Learning Value Functions using Residual Variance in Deep Policy Gradients

Yannis Flet-Berliac*[1,2], based on work with:
Reda Ouhamma*[1,2], Odalric-Ambrym Maillard[1], Philippe Preux[1,2]

[1]SequeL team – Inria Lille – Nord Europe, France
[2]Univ. Lille, CRIStAL, CNRS
*Equal contribution

DeepMind Reading Group **October 23, 2020**

Agenda

- Policy Gradients (PG) and Actor-Critic (AC) methods
  - Contextualization
  - The use of value functions (Critics) in deep PG algorithms
- Problem: empirical *failure* of popular AC methods
  - Critics *do not* actually fit V^π
  - State-action-dependent baselines *fail* to reduce gradient variance
- `AVEC`: learning the Critics using Residual Variance
  - An alternative Critic (+ building motivation)
  - Consistent gradient directions
  - Empirical results: continuous control + sparse-reward tasks
- `AVEC`: does it really work?
  - Estimation error (learning the empirical target $\hat{V}^\pi$)
  - Approximation error (learning the true target V^π)
  - Empirical variance
- Conclusion and perspectives

Reinforcement Learning

Environment (Markov Decision Process):

- State $s \in \mathcal{S}$, action $a \in \mathcal{A}$
- Reward function: $r(s, a)$, transition probabilities: $P(s'|s, a)$

Agent:

- Stochastic policy $\pi_\theta(a|s)$ with parameter θ

An agent in state s_t interacts with an environment by sampling action $a_t \sim \pi_\theta(\cdot|s_t)$, receives reward r_t and transitions to a new state s_{t+1}.

Reinforcement Learning

Environment (Markov Decision Process):

- State $s \in \mathcal{S}$, action $a \in \mathcal{A}$
- Reward function: $r(s, a)$, transition probabilities: $P(s'|s, a)$

Agent:

- Stochastic policy $\pi_\theta(a|s)$ with parameter θ

An agent in state s_t interacts with an environment by sampling action $a_t \sim \pi_\theta(\cdot|s_t)$, receives reward r_t and transitions to a new state s_{t+1}.

Goal: Find π that maximizes

$$J(\pi_\theta) \triangleq \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right)\right]$$

with $\gamma \in [0, 1)$, $s_{t+1} \sim P(\cdot|s_t, a_t)$, $a_t \sim \pi_\theta(\cdot|s_t)$ and trajectory τ.

Policy Gradients

Policy gradient algorithms try to solve the optimization problem

$$\max_{\theta} J(\pi_\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) \right]$$

by taking stochastic gradient ascent on the policy parameters θ, using the policy gradient

$$\nabla_\theta J = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t | s_t) Q^{\pi_\theta}(s_t, a_t) \right]$$

with $Q^{\pi_\theta}(s, a) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t\right) | s_0 = s, a_0 = a \right]$.

Intuition: make the good actions more probable.

Policy Gradients

It is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories ...

... But the corresponding **variance can be extremely high**.

Policy Gradients

It is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories ...

... But the corresponding **variance can be extremely high**.

Subtracting a baseline from the value function in the policy gradient can be very beneficial in reducing variance without damaging the bias [Williams, 1992, Weaver and Tao, 2001].

Policy Gradients

It is possible to obtain an unbiased estimate of the policy gradient from empirical trajectories ...

... But the corresponding **variance can be extremely high**.

Subtracting a baseline from the value function in the policy gradient can be very beneficial in reducing variance without damaging the bias [Williams, 1992, Weaver and Tao, 2001].

In practice, if we denote $\hat{X}$ the empirical estimate of X, the policy gradient becomes

$$\nabla_\theta J = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t | s_t) \hat{A}^{\pi_\theta}(s_t, a_t) \right],$$

with $\hat{A}^{\pi_\theta}(s, a) = \hat{Q}^{\pi_\theta}(s, a) - \hat{V}^{\pi_\theta}(s)$ the advantage estimate which quantifies how an action a is better than the average action in state s.

Critics in Deep Policy Gradients

$\hat{V}^{\pi_\theta}$ is learned using a function estimator.

Let $f_\phi : \mathcal{S} \to \mathbb{R}$ (ϕ its parameter) be an estimator of the empirical return $\hat{V}^{\pi_\theta}$. f_ϕ is traditionally learned through minimizing the MSE against $\hat{V}^{\pi_\theta}$. At update k, the critic minimizes:

$$\mathcal{L}_{\text{AC}} = \mathbb{E}_s \left[\left(f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s) \right)^2 \right],$$

where the states s are collected under policy π_{θ_k}.

