# Off-Policy Actor-Critic

**Thomas Degris, Martha White, Richard S. Sutton**
Flowers Team, INRIA, Talence, ENSTA-ParisTech, Paris, France
RLAI Laboratory, Department of Computing Science, University of Alberta, Edmonton, Canada
thomas.degris@inria.fr, whitem@cs.ualberta.ca, Sutton sutton@cs.ualberta.ca

## 1 Introduction

This paper illustrate how to practically combine the generality and learning potential of off-policy learning with the flexibility in action selection given by actor-critic methods.

The previous actor-critic algorithm was limited to the on-policy environment and did not take advantage of the off-policy time difference gradient. In an off-policy setting, an agent learns about a policy different from the one it is executing.Compared with on-policy setting, the benefits of off-policy setting have the following benefits: First, it can optimize strategies while collecting data with exploratory strategies. Second, the policy used for behavior should be soft. Fourth, the off-policy method is able to learn multiple tasks in parallel from a single sensorimotor interaction with an environment. Fifth, it is able to learn from demonstration. Finally, it is very general and can make full use of samples. Because of this generality, off-policy method is of great interest in many application domains.

Most off-policy method such as the Q-learning algorithm and its various versions cannot handle the continuous action problem well. The best algorithm for handling continuous actions is the policy gradient algorithm, which includes the famous actor-critic algorithm. The actor updates the policy weights and the critic learns an off-policy estimate of the value function for the current actor policy. For many problems, the actor-critic method is more practical than action value method because the policy can be stochastic and utilize a large action space.

There are three contributions. First, it proposed a new algorithm for learning control off-policy, called Off-PAC (Off-Policy Actor Critic). Second, it proved that Off-PAC converges in a standard off-policy setting. Finally, it provides the first empirical evaluations of gradient-TD methods for off-policy control and showed that Off-PAC has the best final performance and the lowest standard error on three benchmark Q($\lambda$),Greedy-GQ, Softmax-GQ.

This paper is the first algorithm for off-policy actor critic in reinforcement learning.It successfully solves the problem of weighted importance sampling exponential increase or decrease. Its algorithm is per-time-step complexity scales linearly with the number of learned weighted. It is a very important cornerstone for the DPG and DDPG algorithm.

## 2 Problem Formulation

### 2.1 Notation

#### 2.1.1 Markov Decision Processes

We consider Markov decision processes with a discrete state space $S$, a discrete action space $A$, a distribution $P : S \times S \times A \to$ [0, 1], where $P(s'|s, a)$is the probability of transitioning into state $s'$ from state $s$ after taking action $a$, and an expected reward function $R : S \times A \times S \to \mathbb{R}$ that provides an expected reward for taking action a in state s and transitioning into $s'$. We observe a stream of

Preprint. Under review.

data, which includes states $s_t \in S$, actions $a_t \in A$ and rewards $r_t \in R$ for t = 1, 2,... with actions selected from a fixed behavior policy, $b(a|s) \in (0, 1]$.

### 2.1.2 Value Function

We define the value function for $\pi : S \times A \to$ [0, 1] to be:

$$V^{\pi,\gamma}(s) = E[r_{t+1} + ...r_{t+T}|s_t = s] \quad \forall s \in S \tag{1}$$

where policy $\pi$ is followed from time step $t$ and terminates at time $t + T$ according to $\gamma$. We assume termination always occurs in a finite number of steps.

### 2.1.3 Action-Value Function

The action-value function, $Q^{\pi,\gamma}(s, a)$, is defined as:

$$Q^{\pi,\gamma}(s, a) = \sum_{s' \in S} P(s'|s, a)[R(s, a, s') + \gamma(s')V^{\pi,\gamma}(s')] \tag{2}$$

for all $a \in A$ and for all $s \in S$. Note that $V^{\pi,\gamma}(s') = \sum_{a \in A} \pi(a|s)Q^{\pi,\gamma}(s, a)$ for all $s \in S$.

### 2.1.4 Objective Function

The aim is to choose $u$ so as to maximize the following scalar objective function:

$$J_\gamma(u) = \sum_{s \in S} d^b(s)V^{\pi_u,\gamma}(s) \tag{3}$$

where $d^b(s) = \lim_{t\to\infty} P(s_t = s|s_0, b)$ is the the limiting distribution of states under $b$ and $P(s_t = s|s_0, b)$ is the probability that $s_t = s$ when starting in $s_0$ and executing $b$. The objective function is weighted by $d_b$ because, in the off-policy setting, data is obtained according to this behavior distribution.

