Reinforcement Learning

Safe and Efficient Off-Policy Reinforcement Learning

Review by Thaïs Cornilleau, Nour El Haddad, Salma Guennouni and Adrien Sardi

February 2024

Abstract

In this report, we review the paper titled "Safe and Efficient Off-Policy Reinforcement Learning", authored by Remi Munos, Thomas Stepleton, Anna Harutyunyan, and Marc G. Bellemare. This article discusses the development of a new algorithm, Retrace(λ), for off-policy reinforcement learning, which addresses the trade-off between return-based and value bootstrap methods, and demonstrates low variance, safe and efficient learning from off-policy data, and convergence guarantees for both policy evaluation and control settings. After analyzing the theoritical results of the paper and their proofs, we proceeded to implement the four distinct algorithms under comparison within the article and compare them using two different GymAI environments, Cliff Walking V0 and Frozen Lake V1, for our simulations.

Contents

1 Notation

In this section, we introduce the notation that we will be using in this report, sticking to the same ones used in the article.

We consider an agent's interaction with a Markov Decision Process (MDP) denoted as $(\mathcal{X}, \mathcal{A}, \gamma, P, r)$ where:

- $\mathcal{X}$ is a finite state space,
- $\mathcal{A}$ denotes the action space,
- $\gamma \in [0, 1)$ represents the discount factor,
- P stands for the transition function that maps state-action pairs $(x, a) \in \mathcal{X} \times \mathcal{A}$ to distributions over $\mathcal{X}$,
- $r : \mathcal{X} \times \mathcal{A} \to [-R_{\mathrm{MAX}}, R_{\mathrm{MAX}}]$ is the reward function.

A policy π is a mapping from $\mathcal{X}$ to a distribution over $\mathcal{A}$ such that $\pi(a|x)$ represents the probability of taking action a from the state x. A Q-function Q maps each state-action pair (x, a) to a real value in $\mathbb{R}$; notably, the reward function r itself is a Q-function.

For a given policy π, we define the operator P^π as follows:

$$\left(P^\pi Q\right)(x, a) := \sum_{x' \in \mathcal{X}} \sum_{a' \in \mathcal{A}} P\left(x' \mid x, a\right) \pi\left(a' \mid x'\right) Q\left(x', a'\right)$$

For a policy π we will denote by Q^π the **value function**, which describes the expected discounted sum of rewards associated with following π from a given state-action pair. We can write this as:

$$Q^\pi := \sum_{t \geq 0} \gamma^t \left(P^\pi\right)^t r$$

We can explicit this relation:

If we consider the expected discounted reward at time $t = 1$, initiated from an initial state-action pair (x, a) at time $t = 0$, we get that it is equal to:

$$\gamma \sum_{x' \in \mathcal{X}} \sum_{a' \in \mathcal{A}} P\left(x' \mid x, a\right) \pi\left(a' \mid x'\right) r\left(x', a'\right)$$

which corresponds exactly to $(\gamma P^\pi r)(x, a)$. By immediate induction, we get that the expected of rewards at associated with following π from a given state-action pair (x, a) at time t is $\gamma^t \left(P^\pi\right)^t r$, hence the definition of the value function.

1.1 Bellmann Operator

We now introduce the notion of **Bellmann Operator** $\mathcal{T}^\pi$ for a policy π, which is defined as follows:

Definition: Bellmann Operator

$$\mathcal{T}^{\pi} Q := r + \gamma P^{\pi} Q$$

Furthermore, in addition to the content presented in the article, we provide a short proof for the following result:

Result: Fixed point of the Bellmann Operator

The fixed point of the Bellmann Operator is Q^{π}, i.e. $\mathcal{T}^{\pi} Q^{\pi} = Q^{\pi} = \left(I - \gamma P^{\pi}\right)^{-1} r$

Proof: From the definition of the value function Q^{π}, we have that:

$$\mathcal{T}^{\pi} Q^{\pi} = r + \gamma P^{\pi} Q^{\pi} = r + \gamma P^{\pi} \sum_{t \geq 0} \gamma^{t} \left(P^{\pi}\right)^{t} r = r + \sum_{t \geq 0} \gamma^{t+1} \left(P^{\pi}\right)^{t+1} r = r + \underbrace{\sum_{t \geq 1} \gamma^{t} \left(P^{\pi}\right)^{t} r}_{= \sum_{t \geq 0} \gamma^{t} (P^{\pi})^{t} r - r = Q^{\pi} - r} = Q^{\pi}$$

Therefore $\mathcal{T}^{\pi} Q^{\pi} = Q^{\pi}$ that is Q^{π} is a fixed point of the Bellmann operator.

For the second equality, we have that $Q^{\pi} = \sum_{t \geq 0} \left(\gamma P^{\pi}\right)^{t} r$ with $||\gamma P^{\pi}|| < 1$. Therefore, the Neumann series $\sum_{t \geq 0} \left(\gamma P^{\pi}\right)^{t}$ converges in the operator norm. Moreover $I - \gamma P^{\pi}$ is invertible with

$$\left(I - \gamma P^{\pi}\right)^{-1} = \sum_{t \geq 0} \left(\gamma P^{\pi}\right)^{t}$$

.
Hence $Q^{\pi} = (I - \gamma P^{\pi})^{-1} r$.

□

1.2 Bellmann optimality operator

We also define the **Bellmann optimality operator** as

Definition: Bellmann optimality operator

$$\mathcal{T} Q := r + \gamma \max_{\pi} P^{\pi} Q.$$

which corresponds to the maximization of the Bellman operator over the set of policies, and whose fixed point Q^{*} is the unique optimal value function that we will be looking to obtain when talking about the "control setting".

1.3 Returned-based operators

We now introduce the notion of **Returned-based operators**. More precisely, the λ-return extension of the Bellman operator is defined as:

$$\mathcal{T}_{\lambda}^{\pi} Q := (1 - \lambda) \sum_{n \geq 0} \lambda^{n} \left[\left(\mathcal{T}^{\pi}\right)^{n} Q\right] = Q + \left(I - \lambda \gamma P^{\pi}\right)^{-1} \left(\mathcal{T}^{\pi} Q - Q\right)$$

where $\mathcal{T}^\pi Q - Q$ is called the **Bellman residual** of Q for policy π. Since Q^π is a fixed point of $\mathcal{T}^\pi$, we can immediately notice that the λ-return extension of the Bellman operator admits Q^π as fixed point.

We denote by μ the **behaviour policy**, and by π the **target policy** that we are trying to evaluate by using trajectories drawn from μ. In particular, the case $\pi = \mu$ corresponds to the **onpolicy** setting, while the other cases correspond to **offpolicy** settings. The trajectories we will be considering are of the form:

$$x_0 = x, a_0 = a, r_0, x_1, a_1, r_1, x_2, a_2, r_2, \ldots$$

with $a_t \sim \mu(\cdot \mid x_t), r_t = r(x_t, a_t)$ and $x_{t+1} \sim P(\cdot \mid x_t, a_t)$. We represent the sequence up to time t as $\mathcal{F}_t$, and denote the expectation with respect to both μ and the transition probabilities of the MDP as $\mathbb{E}_\mu$.

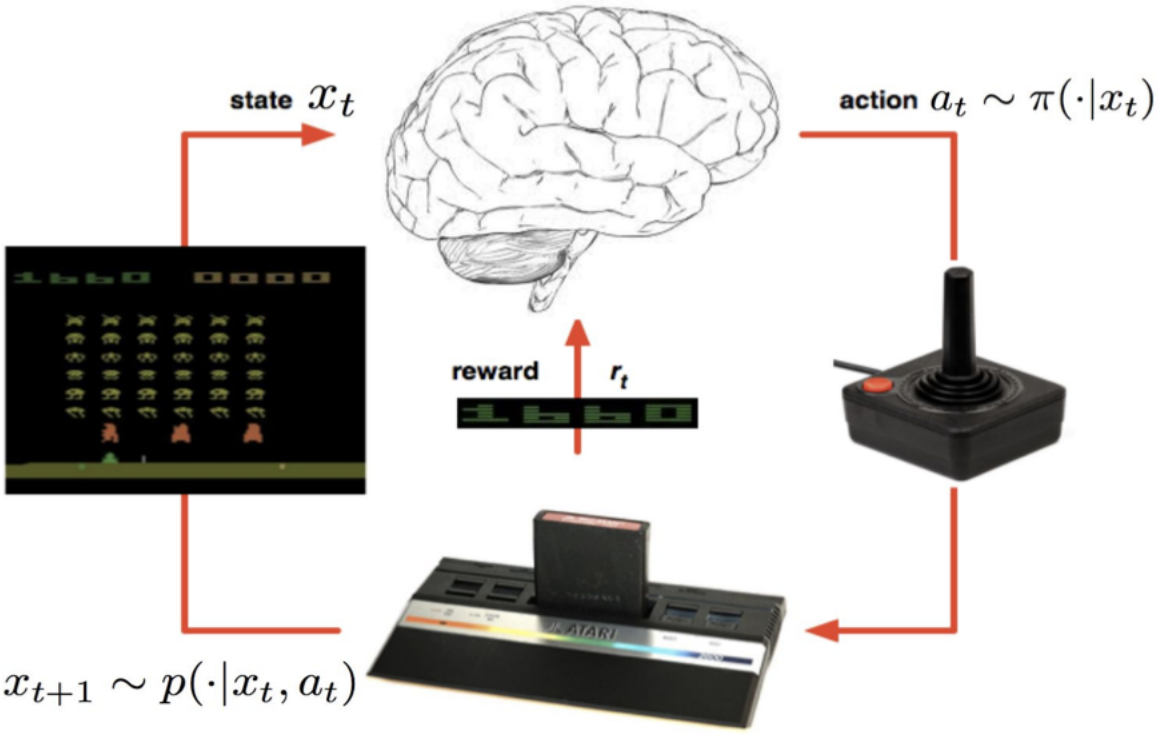
state x_t
action $a_t \sim \pi(\cdot|x_t)$
reward
r_t
1660
1660 0000
$x_{t+1} \sim p(\cdot|x_t, a_t)$
ATARI

Figure 1: The RL agent in its environment [2]

2 Eligibility traces

We introduce the notion of eligibility traces, that were briefly evoked in the article. Eligibility traces are a fundamental mechanism in reinforcement learning, that allows to improve the performance of temporal difference methods. [5]
The trace marks the memory parameters linked to the event as eligible for experiencing learning adjustments. When a TD error occurs, only the states or actions deemed eligible are held accountable for the error.

The two main views for eligibility traces are the forward and the backward view.

The forward view: For each state visited, *we look forward* in time to update the states and better combine the rewards. Thus, as we move on in steps, we do not work with the preceding states anymore.