Setting applicable to e.g. PPO [Schulman et al., 2017] and TRPO [Schulman et al., 2015].

Alternatively use $f_\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ instead to fit $\hat{Q}^{\pi\theta}$ to use in SAC [Haarnoja et al., 2018].

The sad truth about policy gradient variance

In practice, these modifications on the AC framework result in improved performance without a significant variance reduction [Ilyas et al., 2019]. Same conclusion with state-action-dependent baselines [Tucker et al., 2018].

The sad truth about policy gradient variance

In practice, these modifications on the AC framework result in improved performance without a significant variance reduction [Ilyas et al., 2019]. Same conclusion with state-action-dependent baselines [Tucker et al., 2018].

Problem: discrepancy between what motivates AC algorithms and the resulting implementation to obtain maximum gains.

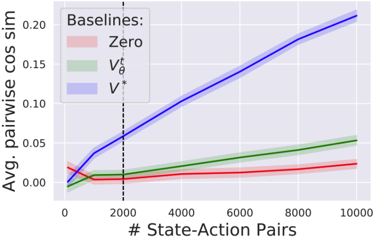
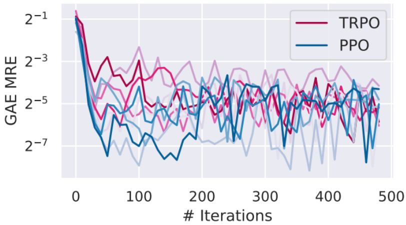


Figure: Empirical variance of the gradient [Ilyas et al., 2019].

NB: the avg. pairwise cos sim is inversely proportional to the gradient variance (the higher the better).

The sad truth about learning the value function

The value network (critic) succeeds in the supervised learning task of fitting $\hat{V}^{\pi}$ but **does not fit** V^{π}.
→ The problem is the approximation error and not the estimator.



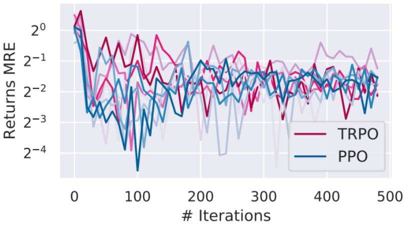


Figure: Quality of value prediction [Ilyas et al., 2019]

The sad truth about learning the value function

- Variance **is not reduced**.
- Approximation error: the value network succeeds in fitting $\hat{V}^{\pi}$ but **does not fit** V^{π}.
- Unbiased equivalents of “novel baseline-based algorithms” **do not improve performance nor reduce variance** (see [Tucker et al., 2018]).

A new approach to learning value functions

In `AVEC`, at update k the critic minimizes:

$$\mathcal{L}_{\mathrm{AC}}(\phi) = \cancel{\mathbb{E}_s\left[\left(f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s)\right)^2\right]}$$

$$\mathcal{L}_{\mathtt{AVEC}}(\phi) = \mathbb{E}_s\left[\left(\left(f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s)\right) - \mathbb{E}_s\left[f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s)\right]\right)^2\right]$$

where the states s are collected under policy π_{θ_k}.

→ `AVEC` minimizes the residual variance.

Using $g_\phi(s) = f_\phi(s) + \mathbb{E}_s[\hat{V}^{\pi_{\theta_k}}(s) - f_\phi(s)]$ provides consistent gradient directions:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{\tau\sim\pi_\theta}\left[\sum_{t=0}^{\infty} \nabla_\theta \log \pi_\theta(a_t|s_t) g_\phi(s_t)\right].$$

Proof. See [Flet-Berliac et al., 2020].

Intuition

The most straightforward intuition comes in the Q-function estimation case: simply replace $\hat{V}^{\pi}(s)$ by $\hat{Q}^{\pi}(s,a)$ and $f_{\phi}(s)$ by $f_{\phi}(s,a)$ in $\mathcal{L}_{\texttt{AVEC}}$.

Idea: the practical use of the Q-function is to disentangle the relative values of actions for each state [Sutton et al., 2000]. $\mathcal{L}_{\texttt{AVEC}}$ focuses on the relative state-action values.