## 2.2 The Off-Policy Actor Critic Algorithm

With the objective function (Equation 3), the next step is to optimize the objective function to obtain weight updates:

$$\begin{aligned} \nabla_u J_\gamma(u) &= \nabla_u \left[ \sum_{s \in S} d^b(s) \sum_{a \in A} \pi(a|s)Q^{\pi,\gamma}(s, a) \right] \\ &= \sum_{s \in S} d^b(s) \sum_{a \in A} [\nabla_u \pi(a|s)Q^{\pi,\gamma}(s, a) + \pi(a|s)\nabla_u Q^{\pi,\gamma}(s, a)] \end{aligned} \tag{4}$$

The final term in this equation, $\nabla_u Q^{\pi,\gamma}(s, a)$, is difficult to estimate in an incremental off-policy setting. The first approximation involved in the theory of Off-PAC is to omit this term.That is, we work with an approximation to the gradient, which we denote $g(u) \in \mathbb{R}^{N_u}$, defined by

$$\nabla_u J_\gamma(u) \approx g(u) = \sum_{s \in S} d^b(s) \sum_{a \in A} \nabla_u \pi(a|s)Q^{\pi,\gamma}(s, a) \tag{5}$$

There is one problem in the equation 5. The source of the action sample is the target policy $\pi$, not the behavior policy $b$. We need to rewrite the formula 5 by the importance sampling:

$$\begin{aligned} g(u) &= E\left[ \sum_{a \in A} \nabla_u \pi(a|s)Q^{\pi,\gamma}(s, a) \middle| s \sim d^b \right] \\ &= E\left[ \sum_{a \in A} b(a|s) \frac{\pi(a|s)}{b(a|s)} \frac{\nabla_u \pi(a|s)}{\pi(a|s)} Q^{\pi,\gamma}(s, a) \middle| s \sim d^b \right] \\ &= E\left[ \rho(s, a)\psi(s, a)Q^{\pi,\gamma}(s, a) \middle| s \sim d^b, a \sim b(\cdot|s) \right] \\ &= E_b\left[ \rho(s_t, a_t)\psi(s_t, a_t)Q^{\pi,\gamma}(s_t, a_t) \right] \end{aligned} \tag{6}$$

where $\rho(s,a) = \frac{\pi(a|s)}{b(a|s)}$ , $\psi(s_t,a_t) = \frac{\nabla_u \pi(a|s)}{\pi(a|s)}$ and we introduce the new notation $E_b[\cdot]$ to denote the expectation implicitly conditional on all the random variables being drawn from their limiting stationary distribution under the behavior policy.

A standard result (e.g., see Sutton et al., 2000) is that an arbitrary function of state can be introduced into these equations as a baseline without changing the expected value. To reduce the variance,We use the approximate state-value function provided by the critic, $\hat{\mathbf{v}}$ , in this way:

$$g(u) = E_b\left[\rho(s_t,a_t)\psi(s_t,a_t)(Q^{\pi,\gamma}(s_t,a_t) - \hat{\mathbf{v}}(s_t))\right] \tag{7}$$

The next step is to replace the action value, $Q^{\pi,\gamma}(s_t,a_t)$, by the off-policy $\lambda$-return.

$$\widehat{\mathbf{g}(\mathbf{u})} \approx g(u) = E\left[\rho(s_t,a_t)\psi(s_t,a_t)(R_t^\lambda - \hat{\mathbf{v}}(s_t))\right] \tag{8}$$

where the off-policy $\lambda$-return is defined by:

$$R_t^\lambda = r_{t+1} + (1-\lambda)\gamma(s_{t+1})\hat{\mathbf{v}}(s_{t+1}) + \lambda\gamma(s_{t+1})\rho(s_{t+1},a_{t+1})R_{t+1}^\lambda \tag{9}$$

Finally, based on this equation, we can write the forward view of Off-PAC:

$$\mathbf{u_{t+1}} - \mathbf{u_t} = \alpha_{\mathbf{u},\mathbf{t}}\rho(\mathbf{s_t},\mathbf{a_t})\psi(\mathbf{s_t},\mathbf{a_t})(\mathbf{R_t^\lambda} - \hat{\mathbf{v}}(\mathbf{s_t})) \tag{10}$$

where $\alpha_{u,t} \in \mathbb{R}$ is a positive step-size parameter.

