Safe and Efficient Off-Policy Reinforcement Learning
NIPS 2016

Yasuhiro Fujita
Preferred Networks Inc.

January 11, 2017
Proposes a new off-policy multi-step RL method: Retrace($\lambda$)

- Good theoretical properties: low-variance, safe and efficient
- It outperforms one-step Q-learning and existing multi-step variants
- Proves the convergence of Watkins’s $Q(\lambda)$ for the first time
Multi-step methods

- Multi-step methods have some advantages over single-step methods
  - They can balance bias and variance
  - They can propagate values more quickly
- Example: SARSA(\(\lambda\))
  - n-step return

\[
R_s^{(n)} = \sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} Q(x_{s+n+1}, a_{s+n+1})
\]

- \(\lambda\)-return based update rule

\[
\Delta Q(x_s, a_s) = \sum_{t\geq s} (\lambda \gamma)^{t-s} \delta_t
\]

\[
\delta_t = r_t + \gamma Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)
\]
Multi-step methods in off-policy settings

- Can we apply multi-step methods to off-policy cases?
  - **Policy evaluation**: estimate $Q^\pi$ from samples collected by $\mu$ ($\pi \neq \mu$)
  - **Control**: estimate $Q^*$ from samples collected by $\mu$

In “Algorithms for Reinforcement Learning”, p. 57

There exist multi-step versions of $Q$-learning (e.g., Sutton and Barto, 1998, Section 7.6). However, these are not as appealing (and straightforward) as the multi-step extensions of TD(0) since $Q$-learning is an inherently off-policy algorithm: the temporal differences underlying $Q$-learning do not telescope even when $X_{t+1} = Y_{t+1}$. 
Watkins’s Q(\(\lambda\)) [Watkins 1989]

- Classic multi-step algorithm for off-policy control
- Cut off traces whenever a non-greedy action is taken

\[
R_s^{(n)} = \sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)
\]

(for any \(n < \tau = \arg\min_u \{ \pi_{s+u} \neq \mu_{s+u} \} \))

- Converges to \(Q^*\) under a mild assumption (proved in this paper)
- Only little faster than one-step Q-learning if non-greedy actions are frequent (i.e. not “efficient”)
Backup diagram of $Q(\lambda)$
General operator $\mathcal{R}$

- To compare off-policy multistep methods, consider the general operator $\mathcal{R}$:

$$\mathcal{R}Q(x, a) := Q(x, a) + \mathbb{E}_\mu \left[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^{t} c_s \right) \left( r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t) \right) \right],$$

(3)

- Different non-negative coefficients $c_s$ (traces) result in different methods

- (Is this equation correct?)
Desired properties

- **Low variance**
  - Variance of the online estimate is small
  - \( \approx \nabla (c_1 \cdots c_t) \) is small
  - \( \approx \nabla (c) \) is small

- **Safe**
  - Convergence to \( Q^\pi \) (policy evaluation) or \( Q^* \) (control) is guaranteed

- **Efficient**
  - Traces are not unnecessarily cut if \( \pi \) and \( \mu \) are close
Comparison of properties

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Definition of $c_s$</th>
<th>Estimation variance</th>
<th>Guaranteed convergence†</th>
<th>Use full returns (near on-policy)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importance sampling</td>
<td>$\frac{\pi(a_s</td>
<td>x_s)}{\mu(a_s</td>
<td>x_s)}$</td>
<td>High</td>
</tr>
<tr>
<td>$Q^\pi(\lambda)$</td>
<td>$\lambda$</td>
<td>Low</td>
<td>for $\pi$ close to $\mu$</td>
<td>yes</td>
</tr>
<tr>
<td>TB($\lambda$)</td>
<td>$\lambda\pi(a_s</td>
<td>x_s)$</td>
<td>Low</td>
<td>for any $\pi, \mu$</td>
</tr>
<tr>
<td>Retrace($\lambda$)</td>
<td>$\lambda \min\left(1, \frac{\pi(a_s</td>
<td>x_s)}{\mu(a_s</td>
<td>x_s)}\right)$</td>
<td>Low</td>
</tr>
</tbody>
</table>

Table 1: Properties of several algorithms defined in terms of the general operator given in (3). †Guaranteed convergence of the expected operator $\mathcal{R}$.

