Divergence-Augmented Policy Optimization

Qing Wang, Yingru Li, Jiechao Xiong, Tong Zhang

Huya AI, CUHK, Tencent AI Lab, HKUST

Overview

We study the divergence augmented policy optimization method in which the optimization is guided by divergence augmented reward $r_i - 1/\eta \log \frac{\pi_\theta}{\pi_{\theta_t}}$ instead of original r_i. The method is to introduce divergence regularization to stabilize policy optimization when off-policy data are reused. We show that the above formation of divergence augmentation can be derived from the Bregman divergence calculated between the state-action distributions of two policies, instead of only on the action probabilities. Empirical experiments on Atari games show that in the data-scarce scenario where the reuse of off-policy data becomes necessary, our method can achieve better performance than other state-of-the-art deep reinforcement learning algorithms.

Preliminaries

We define the Bregman divergence as follows: With respect to a *Legendre* F, we define the Bregman divergence $D_F : \bar{\mathcal{D}} \times \mathcal{D} \to \mathbb{R}$ as

$$D_F(x, y) = F(x) - F(y) - \langle \nabla F(y), x - y \rangle.$$

The inner product is defined as $\langle x, y \rangle = \Sigma_i x_i y_i$. For $\mathcal{K} \subset \bar{\mathcal{D}}$ and $\mathcal{K} \cap \mathcal{D} \neq \emptyset$, the Bregman projection

$$z = \arg\min_{x \in \mathcal{K}} D_F(x, y)$$

exists uniquely for all $y \in \mathcal{D}$. Specifically, for $F(x) = \Sigma_i x_i \log(x_i) - \Sigma_i x_i$, we recover the Kullback-Leibler (KL) divergence as

$$D_{\mathrm{KL}}(\mu', \mu) = \Sigma_{s,a} \mu'(s,a) \log \frac{\mu'(s,a)}{\mu(s,a)}$$

for $\mu, \mu' \in \Delta(\mathcal{S} \times \mathcal{A})$ and $\pi, \pi' \in \Pi$. To measure the distance between two policies π and π', we also use the symbol for conditional "Bregman divergence" associated with state distribution d denoted as

$$D_F^d(\pi', \pi) = \Sigma_s d(s) D_F(\pi'(\cdot|s), \pi(\cdot|s)). \quad (1)$$

Motivation

Given the policy at iteration t as

$$\mu_t(s,a) = d_t(s)\pi_t(a|s)$$

we would like to find a μ_{t+1} which is close to μ_t. Inspired from mirror descent methods, given the gradient of loss as g_t and $\tilde{\mu}_{t+1} \propto \mu_t \exp(-\eta g_t)$, it is known that

$$\mu_{t+1} = \arg\min_{\mu \in \Delta_\Pi} D_{\mathrm{KL}}(\mu, \tilde{\mu}_{t+1}), \quad (2)$$

$$\mu_{t+1} = \arg\min_{\mu \in \Delta_\Pi} D_{\mathrm{KL}}(\mu, \mu_t) + \eta \langle g_t, \mu \rangle \quad (3)$$

are equivalent. In addition, if we reverse the KL divergence in (2), we get

$$\mu_{t+1} = \arg\min_{\mu \in \Delta_\Pi} D_{\mathrm{KL}}(\tilde{\mu}_{t+1}, \mu)$$

which recovers the core algorithm as in MPO and MARWIL; while if we reverse the KL divergence in (3), we get

$$\mu_{t+1} = \arg\min_{\mu \in \Delta_\Pi} D_{\mathrm{KL}}(\mu_t, \mu) + \eta \langle g_t, \mu \rangle,$$

which is related to the TRPO method. Although TRPO and MPO have been widely used in reinforcement learning community, the original formulation in (2) (3) is more natural in this view.

Divergence-Augmented Policy Optimization (DAPO)

Input: $D_F(\mu', \mu)$, total iteration T, batch size M, learning rate α_t.
Initialize : randomly initiate θ_0
For $t = 0$ to T
  (in parallel) Use $\pi_t = \pi_{\theta_t}$ to generate trajectories.
  For $m = 1$ to M
    Sample $(s_i, a_i) \in \mathcal{S} \times \mathcal{A}$ w.p. $d_t \pi_t$.
    Estimate state value v_{s_i} (e.g. by V-trace).
    Calculate Q-value est. $\hat{A}_{s,a}$ and divergence est. $\hat{D}_{s,a}$.
      $\hat{A}_{s,a} = r_i + \gamma v_{i+1} - V_\theta(s_i)$,
      $\hat{D}_{s,a} = f(s_i, a_i) + \Sigma_{j=1}^n \gamma^j (\Pi_{k=0}^{j-1} c_{i+k}) \rho_{i+j} f(s_{i+j}, a_{i+j})$.
    Update θ with respect of policy loss and value loss
      $\theta \leftarrow \theta - \alpha_t(\nabla_\theta L_\pi(\theta) + b \nabla_\theta L_v(\theta))$.
  Set $\theta_{t+1} = \theta$.

