implies TB(\lambda)$ [5]

- Traces: $c_i = \lambda$
- Idea: Do not cut traces
- Low variance
- Issue: not safe
  Convergence only if $\pi \simeq \mu$:

$$\|\pi - \mu\|_1 \leq \frac{1-\gamma}{\lambda\gamma}$$

Tree-Backup $TB(\lambda)$

- Traces: $c_i = \lambda\pi(a_i \mid s_i)$
- Idea: soft cut
- Convergence for any π, μ
- Issue: not efficient
  Traces are cut unnecessarily when almost on-policy

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

Algorithm	Trace c_i	Issue
IS	$\frac{\pi(a_i\|s_i)}{\mu(a_i\|s_i)}$	High variance
$Q^\pi(\lambda)$	λ	Not safe off-policy
$TB(\lambda)$	$\lambda\pi(a_i \mid s_i)$	Not efficient on-policy

We want an algorithm that has **low variance**, is **safe** and **efficient**

Safety

Theorem (Off-policy prediction)

For any π and μ, assuming the state space is finite and all states are visited infinitely often:

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \quad \textit{then} \quad Q \to Q^\pi$$

Theorem (Off-policy prediction)

For any π and μ, assuming the state space is finite and all states are visited infinitely often:

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \quad \textit{then} \quad Q \to Q^\pi$$

Proof sketch: Define the off-policy operator $\mathcal{R}$:

$$\mathcal{R}Q(s,a) \doteq Q(s,a) + \mathop{E}_{a_t \sim \mu}\left[\sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{t} c_i\right)\left(G_t^{(1)} - Q(s_t, a_t)\right)\right]$$

Show that Q^π is the unique fixed point of $\mathcal{R}$
Show that $\|\mathcal{R}Q - Q^\pi\|_\infty \leq \gamma \|Q - Q^\pi\|_\infty$

Theorem (Off-policy prediction)

For any π and μ, assuming the state space is finite and all states are visited infinitely often:

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \quad \textit{then} \quad Q \rightarrow Q^\pi$$

Proof sketch: Define the off-policy operator $\mathcal{R}$:

$$\mathcal{R}Q(s,a) \doteq Q(s,a) + \underset{a_t \sim \mu}{E}\left[\sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{t} c_i\right)\left(G_t^{(1)} - Q(s_t,a_t)\right)\right]$$

Show that Q^π is the unique fixed point of $\mathcal{R}$

Show that $\|\mathcal{R}Q - Q^\pi\|_\infty \leq \gamma \|Q - Q^\pi\|_\infty$

Theorem (Off-policy prediction)

For any π and μ, assuming the state space is finite and all states are visited infinitely often:

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \quad \textit{then} \quad Q \rightarrow Q^\pi$$

Proof sketch: Define the off-policy operator $\mathcal{R}$:

$$\mathcal{R}Q(s,a) \doteq Q(s,a) + \underset{a_t \sim \mu}{E}\left[\sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{t} c_i\right)\left(G_t^{(1)} - Q(s_t, a_t)\right)\right]$$

Show that Q^π is the unique fixed point of $\mathcal{R}$

Show that $\|\mathcal{R}Q - Q^\pi\|_\infty \leq \gamma \|Q - Q^\pi\|_\infty$

Theorem (Off-policy prediction)

For any π and μ, assuming the state space is finite and all states are visited infinitely often:

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \quad \textit{then} \quad Q \to Q^\pi$$

Proof sketch: Define the off-policy operator $\mathcal{R}$:

$$\mathcal{R}Q(s,a) \doteq Q(s,a) + \mathop{E}_{a_t \sim \mu}\left[\sum_{t \geq 0} \gamma^t \left(\prod_{i=1}^{t} c_i\right)\left(G_t^{(1)} - Q(s_t, a_t)\right)\right]$$

Show that Q^π is the unique fixed point of $\mathcal{R}$
Show that $\|\mathcal{R}Q - Q^\pi\|_\infty \leq \gamma \|Q - Q^\pi\|_\infty$