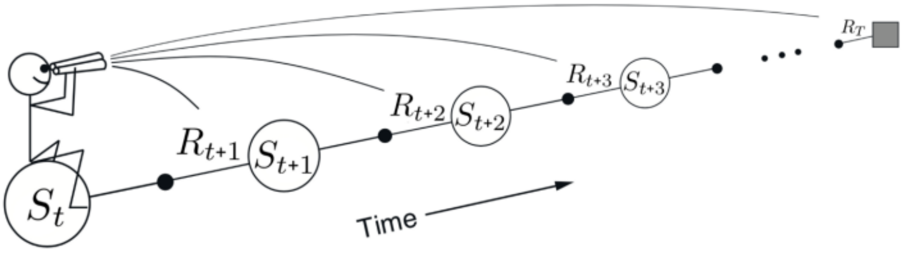

Figure 2: Eligibility traces - forward view [6]

The backward view: On the other hand, with the backward view of temporal difference, *we look backward* in time. For each time, we look at the current error and we assign it to each prior states that are eligibles.

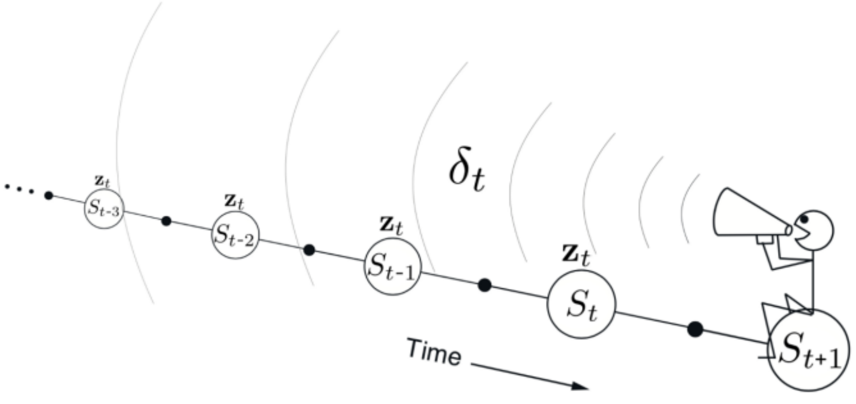

Figure 3: Eligibility traces - backward view [6]

We will see in the next section how the authors of the article extend this idea.

3 Off-policy Algorithms

The article focuses on two off-policy learning problems:

- The **policy evaluation** setting, where we want to estimate Q^{π} given a fixed policy π, using sample trajectories drawn from a behaviour policy μ,
- The **control** setting, where we seek to approximate Q^{*} by considering a sequence of policies that depend on our own sequence of Q-functions (e.g ϵ-greedy policies).

In order to compare several return-based off-policy algorithms, we introduce the following general operator $\mathcal{R}$:

Definition: General operator $\mathcal{R}$

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

where $\mathbb{E}_\pi Q(x,\cdot) := \sum_a \pi(a \mid x) Q(x,a)$ and where the coefficients (c_s) are non-negative and called **traces** of the operator. The choice of this nomenclature can be seen as an extension of the idea of **eligibility traces** that were presented in the previous section. As we will see, those coefficients add a possibility to enhance the algorithm by manipulating their expression.

First, we prove the following result:

Result: Fixed point of $\mathcal{R}$

Q^π is a fixed point of $\mathcal{R}$

Proof:

$$\begin{aligned}
\mathcal{R}Q^\pi(x,a) - Q^\pi(x,a) &= \mathbb{E}_\mu \left[\sum_{t\geq 0} \gamma^t \left(\prod_{s=1}^{t} c_s \right) \left(r_t + \gamma \mathbb{E}_\pi Q^\pi\left(x_{t+1}, \cdot\right) - Q^\pi\left(x_t, a_t\right)\right) \right] \\
&= \sum_{t\geq 0} \gamma^t \mathbb{E}_{\substack{x_{1:t} \\ a_{1:t}}} \left[\left(\prod_{s=1}^{t} c_s \right) \mathbb{E}_{x_{t+1} \sim P(\cdot \mid x_t, a_t)} \left[r_t + \gamma \mathbb{E}_\pi Q^\pi\left(x_{t+1}, \cdot\right) - Q^\pi\left(x_t, a_t\right)\right) \right] \\
&= \sum_{t\geq 0} \gamma^t \mathbb{E}_{\substack{x_{1:t} \\ a_{1:t}}} \left[\left(\prod_{s=1}^{t} c_s \right) \left(r_t\left(x_t, a_t\right) + \gamma \sum_{x' \in \mathcal{X}} \sum_{a' \in \mathcal{A}} P\left(x' \mid x_t, a_t\right) \pi\left(a' \mid x'\right) Q^\pi\left(x', a'\right) - Q^\pi\left(x_t, a_t\right)\right) \right] \\
&= \sum_{t\geq 0} \gamma^t \mathbb{E}_{\substack{x_{1:t} \\ a_{1:t}}} \left[\left(\prod_{s=1}^{t} c_s \right) \left(r_t\left(x_t, a_t\right) + \gamma P^\pi Q^\pi\left(x_t, a_t\right) - Q^\pi\left(x_t, a_t\right)\right) \right] \\
&= \sum_{t\geq 0} \gamma^t \mathbb{E}_{\substack{x_{1:t} \\ a_{1:t}}} \left[\left(\prod_{s=1}^{t} c_s \right) \underbrace{\left(\mathcal{T}^\pi Q^\pi - Q^\pi\right)\left(x_t, a_t\right)}_{=0} \right] \\
&= 0
\end{aligned}$$

since Q^π is a fixed point of $\mathcal{T}^\pi$.

□

3.1 Tradeoff for trace coefficients c_s

The choice of the trace coefficient c_s can be done on the basis of the contraction coefficient and the variance of the estimate.

We define the **contraction coefficient of the expected operator** :

$$\eta := \gamma - (1-\gamma)\mathbb{E}_\mu\left[\sum_{t\geq 1}\gamma^t(c_1...c_t)\right] \in [0,\gamma]$$

We have :

- When $c_s = 0$, $\eta = \gamma - (1-\gamma)\times 0 = \gamma$. This is called one-step Bellman update.
- When $c_s = 1$, $\eta = \gamma - (1-\gamma)\times\sum_{t\geq 1}\gamma^t = \gamma - (1-\gamma)\times\frac{\gamma}{1-\gamma} = 0$. This is called full Monte-Carlo rollouts.

The variance of the estimate depends on the value of c_s.
When c_s is large, the estimate uses multi-steps returns, but the variance is large. On the other hand, a small c_s gives a low variance, but the estimate does not use multi-steps returns. One must notice that the variance of the estimate can be infinite for $c_s = \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}$.

3.2 Definition of the algorithms

Let's now focus on the different types of choices for the coefficients (c_s) that are mentioned in the article:

- The case $c_s = \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}$ corresponds to **Importance sampling (IS)**.
- The case $c_s = \lambda$ corresponds to **Off-policy $\mathbf{Q}^\pi(\lambda)$ and $\mathbf{Q}^*(\lambda)$**
- The case $c_s = \lambda\pi(a_s \mid x_s)$ corresponds to **Tree-backup** TB(λ)
- The case $c_s = \lambda\min(1, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)})$ corresponds to **Retrace**(λ) that is the novel algorithm derived in the article

Let's take a closer look at those algorithms.

We first state and prove the following Lemma, which will be useful in the subsequent sections of this report:

Lemma: difference between $\mathcal{R}Q$ and its fixed point Q^π

$$\mathcal{R}Q(x,a) - Q^\pi(x,a) = \mathbb{E}_\mu\left[\sum_{t\geq 1}\gamma^t\left(\prod_{i=1}^{t-1}c_i\right)\left(\left[\mathbb{E}_\pi\left[\left(Q-Q^\pi\right)(x_t,\cdot)\right] - c_t\left(Q-Q^\pi\right)(x_t,a_t)\right]\right)\right]$$

Proof:

We have that:

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

since the term $t = 0$ in the second sum corresponds to $\mathbb{E}_\mu[Q(x_0,a_0)] = Q(x,a)$

$$
\begin{aligned}
&= \mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^t\left(\prod_{s=1}^{t} c_s\right)\left(r_t + \gamma\mathbb{E}_\pi Q\left(x_{t+1},\cdot\right)\right) - \sum_{t\geq 0}\gamma^{t+1}\left(\prod_{s=1}^{t+1} c_s\right)Q\left(x_{t+1},a_{t+1}\right)\right] \\
&= \mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^t\left(\prod_{s=1}^{t} c_s\right)\left(r_t + \gamma\left[\mathbb{E}_\pi Q\left(x_{t+1},\cdot\right) - c_{t+1}Q\left(x_{t+1},a_{t+1}\right)\right)\right]\right].
\end{aligned}
$$

Now, using the result that Q^π is a fixed point of $\mathcal{R}$, ie $Q^\pi(x,a) = \mathcal{R}Q^\pi(x,a) = \mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^t\left(\prod_{s=1}^{t} c_s\right)\left(r_t + \gamma\left[\mathbb{E}_\pi Q^\pi\left(x_{t+1},\cdot\right) - c_{t+1}Q^\pi\left(x_{t+1},a_{t+1}\right)\right)\right]\right]$, we have that:

$$
\begin{aligned}
\mathcal{R}Q(x,a) - Q^\pi(x,a) &= \mathbb{E}_\mu\left[\sum_{t\geq 0}\gamma^t\left(\prod_{s=1}^{t} c_s\right)\left(r_t + \gamma\left[\mathbb{E}_\pi[\left(Q-Q^\pi\right)\left(x_{t+1},\cdot\right)] - c_{t+1}(Q-Q^\pi)\left(x_{t+1},a_{t+1}\right)\right)\right]\right] \\
&= \mathbb{E}_\mu\left[\sum_{t\geq 1}\gamma^t\left(\prod_{i=1}^{t-1} c_i\right)\left(\left[\mathbb{E}_\pi\left[\left(Q-Q^\pi\right)\left(x_t,\cdot\right)\right] - c_t\left(Q-Q^\pi\right)\left(x_t,a_t\right)\right]\right)\right]
\end{aligned}
$$

which concludes the proof of the lemma.

□

We can now use this Lemma to show that for the choice of (c_s) made in Important Sampling, we have the following corollary:

Corollary: $\mathcal{R}Q$ for IS

$\mathcal{R}Q$ yields Q^π for any Q in the case of Importance Sampling.

