Decision-Aware Actor-Critic with Function Approximation

Sharan Vaswani (Simon Fraser University)

Based on joint works with Olivier Bachem, Simone Totaro, Robert Müller, Shivam Garg, Matthieu Geist, Marlos Machado, Pablo Samuel Castro & Amirreza Kazemi, Reza Babanezhad, Nicolas Le Roux

Vector Institute, Toronto

Motivation

- Policy gradient (PG) methods based on REINFORCE:
  - Each policy update requires recomputing the policy gradient.
  - ✓ Theoretical guarantees [Agarwal et al., 2020] with function approximation.
  - × Each update requires computationally expensive interactions with the environment.

Motivation

- Policy gradient (PG) methods based on REINFORCE:
  - Each policy update requires recomputing the policy gradient.
  - ✓ Theoretical guarantees [Agarwal et al., 2020] with function approximation.
  - × Each update requires computationally expensive interactions with the environment.
- Methods such as TRPO, PPO and MPO:
  - Rely on constructing *surrogate functions* and update the policy to maximize these surrogates.
  - ✓ Support *off-policy updates* – can update the policy without requiring additional environment interactions. Have good empirical performance, and widely used.
  - × Only have theoretical guarantees in the tabular setting, and can fail to converge in simple scenarios [Hsu et al., 2020].

Motivation

- Policy gradient (PG) methods based on REINFORCE:
  - Each policy update requires recomputing the policy gradient.
  - ✓ Theoretical guarantees [Agarwal et al., 2020] with function approximation.
  - × Each update requires computationally expensive interactions with the environment.
- Methods such as TRPO, PPO and MPO:
  - Rely on constructing *surrogate functions* and update the policy to maximize these surrogates.
  - ✓ Support *off-policy updates* – can update the policy without requiring additional environment interactions. Have good empirical performance, and widely used.
  - × Only have theoretical guarantees in the tabular setting, and can fail to converge in simple scenarios [Hsu et al., 2020].

No systematic way to design theoretically principled surrogate functions, or a unified framework to analyze their properties.

Outline

- **Problem Formulation**
- Functional Mirror Ascent for Policy Gradient (FMA-PG) Framework
  - Theoretical Guarantees
  - Instantiating the FMA-PG Framework
- Decision-aware Actor-Critic
  - Instantiating the AC Framework
  - Theoretical Guarantees
- Conclusions and Future Work

Problem Formulation

- Infinite-horizon discounted MDP: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \rho, \gamma \rangle$.

Problem Formulation

- Infinite-horizon discounted MDP: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \rho, \gamma \rangle$.
- Distributions induced by policy π: For each state $s \in \mathcal{S}$, $p^\pi(\cdot|s)$ over actions. State occupancy measure: $d^\pi(s) = (1-\gamma)\sum_{\tau=0}^{\infty} \gamma^\tau \mathbb{P}(s_\tau = s \mid s_0 \sim \rho, a_\tau \sim p^\pi(\cdot|s_\tau))$. State-action occupancy measure: $\mu^\pi(s,a) = d^\pi(s) p^\pi(a|s)$.

Problem Formulation

- Infinite-horizon discounted MDP: $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, p, r, \rho, \gamma \rangle$.
- Distributions induced by policy π: For each state $s \in \mathcal{S}$, $p^\pi(\cdot|s)$ over actions. State occupancy measure: $d^\pi(s) = (1-\gamma)\sum_{\tau=0}^{\infty} \gamma^\tau \mathbb{P}(s_\tau = s \mid s_0 \sim \rho, a_\tau \sim p^\pi(\cdot|s_\tau))$. State-action occupancy measure: $\mu^\pi(s,a) = d^\pi(s)p^\pi(a|s)$.
- Expected discounted return for π: $J(\pi) = \mathbb{E}_{s_0, a_0, \ldots}[\sum_{\tau=0}^{\infty} \gamma^\tau r(s_\tau, a_\tau)]$, where $s_0 \sim \rho, a_\tau \sim p^\pi(\cdot|s_\tau)$, and $s_{\tau+1} \sim p(\cdot|s_\tau, a_\tau)$.
- **Objective**: Given a set of feasible policies Π, $\max_{\pi \in \Pi} J(\pi)$. $\pi^* := \arg\max_{\pi \in \Pi} J(\pi)$.

Functional representation vs Policy parameterization

- **Functional representation**: Specifies a policy's sufficient statistics and is implicit.

  Examples:
  - *Direct functional representation*: Conditional distribution over actions $p^\pi(\cdot|s)$ for each s.
  - *Softmax functional representation*: Logits $z^\pi(s, a)$ such that $p^\pi(a|s) = \frac{\exp(z^\pi(s,a))}{\sum_{a'} \exp(z^\pi(s,a'))}$.

Functional representation vs Policy parameterization

- **Functional representation**: Specifies a policy's sufficient statistics and is implicit. *Examples*:
  - *Direct functional representation*: Conditional distribution over actions $p^\pi(\cdot|s)$ for each s.
  - *Softmax functional representation*: Logits $z^\pi(s,a)$ such that $p^\pi(a|s) = \frac{\exp(z^\pi(s,a))}{\sum_{a'} \exp(z^\pi(s,a'))}$.
- **Policy parameterization**: Practical realization of the sufficient statistics. Determines Π (the set of feasible policies). *Examples*:
  - *Tabular parameterization* for the direct functional representation: $p^\pi(a|s) = \theta(s,a)$.
  - *Linear parameterization* for the softmax functional representation: $z^\pi(s,a) = \langle \theta, \mathrm{X}(s,a) \rangle$, where $\mathrm{X}(s,a)$ are the state-action features and $\theta \in \mathbb{R}^d$ are the parameters of a linear model.

Functional representation vs Policy parameterization

- **Functional representation**: Specifies a policy's sufficient statistics and is implicit.
  Examples:
  - *Direct functional representation*: Conditional distribution over actions $p^\pi(\cdot|s)$ for each s.
  - *Softmax functional representation*: Logits $z^\pi(s,a)$ such that $p^\pi(a|s) = \frac{\exp(z^\pi(s,a))}{\sum_{a'} \exp(z^\pi(s,a'))}$.
- **Policy parameterization**: Practical realization of the sufficient statistics. Determines Π (the set of feasible policies). *Examples*:
  - *Tabular parameterization* for the direct functional representation: $p^\pi(a|s) = \theta(s,a)$.
  - *Linear parameterization* for the softmax functional representation: $z^\pi(s,a) = \langle \theta, \mathrm{X}(s,a) \rangle$, where $\mathrm{X}(s,a)$ are the state-action features and $\theta \in \mathbb{R}^d$ are the parameters of a linear model.
- The functional representation of a policy is independent of its parameterization.