Safety

Theorem (Off-policy prediction)

For any π and μ, assuming the state space is finite and all states are visited infinitely often:

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \quad \textit{then} \quad Q \to Q^\pi$$

- **Safety**: ensured
- **Variance**: maximal when $c_i = \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$
- **Efficiency**: minimal when $c_i = 0$

Retrace(λ) for prediction

$$a_t \sim \mu(\cdot \mid s_t)$$

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \mathbf{c_i} \right) \mathbb{1}_{jt} \right]$$

$$c_i = \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\}$$

Retrace(λ) for prediction

$$a_t \sim \mu(\cdot \mid s_t)$$

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} c_i \right) \mathbb{1}_{jt} \right]$$

$$c_i = \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\}$$

- **Safe** since $0 \leq c_i \leq \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$
- **Low variance** since $c_i \leq 1$
- **Efficient on-policy** since $c_i \to \lambda$ as $\mu \to \pi$

Retrace(λ) for prediction

$$a_t \sim \mu(\cdot \mid s_t)$$

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} c_i \right) \mathbb{1}_{jt} \right]$$

$$c_i = \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\}$$

- **Safe** since $0 \leq c_i \leq \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$
- **Low variance** since $c_i \leq 1$
- **Efficient on-policy** since $c_i \to \lambda$ as $\mu \to \pi$

Retrace(λ) for prediction

$$a_t \sim \mu(\cdot \mid s_t)$$

$$\Delta Q(s_t, a_t) = \alpha_k \sum_{k \geq t} \left[(G_k^{(1)} - Q(s_k, a_k)) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} c_i \right) \mathbb{1}_{jt} \right]$$

$$c_i = \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\}$$

- **Safe** since $0 \leq c_i \leq \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}$
- **Low variance** since $c_i \leq 1$
- **Efficient on-policy** since $c_i \to \lambda$ as $\mu \to \pi$

Retrace(λ) for control

Theorem (Off-policy control)

For any μ_k, if π_k is ***increasingly greedy*** *w.r.t to Q_k:*

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi_k(a_i \mid s_i)}{\mu_k(a_i \mid s_i)} \quad \textit{then} \quad Q_k \to Q^*$$

Retrace(λ) for control

Theorem (Off-policy control)

*For any μ_k, if π_k is **increasingly greedy** w.r.t to Q_k:*

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi_k(a_i \mid s_i)}{\mu_k(a_i \mid s_i)} \quad \textit{then} \quad Q_k \to Q^*$$

Proof sketch: Show that

$$\|\mathcal{R} Q_k - Q^*\|_\infty \leq \gamma \|Q_k - Q^*\|_\infty + \epsilon_k \|Q_k\|_\infty$$

Then $Q_k \to Q^*$ as $\epsilon_k \to 0$

Retrace(λ) for control

Theorem (Off-policy control)

*For any μ_k, if π_k is **increasingly greedy** w.r.t to Q_k:*

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi_k(a_i \mid s_i)}{\mu_k(a_i \mid s_i)} \quad \textit{then} \quad Q_k \to Q^*$$

Proof sketch: Show that

$$\|\mathcal{R} Q_k - Q^*\|_\infty \leq \gamma \|Q_k - Q^*\|_\infty + \epsilon_k \|Q_k\|_\infty$$

Then $Q_k \to Q^*$ as $\epsilon_k \to 0$

Retrace(λ) for control

Theorem (Off-policy control)

For any μ_k, if π_k is ***increasingly greedy*** *w.r.t to Q_k:*

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi_k(a_i \mid s_i)}{\mu_k(a_i \mid s_i)} \quad \textit{then} \quad Q_k \to Q^*$$

Proof sketch: Show that

$$\|\mathcal{R} Q_k - Q^*\|_\infty \leq \gamma \|Q_k - Q^*\|_\infty + \epsilon_k \|Q_k\|_\infty$$

Then $Q_k \to Q^*$ as $\epsilon_k \to 0$

Retrace(λ) for control

Theorem (Off-policy control)