- Retrace($\lambda$) is low-variance, safe and efficient
- Note that Watkins’s $Q(\lambda) \neq Q^\pi(\lambda)$
Importance Sampling (IS) [Precup et al. 2000]

\[ \mathcal{R}Q(x, a) := Q(x, a) + \mathbb{E}_\mu \left[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^{t} c_s \right) \left( r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t) \right) \right], \quad (3) \]

\[ c_s = \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)} \]

- \( \mathcal{R}Q = Q^\pi \) for any \( Q \) in this case
  - If \( Q = 0 \), it just becomes the basic IS estimate
    \[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^{t} c_s \right) r_t \]
  - High variance, mainly due to the variance of the product
    \[ \frac{\pi(a_1|x_1)}{\mu(a_1|x_1)} \ldots \frac{\pi(a_t|x_t)}{\mu(a_t|x_t)} \]
Off-policy $Q^\pi(\lambda)$ and $Q^*(\lambda)$ [Harutyunyan et al. 2016]

$$\mathcal{R}Q(x, a) := Q(x, a) + \mathbb{E}_\mu \left[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^{t} c_s \right) \left( r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t) \right) \right],$$  \hspace{1cm} (3)

$$c_s = \lambda$$

- A very recently proposed alternative
  - $Q^\pi(\lambda)$ for policy evaluation, $Q^*$ for control
- To guarantee convergence, $\pi$ and $\mu$ must be sufficiently close:
  - In policy evaluation, $\lambda < \frac{1-\gamma}{\gamma \epsilon}$, where \( \epsilon := \max_x \| \pi(\cdot | x) - \mu(\cdot | x) \|_1 \)
  - In control, $\lambda < \frac{1-\gamma}{2\gamma}$
- Available even if $\mu$ is unknown and/or non-Markovian
Tree Backup (TB) [Precup et al. 2000]

\[
RQ(x, a) := Q(x, a) + \mathbb{E}_\mu \left[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^{t} c_s \right) \left( r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t) \right) \right],
\]

\[
c_s = \lambda \pi(a_s | x_s)
\]

- The operator defines a contraction, thus is safe
- Not efficient because it cuts traces even if \( \pi = \mu \)
- Available even if \( \mu \) is unknown and/or non-Markovian
Useful table [Harutyunyan et al. 2016]