Proof:

We want to show that

$$\mathcal{R}Q(x,a) - Q^\pi(x,a) = 0 \quad \forall Q$$

i.e., using Lemma 1,

$$\mathbb{E}_\mu\left[\sum_{t\geq 1}\gamma^t\left(\prod_{i=1}^{t-1} c_i\right)\left(\left[\mathbb{E}_\pi\left[\Delta Q\left(x_t,\cdot\right)\right] - c_t\Delta Q\left(x_t,a_t\right)\right]\right)\right] = 0 \quad \forall Q \quad \text{with} \quad \Delta Q := Q - Q^\pi$$

.

$$
\begin{aligned}
\mathcal{R}Q(x,a) - Q^{\pi}(x,a) &= \sum_{t\geq 1} \gamma^t \mathbb{E}_{\substack{x_{1:t}\\a_{1:t}}} \left[\left(\prod_{i=1}^{t-1} c_i \right) \left(\left[\mathbb{E}_{\pi} \Delta Q\left(x_t, \cdot\right) - c_t \Delta Q\left(x_t, a_t\right) \right] \right) \right] \\
&= \sum_{t\geq 1} \gamma^t \mathbb{E}_{\substack{x_{1:t}\\a_{1:t}}} \left[\left(\prod_{i=1}^{t-1} c_i \right) \left(\left[\mathbb{E}_{\pi} \Delta Q\left(x_t, \cdot\right) - \mathbb{E}_{a_t}\left[c_t\left(a_t, \mathcal{F}_t\right) \Delta Q\left(x_t, a_t\right) \mid \mathcal{F}_t \right] \right] \right) \right] \\
&= \sum_{t\geq 1} \gamma^t \mathbb{E}_{\substack{x_{1:t}\\a_{1:t}}} \left[\left(\prod_{i=1}^{t-1} c_i \right) \sum_{b} \left(\pi\left(b \mid x_t\right) - \mu\left(b \mid x_t\right) c_t\left(b, \mathcal{F}_t\right) \right) \Delta Q\left(x_t, b\right) \right] \quad \text{as} \quad a_t \sim \mu\left(\cdot \mid x_t\right)
\end{aligned}
$$

For importance sampling, the traces of the operator are defined as $c_s = \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)} \quad \forall s$. This yields $\pi\left(a \mid x_t\right) - \mu\left(a \mid x_t\right) c_t\left(b, \mathcal{F}_t\right) = 0$. As a consequence, $\mathcal{R}Q(x,a) - Q^{\pi}(x,a) = 0 \quad \forall Q$

□

3.3 Comparison between the algorithms

- **Importance Sampling** is a simple way to compensate the difference between μ policy and π policy when we want to evaluate the Q-value function. It uses the product of the likelihood ratios between π and μ. However, this method increases the variance of the estimate since $\frac{\pi}{\mu}$ may take big values.

- The **Off-policy** $Q^{\pi}(\lambda)$ **and** $Q^{*}(\lambda)$ method converges to Q^{π} exponentially fast when μ and π are close enough. As a result, it avoids the blow-up of the variance of the product of ratios that we encounter with IS. Nevertheless, the convergence also needs a maximality condition on λ which is a restrictive condition.

- The **Tree-backup** algorithm corrects the behaviour discrepancy by multiplying each term of the sum by the product of target policy probabilities. The estimates defines a contraction mapping for any policy μ and π. It is safe since it does not require the knowledge of the behavior policy μ and converges for arbitrary μ and and π. This algorithm however is not efficient when μ and π are similar (near on-policy case).

- As for **Retrace**, it takes the best of the three previous algorithm. In the next sub-section, we'll be taking a closer look at the benefits of this algorithm.

3.4 Interest of Retrace algorithm

3.4.1 Description of the algorithm

The **Retrace(**λ**)** algorithm developped in the article takes the best of the three previous algorithm. It uses an **Importance sampling** ratio truncated at 1. It then does not suffer from the variance explosion of the product of IS ratio. In the near on-policy case, like in $Q^{\pi}(\lambda)$, it does not cut the traces. In the off-policy case, the traces are safely cut like in **TB(λ)**. The algorithm is thus safe in on and off policy cases.

Based on the benefits and drawbacks of each algorithm, Retrace(λ) uses the coefficient c_s defined by :

$$c_s = \lambda \min(1, \frac{\pi\left(a_s \mid x_s\right)}{\mu\left(a_s \mid x_s\right)})$$

We list in the following table the algorithms and their drawbacks: [2]

Algorithm:	Trace coefficient:	Problem:
IS	$c_s = \frac{\pi(a_s\|x_s)}{\mu(a_s\|x_s)}$	high variance
$Q^\pi(\lambda)$	$c_s = \lambda$	not safe (off-policy)
TB(λ)	$c_s = \lambda\pi(a_s\|x_s)$	not efficient (on-policy)
Retrace(λ)	$c_s = \lambda \min\left(1, \frac{\pi(a_s\|x_s)}{\mu(a_s\|x_s)}\right)$	none!

3.4.2 Policy evaluation

We fix a target policy π and a behaviour policy μ.

We want to evaluate the state-value function Q^π when using the Retrace(λ) algorithm.

Through the following theorem, we state that the γ-contraction of the operator $\mathcal{R}$ defined by any set of non-negative coefficients $c_s = c_s(a_s, \mathcal{F}_s) \in \left[0, \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}\right]$:

Theorem 1

The operator $\mathcal{R}$ has unique fixed point Q^π. Furthermore, if $\forall a_s \in \mathcal{A}$ and $\forall$ 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 $\mathcal{Q}$-function Q

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

Proof:

We proved earlier that Q^π is the fixed point of $\mathcal{R}$ (see 1.1).

Now, using the lemma of part 1.1, with $\Delta Q := Q - Q^\pi$, we have:

$$
\begin{aligned}
\mathcal{R}Q(x,a) - Q^\pi(x,a) &= \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t-1}} \left[\left(\prod_{s=1}^{t-1} \right) \left(\mathbb{E}_\pi \Delta Q(x_t,.) - c_t \Delta Q(x_t,a_t) \right) \right] \\
&= \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t-1}} \left[\left(\prod_{s=1}^{t-1} \right) \left(\mathbb{E}_\pi \Delta Q(x_t,.) - \mathbb{E}_{a_t} \left[c_t(a_t,\mathcal{F}_t) \Delta Q(x_t,a_t) | \mathcal{F}_t \right] \right) \right] \\
&= \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t-1}} \left[\left(\prod_{s=1}^{t-1} c_s \right) \sum_b (\pi(b|x_t) - \mu(b|x_t) c_t(b,\mathcal{F}_t)) \Delta Q(x_t,b) \right] \\
&= \sum_{y,b} w_{y,b} \Delta Q(y,b)
\end{aligned}
$$

where $w_{y,b} := \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t-1}} \left[\left(\prod_{s=1}^{t-1} c_s \right) (\pi(b|x_t) - \mu(b|x_t) c_t(b,\mathcal{F}_t)) \mathbb{1}\{x_t = y\}) \right]$.

We chose $0 \leq c_s \leq \frac{\pi}{\mu}$, therefore we have $w_{y,b} \geq 0$. Then:

$$
\begin{aligned}
\sum_{y,b} w_{y,b} &= \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t-1}} \left[\left(\prod_{i=1}^{t-1} c_i \right) \sum_b (\pi(b|x_t) - \mu(b|x_t) c_t(b,\mathcal{F}_t)) \right] \\
&= \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t-1}} \left[\left(\prod_{i=1}^{t-1} c_i \right) \mathbb{E}_{a_t} \left[1 - c_t(a_t,\mathcal{F}_t) | \mathcal{F}_t \right] \right] \\
&= \sum_{t\geq 1} \gamma^t \mathbb{E}_{x_{1:t},a_{1:t}} \left[\left(\prod_{i=1}^{t-1} c_i \right) (1 - c_t) \right] \\
&= \mathbb{E}_\mu \left[\sum_{t\geq 1} \gamma^t \left(\prod_{i=1}^{t-1} c_i \right) - \sum_{t\geq 1} \gamma^t \left(\prod_{i=1}^{t} c_i \right) \right] \\
&= \gamma C - (C-1)
\end{aligned}
$$

With $C := \mathbb{E}_\mu \left[\sum_{t\geq 0} \gamma^t \left(\prod_{i=1}^t c_i \right) \right]$

We have $C \geq 1$, thus $\sum_{y,b} w_{y,b} \leq \gamma$.

Then :

$$\begin{aligned}|\mathcal{R}Q(x,a) - Q^{\pi}(x,a)| &\leq \sum_{y,b} |w_{y,b}||\Delta Q(y,b)| \\ &\leq \left(\sum_{y,b} |w_{y,b}|\right) ||\Delta Q|| \\ &= \left(\sum_{y,b} w_{y,b}\right) ||\Delta Q|| \\ &= \gamma ||\Delta Q||\end{aligned}$$

So we have :

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

□

The theorem means that $\mathcal{R}$ is a contraction mapping around Q^{π}.

3.4.3 Control

In the control policy, the target π is not fixed, we replace it by a sequence of policies (π_k) depending on (Q_k) and the goal is to find the optimal policy π^*. We now define a class of *increasingly greedy* sequences.

Definition: Increasingly greedy sequence

We say that a sequence of policies $(\pi_k : k \in \mathbb{N})$ is increasingly greedy with regard to a sequence $(Q_k : k \in \mathbb{N})$ of Q-functions if the following property holds for all k: $P^{\pi_{k+1}} Q_{k+1} \geq P^{\pi_k} Q_{k+1}$.

Lemma 2

Let ε_k be a non increasing sequence. Then, the sequence of policies (π_k) which are ε_k−greedy with respect to the sequence (Q_k) is increasingly greedy with respect to that sequence.

Proof:

The definition of an ε−greedy policy gives us :

$$\begin{aligned}P^{\pi_{k+1}} Q_{k+1}(x,a) &= \sum_y p(y|x,a)[(1-\varepsilon_{k+1})\max_b Q_{k+1}(y,b) + \varepsilon_{k+1}\frac{1}{A}\sum_b Q_{k+1}(y,b)] \\ &\underbrace{\geq}_{\varepsilon_{k+1} \leq \varepsilon_k} \sum_y p(y|x,a)[(1-\varepsilon_k)\max_b Q_{k+1}(y,b) + \varepsilon_k \frac{1}{A}\sum_b Q_{k+1}(y,b)] \\ &\geq \sum_y p(y|x,a)[(1-\varepsilon_k)Q_{k+1}(y, \max_b Q_k(y,b)) + \varepsilon_k \frac{1}{A}\sum_b Q_{k+1}(y,b)] \\ &= P^{\pi_k} Q_{k+1}\end{aligned}$$

□

The value of ε starts high, allowing for more exploration, and decreases over time, making the policy increasingly greedy.

Lemma 3

Let (β_k) be a non-decreasing sequence of soft-max parameters. Then the sequence of policies (π_k) which are soft-max (with parameter (β_k)) with respect to the sequence of functions (Q_k) is increasingly greedy with respect to that sequence.