**Algorithm 1** The Off-PAC algorithm

Initialize the vectors $\mathbf{e}_v$, $\mathbf{e}_u$, and $\mathbf{w}$ to zero
Initialize the vectors $\mathbf{v}$ and $\mathbf{u}$ arbitrarily
Initialize the state $s$
For each step:
  Choose an action, $a$, according to $b(\cdot|s)$
  Observe resultant reward, $r$, and next state, $s'$
  $\delta \leftarrow r + \gamma(s')\mathbf{v}^\mathsf{T}\mathbf{x}_{s'} - \mathbf{v}^\mathsf{T}\mathbf{x}_s$
  $\rho \leftarrow \pi_\mathbf{u}(a|s)/b(a|s)$
  Update the critic (GTD($\lambda$) algorithm):
    $\mathbf{e}_v \leftarrow \rho\left(\mathbf{x}_s + \gamma(s)\lambda\mathbf{e}_v\right)$
    $\mathbf{v} \leftarrow \mathbf{v} + \alpha_v\left[\delta\mathbf{e}_v - \gamma(s')(1-\lambda)(\mathbf{w}^\mathsf{T}\mathbf{e}_v)\mathbf{x}_s\right]$
    $\mathbf{w} \leftarrow \mathbf{w} + \alpha_w\left[\delta\mathbf{e}_v - (\mathbf{w}^\mathsf{T}\mathbf{x}_s)\mathbf{x}_s\right]$
  Update the actor:
    $\mathbf{e}_u \leftarrow \rho\left[\frac{\nabla_\mathbf{u}\pi_\mathbf{u}(a|s)}{\pi_\mathbf{u}(a|s)} + \gamma(s)\lambda\mathbf{e}_u\right]$
    $\mathbf{u} \leftarrow \mathbf{u} + \alpha_u\delta\mathbf{e}_u$
  $s \leftarrow s'$

Figure 1: The Off-PAC algorithm

Although the forward view is useful for understanding and analyzing algorithms, it is not easy to implement it. We must be converted to a backward view that does not involve the $\lambda$-return.

$$g(u) = E_b\left[\rho(s_t,a_t)\psi(s_t,a_t)(Q^{\pi,\gamma}(s_t,a_t) - \hat{\mathbf{v}}(s_t))\right] = E_b[\delta_t e_t] \tag{11}$$

where $\delta_t = r_{t+1} + \gamma(s_{t+1})\hat{\mathbf{v}}(s_{t+1})) - \hat{\mathbf{v}}(s_t))$ is the temporal difference error, and $e_t \in R^{N_u}$ is the eligibility trace of $\psi$, updated by:

$$e_t = \rho(s_t,a_t)(\psi(s_t,a_t) + \lambda e_{t-1}) \tag{12}$$

Finally, combining the three previous equations, the backward view of the actor update can be written simply as:

$$u_{t+1} - u_t = \alpha_{u,t}\delta_t e_t \tag{13}$$

The complete Off-PAC algorithm is given above as Algorithm 1.With discrete actions, a common policy distribution is the Gibbs distribution, which uses a linear combination of features $\pi(a|s) = \frac{e^{u^t\phi_{s,a}}}{\sum_b e^{u^t\phi_{s,b}}}$ where $\phi_{s,a}$ are state-action features for state $s$, action $a$, and where $\psi(s,a) = \frac{\nabla_u\pi(a|s)}{\pi(a|s)} = \phi_{s,a} - \sum_b \pi(b|s)\phi_{s,b}$.

# 3 Theoretical Analysis

## 3.1 Convergence Analysis

The algorithm has the same recursive stochastic form that the two-timescale off-policy value-function algorithms have:

$$\begin{aligned} u_{t+1} &= u_t + \alpha_t(h(u_t, z_t) + M_t + 1) \\ z_{t+1} &= z_t + \alpha_t(f(u_t, z_t) + N_t + 1) \end{aligned} \tag{14}$$

where $x \in \mathbb{R}^d$, $h : \mathbb{R}^d \to \mathbb{R}^d$ is a differentiable functions, $\{\alpha_t\}_k \geq 0$ is a positive step-size sequence and $\{M_t\}_k \geq 0$ is a noise sequence. Since we have two updates, one for the actor and one for the critic, and those time updates are not linearly separable. In order to satisfy the conditions for the two-timescale analysis,We need to have the following assumptions on features and step-sizes.