For any μ_k, if π_k is ***increasingly greedy*** *w.r.t to Q_k:*

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi_k(a_i \mid s_i)}{\mu_k(a_i \mid s_i)} \quad \textit{then} \quad Q_k \to Q^*$$

Proof sketch: Show that

$$\|\mathcal{R}Q_k - Q^*\|_\infty \leq \gamma \|Q_k - Q^*\|_\infty + \epsilon_k \|Q_k\|_\infty$$

Then $Q_k \to Q^*$ as $\epsilon_k \to 0$

Retrace(λ) for control

Theorem (Off-policy control)

*For any μ_k, if π_k is **increasingly greedy** w.r.t to Q_k:*

$$\textit{If} \quad 0 \leq c_i \leq \frac{\pi_k(a_i \mid s_i)}{\mu_k(a_i \mid s_i)} \quad \textit{then} \quad Q_k \to Q^*$$

Remarks

- No GLIE assumption on μ
- Extends to continuous action spaces
- c_i must be Markovian

Solving open problems as a corollary

$$\alpha_k \sum_{k \geq t} \left[\left(G_k^{(1)} - Q(s_k, a_k)\right) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \lambda \min\left\{1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)}\right\} \right) \mathbb{1}_{jt} \right]$$

$$\alpha_k \sum_{k \geq t} \left[\left(G_k^{(1)} - Q(s_k, a_k) \right) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\} \right) \mathbb{1}_{jt} \right]$$

- when π is a greedy policy, $G_t^{(1)}$ is just G_t^*
- when π is a greedy policy, $c_i = \lambda \mathbb{1} \{ \mu_k(s_i) = \pi_{\text{greedy}}(s_i) \}$

Solving open problems as a corollary

$$\alpha_k \sum_{k \geq t} \left[\left(G_k^{(1)} - Q(s_k, a_k)\right) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\} \right) \mathbb{1}_{jt} \right]$$

- when π is a greedy policy, $G_t^{(1)}$ is just G_t^*
- when π is a greedy policy, $c_i = \lambda \mathbb{1} \left\{ \mu_k(s_i) = \pi_{\text{greedy}}(s_i) \right\}$

$$\alpha_k \sum_{k \geq t} \left[\left(G_k^{(1)} - Q(s_k, a_k)\right) \sum_{j=t}^{k} \gamma^{k-j} \left(\prod_{i=j+1}^{k} \lambda \min \left\{ 1, \frac{\pi(a_i \mid s_i)}{\mu(a_i \mid s_i)} \right\} \right) \mathbb{1}_{jt} \right]$$

- when π is a greedy policy, $G_t^{(1)}$ is just G_t^*
- when π is a greedy policy, $c_i = \lambda \mathbb{1}\left\{\mu_k(s_i) = \pi_{\text{greedy}}(s_i)\right\}$

Watkins' $Q(\lambda)$ is a special case of Retrace(λ)

Convergence of Watkins' $Q(\lambda)$ proved after 27 years!

Retrace(λ) in practice

Retrace(λ) is more general than Watkins' $Q(\lambda)$

- π_k and μ_k can both be increasingly greedy policies, with π_k converging faster than $\mu_k \implies$ A3C [2]
- μ can be a snapshot of $\pi \implies$ DQN with memory replay [3]

Retrace(λ) in practice

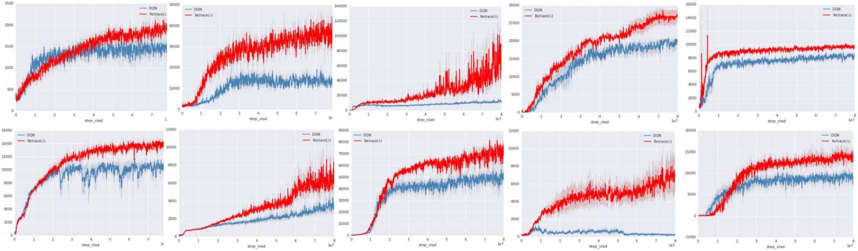
Retrace(λ) is more general than Watkins' $Q(\lambda)$