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>$n$-step return</th>
<th>Update rule for the $\lambda$-return</th>
<th>FP</th>
</tr>
</thead>
<tbody>
<tr>
<td>TD($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} V(x_{s+n+1})$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t$ $\delta_t = r_t + \gamma V(x_{t+1}) - V(x_t)$</td>
<td>$V^\mu$</td>
</tr>
<tr>
<td>(on-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SARS($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} Q(x_{s+n+1}, a_{s+n+1})$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t$ $\delta_t = r_t + \gamma Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)$</td>
<td>$Q^\mu$</td>
</tr>
<tr>
<td>(on-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>E SARS($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} E_\mu Q(x_{s+n+1}, \cdot)$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t + E_\mu Q(x_s, \cdot) - Q(x_s, a_s)$ $\delta_t = r_t + \gamma E_\mu Q(x_{t+1}, \cdot) - E_\mu Q(x_t, \cdot)$</td>
<td>$Q^\mu$</td>
</tr>
<tr>
<td>(on-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General $Q(\lambda)$</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} E_\pi Q(x_{s+n+1}, \cdot)$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t + E_\pi Q(x_s, \cdot) - Q(x_s, a_s)$ $\delta_t = r_t + \gamma E_\pi Q(x_{t+1}, \cdot) - E_\pi Q(x_t, \cdot)$</td>
<td>$Q^\mu, \pi$</td>
</tr>
<tr>
<td>(off-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PDIS($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t \prod_{i=s+1}^{t} \rho_i + \gamma^{n+1} Q(x_{s+n+1}, a_{s+n+1}) \prod_{i=s}^{s+n} \rho_i$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t \prod_{i=s+1}^{t} \rho_i$ $\delta_t = r_t + \gamma \rho_{t+1} Q(x_{t+1}, a_{t+1}) - Q(x_t, a_t)$</td>
<td>$Q^\pi$</td>
</tr>
<tr>
<td>(off-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TB($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \prod_{i=s+1}^{t} \tau_i [r_t + \gamma E_\pi a_{t+1} Q(x_{t+1}, \cdot)] + \gamma^{n+1} \prod_{i=s}^{s+n} \tau_i Q(x_{s+n+1}, a_{s+n+1})$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t \prod_{i=s+1}^{t} \tau_i$ $\delta_t = r_t + \gamma E_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t)$</td>
<td>$Q^\pi$</td>
</tr>
<tr>
<td>(off-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q^\pi(\lambda)$</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} [r_t + E_\pi Q(x_t, \cdot) - Q(x_t, a_t)] + \gamma^{n+1} E_\pi Q(x_{s+n+1}, \cdot)$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t$ $\delta_t = r_t + \gamma E_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t)$</td>
<td>$Q^\pi$</td>
</tr>
<tr>
<td>(on/off-policy)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)$ (for any $n &lt; \tau = \arg \min_a {\pi_s + \mu_s + u}$)</td>
<td>$\sum_{t=s}^{s+n} (\lambda \gamma)^{t-s} \delta_t \prod_{i=s+1}^{t} \tau_i$ $\delta_t = r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)$</td>
<td>$Q^*$</td>
</tr>
<tr>
<td>(Watkins’s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Q($\lambda$)</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} r_t + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)$</td>
<td>$\sum_{t=s}^{s+n} (\lambda \gamma)^{t-s} \delta_t + \max_a Q(x_s, a) - Q(x_s, a_s)$ $\delta_t = r_t + \gamma \max_a Q(x_{t+1}, a) - \max_a Q(x_t, a)$</td>
<td>$Q^\mu, *$</td>
</tr>
<tr>
<td>(P &amp; W’s)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$Q^*(\lambda)$</td>
<td>$\sum_{t=s}^{s+n} \gamma^{t-s} [r_t + \max_a Q(x_t, a) - Q(x_t, a_t)] + \gamma^{n+1} \max_a Q(x_{s+n+1}, a)$</td>
<td>$\sum_{t \geq s} (\lambda \gamma)^{t-s} \delta_t$ $\delta_t = r_t + \gamma \max_a Q(x_{t+1}, a) - Q(x_t, a_t)$</td>
<td>$Q^*$</td>
</tr>
</tbody>
</table>
Retrace($\lambda$)

\[
RQ(x, a) := Q(x, a) + \mathbb{E}_\mu \left[ \sum_{t \geq 0} \gamma^t \left( \prod_{s=1}^{t} c_s \right) \left( r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t) \right) \right],
\]

\[
c_s = \lambda \min(1, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)})
\]

- Proposed by this paper
- IS ratio truncated at 1
- If $\pi$ is close to $\mu$, $c_s$ is close to 1, avoid unnecessarily cutting traces
Experiments on Atari 2600

- Trained asynchronously with 16 threads
- Each thread has private replay memory holding 62,500 transitions
- Q-learning uses a minibatch of 64 transitions
- Retrace, TB and Q* use four 16-step sequences
Performance comparison

Figure 1: Inter-algorithm score distribution for $\lambda$-return ($\lambda = 1$) variants and Q-Learning ($\lambda = 0$).

- 0 and 1 of inter-algorithm scores respectively correspond to the worst and best scores for a particular game.
- Retrace($\lambda$) performs best on 30 out of 60 games.
Sensitivity to $\lambda$

![Graph showing sensitivity to $\lambda$](image)

Figure 2: Average inter-algorithm scores for each value of $\lambda$. The DQN scores are fixed across different $\lambda$, but the corresponding inter-algorithm scores varies depending on the worst and best performer within each $\lambda$. **Note that average scores are not directly comparable across different values of $\lambda$.**

- Note that the Q-learning scores are fixed across different $\lambda$
- $Q^*$ performs best for small values of $\lambda$
Conclusions

- Retrace($\lambda$)
  - is low-variance, safe and efficient
  - outperforms one-step Q-learning and existing multi-step variants on Atari 2600
  - (is already applied to A3C in another paper [Wang et al. 2016])
- Watkins’s $Q(\lambda)$ now has a convergence guarantee
Future work