Functional representation vs Policy parameterization

- **Functional representation**: Specifies a policy's sufficient statistics and is implicit. *Examples*:
  - *Direct functional representation*: Conditional distribution over actions $p^\pi(\cdot|s)$ for each s.
  - *Softmax functional representation*: Logits $z^\pi(s,a)$ such that $p^\pi(a|s) = \frac{\exp(z^\pi(s,a))}{\sum_{a'}\exp(z^\pi(s,a'))}$.
- **Policy parameterization**: Practical realization of the sufficient statistics. Determines Π (the set of feasible policies). *Examples*:
  - *Tabular parameterization* for the direct functional representation: $p^\pi(a|s) = \theta(s,a)$.
  - *Linear parameterization* for the softmax functional representation: $z^\pi(s,a) = \langle\theta, \mathrm{X}(s,a)\rangle$, where $\mathrm{X}(s,a)$ are the state-action features and $\theta \in \mathbb{R}^d$ are the parameters of a linear model.
- The functional representation of a policy is independent of its parameterization.
- **Standard PG approach**: Use a model (with parameters θ) to parameterize (the functional representation of) π and directly optimize $J(\pi(\theta))$ w.r.t. θ.

Outline

- Problem Formulation
- **Functional Mirror Ascent for Policy Gradient (FMA-PG) Framework**
  - Theoretical Guarantees
  - Instantiating the FMA-PG Framework
- Decision-aware Actor-Critic
  - Instantiating the AC Framework
  - Theoretical Guarantees
- Conclusions and Future Work

Functional Mirror Ascent

- Idea: Iteratively optimize J w.r.t π and project onto Π (depends on the parameterization).

Functional Mirror Ascent

- Idea: Iteratively optimize J w.r.t π and project onto Π (depends on the parameterization).
- Overload π to be a general functional representation, with $\pi(\theta)$ as its parametric realization.

Functional Mirror Ascent

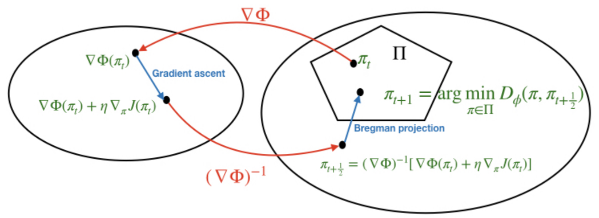
- Idea: Iteratively optimize J w.r.t π and project onto Π (depends on the parameterization).
- Overload π to be a general functional representation, with $\pi(\theta)$ as its parametric realization.
- For a strictly convex, differentiable function Φ (*mirror map*), $D_\Phi(\pi, \pi')$ is the *Bregman divergence* between policies π and π'. $D_\Phi(\pi, \pi') := \Phi(\pi) - \Phi(\pi') - \langle \nabla\Phi(\pi'), \pi - \pi' \rangle$.
- E.g. If $\Phi(\pi) = \frac{1}{2} \|\pi\|_2^2$, $D_\Phi(\pi, \pi') = \frac{1}{2} \|\pi - \pi'\|_2^2$.

Functional Mirror Ascent

- Idea: Iteratively optimize J w.r.t π and project onto Π (depends on the parameterization).
- Overload π to be a general functional representation, with $\pi(\theta)$ as its parametric realization.
- For a strictly convex, differentiable function Φ (*mirror map*), $D_\Phi(\pi, \pi')$ is the *Bregman divergence* between policies π and π'. $D_\Phi(\pi, \pi') := \Phi(\pi) - \Phi(\pi') - \langle \nabla\Phi(\pi'), \pi - \pi' \rangle$.
- E.g. If $\Phi(\pi) = \frac{1}{2}\|\pi\|_2^2$, $D_\Phi(\pi, \pi') = \frac{1}{2}\|\pi - \pi'\|_2^2$.

In each iteration $t \in [T]$ of *functional mirror ascent* (FMA), with *step-size* η,

$$\pi_{t+1/2} = (\nabla\Phi)^{-1}\left(\nabla\Phi(\pi_t) + \eta\nabla_\pi J(\pi_t)\right) \quad ; \quad \pi_{t+1} = \arg\min_{\pi \in \Pi} D_\Phi(\pi, \pi_{t+1/2})$$



Functional Mirror Ascent

- Idea: Iteratively optimize J w.r.t π and project onto Π (depends on the parameterization).
- Overload π to be a general functional representation, with $\pi(\theta)$ as its parametric realization.
- For a strictly convex, differentiable function Φ (*mirror map*), $D_\Phi(\pi, \pi')$ is the *Bregman divergence* between policies π and π'. $D_\Phi(\pi, \pi') := \Phi(\pi) - \Phi(\pi') - \langle \nabla \Phi(\pi'), \pi - \pi' \rangle$.
- E.g. If $\Phi(\pi) = \frac{1}{2} \|\pi\|_2^2$, $D_\Phi(\pi, \pi') = \frac{1}{2} \|\pi - \pi'\|_2^2$.

In each iteration $t \in [T]$ of *functional mirror ascent* (FMA), with *step-size* η,

$$\pi_{t+1/2} = (\nabla \Phi)^{-1} \left(\nabla \Phi(\pi_t) + \eta \nabla_\pi J(\pi_t)\right) \quad ; \quad \pi_{t+1} = \arg\min_{\pi \in \Pi} D_\Phi(\pi, \pi_{t+1/2})$$

$$\pi_{t+1} = \arg\max_{\pi \in \Pi} \left[\langle \pi, \, \nabla_\pi J(\pi_t) \rangle - \frac{1}{\eta} D_\Phi(\pi, \pi_t) \right]$$

FMA-PG Framework

- The complexity of the projection onto Π depends on the parameterization. *Examples*:
  - For a tabular parameterization, Π allows all memoryless policies.
  - For a linear parameterization, Π is restricted, but is a convex set in θ.
  - For a neural network, Π is restricted and non-convex, making the projection ill-defined.

FMA-PG Framework

- The complexity of the projection onto Π depends on the parameterization. *Examples*:
  - For a tabular parameterization, Π allows all memoryless policies.
  - For a linear parameterization, Π is restricted, but is a convex set in θ.
  - For a neural network, Π is restricted and non-convex, making the projection ill-defined.

If Π consists of policies realizable by a parametric model, then

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} D_\Phi(\pi, \pi_{t+1/2}) = \arg\min_{\theta \in \mathbb{R}^d} D_\Phi(\pi(\theta), \pi_{t+1/2}) \qquad \text{(\textbf{Reparameterization})}$$

Ensures that $\pi_{t+1} \in \Pi$.

FMA-PG Framework

- The complexity of the projection onto Π depends on the parameterization. *Examples*:
  - For a tabular parameterization, Π allows all memoryless policies.
  - For a linear parameterization, Π is restricted, but is a convex set in θ.
  - For a neural network, Π is restricted and non-convex, making the projection ill-defined.

If Π consists of policies realizable by a parametric model, then

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} D_\Phi(\pi, \pi_{t+1/2}) = \arg\min_{\theta \in \mathbb{R}^d} D_\Phi(\pi(\theta), \pi_{t+1/2}) \qquad (\textbf{Reparameterization})$$

Ensures that $\pi_{t+1} \in \Pi$.