Proof:

For any Q and y, define $\pi_\beta(b) = \frac{e^{\beta Q(y,b)}}{\sum_{b'} e^{\beta Q(y,b')}}$ and $f(\beta) = \sum_b \pi_\beta(b) Q(y,b)$. Then, we have

$$\begin{aligned} f'(\beta) &= \sum_b [\pi_\beta(b)Q(y,b) - \pi_\beta(b)\sum_{b'}\pi_\beta(b')Q(y,b')]Q(y,b) \\ &= \sum_b \pi_\beta(b)Q(y,b)^2 - \left(\sum_b \pi_\beta(b)Q(y,b)\right)^2 \\ &= \mathbb{V}_{b\sim\pi_\beta}[Q(y,b)] \geq 0. \end{aligned}$$

Thus $\beta \mapsto f(\beta)$ is a non decreasing function, and since $\beta_{k+1} \geq \beta_k$, we have :

$$\begin{aligned} P^{\pi_{k+1}}Q_{k+1}(x,a) &= \sum_y p(y|x,a) \sum_b \frac{e^{\beta_{k+1}Q_{k+1}(y,b)}}{\sum_{b'} e^{\beta_{k+1}Q_{k+1}(y,b')}} Q_{k+1}(y,b) \\ &\underbrace{\geq}_{\beta_{k+1}\geq\beta_k} \sum_y p(y|x,a) \sum_b \frac{e^{\beta_k Q_{k+1}(y,b)}}{\sum_{b'} e^{\beta_k Q_{k+1}(y,b')}} Q_{k+1}(y,b) \\ &= P^{\pi_k}Q_{k+1}(x,a). \end{aligned}$$

□

In other words, this policy transforms the values of the expected rewards for each action into a probability distribution. The sequence of policies (π_k) , which are softmax with parameter (β_k) with respect to the sequence of functions (Q_k), being "increasingly greedy" with respect to that sequence, means that as k increases, the policies increasingly favor the actions that the current Q_k function estimates to be the best, while still allowing for some exploration based on the softmax distribution influenced by β_k . Therefore, the agent becomes more confident in its decision-making, and more likely to choose the action with the maximum expected reward.

This approach helps in finding a balance between exploring to find better action-value estimates and exploiting the current knowledge to maximize returns.

Another example of increasingly greedy policy is the **Upper-Confidence Bound (UCB)** that we already saw in class [4]. The UCB policy selects actions based on the confidence intervals of the expected rewards for each action and state, rather than the expected rewards themselves. At each step in the UCB policy, the agent selects the action with the highest UCB, i.e

$$\operatorname{argmax}_a \left(Q_k(x,a) + c\sqrt{\frac{ln(k)}{N_k(a)}} \right).$$

Where :

- $Q_k(x,a)$ is the current estimate for action a and state x at time k.
- $N_k(a)$ is the number of times action a is taken.

The addition of the confidence term accounts for the inherent uncertainty in the precision of the Q function estimates, which arises from using a sampled set of rewards. Furthermore, it guarantees that the UCB policy undertakes exploration of the environment and attempts new actions, regardless of their potentially lower anticipated rewards.

Example of increasingly greedy policy: UCB

The UCB policy is increasingly greedy policy.

Proof:

We denote $s^*_{k+1} = \operatorname{argmax}_s \left(Q_{k+1}(x,a) + c\sqrt{\frac{ln(k+1)}{N_{k+1}(a)}} \right)$ and $s^*_k = \operatorname{argmax}_s \left(Q_k(x,a) + c\sqrt{\frac{ln(k)}{N_k(a)}} \right)$

Then follows :

$$\begin{aligned} P^{\pi_{k+1}} Q_{k+1}(x,a) &= \sum_y P(y|x,a) \sum_s \pi_{k+1}(s|y) Q_{k+1}(y,s) \\ &= \sum_y P(y|x,a) Q_{k+1}(y, s^*_{k+1}) \\ &\geq \sum_y P(y|x,a) Q_{k+1}(y, s^*_k) \\ &= P^{\pi_k} Q_{k+1}(x,a). \end{aligned}$$

□

We are interested now in the study of the convergence in the control case: First, we assume that $c_s = c_s(as, \mathcal{F}_s)$ is Markovian, i.e it does not depend on the full past history, but rather only on x_s, a_s $c_s = c(a_s, x_s)$. This allows us to define the (sub)-probability transition operator

$$(P^{c\mu}Q)(x,a) := \sum_{x'} \sum_{a'} p(x'|x,a) \mu(a'|x') c(a',x') Q(x',a')$$

A second additional assumption for the convergence in the control case, we assume that Q_0 satisfies $\mathcal{T}^{\pi_0} Q_0 \geq Q_0$ (this can be achieved by a pessimistic initialization $Q_0 = \frac{-R_{MAX}}{(1-\gamma)}$).

Theorem 2

Consider an arbitrary sequence of behaviour policies (μ_k) (which may depend on (Q_k)) and a sequence of target policies (π_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 (π_k) and μ_k and a Markovian $c_s = c(a_s, x_s) \in \left[0, \frac{\pi_k(a_s|x_s)}{\mu_k(a_s|x_s)}\right]$. Assume the target policies π_k are ε_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 $\varepsilon_k \longrightarrow 0$ then $Q_k \longrightarrow Q^*$.

Proof:

The proof can be found in the appendix of the paper [3].

3.4.4 Online algorithms

In this part, we will explore online algorithms designed to learn from observed sample paths. Our focus will be on the Retrace(λ) algorithm, which utilizes a specific coefficient c defined by $c = \lambda \min\left(1, \frac{\pi}{\mu}\right)$. This definition of c enables us to express $P^{c\mu}$ in the form of

$$\lambda P^{\pi \wedge \mu} := \sum_y p(y \mid x, a) \sum_b \min(\pi(b \mid y), \mu(b \mid y)) Q(y, b).$$

It's important to note that the retrace operator is given by:

$$\mathcal{R}Q = Q + (I - \lambda\gamma P^{\pi \wedge \mu})^{-1} (\mathcal{T}^\pi Q - Q).$$

This formulation is essential for understanding the mechanism through which the algorithm adjusts its learning process based on the sample trajectories.

We first state the Robbins-Munro conditions, which will be useful for the next theorem.

Definition: Robbins-Munro conditions

Let (α_n) be a sequence of step sizes. The Robbins-Munro conditions for (α_n) are given by:

$$1.\ \sum_{n=1}^{\infty} \alpha_n = \infty$$

$$2.\ \sum_{n=1}^{\infty} \alpha_n^2 < \infty$$

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 $\mu_k : a_t \sim \mu_k(\cdot \mid x_t)$. For each (x, a) along this trajectory, with s being the time of first occurence of (x, a), update

$$Q_{k+1}(x,a) \leftarrow Q_k(x,a) + \alpha_k \sum_{t \geq s} \delta_t^{\pi_k} \sum_{j=s}^{t} \gamma^{t-j} \left(\prod_{i=j+1}^{t} c_i \right) \mathbb{I}\{x_j, a_j = x, a\} \tag{2}$$

where $\delta_t^{\pi_k} := r_t + \gamma \mathbb{E}_{\pi_k} Q_k(x_{t+1}, .) - Q_k(x_t, a_t), \alpha_k = \alpha_k(x_s, a_s)$. We consider the Retrace (λ) algorithm where $c_i = \lambda \min\left(1, \frac{\pi(a_i|x_i)}{\mu(a_i|x_i)}\right)$. Assume that (π_k) are increasingly greedy w.r.t (Q_k) and are each ϵ_k-away from the greedy policies (π_{Q_k}), i.e. $\max_x \|\pi_k(\cdot \mid x) - \pi_{Q_k}(\cdot \mid x)\|_1 \leq \epsilon_k$, with $\epsilon_k \to 0$. Assume that P^{π_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: $\sum_{t \geq 0} \mathbb{P}\{x_t, a_t = x, a\} \geq D > 0$,
2. The sample trajectories are finite in terms of the second moment of their lengths T_k :

$$\mathbb{E}_{\mu_k} T_k^2 < \infty,$$

3. The stepsizes obey the usual Robbins-Munro conditions.

Then,

$$Q_k \to Q^* \text{ a.s.}$$

Proof:

Prior to unveiling the theorem's proof, we shall initially introduce several pertinent lemmas and findings, mirroring those documented in the appendix of paper [3].

Lemma 4

Let (π_k) and (μ_k) two sequences of policies. If there exists α such that for all x, a

$$\min(\pi_k(a|x), \mu_k(a|x)) = \alpha \pi(k)(a|x) + o(1), \tag{1}$$

Then the transition matrices P^{π_k} and $P^{\pi_k \wedge \mu_k}$ asymptotically commute :

$$||P^{\pi_k} P^{\pi_k \wedge \mu_k} - P^{\pi_k \wedge \mu_k} P^{\pi_k}|| = o(1)$$

Proof:

For any Q, we have :

$$\begin{aligned}
\left(P^{\pi_k} P^{\pi_k \wedge \mu_k}\right) Q(x,a) &= \sum_y p(y \mid x,a) \sum_b \pi_k(b \mid y) \sum_z p(z \mid y,b) \sum_c (\pi_k \wedge \mu_k)(c \mid z) Q(z,c) \\
&= \alpha \sum_y p(y \mid x,a) \sum_b \pi_k(b \mid y) \sum_z p(z \mid y,b) \sum_c \pi_k(c \mid z) Q(z,c) + \|Q\| o(1) \\
&= \sum_y p(y \mid x,a) \sum_b (\pi_k \wedge \mu_k)(b \mid y) \sum_z p(z \mid y,b) \sum_c \pi_k(c \mid z) Q(z,c) + \|Q\| o(1) \\
&= \left(P^{\pi_k \wedge \mu_k} P^{\pi_k}\right) Q(x,a) + \|Q\| o(1)
\end{aligned}$$

End of Proof Lemma 4 □

Lemma 5

Let (π_{Q_k}) a sequence of (deterministic) greedy policies w.r.t. a sequence (Q_k). Let (π_k) a sequence of policies that are ϵ_k away from (π_{Q_k}), in the sense that, for all x,

$$\left\|\pi_k(. \mid x) - \pi_{Q_k}(x)\right\|_1 := 1 - \pi_k\left(\pi_{Q_k(x)|x}\right) + \sum_{a \neq \pi_{Q_k}(x)} \pi_k(a \mid x) \leq \epsilon_k$$

Let μ_k a sequence of policies defined by:

$$\mu_k(a \mid x) = \frac{\alpha \mu(a \mid x)}{1 - \mu\left(\pi_{Q_k(x)|x}\right)} \mathbb{1}\left\{a \neq \pi_{Q_k}(x)\right\} + (1-\alpha) \mathbb{1}\left\{a = \pi_{Q_k}(x)\right\}$$

for some arbitrary policy μ and $\alpha \in [0,1]$. Assume $\epsilon_k \to 0$. Then the transition matrices P^{π_k} and $P^{\pi_k \wedge \mu_k}$ asymptotically commute.

Proof:

The intuition is that asymptotically π_k gets very close to the deterministic policy π_{Q_k}. In that case, the minimum distribution $(\pi_k \wedge \mu_k)(. \mid x)$ puts a mass close to $1-\alpha$ on the greedy action $\pi_{Q_k}(x)$, and no mass on other actions, thus $(\pi_k \wedge \mu_k)$ gets very close to $(1-\alpha)\pi_k$, and Lemma 4 applies (with multiplicative constant $(1-\alpha)$).

