Retrace: Safe and Efficient Off-Policy RL

Alex Lamb, Nan Rosemary Ke

Off-Policy Model-free Learning

- **Why** do we want to do Off-Policy learning?
  - Learn from observing humans or other agents
  - Re-use past experience generated from older policies
  - Learn multiple policies while following one policy

- Follow **Behaviour** policy $\boldsymbol{\mu}$, evaluate **Target** policy $\boldsymbol{\pi}$

Importance Sampling

$$G_t^{\frac{\pi}{\mu}} = \frac{\pi(A_t|S_t)}{\mu(A_t|S_t)} \frac{\pi(A_{t+1}|S_{t+1})}{\mu(A_{t+1}|S_{t+1})} \cdots \frac{\pi(A_T|S_T)}{\mu(A_T|S_T)} G_t \quad (4)$$

$$V(S_t) = V(S_t) + \alpha(G_t^{\frac{\pi}{\mu}} - V(S_t)) \quad (5)$$

- **Pros**
  - Unbiased. If $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ match, then performs perfectly.
- **Cons**
  - High variance. Not practical, $\boldsymbol{\mu}$ and $\boldsymbol{\pi}$ never match. If sequence is very unlikely under $\boldsymbol{\mu}$ compared to $\boldsymbol{\pi}$, then importance weight update will be large.

Retrace

- Retrace $$c_s = \lambda \min\left(1, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}\right)$$
-
- Calculate importance weights but clip them so that they can't go below 1.0.
- Appealing properties:

  -Safe (converges for any target and behavior policies)

  -Low variance

  -Performs ideally when behavior and target policies are the same.

-I find it surprising that this isn't an already well known trick (even without the theory).

Intuition for what Retrace is doing

-We still sample from the behavioral policy (i.e. off policy)

-We do a procedure like importance sampling, but wherever the behavior policy is *less likely* than the target policy, we treat it as if it were *just as likely* as the target policy.

-So if there are two paths with 10% probability under the target policy, one has 1% probability under the behavior policy and one has 0.0001% probability under the behavior policy, we give them the same reweighting!

-So policies that are very rare under the behavior policy end up getting underweighted and counting for less when we compute our value function for the target policy.

MDP where importance sampling blows up

-Have a certain action with very low probability under the behavior policy but high probability under the target policy.

Importance Sampling:

Bias: 1.09

Variance (std): 3167.55

Retrace:

