An Alternate Policy Gradient Estimator for Softmax Policies

Shivam Garg[1] Samuele Tosatto[1] Yangchen Pan[2]

Martha White[1*] A. Rupam Mahmood[1*]

[1]University of Alberta. [2]Noah's Ark Lab, Huawei.
[*]CIFAR AI Chair, Alberta Machine Intelligence Institute (Amii).

Policy Gradient Methods

The goal of reinforcement learning is to learn a policy which maximizes the objective

$$\mathcal{J}_\pi := \mathbb{E}_\pi \left[\sum_{t=0}^{T} \gamma^t R_{t+1} \right].$$

Softmax policy gradient (PG) achieves this by using a softmax policy

$$\pi_{\mathbf{w}}(a|s) = \frac{e^{[\theta_{\mathbf{w}}(s)]_a}}{\sum_{b \in \mathcal{A}} e^{[\theta_{\mathbf{w}}(s)]_b}},$$

which is updated via stochastic gradient ascent:

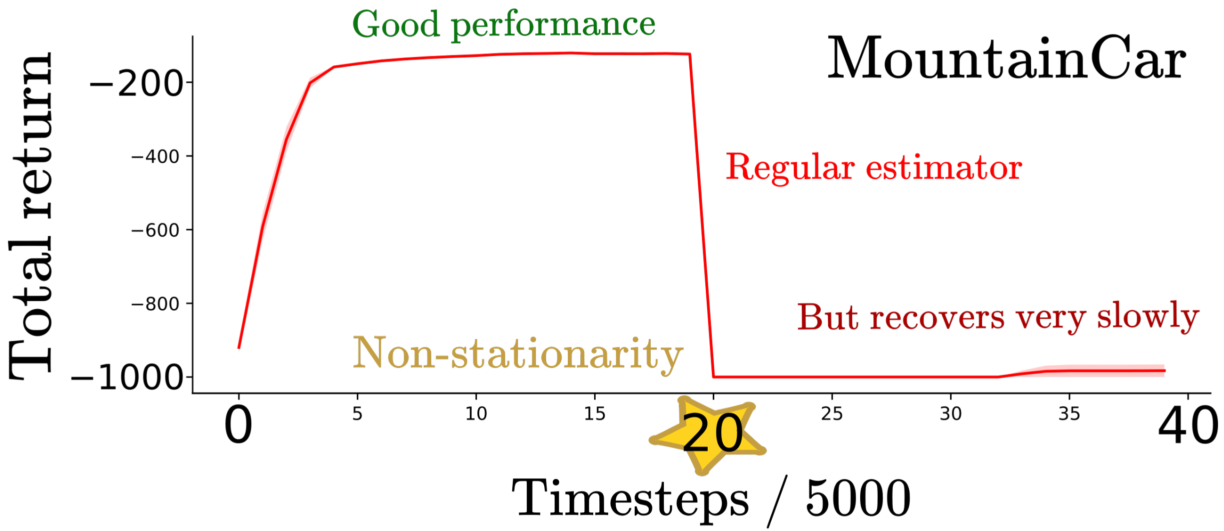
$$\mathbf{w} \leftarrow \mathbf{w} + \alpha \mathbf{g}, \quad \text{with} \quad \mathbb{E}_\pi [\mathbf{g}] = \nabla \mathcal{J}_\pi.$$

Problems with the Regular Softmax PG Estimator

The regularly used softmax estimator

$$\mathbf{g}^{\mathrm{REG}}(S, A) := \nabla_{\mathbf{w}} \log \pi(A|S) \cdot (q_\pi(S, A) - v_\pi(S)),$$