- Estimate \( \mu \) if it is unknown
- Relaxing the Markov assumption in the control case to allow \( c_s > 1 \):

\[
c_s = \lambda \min\left( \frac{1}{c_1 \cdots c_{s-1}}, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)} \right)
\]
Theorem 1. The operator $\mathcal{R}$ defined by (3) has a unique fixed point $Q^\pi$. Furthermore, if for each $a_s \in \mathcal{A}$ and each history $\mathcal{F}_s$ we have $c_s = c_s(a_s, \mathcal{F}_s) \in \left[0, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}\right]$, then for any $Q$-function $Q$,

$$\|\mathcal{R}Q - Q^\pi\| \leq \gamma \|Q - Q^\pi\|.$$  

- $\pi$ and $\mu$ are stationary
- $c_s$ can be non-Markovian
**Theorem 2.** Consider an arbitrary sequence of behaviour policies \((\mu_k)\) (which may depend on \((Q_k)\)) and a sequence of target policies \((\pi_k)\) that are increasingly greedy w.r.t. the sequence \((Q_k)\):

\[
Q_{k+1} = \mathcal{R}_k Q_k,
\]

where the return operator \(\mathcal{R}_k\) is defined by (3) for \(\pi_k\) and \(\mu_k\) and a Markovian \(c_s = c(a_s, x_s) \in [0, \frac{\pi_k(a_s|x_s)}{\mu_k(a_s|x_s)}]\). Assume the target policies \(\pi_k\) are \(\varepsilon_k\)-away from the greedy policies w.r.t. \(Q_k\), in the sense that \(\mathcal{T}^{\pi_k} Q_k \geq \mathcal{T} Q_k - \varepsilon_k \|Q_k\| e\), where \(e\) is the vector with 1-components. Further suppose that \(\mathcal{T}^{\pi_0} Q_0 \geq Q_0\). Then for any \(k \geq 0\),

\[
\|Q_{k+1} - Q^*\| \leq \gamma \|Q_k - Q^*\| + \varepsilon_k \|Q_k\|.
\]

In consequence, if \(\varepsilon_k \to 0\) then \(Q_k \to Q^*\).

- \(\pi\) is not stationary
- \(c_s\) must be Markovian
Theorem 3. Consider a sequence of sample trajectories, with the $k^{th}$ trajectory $x_0, a_0, r_0, x_1, a_1, r_1, \ldots$ generated by following $\mu_k$: $a_t \sim \mu_k(\cdot|x_t)$. For each $(x, a)$ along this trajectory, with $s$ being the time of first occurrence of $(x, a)$, update

$$Q_{k+1}(x, a) \leftarrow Q_k(x, a) + \alpha_k \sum_{t \geq s} \delta^\pi_t \sum_{j=s}^{t} \gamma^{t-j} \left( \prod_{i=j+1}^{t} c_i \right) 1\{x_j, a_j = x, a\},$$

(7)

where $\delta^\pi_t := r_t + \gamma E_{\pi_k} Q_k(x_{t+1}, \cdot) - Q_k(x_t, a_t)$, $\alpha_k = \alpha_k(x_s, a_s)$. We consider the Retrace($\lambda$) algorithm where $c_i = \lambda \min (1, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)})$. Assume that $(\pi_k)$ are increasingly greedy w.r.t. $(Q_k)$ and are each $\varepsilon_k$-away from the greedy policies $(\pi_{Q_k})$, i.e. $\max_x \|\pi_k(\cdot|x) - \pi_{Q_k}(\cdot|x)\|_1 \leq \varepsilon_k$, with $\varepsilon_k \to 0$. Assume that $P^\pi_k$ and $P^\pi_k \wedge \mu_k$ asymptotically commute: $\lim_k \|P^\pi_k P^\pi_k \wedge \mu_k - P^\pi_k \wedge \mu_k P^\pi_k\| = 0$. Assume further that (1) all states and actions are visited infinitely often: $\sum_{t \geq 0} P\{x_t, a_t = x, a\} \geq D > 0$, (2) the sample trajectories are finite in terms of the second moment of their lengths $T_k$: $E_{\mu_k} T_k^2 < \infty$, (3) the stepsizes obey the usual Robbins-Munro conditions. Then $Q_k \to Q^*$ a.s.

- Convergence of sample-based online algorithm
- As a corollary, Watkins’s $Q(\lambda)$ converges to $Q^*$
  - Only $c_s$ is different
参考文献