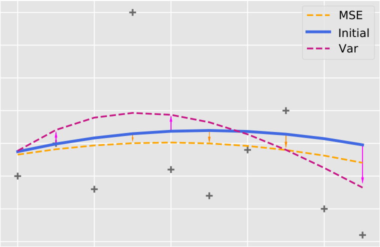


Figure: Comparison of simple models derived when $\mathcal{L}_{\texttt{AVEC}}$ is used instead of the MSE.

→ In minimizing the residual variance,
`AVEC` allows a better recovery of the "shape" of the target near extrema.

Intuition 2

Recall the approximation error $\|\hat{V}^{\pi} - V^{\pi}\|$ is problematic [Tucker et al., 2018, Ilyas et al., 2019].
$\rightarrow$ Suggests that the variance of the empirical targets V^{π} is high.

Idea: optimizing the critic with $\mathcal{L}_{\texttt{AVEC}}$ can be interpreted as fitting $\hat{V}'^{\pi}(s) = \hat{V}^{\pi}(s) - \mathbb{E}_{s'}[\hat{V}^{\pi}(s')]$ using the MSE.

In the independent case, $\hat{V}'^{\pi}$ is a better approximations of $V'^{\pi}(s) = V^{\pi}(s) - \mathbb{E}_{s'}[V^{\pi}(s')]$ than $\hat{V}^{\pi}$ is of V^{π} (see [Flet-Berliac et al., 2020]).

$\rightarrow$ $\hat{V}'^{\pi}$ has a more compact span, and is consequently **easier to fit**.

Implementation

Algorithm 1 AVEC coupled with PPO.

```
Input parameters: $\lambda_\pi \geq 0, \lambda_V \geq 0$
Initialize policy parameter $\theta$ and value function parameter $\phi$
for each update step do
    batch $\mathcal{B} \leftarrow \emptyset$
    for each environment step do
        $a_t \sim \pi_\theta(\cdot|s_t)$
        $s_{t+1} \sim \mathcal{P}(s_t, a_t)$
        $\mathcal{B} \leftarrow \mathcal{B} \cup \{(s_t, a_t, r_t, s_{t+1})\}$
    end for
    for each gradient step do
        $\theta \leftarrow \theta - \lambda_\pi \hat{\nabla}_\theta J^{\text{PPO}}(\pi_\theta)$
        ~~$\phi \leftarrow \phi - \lambda_V \hat{\nabla}_\phi \mathcal{L}_{\text{AC}}(\phi)$~~    ~~$\mathcal{L}_{\text{AC}}(\phi) = \mathbb{E}_s\left[(f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s))^2\right]$~~
        $\phi \leftarrow \phi - \lambda_V \hat{\nabla}_\phi \mathcal{L}_{\texttt{AVEC}}(\phi)$    $\mathcal{L}_{\texttt{AVEC}}(\phi) = \mathbb{E}_s\left[\left((f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s)) - \mathbb{E}_s\left[f_\phi(s) - \hat{V}^{\pi_{\theta_k}}(s)\right]\right)^2\right]$
    end for
end for
```

Before we go further...

Do you have questions?

Continuous Control

On average, using `AVEC` improves SAC by 26% and PPO by 40%.

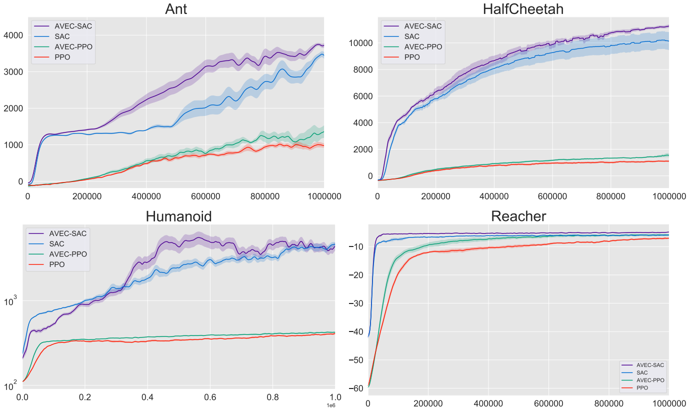


Figure: `AVEC` coupled with SAC and PPO on MuJoCo tasks. X-axis: number of timesteps. Y-axis: average total reward.

Sparse Reward tasks

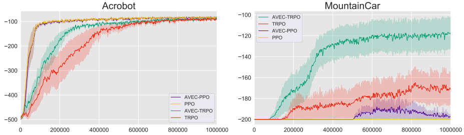


Figure: Comparative evaluation of AVEC in sparse reward tasks. X-axis: number of timesteps. Y-axis: average total reward.