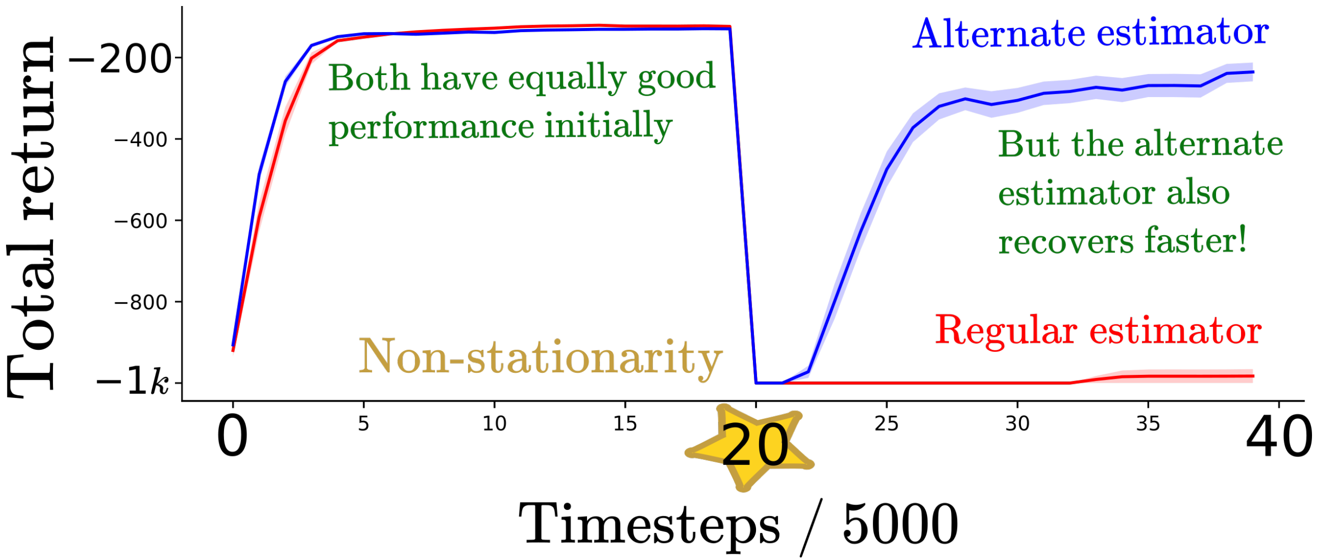
can take a large amount of time to overcome policy saturation.



Alternate PG Estimator

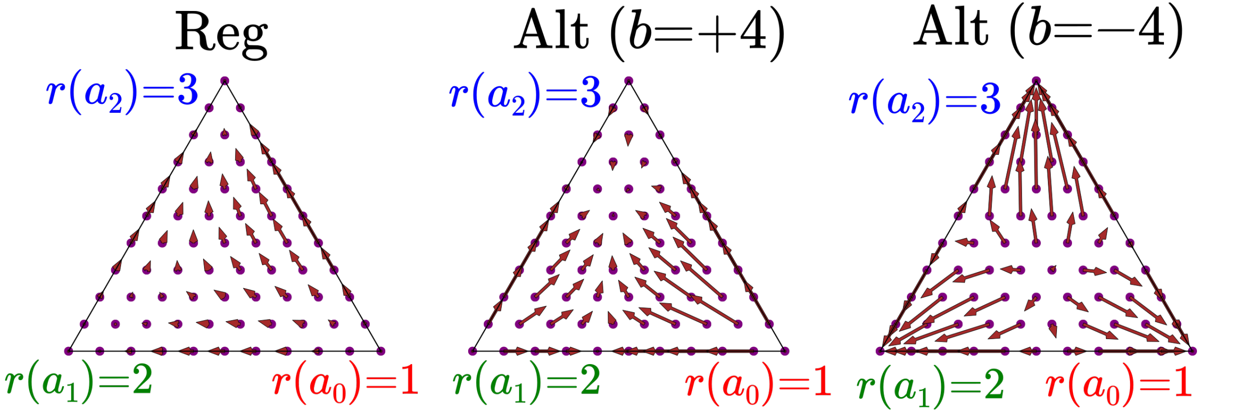
We present an alternate PG estimator that makes softmax policies robust to policy saturation:

$$\mathbf{g}^{\mathrm{ALT}}(S, A) := \nabla_{\mathbf{w}}[\theta(S)]_A \cdot (q_\pi(S, A) - v_\pi(S)).$$



Why does it work this way?