Indeed, from our assumption that π_k is ϵ-away from π_{Q_k} we have :

$$\pi_k\left(\pi_{Q_k}(x) \mid x\right) \geq 1 - \epsilon_k, \text{ and } \pi_k\left(a \neq \pi_{Q_k}(x) \mid x\right) \leq \epsilon_k$$

We deduce that

$$\begin{aligned}
\left(\pi_k \wedge \mu_k\right)\left(\pi_{Q_k}(x) \mid x\right) &= \min\left(\pi_k\left(\pi_{Q_k}(x) \mid x\right), 1-\alpha\right) \\
&= 1 - \alpha + O\left(\epsilon_k\right) \\
&= (1-\alpha)\pi_k\left(\pi_{Q_k}(x) \mid x\right) + O\left(\epsilon_k\right)
\end{aligned}$$

and

$$\begin{aligned}
\left(\pi_k \wedge \mu_k\right)\left(a \neq \pi_{Q_k}(x) \mid x\right) &= O\left(\epsilon_k\right) \\
&= (1-\alpha)\pi_k(a \mid x) + O\left(\epsilon_k\right)
\end{aligned}$$

Thus, Lemma 4 applies (with a multiplicative constant $(1-\alpha)$) and P^{π_k} and $P^{\pi_k \wedge \mu_k}$ asymptotically commute.

End of Proof Lemma 5 □

It is now important to recall a valuable proposition, specifically Proposition 4.5, as outlined by Bertsekas and Tsitsiklis (1996) [1], before we introduce a broader algorithm for the online context.

Proposition : Proposition 4.5 of Bertsekas and Tsitsiklis (1996)

Consider the iteration

$$r_{t+1} := (1-\gamma_t(x))\, r_t(x) + \gamma_t(x)\left((U r_t)(x) + \omega_t(x) + u_t(x)\right).$$

Let $\mathcal{F}_t = \{r_0(x), \ldots, r_t(x), \omega_0(x), \ldots, \omega_{t-1}(x), \gamma_0(x), \ldots, \gamma_t(x), \forall x\}$ be the entire history of the iteration. If

- $\gamma_t(i) \geq 0, \sum_{t=0}^{\infty} \gamma_t(i) = \infty, \sum_{t=0}^{\infty} \gamma_t^2(i) < \infty$
- For every i and t the noise terms $\omega_t(i)$ satisfy $\mathbb{E}\left[\omega_t(i) \mid \mathcal{F}_t\right] = 0$
- Given any norm $\|\cdot\|$ on $\mathbb{R}^n$, there exist constants A and B such that $\mathbb{E}\left[\omega_t^2(i) \mid \mathcal{F}_t\right] \leq A + B\left\|r_t\right\|^2$.
- There exists a vector r^*, a positive vector ξ, and a scalar $\beta \in [0,1)$, such that for all t,

$$\left\|U r_t - r^*\right\|_{\xi} \leq \beta \left\|r_t - r^*\right\|_{\xi}$$

- There exists a nonnegative random sequence θ_t that converges to zero with probability 1 and is such that for all t,

$$\left|u_t(x)\right| \leq \theta_t\left(\left\|r_t\right\|_{\xi} + 1\right)$$

Then r_t converges to r^* with probability 1 . The notation $\|.\|_{\xi}$ denotes a weighted maximum norm

$$\|A\|_{\xi} = \max_x \frac{|A(x)|}{\xi(x)}$$

Next, we will introduce a broader theorem on the convergence of online algorithms, which will subsequently be applied in the proof of Theorem 4.

Theorem 4

Consider the algorithm

$$Q_{k+1} = (1-\alpha_k(x,a))\, Q_k(x,a) + \alpha_k(x,a)\left(\mathcal{R}_k Q_k(x,a) + \omega_k(x,a) + \upsilon_k(x,a)\right), \tag{4}$$

and assume that

1. ω_k is a centered, $\mathcal{F}_k$-measurable noise term of bounded variance
2. υ_k is bounded from above by $\theta_k\left(\|Q_k\| + 1\right)$, where (θ_k) is a random sequence that converges to 0 a.s.

Then, under the same assumptions as in Theorem 2, we have $Q_k \to Q^*$ almost surely.

Proof:

We remind the reader that in our case, the retrace operator is defined as:

$$\mathcal{R}Q = Q + \left(I - \lambda\gamma P^{\pi\wedge\mu}\right)^{-1}\left(\mathcal{T}^{\pi} Q - Q\right)$$

Upper bound on $\mathcal{R}Q_k - Q^*$.

We have :

$$\begin{aligned}\mathcal{R}Q_k - Q^* &= Q_k - Q^* + \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[\mathcal{T}^{\pi_k} Q_k - Q_k\right]\\
&= \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[\mathcal{T}^{\pi_k} Q_k - Q_k + \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)\left(Q_k - Q^*\right)\right]\\
&= \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[\mathcal{T}^{\pi_k} Q_k - Q^* - \lambda\gamma P^{\pi_k \wedge \mu_k}\left(Q_k - Q^*\right)\right]\\
&= \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[\mathcal{T}^{\pi_k} Q_k - \mathcal{T}Q^* - \lambda\gamma P^{\pi_k \wedge \mu_k}\left(Q_k - Q^*\right)\right]\\
&\leq \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[\gamma P^{\pi_k}\left(Q_k - Q^*\right) - \lambda\gamma P^{\pi_k \wedge \mu_k}\left(Q_k - Q^*\right)\right]\\
&= \gamma\left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right]\left(Q_k - Q^*\right)\\
&= A_k\left(Q_k - Q^*\right)\end{aligned}$$

Where $A_k := \gamma\left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1}\left[P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right]$

In the above, we used the fact that $\mathcal{T}Q^* = Q^*$ and we also used :

$$\begin{aligned}\mathcal{T}^{\pi_k} Q_k - \mathcal{T}Q^* &= r + \gamma P^{\pi_k} Q_k - r - \max_{\pi} P^{\pi} Q^*\\
&= \gamma P^{\pi_k} Q_k - \max_{\pi} P^{\pi} Q^*\\
&\leq \gamma P^{\pi_k}\left(Q_k - Q^*\right)\end{aligned}$$

We also have : $A_k = \gamma \sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t \left(P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right)$. Let e be the vector with 1-components. We have :

$$\left(P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right) e(x,a) = \sum_{x'} \sum_{a'} p\left(x' \mid x, a\right)\left[\pi_k\left(a' \mid x'\right) - \lambda\left(\pi_k \wedge \mu_k\right)\left(a' \mid x'\right)\right] \geq 0$$

The expression above is greater or equal than 0 because we have for any traces (thus including the Retrace(λ)) : $0 \leq c_s \leq \frac{\pi(a_s|x_s)}{\mu(a_s|x_s)}$.

It becomes clear that A_k has only non-negative elements. Moreover, we have:

$$\begin{aligned}A_k e &= \gamma \sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t \left[P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right] e\\
&= \gamma \sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t e - \sum_{t \geq 0} \gamma^{t+1} \left(\lambda P^{\pi_k \wedge \mu_k}\right)^{t+1} e\\
&= \gamma \sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t e - \sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t e + e\\
&= e - (1-\gamma) \sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t e\\
&\leq \gamma e\end{aligned}$$

We used in the above the fact that : $P^{\pi_k} e = e$ and $\sum_{t \geq 0} \gamma^t \left(\lambda P^{\pi_k \wedge \mu_k}\right)^t e \geq e$.

Taking all the above into accounts, we deduce that :

$$\mathcal{R}Q_k - Q^* \leq \gamma \left\| Q_k - Q^* \right\| e. \tag{2}$$

Lower bound on $\mathcal{R}Q_k - Q^*$.

We have :

$$\begin{aligned}
\mathcal{R}Q_k &= Q_k + \left(I - \lambda\gamma P^{\pi_k \wedge \mu_k}\right)^{-1} \left[\mathcal{T}^{\pi_k} Q_k - Q_k\right] \\
&= Q_k + \sum_{i \geq 0} \gamma^i \left(\lambda P^{\pi_k \wedge \mu_k}\right)^i \left[\mathcal{T}^{\pi_k} Q_k - Q_k\right] \\
&= \mathcal{T}^{\pi_k} Q_k + \sum_{i \geq 1} \gamma^i \left(\lambda P^{\pi_k \wedge \mu_k}\right)^i \left[\mathcal{T}^{\pi_k} Q_k - Q_k\right] \\
&= \mathcal{T}^{\pi_k} Q_k + \gamma\lambda P^{\pi_k \wedge \mu_k} \left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1} \left[\mathcal{T}^{\pi_k} Q_k - Q_k\right]
\end{aligned}$$

Using the fact that : $\mathcal{T}^{\pi_k} Q_k \geq \mathcal{T} Q_k - \epsilon_k \left\| Q_k \right\| e \geq \mathcal{T}^{\pi^*} Q_k - \epsilon_k \left\| Q_k \right\| e$, we obtain :

$$\begin{aligned}
\mathcal{R}Q_k - Q^* &= Q_{k+1} - \mathcal{T}^{\pi_k} Q_k + \mathcal{T}^{\pi_k} Q_k - \mathcal{T}^{\pi^*} Q_k + \mathcal{T}^{\pi^*} Q_k - \mathcal{T}^{\pi^*} Q^* \\
&\geq Q_{k+1} - \mathcal{T}^{\pi_k} Q_k + \gamma P^{\pi^*} \left(Q_k - Q^*\right) - \epsilon_k \left\| Q_k \right\| e
\end{aligned}$$

Now, we obtain :

$$\mathcal{R}Q_k - Q^* \geq \gamma\lambda P^{\pi_k \wedge \mu_k} \left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1} \left(\mathcal{T}^{\pi_k} Q_k - Q_k\right) + \gamma P^{\pi^*} \left(Q_k - Q^*\right) - \epsilon_k \left\| Q_k \right\|. \tag{3}$$

Lower-bound on $\mathcal{T}^{\pi_k} Q_k - Q_k$.

This segment of the proof will remain identical to its presentation in the paper.

Since the sequence of policies (π_k) is increasingly greedy w.r.t. (Q_k), we have

$$\begin{aligned}
\mathcal{T}^{\pi_{k+1}} Q_{k+1} - Q_{k+1} \geq \mathcal{T}^{\pi_k} Q_{k+1} - Q_{k+1} & \\
&= \left(1 - \alpha_k\right) \mathcal{T}^{\pi_k} Q_k + \alpha_k \mathcal{T}^{\pi_k} \left(\mathcal{R}Q_k + w_k + \upsilon_k\right) - Q_{k+1} \\
&= \left(1 - \alpha_k\right) \left(\mathcal{T}^{\pi_k} Q_k - Q_k\right) + \alpha_k \left[\mathcal{T}^{\pi_k} \mathcal{R}Q_k - \mathcal{R}Q_k + \omega_k' + \upsilon_k'\right]
\end{aligned}$$

where $\omega_k' := \left(\gamma P^{\pi_k} - I\right) \omega_k$ and $\upsilon_k' := \left(\gamma P^{\pi_k} - I\right) \upsilon_k$. It is easy to see that both ω_k' and υ_k' continue to satisfy the assumptions on ω_k and υ_k. Indeed, ω_k' stays a centered, $\mathcal{F}_k$-measurable noise term of bounded variance, and υ_k' stays bounded from above by $\theta_k \left(\left\| Q_k \right\| + 1\right)$,

Now, we have

$$\begin{aligned}\mathcal{T}^{\pi_k}\mathcal{R}Q_k - \mathcal{R}Q_k &= r + (\gamma P^{\pi_k} - I)\,\mathcal{R}Q_k \\ &= r + (\gamma P^{\pi_k} - I)\left[Q_k + \left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1}\left(\mathcal{T}^{\pi_k}Q_k - Q_k\right)\right] \\ &= \mathcal{T}^{\pi_k}Q_k - Q_k + (\gamma P^{\pi_k} - I)\left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1}\left(\mathcal{T}^{\pi_k}Q_k - Q_k\right) \\ &= \gamma\left(P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right)\left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1}\left(\mathcal{T}^{\pi_k}Q_k - Q_k\right)\end{aligned}$$

Using this equality and the previous one and writing $\xi_k := \mathcal{T}^{\pi_k}Q_k - Q_k$, we have

$$\xi_{k+1} \geq (1 - \alpha_k)\,\xi_k + \alpha_k\left[B_k\xi_k + \omega'_k + \upsilon'_k\right] \tag{4}$$

where $B_k := \gamma\left(P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right)\left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1}$. The matrix B_k is non-negative but may not be contraction mapping (the sum of its components per row may be larger than 1). Thus we cannot directly apply Proposition 4.5 of Bertsekas and Tsitsiklis (1996) (see above for a reminder of the proposition).