With this reparameterization, the FMA update can be rewritten as:

$$\pi_{t+1} = \pi(\theta_{t+1}) \quad ; \quad \theta_{t+1} = \arg\max_{\theta \in \mathbb{R}^d} \underbrace{\left[J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t)) \right]}_{\text{Surrogate function } \ell_t^{\pi,\Phi,\eta}(\theta)}$$

FMA-PG Framework

- The complexity of the projection onto Π depends on the parameterization. *Examples*:
  - For a tabular parameterization, Π allows all memoryless policies.
  - For a linear parameterization, Π is restricted, but is a convex set in θ.
  - For a neural network, Π is restricted and non-convex, making the projection ill-defined.

If Π consists of policies realizable by a parametric model, then

$$\pi_{t+1} = \arg\min_{\pi \in \Pi} D_\Phi(\pi, \pi_{t+1/2}) = \arg\min_{\theta \in \mathbb{R}^d} D_\Phi(\pi(\theta), \pi_{t+1/2}) \qquad \text{(\textbf{Reparameterization})}$$

Ensures that $\pi_{t+1} \in \Pi$.

With this reparameterization, the FMA update can be rewritten as:

$$\pi_{t+1} = \pi(\theta_{t+1}) \quad ; \quad \theta_{t+1} = \arg\max_{\theta \in \mathbb{R}^d} \underbrace{\left[J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t)) \right]}_{\text{Surrogate function } \ell_t^{\pi,\Phi,\eta}(\theta)}$$

$\ell_t(\theta)$ is non-concave in general, and we optimize it using a gradient-based method.

FMA-PG Framework – Algorithm

Algorithm 1: Generic policy optimization

Input: π (functional representation), θ_0 (initial policy parameterization), T (PG iterations), m (inner-loops), η (step-size for functional update), α (step-size for parametric update)

for $t \leftarrow 0$ **to** $T-1$ **do**

  Compute $\nabla_\pi J(\pi_t)$ and form the surrogate $\ell_t^{\pi,\Phi,\eta}(\theta)$.

  Initialize inner-loop: $\omega_0 = \theta_t$

  for $k \leftarrow 0$ **to** m **do**

    $\omega_{k+1} = \omega_k + \alpha \nabla_\omega \ell_t^{\pi,\Phi,\eta}(\omega_k)$ /* `Off-policy actor updates` */

  $\theta_{t+1} = \omega_m$

  $\pi_{t+1} = \pi(\theta_{t+1})$

Return θ_T

Theoretical Guarantees

- Recall that, $\ell_t(\theta) = J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t))$.
- Sufficient conditions to ensure monotonic policy improvement, i.e. $J(\pi_{t+1}) \geq J(\pi_t)$:
  (i) $\ell_t(\theta_{t+1}) \geq \ell_t(\theta_t)$, [Inner-loop improves the surrogate value]
  (ii) $\ell_t(\theta) \leq J(\pi(\theta))$ *for all* θ. [Surrogate is a global lower bound on $J(\pi(\theta))$]

Theoretical Guarantees

- Recall that, $\ell_t(\theta) = J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t))$.
- Sufficient conditions to ensure monotonic policy improvement, i.e. $J(\pi_{t+1}) \geq J(\pi_t)$:
  (i) $\ell_t(\theta_{t+1}) \geq \ell_t(\theta_t)$, [Inner-loop improves the surrogate value]
  (ii) $\ell_t(\theta) \leq J(\pi(\theta))$ *for all* θ. [Surrogate is a global lower bound on $J(\pi(\theta))$]

Theoretical Guarantees

- Recall that, $\ell_t(\theta) = J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t))$.
- Sufficient conditions to ensure monotonic policy improvement, i.e. $J(\pi_{t+1}) \geq J(\pi_t)$:
  (i) $\ell_t(\theta_{t+1}) \geq \ell_t(\theta_t)$, [Inner-loop improves the surrogate value]
  (ii) $\ell_t(\theta) \leq J(\pi(\theta))$ *for all* θ. [Surrogate is a global lower bound on $J(\pi(\theta))$]

If these conditions are satisfied, then,

$$J(\pi_{t+1}) \overset{Def}{=} J(\pi(\theta_{t+1})) \overset{(ii)}{\geq} \ell_t(\theta_{t+1}) \overset{(i)}{\geq} \ell_t(\theta_t) \overset{Def}{=} J(\pi(\theta_t)) \overset{Def}{=} J(\pi_t)$$

Since $J(\pi)$ is upper-bounded by $\frac{1}{1-\gamma}$, this guarantees convergence to a stationary point for any complicated policy parameterization.

Theoretical Guarantees

- Recall that, $\ell_t(\theta) = J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t))$.
- Sufficient conditions to ensure monotonic policy improvement, i.e. $J(\pi_{t+1}) \geq J(\pi_t)$:
  (i) $\ell_t(\theta_{t+1}) \geq \ell_t(\theta_t)$, [Inner-loop improves the surrogate value]
  (ii) $\ell_t(\theta) \leq J(\pi(\theta))$ *for all* θ. [Surrogate is a global lower bound on $J(\pi(\theta))$]

If these conditions are satisfied, then,

$$J(\pi_{t+1}) \overset{Def}{=} J(\pi(\theta_{t+1})) \overset{(ii)}{\geq} \ell_t(\theta_{t+1}) \overset{(i)}{\geq} \ell_t(\theta_t) \overset{Def}{=} J(\pi(\theta_t)) \overset{Def}{=} J(\pi_t)$$

Since $J(\pi)$ is upper-bounded by $\frac{1}{1-\gamma}$, this guarantees convergence to a stationary point for any complicated policy parameterization.

- (i) is satisfied by setting the *parametric* step-size α according to the smoothness of $\ell_t(\theta)$. Specifically, if $\ell_t(\theta)$ is β-smooth, any $\alpha \leq \frac{1}{\beta}$ and $m \geq 1$ guarantees (i).

Theoretical Guarantees

- Recall that, $\ell_t(\theta) = J(\pi(\theta_t)) + \langle \pi(\theta) - \pi(\theta_t), \nabla_\pi J(\pi(\theta_t)) \rangle - \frac{1}{\eta} D_\Phi(\pi(\theta), \pi(\theta_t))$.
- Sufficient conditions to ensure monotonic policy improvement, i.e. $J(\pi_{t+1}) \geq J(\pi_t)$:
  (i) $\ell_t(\theta_{t+1}) \geq \ell_t(\theta_t)$, [Inner-loop improves the surrogate value]
  (ii) $\ell_t(\theta) \leq J(\pi(\theta))$ *for all* θ. [Surrogate is a global lower bound on $J(\pi(\theta))$]

If these conditions are satisfied, then,

$$J(\pi_{t+1}) \overset{Def}{=} J(\pi(\theta_{t+1})) \overset{(ii)}{\geq} \ell_t(\theta_{t+1}) \overset{(i)}{\geq} \ell_t(\theta_t) \overset{Def}{=} J(\pi(\theta_t)) \overset{Def}{=} J(\pi_t)$$

Since $J(\pi)$ is upper-bounded by $\frac{1}{1-\gamma}$, this guarantees convergence to a stationary point for any complicated policy parameterization.