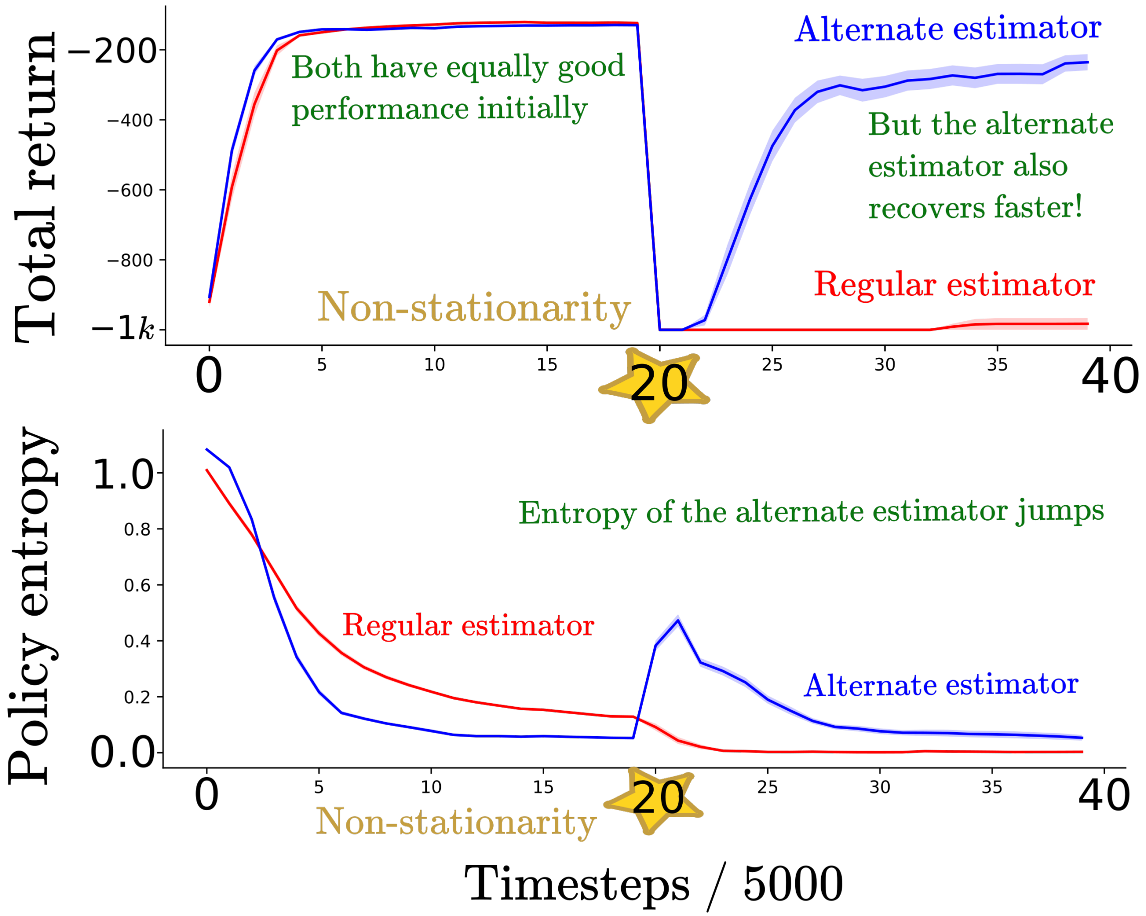
The regular has small gradients when the policy is saturated (simplex corners), leading to slow policy updates.



Expected policy updates (3-armed bandits)

The alternate estimator with an inaccurate critic estimator becomes biased, and an optimistic baseline increases the policy entropy. This enables it to overcome policy saturation.

This behavior is corroborated by the entropy plots



Conclusions

The alternate estimator makes softmax PG more suitable for non-stationary tasks.

It can also be adapted to work with various PG algo-rithms (REINFORCE, online actor-critic, PPO).

It works well with different function approximation schemes (tabular, linear, and neural architectures).

See you at the poster session!