(A1) The policy function, $\pi(\cdot)(a|s) : \mathbb{R}^{N_u} \to [0, 1]$, is continuously differentiable in $u$ ,$\forall s \in S, a \in A$.

(A2) The update on $u_t$ includes a projection operator, $\Gamma : \mathbb{R}^{N_u} \to \mathbb{R}^{N_u}$ that projects any $u$ to a compact set $U = \{\mathbf{u} \mid \mathbf{q_i(u)} \leq \mathbf{0}, \mathbf{i} = \mathbf{1}, ..., \mathbf{s}\} \in \mathbb{R}^{\mathbf{N_u}}$, where $q_i(\cdot) : \mathbb{R}^{N_u} \to \mathbb{R}$ are continuously differentiable functions specifying the constraints of the compact region.For each $u$ on the boundary of $U$, the gradients of the active $q_i$ are considered to be linearly independent. Assume that the compact region, $U$, is large enough to contain at least one local maximum of $J_\gamma$.

(A3) The behavior policy has a minimum positive weight for all actions in every state, in other words, $b(a|s) \geq b_m in \forall s \in S, a \in A$, for some $b_m in \in (0, 1]$.

(A4) The sequence $(x_t, x_t + 1, r_t + 1)_{t\geq 0}$ is i.i.d. and has uniformly bounded second moments.

(A5) For every $u \in U$(the compact region to which $u$ is projected), $V^{\pi_u,\gamma} : S \to \mathbb{R}$ is bounded.

(P1) $||x_t||_\infty < \infty, \forall t$ where $x_t \in \mathbb{R}^{N_v}$

(P2) Matrices $C = E[x_t x_t^T]$,$A = E[x_t(x_t - \gamma x_{t+1})^T]$ are non-singular and uniformly bounded. $A$, $C$ and $E[r_{t+1}, x_t]$ are well-defined because the distribution of $(x_t, x_{t+1}, r_{t+1})$ does not depend on $t$.

(S1) $\alpha_{v,t}$,$\alpha_{w,t}$,$\alpha_{u,t} > 0$,$\forall t$ are deterministic such that $\sum_t \alpha_{v,t} = \sum_t \alpha_{w,t} = \sum_t \alpha_{u,t} = \infty$ and $\sum_t \alpha_{v,t}^2 < \infty$, $\sum_t \alpha_{w,t}^2 < \infty$ and $\sum_t \alpha_{u,t}^2 < \infty$ with $\frac{\alpha_{u,t}}{\alpha_{v,t}} \to 0$.

(S2) Define $H(A) \doteq (A + A^T)/2$ and let $\lambda_m in(C^{-1}H(A))$ be the minimum eigenvalue of the matrix $(C^-1H(A))^4$. Then $\alpha_{w,t} = \eta\alpha_{v,t}$ for some $\eta > max(0, \lambda_m in(C^{-1}H(A)))$.

**Theorem 1 (Convergence of Off-PAC):** Let $\lambda = 0$ and consider the Off-PAC iterations for the critic (GTD($\lambda$),i.e., TDC with importance sampling correction) and the actor (for weights $u_t$). Assume that (A1)-(A5), (P1)-(P2) and (S1)-(S2) hold. Then the policy weights, $u_t$, converge to $\hat{\mathbf{Z}} = \{u \in U | \hat{g(u)} = 0\}$ and the value function weights, $v_t$, converge to the corresponding TD-solution with probability one.

## 3.2 Results

This section compares the performance of Off-PAC to three other off-policy algorithms with linear memory and computational complexity: 1) Q($\lambda$) (called QLearning when $\lambda$ = 0), 2) Greedy-GQ (GQ($\lambda$) with a greedy target policy), and 3) Softmax-GQ (GQ($\lambda$) with a Softmax target policy).

It used three benchmarks: mountain car, a pendulum problem and a continuous grid world. These problems all have a discrete action space and a continuous state space.