However, as we have seen above, the matrix $A_k := \gamma\left(I - \gamma\lambda P^{\pi_k \wedge \mu_k}\right)^{-1}\left(P^{\pi_k} - \lambda P^{\pi_k \wedge \mu_k}\right)$ is a γ-contraction mapping. So now we relate B_k to A_k using our assumption that P^{π_k} and $P^{\pi_k \wedge \mu_k}$ commute asymptotically, i.e. $\|P^{\pi_k}P^{\pi_k \wedge \mu_k} - P^{\pi_k \wedge \mu_k}P^{\pi_k}\| = \eta_k$ with $\eta_k \to 0$.

For any (sub)-transition matrices U, and V, we have

$$\begin{aligned}U(I - \lambda\gamma V)^{-1} &= \sum_{t\geq 0}(\lambda\gamma)^t U V^t \\ &= \sum_{t\geq 0}(\lambda\gamma)^t\left[\sum_{s=0}^{t-1} V^s(UV - VU)V^{t-s-1} + V^t U\right] \\ &= (I - \lambda\gamma V)^{-1}U + \sum_{t\geq 0}(\lambda\gamma)^t\sum_{s=0}^{t-1} V^s(UV - VU)V^{t-s-1}\end{aligned}$$

Replacing U by P^{π_k} and V by $P^{\pi_k \wedge \mu_k}$, we deduce

$$\|B_k - A_k\| \leq \gamma\sum_{t\geq 0} t(\lambda\gamma)^t\eta_k = \gamma\frac{1}{(1-\lambda\gamma)^2}\eta_k$$

Thus,

$$\xi_{k+1} \geq (1 - \alpha_k)\,\xi_k + \alpha_k\left[A_k\xi_k + \omega'_k + \upsilon''_k\right] \tag{5}$$

where $\upsilon''_k := \upsilon'_k + \gamma\sum_{t\geq 0} t(\lambda\gamma)^t\eta_k\,\|\xi_k\|$ continues to satisfy the assumptions on υ'_k (since $\eta_k \to 0$) and therefore continue to verify the assumption on υ_k.

Now, let us define another sequence ξ'_k as follows: $\xi'_0 = \xi_0$ and

$$\xi'_{k+1} = (1-\alpha_k)\,\xi'_k + \alpha_k\left(A_k\xi'_k + \omega'_k + v''_k\right)$$

We can now apply Proposition 4.5 of Bertsekas and Tsitsiklis (1996) [1] to the sequence (ξ'_k). The matrices A_k are non-negative, and the sum of their coefficients per row is bounded by γ. This A_k are γ-contraction mappings and have the same fixed point which is 0 . The noise ω'_k is centered and $\mathcal{F}_k$-measurable and satisfies the bounded variance assumption, and v''_k is bounded above by $(1+\gamma)\theta'_k\left(\|Q_k\|+1\right)$ for some $\theta'_k \to 0$. Thus, $\lim_k \xi'_k = 0$ almost surely.

Now, it is straightforward to see that $\xi_k \geq \xi'_k$ for $k \geq 0$. Indeed by induction, let us assume that $\xi_k \geq \xi'_k$. Then

$$\begin{aligned}\xi_{k+1} &\geq (1-\alpha_k)\,\xi_k + \alpha_k\left(A_k\xi_k + \omega'_k + v''_k\right)\\ &\geq (1-\alpha_k)\,\xi'_k + \alpha_k\left(A_k\xi'_k + \omega'_k + v''_k\right)\\ &= \xi'_{k+1}\end{aligned}$$

since all elements of the matrix A_k are non-negative. Thus we deduce that

$$\lim_{k\to\infty}\inf \xi_k \geq \lim_{k\to\infty} \xi'_k = 0 \tag{6}$$

Conclusion. Using (8) in (4) we deduce the lower bound :

$$\lim_{k\to\infty}\inf \mathcal{R}Q_k - Q^* \geq \lim_{k\to\infty}\inf \gamma P^{\pi^*}\left(Q_k - Q^*\right) \tag{7}$$

almost surely. Now combining with the upper bound (3) we deduce that

$$\|\mathcal{R}Q_k - Q^*\| \leq \gamma\,\|Q_k - Q^*\| + O\left(\epsilon_k\,\|Q_k\|\right) + O\left(\xi_k\right). \tag{8}$$

The last two terms can be incorporated to the $v_k(x,a)$ and $\omega_k(x,a)$ terms, respectively; we thus again apply Propostion 4.5 of Bertsekas and Tsitsiklis (1996) [1] to the sequence (Q_k) defined in Theorem 4 and deduce that $Q_k \to Q^*$ almost surely.

End of Proof Theorem 4 □

We still need to rewrite the update from Theorem 3 to match the format of the update in Theorem 4.

Let $z^k_{s,t}$ denote the accumulating trace (Sutton and Barto, 1998) [6]

$$z^k_{s,t} := \sum_{j=s}^{t}\gamma^{t-j}\left(\prod_{i=j+1}^{t} c_i\right)\mathbb{I}\left\{(x_j,a_j)=(x_s,a_s)\right\}$$

Let us write $Q^o_{k+1}(x_s,a_s)$ to emphasize the online setting. Then the update of Theorem 3 can be written

$$Q^o_{k+1}\left(x_s, a_s\right) \leftarrow Q^o_k\left(x_s, a_s\right) + \alpha_k\left(x_s, a_s\right) \sum_{t \geq s} \delta_t^{\pi_k} z^k_{s,t} \tag{9}$$

$$\delta_t^{\pi_k} := r_t + \gamma \mathbb{E}_{\pi_k} Q^o_k\left(x_{t+1}, .\right) - Q^o_k\left(x_t, a_t\right)$$

Using our assumption on finite trajectories and $c_i \leq 1$, we can show that :

$$\mathbb{E}\left[\sum_{t \geq s} z^k_{s,t} \mid \mathcal{F}_k\right] < \mathbb{E}\left[T_k^2 \mid \mathcal{F}_k\right] < \infty \tag{10}$$

Where T_k denotes trajectory length. Now, let $D_k := D_k\left(x_s, a_s\right) := \sum\limits_{t \geq s} \mathbb{P}\left(x_t, a_t\right) = \left(x_s, a_s\right)$.

Then, using (10), we can show that the total update is bounded, and rewrite

$$\mathbb{E}_{\mu_k}\left[\sum_{t \geq s} \delta_t^{\pi_k} z^k_{s,t}\right] = D_k\left(x_s, a_s\right)\left(\mathcal{R}Q_k\left(x_s, a_s\right) - Q\left(x_s, a_s\right)\right)$$

Finally, using the above, and writing $\alpha_k = \alpha_k\left(x_s, a_s\right)$, (9) can be rewritten in the desired form :

$$Q^o_{k+1}\left(x_s, a_s\right) \leftarrow \left(1 - \tilde{\alpha_k}\right) Q^o_k\left(x_s, a_s\right) + \tilde{\alpha_k}\left(\mathcal{R}_k Q^o_k\left(x_s, a_s\right) + \omega_k\left(x_s, a_s\right) + v_k\left(x_s, a_s\right)\right), \tag{11}$$

$$\begin{aligned}
\omega_k\left(x_s, a_s\right) &:= \left(D_k\right)^{-1}\left(\sum_{t \geq s} \delta_t^{\pi_k} z^k_{s,t} - \mathbb{E}_{\mu_k}\left[\sum_{t \geq s} \delta_t^{\pi_k} z^k_{s,t}\right]\right), \\
v_k\left(x_s, a_s\right) &:= \left(\tilde{\alpha_k}\right)^{-1}\left(Q^o_{k+1}\left(x_s, a_s\right) - Q_{k+1}\left(x_s, a_s\right)\right), \\
\tilde{\alpha_k} &:= \alpha_k D_k.
\end{aligned}$$

It can be shown that the variance of the noise term ω_k is bounded, using (10) and the fact that the reward function is bounded. It follows from Assumptions 1-3 that the modified stepsize sequence $(\tilde{\alpha_k})$ satisfies the conditions of Assumption 1. The second noise term $v_k\left(x_s, a_s\right)$ measures the difference between online iterates and the corresponding offline values, and can be shown to satisfy the required assumption analogously to the argument in the proof of Proposition 5.2 in Bertsekas and Tsitsiklis (1996)[1].

The proof relies on the eligibility coefficients (10) and rewards being bounded, the trajectories being finite, and the conditions on the stepsizes being satisfied.

We can thus apply Theorem 4 to (11), and conclude that the iterates

$$\boxed{Q^o_k \rightarrow Q^*}$$

as $k \rightarrow \infty$, with probability 1.

End of Proof Theorem 3 □

4 Code

We now write a code in `Python` to reproduce the results in the paper, by simulating each of the four algorithms mentioned. The code is composed of 3 functions. The first calculates the trace, i.e. the c_s coefficients, the second repeats the reward and Q calculation steps, and the last allows you to play the game and see the final strategy. The full code is available at the following link: `https://colab.research.google.com/drive/1GiPh-pMOXICNdrhlJqhoJgDB6CYyPIoe`. In the following sections, we will briefly describe how the code works, and some of the results we have obtained.

4.1 Simulator function

The core function is the simulation function, which is used to obtain the strategy. A pseudo-code description of the function is given below. To sum up, the algorithm takes several parameters as input: the environment, the algorithm, the max_step and several calibration parameters. It then returns an array of rewards and Q values. The principle of the function is simple. Run several episodes, each time making choices about exploring or exploiting the data already present. Once this has been done, the various parameters are updated and a new episode is started, unless the episode loop stops.