Bias: 20.55

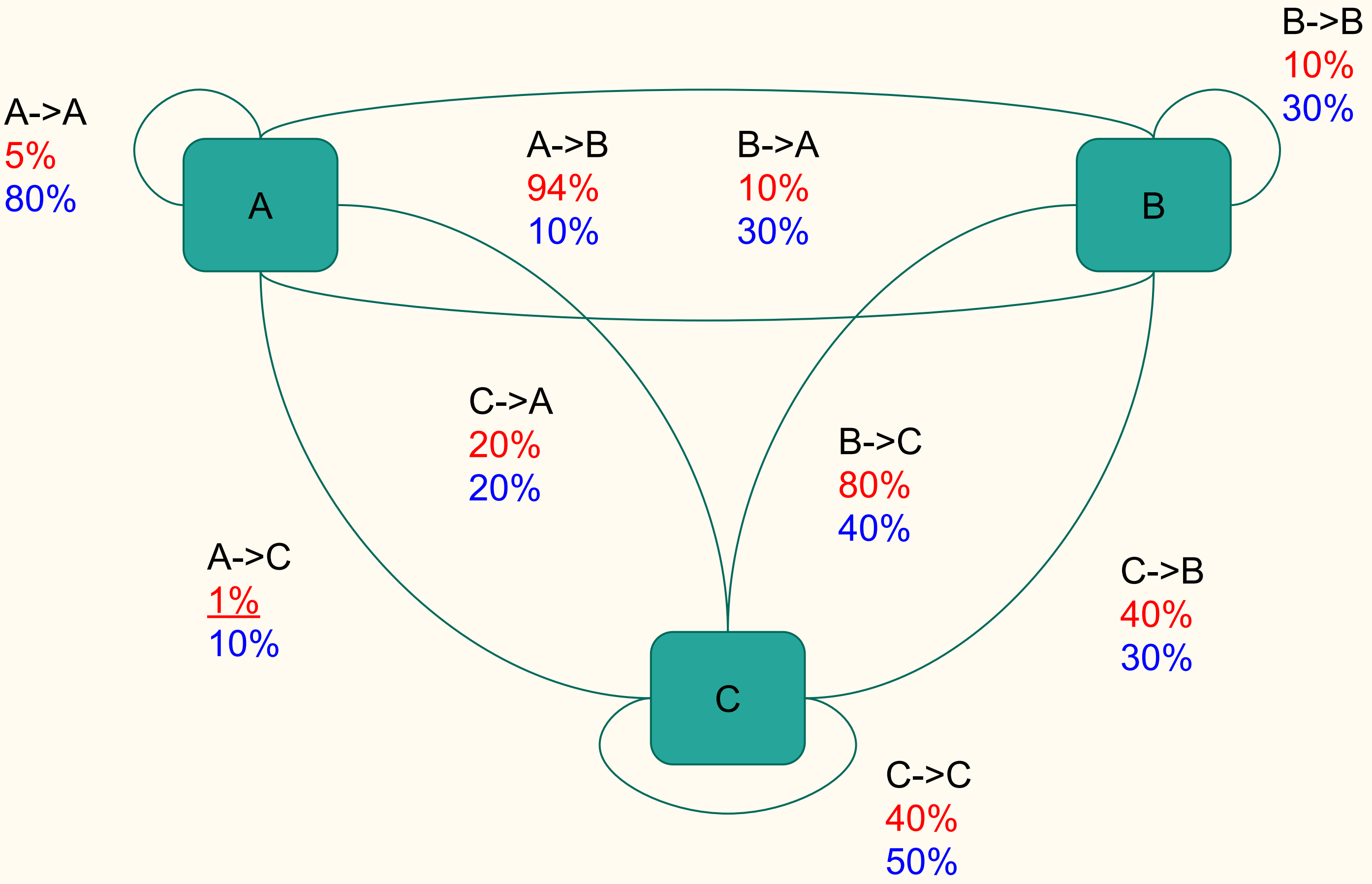
Variance (std): 0.65

Sample Behavior and Set Importance Weights to 1.0:

Bias: 22.5

Variance (std): 12.9

The Actual Structure of that MDP



Rewards

A : 0

B : 1

C : -10

Behavior is Red

Target is Blue

MDP with more balanced transition probabilities

Importance Sampling:

Bias: 0.0054

Variance (standard deviation): 3.298

Error: 10.88

Retrace:

Bias: 0.909

Variance (standard deviation): 0.74

Error: 1.37

Sample Behavior and Set Importance Weights to 1.0:

Bias: 0.423

Variance (standard deviation): 2.626

Error: 7.07

Bias-Variance Tradeoff

-Retrace has lower variance but is biased

Sometimes more biased than just estimating with the behavior policy!

-Why does retrace still have good properties?

Theory (Prediction)

Theorem 1. Assume finite state space. Generate trajectories according to behaviour policy $\boldsymbol{\mu}$. Update all trajectories according to

$$Q_{k+1}(x,a) = Q_k(x,a) + \alpha_k \sum_{t \geq 0} \gamma^t (c_1 \ldots c_t) \big(r_t + \gamma \mathbb{E}_\pi Q_k(x_{t+1}, \cdot) - Q_k(x_t, a_t) \big)$$

Then, If $0 \leq c_s \leq \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}$ then $Q_k \to Q^\pi$ a.s.

The algorithm is safe.

Lemma (Prediction)

$$Q_{k+1}(x,a) = Q_k(x,a) + \alpha_k \sum_{t \geq 0} \gamma^t (c_1 \ldots c_t)\big(r_t + \gamma \mathbb{E}_\pi Q_k(x_{t+1}, \cdot) - Q_k(x_t, a_t)\big)$$

The update follows a **contraction mapping**

$$\|\mathcal{R}Q_1 - \mathcal{R}Q_2\|_\infty \leq \gamma \|Q_1 - Q_2\|_\infty$$

Proof:

$$\begin{aligned}
\mathcal{R}Q(x,a) &= Q(x,a) + \mathbb{E}_\mu\Big[\sum_{t \geq 0} \gamma^t (c_1 \ldots c_t)\big(r_t + \gamma \mathbb{E}_\pi Q(x_{t+1}, \cdot) - Q(x_t, a_t)\big)\Big] \\
&= \mathbb{E}_\mu\Big[\sum_{t \geq 0} \gamma^t (c_1 \ldots c_t)\big(r_t + \gamma\big[\mathbb{E}_\pi Q(x_{t+1}, \cdot) - c_{t+1} Q(x_{t+1}, a_{t+1})\big]\big)\Big]
\end{aligned}$$

$$\begin{aligned}
(\mathcal{R}Q_1 - \mathcal{R}Q_2)(x,a) &= \mathbb{E}_\mu\Big[\sum_{t \geq 0} \gamma^{t+1} (c_1 \ldots c_t)\Big(\mathbb{E}_\pi (Q_1 - Q_2)(x_{t+1}, \cdot) - c_{t+1}(Q_1 - Q_2)(x_{t+1}, a_{t+1})\Big)\Big] \\
&= \mathbb{E}_\mu\Big[\sum_{t \geq 0} \gamma^{t+1} (c_1 \ldots c_t) \sum_a \big(\pi(a|x_{t+1}) - \mu(a|x_{t+1}) c_{t+1}(a)\big)(Q_1 - Q_2)(x_{t+1}, a)\Big]
\end{aligned}$$

Proof (Prediction)

Proof:

$$
\begin{aligned}
&= \mathbb{E}_\mu\Big[\sum_{t\geq 0}\gamma^{t+1}(c_1\dots c_t)\sum_a\big(\pi(a|x_{t+1})-\mu(a|x_{t+1})c_{t+1}(a)\big)\Big]\\
&= \mathbb{E}_\mu\Big[\sum_{t\geq 0}\gamma^{t+1}(c_1\dots c_t)(1-c_{t+1})\Big]\\
&= \gamma-(1-\gamma)\mathbb{E}_\mu\Big[\sum_{t\geq 1}\gamma^{t}(c_1\dots c_t)\Big]\\
&\in [0,\gamma]
\end{aligned}
$$

Therefore $\boxed{\|\mathcal{R}Q_1-\mathcal{R}Q_2\|_\infty \leq \gamma\|Q_1-Q_2\|_\infty}$