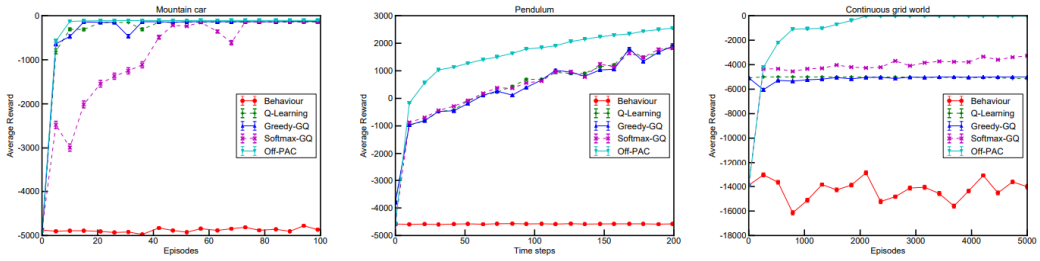


Figure 2: Performance of Off-PAC compared to the performance of Q($\lambda$), Greedy-GQ, and Softmax-GQ.

Figure 2 shows results on three problems.Off-PAC performed best on all problems. On the continuous grid world, Off-PAC was the only algorithm able to learn a policy that reliably found the goal . On all problems, Off-PAC had the lowest standard error.

Off-PAC, like other two-timescale update algorithms, can be sensitive to parameter choices, particularly the step-sizes. Off-PAC has four parameters: $\lambda$ and the three step sizes, $\alpha_v$ and $\alpha_w$ for the critic and $\alpha_u$ for the actor. The value of $\lambda$, as with other algorithms, will depend on the problem and it is often better to start with low values.A common heuristic is to set $\alpha_v$ to 0.1 divided by the norm of the feature vector, while keeping the value of $\alpha_w$ low.Once GTD($\lambda$) is stable learning the value function with $\alpha_u = 0$, $\alpha_u$ can be increased so that the policy of the actor can be improved.

## 4 Conclusion

This paper presents the first actor-critic algorithm for off-policy reinforcement learning. The off-policy actor critic combines the generality and learning potential of off-policy learning with the flexibility in action selection given by actor-critic methods. It showed that Off-PAC has the best final performance on three benchmark problems and consistently has the lowest standard error.

Off-policy learning with experience replay may appear to be an obvious strategy for improving the sample efficiency of actor-critics. However, controlling the variance and stability of off-policy estimators is notoriously hard.Importance sampling is one of the most popular approaches for off-policy learning[1]. Then, the importance weighted policy gradient is given by:

$$g(u)^{imp} = (\prod_{t=0}^{k} \rho_t) \sum_{t=0}^{k} (\sum_{i=0}^{k-t} \gamma^i r_{t+i}) \nabla_\theta log \pi_\theta(a_t, x_t) \tag{15}$$

Off-PAC (Off-Policy Actor Critic) attacked this problem by using marginal value functions over the limiting distribution of the process to yield the following approximation of the gradient:

$$g(u)^{off-PAC} = E_b \left[ \rho(s_t, a_t) \nabla_\theta log \pi_\theta(a_t, x_t) Q^{\pi,\gamma}(s_t, a_t) \right] \tag{16}$$

However, the marginal importance weights in Equation (6) can become large, thus causing instability.To safe-guard against high variance, [2] propose to truncate the importance weights and introduce a correction term via the following decomposition of $g^{ACER}$:

$$\begin{aligned} g_t^{ACER} &= \overline{\rho_t} \nabla_\theta log \pi_\theta(a_t, x_t) [Q^{ret}(x_t, a_t) - V_{\theta_v}(x_t)] \\ &+ E_{a \sim \pi} ([\frac{\rho_t(a) - c}{\rho_t(a)}]_+ \nabla_\theta log \pi_\theta(a|x_t) [Q_{\theta_v}(x_t, a_t) - V_{\theta_v}(x_t)]) \end{aligned} \tag{17}$$

Off-PAC is a promising direction for extending off-policy learning to a more general setting such as continuous action spaces[3]. The objective function is

$$\nabla_\theta J(\pi_\theta) = \int_s d^\pi(s) \nabla_\theta \pi_\theta(s) \nabla_a Q^\pi(s, a)|_{a=\pi_\theta(s)} ds = E_{s \sim d^\pi} [\nabla_\theta \pi_\theta(s) \nabla_a Q^\pi(s, a)|_{a=\pi_\theta(s)} \pi_\theta(s)] \tag{18}$$

## References


[1] Meuleau, L. Peshkin, L. P. Kaelbling, and K. Kim. Off-policy policy search. Technical report, MIT AI Lab, 2000.

[2] Wang, Ziyu, et al. "Sample efficient actor-critic with experience replay." arXiv preprint arXiv:1611.01224 (2016).

[3] Lillicrap, Timothy P., et al. "Continuous control with deep reinforcement learning." arXiv preprint arXiv:1509.02971 (2015).