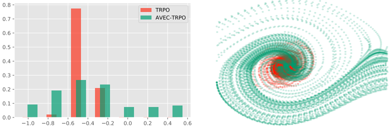


Figure: Left: State visitation frequency. Right: Phase portrait of visited states of AVEC-TRPO (green) and TRPO (red) in MountainCar.

→ AVEC is able to pick up on experienced positive reward more easily.
→ Suggests that the reconstructed shape of the value function is more accurate around such rewarding states: pushes the agent to explore further around experienced states with high values.

Does it really work? Let's examine the new value function

1. What is the estimation error? (learning the empirical target $\hat{V}^{\pi}$)
2. What is the approximation error? (learning the true target V^{π})
3. What is the resulting empirical variance?

Does it really work? Let's examine the new value function

1. What is the **estimation error**? (learning the empirical target)

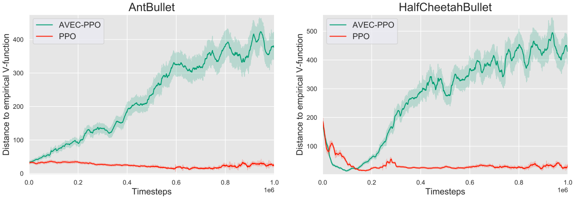


Figure: L_2 distance to $\hat{V}^{\pi}$.

→ Expected result since vanilla PPO optimizes the MSE directly.

Does it really work? Let's examine the new value function

2. What is the **approximation error**? (learning the true target)

NB: to approximate the true value function, we fit the returns sampled from the current policy using a large number of transitions ($3 \cdot 10^5$).

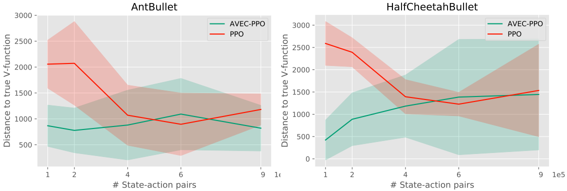


Figure: L_2 distance to V^π. X-axis: we run the algorithm and $\forall t \in \{1, 2, 4, 6, 9\} \cdot 10^5$ we stop training, use the current policy to collect $3 \cdot 10^5$ transitions to estimate V^π.

→ Our estimator is far closer to the true value function half of the time (horizon is 10^6) than the estimator obtained with MSE, then as close to it.

Does it really work? Let's examine the new value function

3. What is the resulting **empirical variance**?

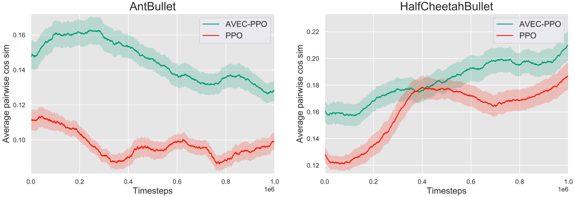


Figure: Average pairwise cosine similarity.

Higher average (10 batches per iteration) pairwise cosine sim. $\implies$ Closer batch-estimates of the gradient $\implies$ Smaller gradient variance.

Conclusion and perspectives

This paper ...:

◦ introduces a modification in the training objective for the critic in actor-critic algorithms which (a) is well-motivated by recent analysis of deep PG algorithms (b) produces considerable gains in performance

◦ highlights the benefits of a more thorough analysis of the critic objective in policy gradient methods

◦ is not a claim that the residual variance is the optimal loss for the state-value or the state-action-value functions, there might be better estimators hiding in the nature!

◦ could be followed-up by further analysis of the bias-variance trade-off (ablation study indicates that `AVEC` works best with $\alpha = 0$ in $\mathcal{L}_\alpha = \mathrm{Var} + \alpha \mathrm{Bias}^2$)

◦ could be extended to stochastic environments

Thank you!

Questions?

- Flet-Berliac, Y., Ouhamma, R., Maillard, O.-A., and Preux, P. (2020). Is standard deviation the new standard? revisiting the critic in deep policy gradients.
  arXiv preprint arXiv:2010.04440.

- Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. (2018).
  Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor.
  In International Conference on Machine Learning, pages 1856–1865.

- Ilyas, A., Engstrom, L., Santurkar, S., Tsipras, D., Janoos, F., Rudolph, L., and Madry, A. (2019).
  A closer look at deep policy gradients.
  In International Conference on Learning Representations.