- π_k and μ_k can both be increasingly greedy policies, with π_k converging faster than $\mu_k \implies$ A3C [2]
- μ can be a snapshot of $\pi \implies$ DQN with memory replay [3]

Retrace(λ) is more general than Watkins' $Q(\lambda)$

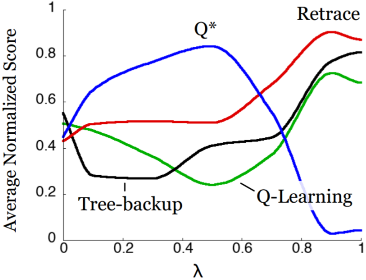
- π_k and μ_k can both be increasingly greedy policies, with π_k converging faster than $\mu_k \implies$ A3C [2]
- μ can be a snapshot of $\pi \implies$ DQN with memory replay [3]

Atari Games: Retrace(λ) vs DQN



Asteroids, Defender, Demon Attack, Hero, Krull, River Raid, Space Invaders, Star Gunner, Wizard of Wor, Zaxxon

Atari Games: Full Comparison



Algorithm	Times Best
DQN	12
$Q^*(\lambda)$	2
TB(λ)	16
Retrace(λ)	**30**

Conclusion

- Retrace(λ) combines **off-policy** learning and **multi-step returns** in a **safe** and **efficient** way
- In practice, it propagates sparse rewards faster than TB(λ) without the convergence problems of naive Q(λ)
- With $\pi =$ greedy(Q) we have Watkin's Q-learning
- With $\lambda = 1$ we have truncated importance sampling
- The theorems leave room for improvement

Questions?

- Anna Harutyunyan, Marc G Bellemare, Tom Stepleton, and Rémi Munos.
  Q (\lambda) with off-policy corrections.
  In *International Conference on Algorithmic Learning Theory*, pages 305–320. Springer, 2016.

- Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu.
  Asynchronous methods for deep reinforcement learning.
  In *International Conference on Machine Learning*, pages 1928–1937, 2016.

- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al.
  Human-level control through deep reinforcement learning.
  Nature, 518(7540):529–533, 2015.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc Bellemare.
Safe and efficient off-policy reinforcement learning.
In *Advances in Neural Information Processing Systems*, pages 1054–1062, 2016.

Doina Precup.
Eligibility traces for off-policy policy evaluation.
Computer Science Department Faculty Publication Series, page 80, 2000.

Harm Seijen and Rich Sutton.
True online td (lambda).
In *International Conference on Machine Learning*, pages 692–700, 2014.

Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering.
A theoretical and empirical analysis of expected sarsa.
In *Adaptive Dynamic Programming and Reinforcement Learning, 2009. ADPRL'09. IEEE Symposium on*, pages 177–184. IEEE, 2009.

The Semigradient Version

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \alpha\delta_t \boldsymbol{e}$$

$$e \leftarrow \nabla \hat{Q}_{\boldsymbol{\theta}}(s_t, a_{,t}) + \gamma\lambda \boldsymbol{e}$$

Accumulating Traces

Update at step t:

$$e(s,a) \leftarrow \mathbb{1}\{s = s_t, a = a_t\} + \gamma\lambda e(s,a)$$
$$\Delta Q(s,a) = \alpha e(s,a)(G_t^{(1)} - Q(s,a))$$

Accumulating Traces

Update at step t:

$$e \leftarrow \mathbb{1}_t + \gamma\lambda e(s,a)$$
$$\Delta Q = \alpha e \delta_t$$

Replacing Traces

Update at step t:

$$e \leftarrow (1 - \mathbb{1}_t)\gamma\lambda e + \mathbb{1}_t$$
$$\Delta Q = \alpha e \delta_t$$

Dutch Traces

Update at step t:

$$e \leftarrow (1 - \alpha \mathbb{1}_t)\gamma\lambda e + \mathbb{1}_t$$
$$\Delta Q = \alpha e \delta_t$$

Seijen and Sutton, 2014 [6]