- (i) is satisfied by setting the *parametric* step-size α according to the smoothness of $\ell_t(\theta)$. Specifically, if $\ell_t(\theta)$ is β-smooth, any $\alpha \leq \frac{1}{\beta}$ and $m \geq 1$ guarantees (i).
- (ii) is satisfied by setting the *functional* step-size η according to the relative smoothness of $J(\pi)$ w.r.t D_Φ. Specifically, any η that ensures $J + \frac{1}{\eta}\Phi$ is a convex function guarantees (ii).

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^{\pi}(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^\pi(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_\phi(p^\pi(\cdot|s), p^{\pi'}(\cdot|s))$.

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^{\pi}(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.
- We choose $D_{\Phi}(\pi, \pi') = \sum_s d^{\pi}(s)\, D_{\phi}(p^{\pi}(\cdot|s), p^{\pi'}(\cdot|s))$.

Since $\frac{\partial J(\pi)}{\partial p^{\pi}(a|s)} = d^{\pi}(s) Q^{\pi}(s, a)$, the surrogate function at iteration t is given by,

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^\pi(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_\phi(p^\pi(\cdot|s), p^{\pi'}(\cdot|s))$.

Since $\frac{\partial J(\pi)}{\partial p^\pi(a|s)} = d^\pi(s) Q^\pi(s, a)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[D_\phi(p^\pi(\cdot|s,\theta), p^\pi(\cdot|s,\theta_t))\right] + C.$$

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^\pi(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_\phi(p^\pi(\cdot|s), p^{\pi'}(\cdot|s))$.

Since $\frac{\partial J(\pi)}{\partial p^\pi(a|s)} = d^\pi(s) Q^\pi(s,a)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[D_\phi(p^\pi(\cdot|s,\theta), p^\pi(\cdot|s,\theta_t))\right] + C.$$

For the negative entropy mirror-map i.e. when $\phi(p^\pi(\cdot|s)) = \sum_a p^\pi(a|s) \log p^\pi(a|s)$,

$$\ell_t^{\pi,\mathrm{NE},\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[\mathrm{KL}\left(p^\pi(\cdot|s,\theta)||p^\pi(\cdot|s,\theta_t)\right)\right] + C.$$

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^\pi(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_\phi(p^\pi(\cdot|s), p^{\pi'}(\cdot|s))$.

Since $\frac{\partial J(\pi)}{\partial p^\pi(a|s)} = d^\pi(s) Q^\pi(s,a)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[D_\phi(p^\pi(\cdot|s,\theta), p^\pi(\cdot|s,\theta_t))\right] + C.$$

For the negative entropy mirror-map i.e. when $\phi(p^\pi(\cdot|s)) = \sum_a p^\pi(a|s)\log p^\pi(a|s)$,

$$\ell_t^{\pi,\mathrm{NE},\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[\mathrm{KL}\left(p^\pi(\cdot|s,\theta)||p^\pi(\cdot|s,\theta_t)\right)\right] + C.$$

Setting η for the direct functional representation with negative entropy mirror map

For any policy parameterization, $\forall\theta$, $J(\pi(\theta)) \geq \ell_t^{\pi,\mathrm{NE},\eta}(\theta)$ for $\eta \leq \frac{(1-\gamma)^3}{2\gamma|A|}$.

Instantiating FMA-PG – Direct functional representation

- Policy is represented by distributions $p^\pi(\cdot|s)$ over actions for each state $s \in \mathcal{S}$.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_\phi(p^\pi(\cdot|s), p^{\pi'}(\cdot|s))$.

Since $\frac{\partial J(\pi)}{\partial p^\pi(a|s)} = d^\pi(s) Q^\pi(s,a)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[D_\phi(p^\pi(\cdot|s,\theta), p^\pi(\cdot|s,\theta_t))\right] + C.$$

For the negative entropy mirror-map i.e. when $\phi(p^\pi(\cdot|s)) = \sum_a p^\pi(a|s) \log p^\pi(a|s)$,

$$\ell_t^{\pi,\mathrm{NE},\eta}(\theta) = \mathbb{E}_{(s,a)\sim\mu^{\pi_t}}\left[\left(Q^{\pi_t}(s,a)\,\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right)\right] - \frac{1}{\eta}\mathbb{E}_{s\sim d^{\pi_t}}\left[\mathrm{KL}\left(p^\pi(\cdot|s,\theta)||p^\pi(\cdot|s,\theta_t)\right)\right] + C.$$

Setting η for the direct functional representation with negative entropy mirror map

For any policy parameterization, $\forall\theta$, $J(\pi(\theta)) \geq \ell_t^{\pi,\mathrm{NE},\eta}(\theta)$ for $\eta \leq \frac{(1-\gamma)^3}{2\gamma|A|}$.

- × Involves the importance-sampling ratio $\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}$ that could be potentially large.
- × Involves the reverse KL divergence making it *mode seeking* hindering exploration.

Instantiating FMA-PG – Softmax functional representation

- Policy is represented by the logits $z^\pi(s, a)$ such that $p^\pi(a|s) \propto \exp(z^\pi(s, a))$ for each state.

Instantiating FMA-PG – Softmax functional representation

- Policy is represented by the logits $z^\pi(s, a)$ such that $p^\pi(a|s) \propto \exp(z^\pi(s, a))$ for each state.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_{\phi_z}(z(s, \cdot), z'(s, \cdot))$.

Instantiating FMA-PG – Softmax functional representation

- Policy is represented by the logits $z^\pi(s,a)$ such that $p^\pi(a|s) \propto \exp(z^\pi(s,a))$ for each state.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_{\phi_z}(z(s,\cdot), z'(s,\cdot))$.

Since $\frac{\partial J(\pi)}{\partial z^\pi(s,a)} = d^\pi(s) A^\pi(s,a) p^\pi(a|s)$, the surrogate function at iteration t is given by,

Instantiating FMA-PG – Softmax functional representation

- Policy is represented by the logits $z^\pi(s,a)$ such that $p^\pi(a|s) \propto \exp(z^\pi(s,a))$ for each state.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_{\phi_z}(z(s,\cdot), z'(s,\cdot))$.