Algorithm 1 Simulate reinforcement learning with epsilon-greedy strategy for the 4 algorithms

```
Input: Environment, Algorithm, Exploration rate, Decay rate, Total episodes, Max steps, Lambda, Discount factor
Outputs: List of total rewards per episode and the final Q-table.
Initialize
$Q \leftarrow$ array of zeros with shape $(n\_states, n\_actions)$
$rewards \leftarrow$ empty list
for $episode$ in $range$(total_episodes) do
  Initialization of step, cs, rewards_tot
  while $step <$ max_steps do
    $step \leftarrow step + 1$, Set $\mu \leftarrow 1$
    if rand() $>$ $Exploration\ rate$ then
      //Exploitation
      (action $\leftarrow$ argmax($Q$[state, :])) & ($\mu \leftarrow 1 - \epsilon$)
    else
      //Exploration
      (action $\leftarrow$ env.action_space.sample()) & ($\mu \leftarrow \frac{\epsilon}{\text{num\_actions}}$)
    end if
    // UPDATE
    next_state, reward, done, _ $\leftarrow$ env.step(action) // Take action and observe new state, reward, and termination flag
    Update policy $\pi$
    $Q_{\text{step}} \leftarrow Q_{\text{step}} + \gamma^{\text{step}} \left(\prod_{s=1}^{\text{step}} c_s\right) (r_{\text{step}} + \gamma \mathbb{E}_\pi Q_{\text{step}+1} - Q_{\text{step}})$
    Update rewards_tot, Update state
    if $termination\ flag$ then
      break
    end if
  end while
end for
return rewards, $Q$
```

4.2 Environment

4.2.1 Cliff Walking environment

We test the performance of the algorithms using the Cliff Walking environment available in `https://www.gymlibrary.dev/environments/toy_text/`.

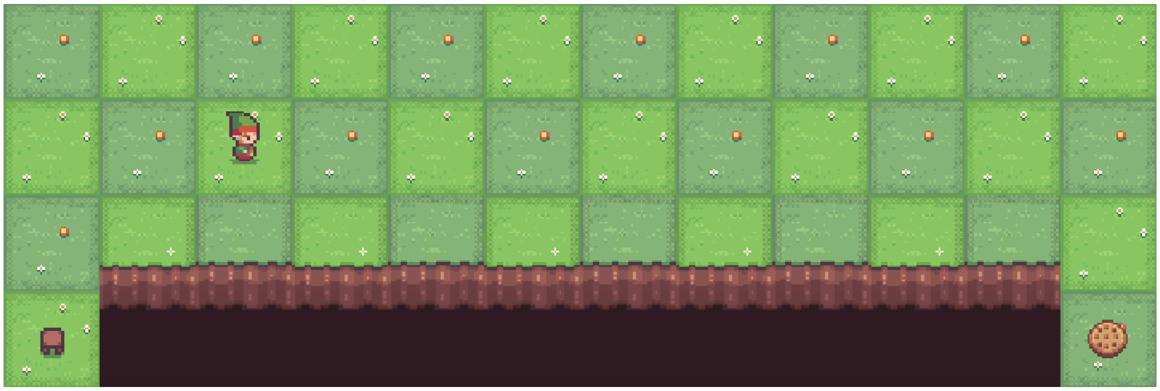

Figure 4: Cliff Walking environment

The agent starts at the bottom-left corner of the grid and must reach the bottom-right corner, by navigating a grid of size 12 × 4. The grid contains a cliff, which is represented by a row of cells along the bottom edge.

The action space is discrete and consists of these four possible actions. At each time step, the agent can go:

- **Up**: Move one cell upwards (towards decreasing row index).
- **Down**: Move one cell downwards (towards increasing row index).
- **Left**: Move one cell to the left (towards decreasing column index).
- **Right**: Move one cell to the right (towards increasing column index).

The agent receives a reward of -1 for each step taken. If the agent falls into the cliff (i.e., enters any cell in the cliff row), it receives a penalty of -100 and is returned to the start state (bottom-left corner).

The episode terminates when the agent reaches the goal state or falls into the cliff.

4.2.2 Frozen Lake environment

We also test the performance of the algorithms using the Frozen Lake environment available in `https://www.gymlibrary.dev/environments/toy_text/`.

The agent starts at the top-left corner of the grid and must reach the bottom-right corner, by navigating a grid of size 4×4. The grid contains holes in the ice, that are distributed in random locations. The possible actions are similar to the Cliff Walking example. The agent receives a reward of $+1$ when reaching the goal,

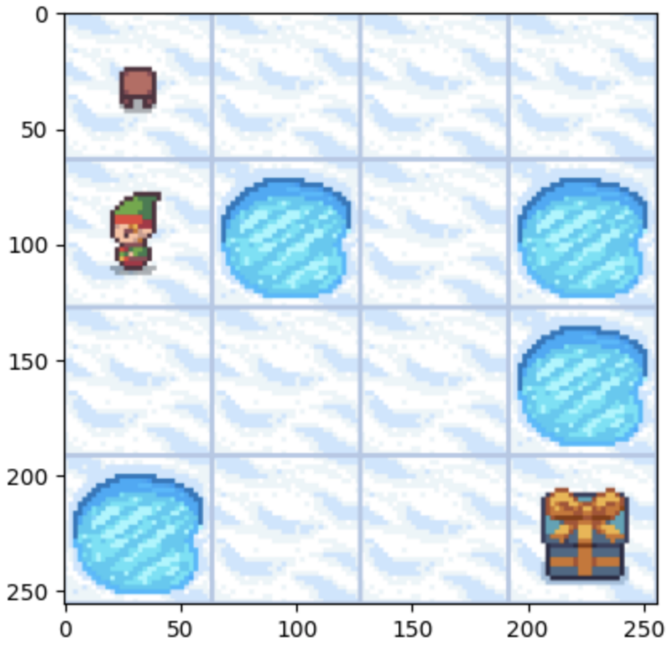

Figure 5: Frozen Lake environment

and 0 otherwise (when reaching hole or frozen). The episode terminates when the agent reaches the goal state or moves into a hole.

4.3 Comparison of the reward per episode

The hyperparameters used for the comparison are the following: $\epsilon = 0.2, \lambda = 0.9, \gamma = 0.99$, `decay_rate`=0.99, `max_steps`=1000, `total_episodes`=30000. Initially, it may be useful to compare algorithms by comparing the evolution of rewards over the course of different episodes. This is what we do in Figure 6&7, where we plot the rewards every 1000 episode.

We observe in Figure 6 that for the Cliff Walking environement, all the algorithms seem to converge quickly and then remain stable, indicating consistent performance across episodes. The rewards quickly rise, indicating learning and adaptation, and then flatten out, showing that the algorithms have reached some level of steady performance. However, the final performance of Importance Sampling is significantly lower than that of the other methods, which suggests that it is not learning an optimal policy as effectively.

In the Figure 7 for Frozen Lake environement, which has binary rewards (0 or 1), the Retrace algorithm appears to consistently achieve the reward and maintains that performance, showing it is the most stable and effective among the tested algorithms. The other algorithms (IS, Off-policy $Q^{\pi}(\lambda)$, and TB) show very poor performance, with the total reward almost always at 0, indicating that they rarely or never reach the goal successfully within the episodes shown. Therefore, those figures show the limits of the other algorithms, contrary to Retrace which seem to perform well in the two examples.

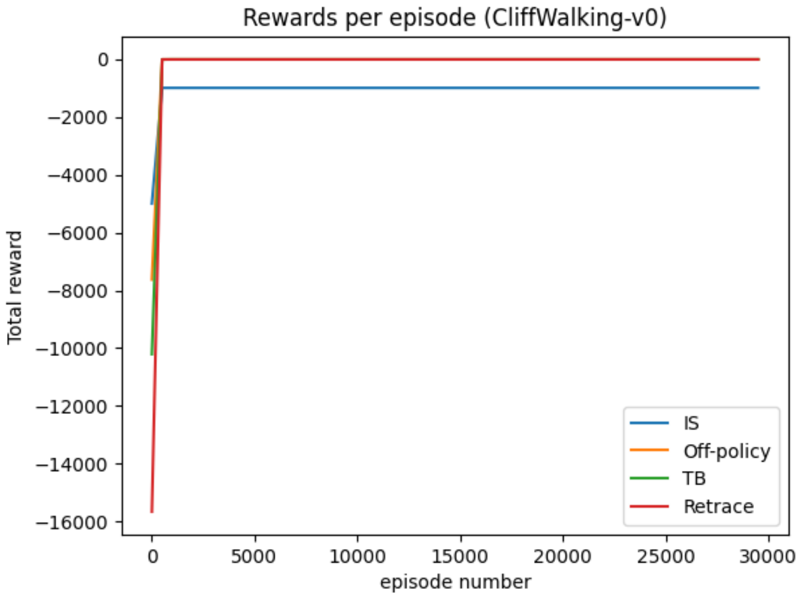

Figure 6: Comparison of the rewards (Cross Walking)

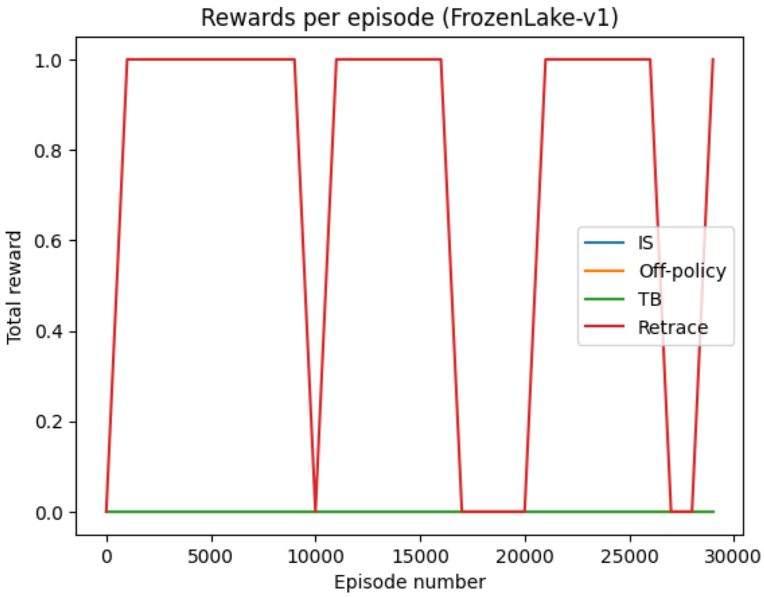

Figure 7: Comparison of the rewards (Frozen Lake)

Having gained a preliminary insight into the results, we can proceed to visualize the rewards averaged across a greater number of simulations. This will afford us a broader perspective on the overall performance of the algorithms.

By choosing a number of simulations equal to 10, the new graph for Cliff Walking in Figure 8 confirms the previous observations and illustrates once again a poorer performance for Importance sampling. However, it fails to discriminate between the performances of the three other algorithms, which all seem to give satisfactory results.