Divergence Augmentation

Solving (3) is usually hard. We consider the optimization of (3) in the (parametric) policy space without explicit projection to Δ_Π. Specifically, we consider μ_π as a function of π, and π parametrized as π_θ. The Formula (3) can be written as

$$\pi_{t+1} = \arg\min_\pi D_F(\mu_\pi, \mu_t) + \eta \langle g_t, \mu_\pi \rangle. \quad (4)$$

Instead of solving globally, we approximate Formula (4) with gradient descent on π. From the celebrated policy gradient theorem, we have

$$\Sigma_{s,a} f(s,a) \nabla_\theta \mu_\pi(s,a) = \Sigma_{s,a} d_\pi(s) \mathcal{Q}^\pi(f)(s,a) \nabla_\theta \pi(a|s),$$

for any state-action function $f : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$, and $\mathcal{Q}^\pi(\cdot)$ defined as an operator such that

$$\mathcal{Q}^\pi(f)(s,a) = \mathbb{E}_{\pi | s_t = s, a_t = a} \Sigma_{l=0}^\infty \gamma^l f(s_{t+l}, a_{t+l}).$$

Decomposing the loss and divergence in two parts (4), we have

$$\nabla_\theta \langle g_t, \mu_\pi \rangle = \langle d_\pi \mathcal{Q}^\pi(g_t), \nabla_\theta \pi(a|s) \rangle, \quad (5)$$

which is the usual policy gradient, and

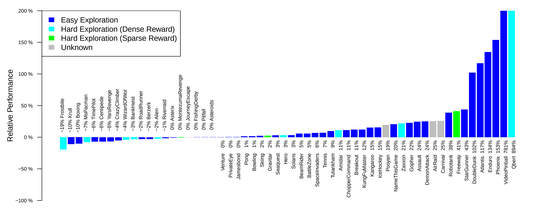
$$\nabla_\theta D_F(\mu_\pi, \mu_t) = \langle d_\pi \mathcal{Q}^\pi \left(\nabla F(\mu_\pi) - \nabla F(\mu_t) \right), \nabla_\theta \pi(a|s) \rangle. \quad (6)$$

We utilize the V-trace estimation of $\mathcal{Q}^\pi(g_t)$ and $\mathcal{Q}^\pi(\nabla F(\mu_\pi) - \nabla F(\mu_t))$ as $\hat{A}_{s,a}$ and $\hat{D}_{s,a}$. The gradient of policy loss is

$$\nabla_\theta L_\pi(\theta) = \mathbb{E}_{\pi_t, d_t} \frac{\pi}{\pi_t} (\hat{D}_{s,a} - \eta \hat{A}_{s,a}) \nabla_\theta \log \pi. \quad (7)$$

which explains the name "divergence augmentation".

Performance Comparison



PPO with and without Divergence Augmentation. We see relatively better performance in data-scarce scenarios.

Experiments

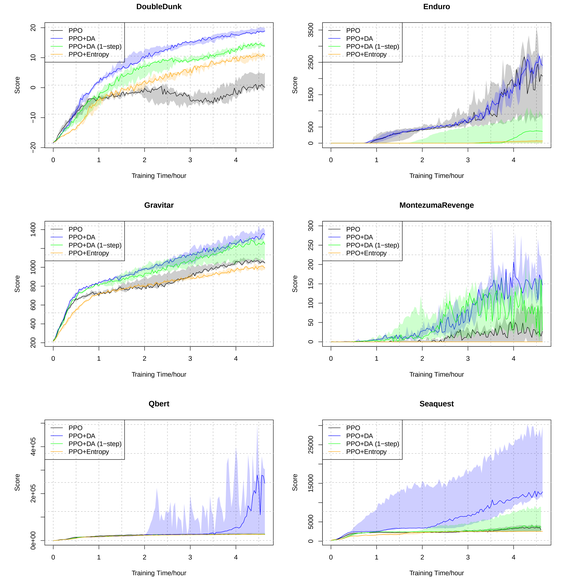


Figure 1: Performance comparison on selected environments of Atari games. The performance of **PPO**, PPO+DA, PPO+DA (1-step), and PPO+Entropy are plotted in different colors.

Conclusion

In short, we showed that divergence augmentation can be viewed as imposing Bregman divergence constraint on the state-action space, which is related to online mirror descent methods. Experiments results showed that divergence augmentation is effective when data generated by previous policies are reused in policy optimization.