Since $\frac{\partial J(\pi)}{\partial z^\pi(s,a)} = d^\pi(s) A^\pi(s,a) p^\pi(a|s)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = E_{(s,a)\sim\mu^{\pi_t}}\left[A^{\pi_t}(s,a)\, z^\pi(s,a|\theta_t)\right] - \frac{1}{\eta}\sum_s d^{\pi_t}(s)\, D_{\phi_z}\left(z^\pi(s,\cdot|\theta), z^\pi(s,\cdot|\theta_t)\right) + C.$$

Instantiating FMA-PG – Softmax functional representation

- Policy is represented by the logits $z^\pi(s,a)$ such that $p^\pi(a|s) \propto \exp(z^\pi(s,a))$ for each state.
- We choose $D_\Phi(\pi,\pi') = \sum_s d^\pi(s)\, D_{\phi_z}(z(s,\cdot), z'(s,\cdot))$.

Since $\frac{\partial J(\pi)}{\partial z^\pi(s,a)} = d^\pi(s) A^\pi(s,a) p^\pi(a|s)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = E_{(s,a)\sim\mu^{\pi_t}}\left[A^{\pi_t}(s,a)\, z^\pi(s,a|\theta_t)\right] - \frac{1}{\eta}\sum_s d^{\pi_t}(s)\, D_{\phi_z}\left(z^\pi(s,\cdot|\theta), z^\pi(s,\cdot|\theta_t)\right) + C.$$

For the log-sum-exp mirror-map i.e. when $\phi_z(z(s,\cdot)) = \log\left(\sum_a \exp(z^\pi(s,a))\right)$,

$$\ell_t^{\pi,\mathrm{LSE},\eta}(\theta) = E_{(s,a)\sim\mu^{\pi_t}}\left[\left(A^{\pi_t}(s,a) + \frac{1}{\eta}\right)\log\frac{p^\pi(a|s,\theta)}{p^\pi(a|s,\theta_t)}\right] + C.$$

Instantiating FMA-PG – Softmax functional representation

- Policy is represented by the logits $z^\pi(s, a)$ such that $p^\pi(a|s) \propto \exp(z^\pi(s, a))$ for each state.
- We choose $D_\Phi(\pi, \pi') = \sum_s d^\pi(s)\, D_{\phi_z}(z(s, \cdot), z'(s, \cdot))$.

Since $\frac{\partial J(\pi)}{\partial z^\pi(s,a)} = d^\pi(s) A^\pi(s, a) p^\pi(a|s)$, the surrogate function at iteration t is given by,

$$\ell_t^{\pi,\Phi,\eta}(\theta) = E_{(s,a)\sim\mu^{\pi_t}}\left[A^{\pi_t}(s, a)\, z^\pi(s, a|\theta_t)\right] - \frac{1}{\eta}\sum_s d^{\pi_t}(s)\, D_{\phi_z}\left(z^\pi(s, \cdot|\theta), z^\pi(s, \cdot|\theta_t)\right) + C.$$

For the log-sum-exp mirror-map i.e. when $\phi_z(z(s, \cdot)) = \log\left(\sum_a \exp(z^\pi(s, a))\right)$,

$$\ell_t^{\pi,\text{LSE},\eta}(\theta) = E_{(s,a)\sim\mu^{\pi_t}}\left[\left(A^{\pi_t}(s, a) + \frac{1}{\eta}\right) \log \frac{p^\pi(a|s, \theta)}{p^\pi(a|s, \theta_t)}\right] + C.$$

Setting η for the softmax functional representation with log-sum-exp mirror map

For any policy parameterization, $\forall\theta$, $J(\pi(\theta)) \geq \ell_t^{\pi,\text{LSE},\eta}(\theta)$ for $\eta \leq 1 - \gamma$.

Instantiating FMA-PG – Softmax functional representation

The surrogate can be rewritten as

$$\ell_t^{\pi,\mathsf{LSE},\eta}(\theta) = \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}}\left(A^{\pi_t}(s,a)\log\frac{p^{\pi}(a|s,\theta)}{p^{\pi}(a|s,\theta_t)}\right) - \frac{1}{\eta}\mathrm{KL}(p^{\pi}(\cdot|s,\theta_t)||p^{\pi}(\cdot|s,\theta))\right] + C.$$

Instantiating FMA-PG – Softmax functional representation

The surrogate can be rewritten as

$$\ell_t^{\pi,\text{LSE},\eta}(\theta) = \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}}\left(A^{\pi_t}(s,a)\log\frac{p^{\pi}(a|s,\theta)}{p^{\pi}(a|s,\theta_t)}\right) - \frac{1}{\eta}\text{KL}(p^{\pi}(\cdot|s,\theta_t)||p^{\pi}(\cdot|s,\theta))\right] + C.$$

- ✓ Compared to $\ell_t^{\pi,\text{NE},\eta}(\theta)$, the above surrogate depends on the log of the importance sampling ratio.
- ✓ Surrogate involves the forward KL divergence making it *mode covering* encouraging exploration.

Instantiating FMA-PG – Softmax functional representation

The surrogate can be rewritten as

$$\ell_t^{\pi,\mathsf{LSE},\eta}(\theta) = \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}}\left(A^{\pi_t}(s,a)\log\frac{p^{\pi}(a|s,\theta)}{p^{\pi}(a|s,\theta_t)}\right) - \frac{1}{\eta}\mathrm{KL}(p^{\pi}(\cdot|s,\theta_t)||p^{\pi}(\cdot|s,\theta))\right] + C.$$

- ✓ Compared to $\ell_t^{\pi,\mathsf{NE},\eta}(\theta)$, the above surrogate depends on the log of the importance sampling ratio.
- ✓ Surrogate involves the forward KL divergence making it *mode covering* encouraging exploration.

Compared to TRPO: $\max_{\theta\in\mathbb{R}^d}\mathbb{E}_{(s,a)\sim\mu^{\pi_t}}[A^{\pi_t}(s,a)\,\frac{p^{\pi}(a|s,\theta)}{p^{\pi}(a|s,\theta_t)}]$ s.t. $\mathbb{E}_{s\sim d^{\pi_t}}[\mathrm{KL}(p^{\pi_t}(\cdot|s,\theta_t)||p^{\pi}(\cdot|s,\theta))]\leq\delta$,

- $\ell_t^{\pi,\mathsf{LSE},\eta}(\theta)$ involves the log of the importance sampling ratio, and enforces proximity between policies using a regularization (with parameter $1/\eta$) rather than a constraint.
- we ensure monotonic policy improvement for any policy parameterization.

Conclusion

✓ Used functional mirror ascent to propose FMA-PG, a systematic way to define surrogate functions for generic policy optimization. Ensures monotonic policy improvement for arbitrary policy parameterization.

Conclusion

- ✓ Used functional mirror ascent to propose FMA-PG, a systematic way to define surrogate functions for generic policy optimization. Ensures monotonic policy improvement for arbitrary policy parameterization.
- ✓ Can use the FMA-PG framework to "lift" existing theoretical guarantees [Mei et al., 2020, Xiao, 2022] for policy optimization algorithms in the tabular setting to use off-policy updates and function approximation.