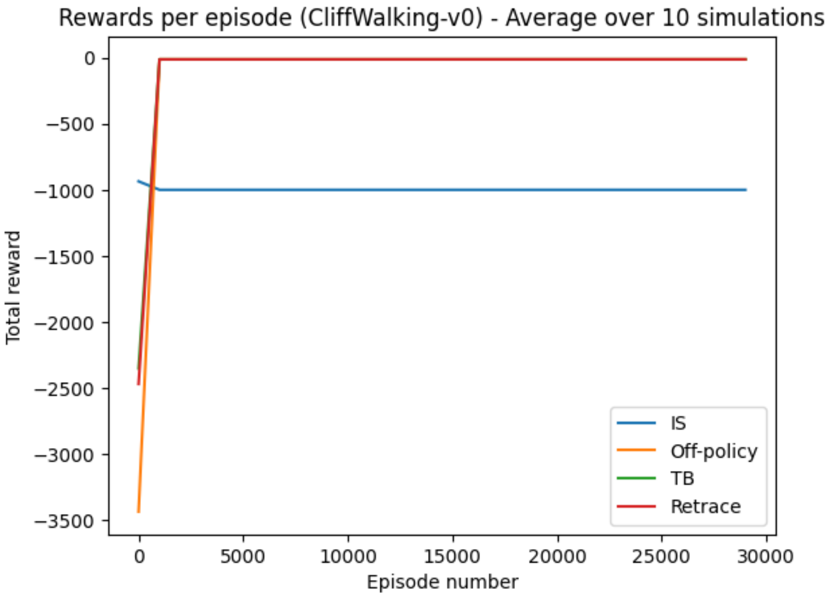

Figure 8: Comparison of the rewards (Cross Walking) over 10 simulations

Hence, to assess the robustness of Retrace algorithm, we proceeded to evaluate its performance within the Frozen Lake environment. In this situation, we were able to carry out a larger number of simulations than previously because the computational time was shorter. Over 50 simulations, we have the following results:

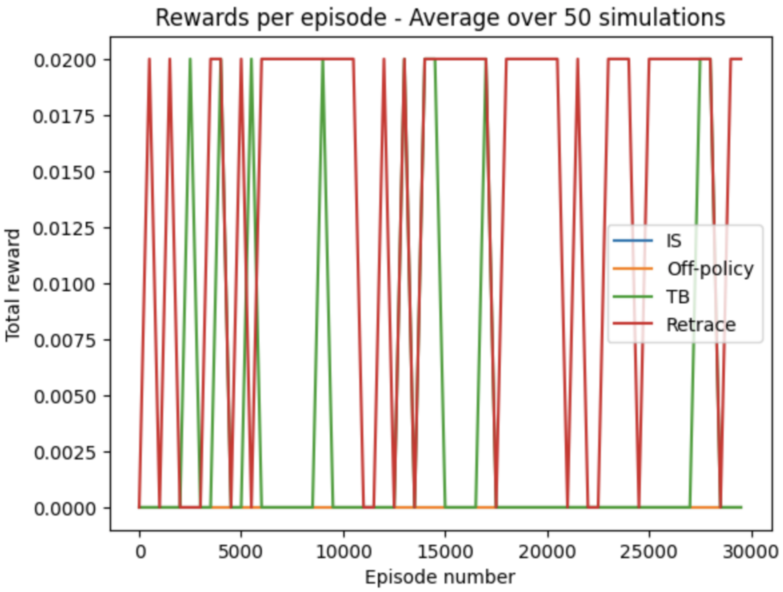

Figure 9: Comparison of the rewards (Frozen Lake) over 50 simulations

Here, we can see that only Tree-backup and Retrace have a non-nul performance, with Retrace exhibiting superior performances among all four algorithms.

4.4 Algorithmic performance in function of λ

As done in the article, we now aim to compare the performances of our algorithms for different values of λ. More precisely, we aim to find its best value. In order to do that we have been calculated the performances of the algorithms on the Cliff Walking environement. Figure 10 shows our results. To obtain this graph, we have normalized our data. Furthermore, we could only conduct the comparison on three out of the four algorithms due to the prolonged execution time of the IS performances for larger lambda values.

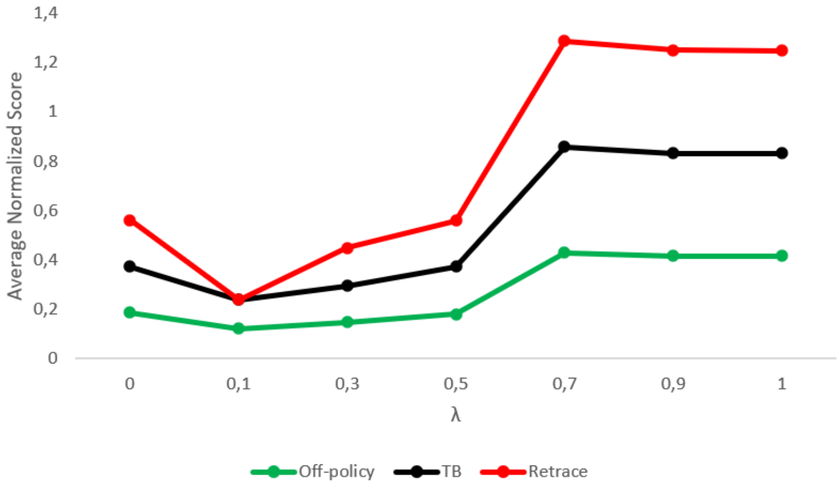

Figure 10: Average inter-algorithm scores for each value of λ

As a comparison, we recall the results of the paper [3], which are the following:

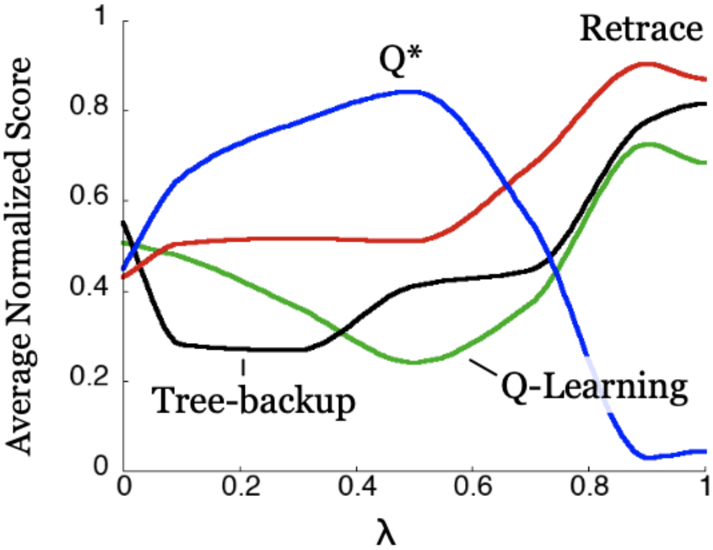

Figure 11: Average inter-algorithm scores for each value of λ in [3]

Performance trends as a function of lambda are similar to those obtained in the article. Notably, we can see a change in trend from $\lambda = 0.5$ where the performance of the different algorithms increases sharply. Furthermore, the findings of this study reveal that the ideal parameter value, denoted as λ_{opt}, is close to the results presented in the article. While our specific experimentation indicates an optimal value of 0.7 versus 0.9 in [3], it underscores the broader strategy of leveraging lambda values that are close to 1 for optimal performance.

We would also like to point out that the Retrace algorithm stands out from the rest. This confirms the advantages of the Retrace algorithm introduced in the paper.

4.5 Algorithmic performance in function of ϵ

To go further, we want to evaluate the impact of the parameter ϵ. As a result, we fixed $\lambda = 0.9$, `decay_rate` $= 0.99$ and evaluated the score for different value of epsilon. We can see on Figure 12 what we obtained:

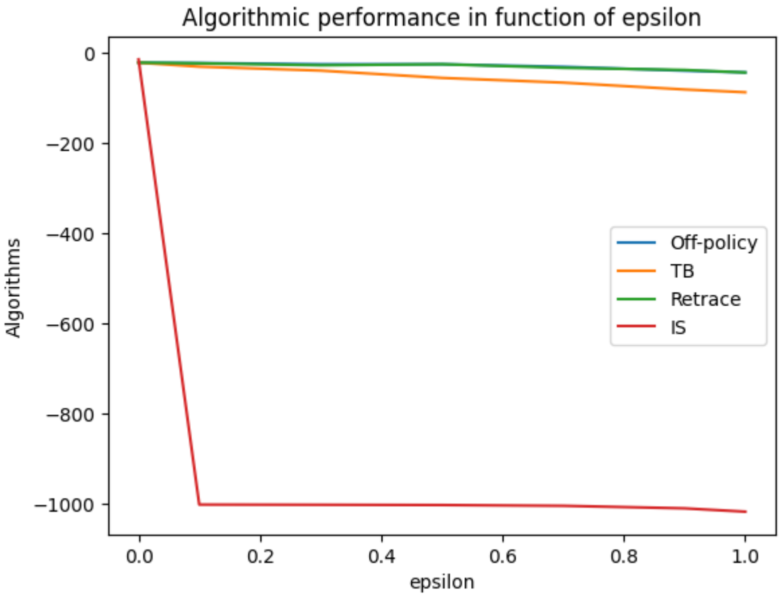

Figure 12: Average Normalized Score in function of ϵ

The best performances were achieved for low values of epsilon. The interpretation we can make is that the score is better when we prioritize exploiting the information we have over exploration. We can also see that this is particularly true for the IS algorithm, but not so much for the other three. Indeed, for these algorithms, the difference remains small.

5 Conclusion

In this paper, we have reviewed several algorithms and their properties, focusing on the advantages that the algorithm Retrace introduced in [3] has to offer. Through a series of computational experiments, we were able to contrast the performance and efficiency of these algorithms. Our empirical findings align with those reported in [3], confirming the efficacy and relative stability of the Retrace(λ) algorithm.

Moreover, the theoretical underpinnings we have established and analyzed offer additional insight into the observed performance. These theoretical contributions not only reinforce the practical observations but also provide a deeper understanding of why certain algorithms, particularly Retrace(λ), exhibit superior behavior in our selected environments.

The knowledge and understanding acquired in our coursework [4] have provided us with a foundation for a more thorough analysis of the article. By drawing on concepts, theories, and methodologies covered in class, we have delved deeper into the nuances of the proposed algorithm and its implications. This has allowed us to critically evaluate the paper's contributions and assess its alignment with established principles in reinforcement learning.

References

[1] Dimitri Bertsekas and John N Tsitsiklis. Neuro-dynamic programming. Athena Scientific, 1996.

[2] Rémi Munos. off-policy deep RL. URL: `https://project.inria.fr/paiss/files/2018/07/munos-off-policy-dRL.pdf`.

[3] Rémi Munos et al. "Safe and efficient off-policy reinforcement learning". In: 29 (2016).

[4] Erwan Le Pennec. Introduction to Reinforcement Learning (M2 DS). Course. Institut Polytechnique de Paris, 2023. URL: `http://www.cmap.polytechnique.fr/~lepennec/fr/teaching/`.

[5] Satinder P Singh and Richard S Sutton. "Reinforcement learning with replacing eligibility traces". In: Machine learning 22 (1996), pp. 123–158.

[6] Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.