- Puterman, M. (1994).
  Markov Decision Processes: Discrete Stochastic Dynamic Programming.
  John Wiley & Sons.

- Schulman, J., Levine, S., Abbeel, P., Jordan, M., and Moritz, P. (2015).
  Trust region policy optimization.
  In International Conference on Machine Learning, pages 1928–1937.

- Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017).
  Proximal policy optimization algorithms.
  arXiv preprint arXiv:1707.06347.

- Sutton, R. S., McAllester, D. A., Singh, S. P., and Mansour, Y. (2000).
  Policy gradient methods for reinforcement learning with function approximation.
  In Advances in Neural Information Processing Systems.

Tucker, G., Bhupatiraju, S., Gu, S., Turner, R., Ghahramani, Z., and Levine, S. (2018).
The mirage of action-dependent baselines in reinforcement learning.
In International Conference on Machine Learning, pages 5015–5024.

Weaver, L. and Tao, N. (2001).
The optimal reward baseline for gradient · based reinforcement learning.
In Advances in Neural Information Processing Systems.

Williams, R. (1992).
Simple statistical gradient-following algorithms for connectionist reinforcement learning.
Machine Learning, 8(3-4):229–256.

More results: approximation error

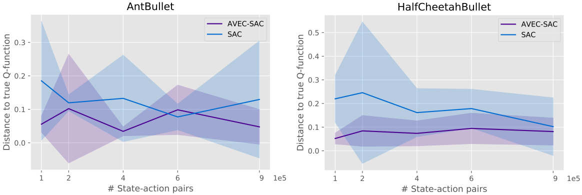


Figure: L_2 distance to V^{π}. X-axis: we run the algorithm and $\forall t \in \{1, 2, 4, 6, 9\} \cdot 10^5$ we stop training, use the current policy to collect $3 \cdot 10^5$ transitions to estimate V^{π}.

More results: empirical variance

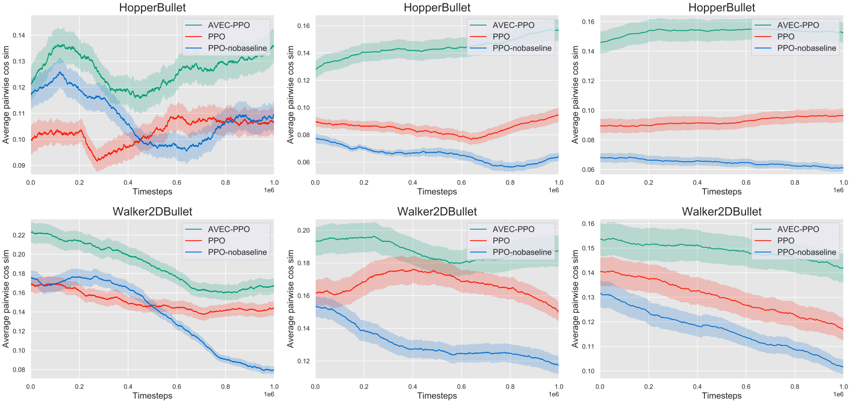
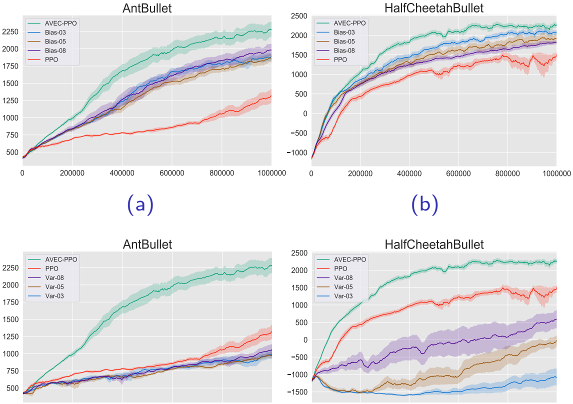


Figure: Average cosine similarity between gradient measurements. `AVEC` empirically reduces the variance compared to PPO or PPO-nobaseline. Trajectory size used in estimation of the gradient variance: 3000 (upper row), 6000 (middle row), 9000 (lower row).

More results: ablation study



(a)

(b)

(c)

(d)

Figure: Sensitivity of `AVEC`-PPO with respect to (a,b): the bias; (c,d): the variance. X-axis: number of timesteps. Y-axis: average total reward.