Conclusion

- ✓ Used functional mirror ascent to propose FMA-PG, a systematic way to define surrogate functions for generic policy optimization. Ensures monotonic policy improvement for arbitrary policy parameterization.
- ✓ Can use the FMA-PG framework to “lift” existing theoretical guarantees [Mei et al., 2020, Xiao, 2022] for policy optimization algorithms in the tabular setting to use off-policy updates and function approximation.
- ✓ Show experimental evidence that on simple tabular MDPs, the algorithms instantiated with FMA-PG are competitive with popular PG algorithms such as TRPO, PPO. The framework suggests an alternative method, `sPPO` that out-performs PPO on the MuJoCo suite.

Motivation

- × FMA-PG relies on the knowledge of the true gradient $\nabla_\pi J(\pi)$, which involves either the action-value (Q^π) or the advantage (A^π) functions. This information is rarely available, making FMA-PG impractical in realistic settings.

Motivation

- × FMA-PG relies on the knowledge of the true gradient $\nabla_\pi J(\pi)$, which involves either the action-value (Q^π) or the advantage (A^π) functions. This information is rarely available, making FMA-PG impractical in realistic settings.
- Can estimate $\nabla_\pi J(\pi)$ using Monte-Carlo samples obtained via environment interactions [Williams, 1992] and use the estimated gradient.

Motivation

- × FMA-PG relies on the knowledge of the true gradient $\nabla_\pi J(\pi)$, which involves either the action-value (Q^π) or the advantage (A^π) functions. This information is rarely available, making FMA-PG impractical in realistic settings.
- Can estimate $\nabla_\pi J(\pi)$ using Monte-Carlo samples obtained via environment interactions [Williams, 1992] and use the estimated gradient.
  - × Resulting estimator has high variance, leading to higher sample-complexity.

Motivation

× FMA-PG relies on the knowledge of the true gradient $\nabla_\pi J(\pi)$, which involves either the action-value (Q^π) or the advantage (A^π) functions. This information is rarely available, making FMA-PG impractical in realistic settings.

- Can estimate $\nabla_\pi J(\pi)$ using Monte-Carlo samples obtained via environment interactions [Williams, 1992] and use the estimated gradient.
  - × Resulting estimator has high variance, leading to higher sample-complexity.
- Can estimate $\nabla_\pi J(\pi)$ using a value-based method ("critic"). Results in a low-variance, but biased estimate.

Motivation

- × FMA-PG relies on the knowledge of the true gradient $\nabla_\pi J(\pi)$, which involves either the action-value (Q^π) or the advantage (A^π) functions. This information is rarely available, making FMA-PG impractical in realistic settings.
- Can estimate $\nabla_\pi J(\pi)$ using Monte-Carlo samples obtained via environment interactions [Williams, 1992] and use the estimated gradient.
  - × Resulting estimator has high variance, leading to higher sample-complexity.
- Can estimate $\nabla_\pi J(\pi)$ using a value-based method ("critic"). Results in a low-variance, but biased estimate.
  - × Critic is usually trained by minimizing the TD error, an objective that is potentially decorrelated with the true goal of achieving a high reward with the actor.

Motivation

× FMA-PG relies on the knowledge of the true gradient $\nabla_\pi J(\pi)$, which involves either the action-value (Q^π) or the advantage (A^π) functions. This information is rarely available, making FMA-PG impractical in realistic settings.

- Can estimate $\nabla_\pi J(\pi)$ using Monte-Carlo samples obtained via environment interactions [Williams, 1992] and use the estimated gradient.
    × Resulting estimator has high variance, leading to higher sample-complexity.
- Can estimate $\nabla_\pi J(\pi)$ using a value-based method ("critic"). Results in a low-variance, but biased estimate.
    × Critic is usually trained by minimizing the TD error, an objective that is potentially decorrelated with the true goal of achieving a high reward with the actor.

Lack of theoretically principled objectives to *jointly* train the actor and critic in order to learn good policies.

Outline

- Problem Formulation
- Functional Mirror Ascent for Policy Gradient (FMA-PG) Framework
  - Theoretical Guarantees
  - Instantiating the FMA-PG Framework
- **Decision-aware Actor-Critic**
  - Instantiating the AC Framework
  - Theoretical Guarantees
- Conclusions and Future Work

Decision-aware Actor Critic

Idea: Generalize the lower-bound on $J(\pi)$ to handle inexact gradients.

Decision-aware Actor Critic

Idea: Generalize the lower-bound on $J(\pi)$ to handle inexact gradients.

Generic lower-bound on $J(\pi)$

For any gradient estimator $\hat{g}_t$ at iteration t of FMA-PG, for $c > 0$ and η such that $J + \frac{1}{\eta}\Phi$ is convex in π, if $\Phi^*(y) := \max_\pi[\langle y, \pi\rangle - \Phi(\pi)]$ is the Fenchel conjugate of Φ, we have **inequality (I)**: $J(\pi) - J(\pi_t) \geq$

$$\underbrace{\langle \hat{g}_t, \pi(\theta) - \pi_t \rangle - \left(\frac{1}{\eta} + \frac{1}{c}\right) D_\Phi(\pi(\theta), \pi_t)}_{\text{Surrogate function that can be maximized as before}} - \underbrace{\frac{1}{c} D_{\Phi^*}\left(\nabla\Phi(\pi_t) - c[\nabla J(\pi_t) - \hat{g}_t], \nabla\Phi(\pi_t)\right)}_{\text{Error in } Q^\pi \text{ or } A^\pi \text{ estimation. Can be minimized by training a critic}} .$$

Decision-aware Actor Critic

Idea: Generalize the lower-bound on $J(\pi)$ to handle inexact gradients.

Generic lower-bound on $J(\pi)$

For any gradient estimator $\hat{g}_t$ at iteration t of FMA-PG, for $c > 0$ and η such that $J + \frac{1}{\eta}\Phi$ is convex in π, if $\Phi^*(y) := \max_\pi[\langle y, \pi\rangle - \Phi(\pi)]$ is the Fenchel conjugate of Φ, we have **inequality (I)**: $J(\pi) - J(\pi_t) \geq$

$$\underbrace{\langle \hat{g}_t, \pi(\theta) - \pi_t\rangle - \left(\frac{1}{\eta} + \frac{1}{c}\right) D_\Phi(\pi(\theta), \pi_t) -}_{\text{Surrogate function that can be maximized as before}} \underbrace{\frac{1}{c} D_{\Phi^*}\left(\nabla\Phi(\pi_t) - c[\nabla J(\pi_t) - \hat{g}_t], \nabla\Phi(\pi_t)\right)}_{\text{Error in } Q^\pi \text{ or } A^\pi \text{ estimation. Can be minimized by training a critic}} .$$

- To maximize policy improvement, an algorithm should (i) learn $\hat{g}_t$ to minimize the blue term (critic objective) and (ii) compute $\pi \in \Pi$ that maximizes the green term (actor objective).

Decision-aware Actor Critic

Idea: Generalize the lower-bound on $J(\pi)$ to handle inexact gradients.

Generic lower-bound on $J(\pi)$

For any gradient estimator $\hat{g}_t$ at iteration t of FMA-PG, for $c > 0$ and η such that $J + \frac{1}{\eta}\Phi$ is convex in π, if $\Phi^*(y) := \max_\pi [\langle y, \pi \rangle - \Phi(\pi)]$ is the Fenchel conjugate of Φ, we have **inequality (I)**: $J(\pi) - J(\pi_t) \geq$

$$\underbrace{\langle \hat{g}_t, \pi(\theta) - \pi_t \rangle - \left(\frac{1}{\eta} + \frac{1}{c}\right) D_\Phi(\pi(\theta), \pi_t)}_{\text{Surrogate function that can be maximized as before}} - \underbrace{\frac{1}{c} D_{\Phi^*}\left(\nabla \Phi(\pi_t) - c[\nabla J(\pi_t) - \hat{g}_t], \nabla \Phi(\pi_t)\right)}_{\text{Error in } Q^\pi \text{ or } A^\pi \text{ estimation. Can be minimized by training a critic}}.$$

- To maximize policy improvement, an algorithm should (i) learn $\hat{g}_t$ to minimize the blue term (critic objective) and (ii) compute $\pi \in \Pi$ that maximizes the green term (actor objective).
- c is a parameter relating the critic error to the permissible movement in the actor update.

Decision-aware Actor-Critic – Algorithm

Algorithm 2: Generic actor-critic algorithm

Input: π (choice of functional representation), θ_0 (initial policy parameters), $\omega_{(-1)}$ (initial critic parameters), T (AC iterations), m_a (actor inner-loops), m_c (critic inner-loops), η (functional step-size for actor), c (trade-off parameter), α_a (parametric step-size for actor), α_c (parametric step-size for critic)

Initialization: $\pi_0 = \pi(\theta_0)$

for $t \leftarrow 0$ **to** $T-1$ **do**

  Estimate $\widehat{\nabla}_\pi J(\pi_t)$ and form $\mathcal{L}_t(\omega) := \frac{1}{c}\, D_{\Phi^*}\left(\nabla\Phi(\pi_t) - c\,[\widehat{\nabla}_\pi J(\pi_t) - \hat{g}_t(\omega)], \nabla\Phi(\pi_t)\right)$

  Initialize inner-loop: $\upsilon_0 = \omega_{t-1}$

  for $k \leftarrow 0$ **to** $m_c - 1$ **do**

    $\upsilon_{k+1} = \upsilon_k - \alpha_c\, \nabla_\upsilon\, \mathcal{L}_t(\upsilon_k)$ `/* Critic Updates */`

  $\omega_t = \upsilon_{m_c}$; $\hat{g}_t = \hat{g}_t(\omega_t)$

  Form $\ell_t(\theta) := \langle \hat{g}_t, \pi(\theta) - \pi_t \rangle - \left(\frac{1}{\eta} + \frac{1}{c}\right) D_\Phi(\pi(\theta), \pi_t)$

  Initialize inner-loop: $\nu_0 = \theta_t$

  for $k \leftarrow 0$ **to** $m_a - 1$ **do**

    $\nu_{k+1} = \nu_k + \alpha_a\, \nabla_\nu\, \ell_t(\nu_k)$ `/* Off-policy actor updates */`

  $\theta_{t+1} = \nu_{m_a}$; $\pi_{t+1} = \pi(\theta_{t+1})$

Return $\pi_T = \pi(\theta_T)$

Instantiating Decision-aware Actor-Critic – Direct functional representation

Lower-bound for direct representation

For the direct representation and negative entropy mirror map, $c > 0$, $\eta \leq \frac{(1-\gamma)^3}{2\gamma\,|A|}$,

$$J(\pi) - J(\pi_t) \geq C + \mathbb{E}_{s \sim d^{\pi_t}} \left[\mathbb{E}_{a \sim p^{\pi_t}(\cdot|s)} \left[\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)} \left(\hat{Q}^{\pi_t}(s,a) - \left(\frac{1}{\eta} + \frac{1}{c} \right) \log \left(\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)} \right) \right) \right] \right]$$

$$- \mathbb{E}_{s \sim d^{\pi_t}} \left[\mathbb{E}_{a \sim p^{\pi_t}(\cdot|s)} \left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a) \right] + \frac{1}{c} \log \left(\mathbb{E}_{a \sim p^{\pi_t}(\cdot|s)} \left[\exp \left(-c \left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a) \right] \right) \right] \right) \right]$$

Instantiating Decision-aware Actor-Critic – Direct functional representation

Lower-bound for direct representation

For the direct representation and negative entropy mirror map, $c > 0$, $\eta \leq \frac{(1-\gamma)^3}{2\gamma\,|A|}$,

$$J(\pi) - J(\pi_t) \geq C + \mathbb{E}_{s \sim d^{\pi_t}} \left[\mathbb{E}_{a \sim p^{\pi_t}(\cdot|s)} \left[\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)} \left(\hat{Q}^{\pi_t}(s,a) - \left(\frac{1}{\eta} + \frac{1}{c} \right) \log \left(\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)} \right) \right) \right] \right]$$

$$- \mathbb{E}_{s \sim d^{\pi_t}} \left[\mathbb{E}_{a \sim p^{\pi_t}(\cdot|s)} [Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a)] + \frac{1}{c} \log \left(\mathbb{E}_{a \sim p^{\pi_t}(\cdot|s)} \left[\exp \left(-c \left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a) \right] \right) \right] \right) \right]$$

- Lower-bound holds for any policy or critic parameterization i.e. $p^{\pi}(\cdot|s) = p^{\pi}(\cdot|s,\theta)$, $\hat{Q}^{\pi}(s,a) = Q^{\pi}(s,a|\omega)$, and instantiates the actor and critic objectives at iteration t.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Lower-bound for direct representation

For the direct representation and negative entropy mirror map, $c > 0$, $\eta \leq \frac{(1-\gamma)^3}{2\gamma\,|A|}$,

$$J(\pi) - J(\pi_t) \geq C + \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)}\left(\hat{Q}^{\pi_t}(s,a) - \left(\frac{1}{\eta} + \frac{1}{c}\right)\log\left(\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)}\right)\right)\right]\right]$$
$$- \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a)\right] + \frac{1}{c}\log\left(\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[\exp\left(-c\left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a)\right]\right)\right]\right)\right]$$

- Lower-bound holds for any policy or critic parameterization i.e. $p^{\pi}(\cdot|s) = p^{\pi}(\cdot|s,\theta)$, $\hat{Q}^{\pi}(s,a) = Q^{\pi}(s,a|\omega)$, and instantiates the actor and critic objectives at iteration t.
- The blue term is referred to as the decision-aware critic loss since minimizing it directly improves the lower-bound on $J(\pi)$ and can result in policy improvement.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Lower-bound for direct representation

For the direct representation and negative entropy mirror map, $c > 0$, $\eta \leq \frac{(1-\gamma)^3}{2\gamma\,|A|}$,

$$J(\pi) - J(\pi_t) \geq C + \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)}\left(\hat{Q}^{\pi_t}(s,a) - \left(\frac{1}{\eta}+\frac{1}{c}\right)\log\left(\frac{p^{\pi}(a|s)}{p^{\pi_t}(a|s)}\right)\right)\right]\right]$$
$$- \mathbb{E}_{s\sim d^{\pi_t}}\left[\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a)\right] + \frac{1}{c}\log\left(\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[\exp\left(-c\left[Q^{\pi_t}(s,a) - \hat{Q}^{\pi_t}(s,a)\right]\right)\right]\right)\right]$$

- Lower-bound holds for any policy or critic parameterization i.e. $p^{\pi}(\cdot|s) = p^{\pi}(\cdot|s,\theta)$, $\hat{Q}^{\pi}(s,a) = Q^{\pi}(s,a|\omega)$, and instantiates the actor and critic objectives at iteration t.
- The blue term is referred to as the decision-aware critic loss since minimizing it directly improves the lower-bound on $J(\pi)$ and can result in policy improvement.
- Critic loss is asymmetric and penalizes the under/over-estimation of the Q^{π} function differently. Unlike the standard squared critic loss: $E_{s\sim d^{\pi_t}}\mathbb{E}_{a\sim p^{\pi_t}(\cdot|s)}\left[Q^{\pi_t}(s,a) - Q^{\pi_t}(s,a|\omega)\right]^2$.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Importance of the decision-aware critic loss

Consider a two-armed bandit example with deterministic rewards where arm 1 is optimal and has reward $r_1 = Q_1 = 2$, whereas arm 2 has reward $r_2 = Q_2 = 1$. Using a linear parameterization for the critic, Q function is estimated as: $\hat{Q}_i = x_i \, \omega$. Set $x_1 = -2$ and $x_2 = 1$ and let p_t be the probability of pulling the optimal arm at iteration t.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Importance of the decision-aware critic loss

Consider a two-armed bandit example with deterministic rewards where arm 1 is optimal and has reward $r_1 = Q_1 = 2$, whereas arm 2 has reward $r_2 = Q_2 = 1$. Using a linear parameterization for the critic, Q function is estimated as: $\hat{Q}_i = x_i\,\omega$. Set $x_1 = -2$ and $x_2 = 1$ and let p_t be the probability of pulling the optimal arm at iteration t.

Consider minimizing two alternative objectives to estimate ω:

(1) Squared loss: $\omega_t^{(1)} := \arg\min \mathrm{TD}(\omega) := \arg\min \left\{ \frac{p_t}{2} [\hat{Q}_1(\omega) - Q_1]^2 + \frac{1-p_t}{2} [\hat{Q}_2(\omega) - Q_2]^2 \right\}$.

(2) Decision-aware critic loss: $\omega_t^{(2)} := \arg\min \mathcal{L}_t(\omega) := p_t\,[Q_1 - \hat{Q}_1(\omega)] + (1-p_t)\,[Q_2 - \hat{Q}_2(\omega)] + \frac{1}{c} \log\left(p_t \exp\left(-c\,[Q_1 - \hat{Q}_1(\omega)] + (1-p_t) \exp\left(-c\,[Q_2 - \hat{Q}_2(\omega)]\right)\right)\right]$.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Importance of the decision-aware critic loss

Consider a two-armed bandit example with deterministic rewards where arm 1 is optimal and has reward $r_1 = Q_1 = 2$, whereas arm 2 has reward $r_2 = Q_2 = 1$. Using a linear parameterization for the critic, Q function is estimated as: $\hat{Q}_i = x_i\,\omega$. Set $x_1 = -2$ and $x_2 = 1$ and let p_t be the probability of pulling the optimal arm at iteration t.

Consider minimizing two alternative objectives to estimate ω:

(1) Squared loss: $\omega_t^{(1)} := \arg\min \mathrm{TD}(\omega) := \arg\min \left\{ \frac{p_t}{2}\,[\hat{Q}_1(\omega) - Q_1]^2 + \frac{1-p_t}{2}\,[\hat{Q}_2(\omega) - Q_2]^2 \right\}$.

(2) Decision-aware critic loss: $\omega_t^{(2)} := \arg\min \mathcal{L}_t(\omega) := p_t\,[Q_1 - \hat{Q}_1(\omega)] + (1 - p_t)\,[Q_2 - \hat{Q}_2(\omega)] + \frac{1}{c}\log\left(p_t \exp\left(-c\,[Q_1 - \hat{Q}_1(\omega)] + (1 - p_t)\exp\left(-c\,[Q_2 - \hat{Q}_2(\omega)]\right)\right)\right]$.

Using the tabular parameterization for the actor, the policy update at iteration t is given by:

$p_{t+1} = \frac{p_t\ \exp(\eta \hat{Q}_1)}{p_t\ \exp(\eta \hat{Q}_1) + (1 - p_t)\ \exp(\eta \hat{Q}_2)}$.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Importance of the decision-aware critic loss

Consider a two-armed bandit example with deterministic rewards where arm 1 is optimal and has reward $r_1 = Q_1 = 2$, whereas arm 2 has reward $r_2 = Q_2 = 1$. Using a linear parameterization for the critic, Q function is estimated as: $\hat{Q}_i = x_i\,\omega$. Set $x_1 = -2$ and $x_2 = 1$ and let p_t be the probability of pulling the optimal arm at iteration t.

Consider minimizing two alternative objectives to estimate ω:

(1) Squared loss: $\omega_t^{(1)} := \arg\min \mathrm{TD}(\omega) := \arg\min \left\{ \frac{p_t}{2} [\hat{Q}_1(\omega) - Q_1]^2 + \frac{1-p_t}{2} [\hat{Q}_2(\omega) - Q_2]^2 \right\}$.

(2) Decision-aware critic loss: $\omega_t^{(2)} := \arg\min \mathcal{L}_t(\omega) := p_t\,[Q_1 - \hat{Q}_1(\omega)] + (1-p_t)\,[Q_2 - \hat{Q}_2(\omega)] + \frac{1}{c} \log\left(p_t \exp\left(-c\,[Q_1 - \hat{Q}_1(\omega)] + (1-p_t) \exp\left(-c\,[Q_2 - \hat{Q}_2(\omega)]\right)\right)\right]$.

Using the tabular parameterization for the actor, the policy update at iteration t is given by:

$p_{t+1} = \frac{p_t \exp(\eta \hat{Q}_1)}{p_t \exp(\eta \hat{Q}_1) + (1-p_t) \exp(\eta \hat{Q}_2)}$.

For any η, for $p_0 < \frac{2}{5}$, minimizing the squared loss results in convergence to the sub-optimal action, while minimizing the decision-aware loss (for any $c, p_0 > 0$) results in convergence to the optimal action.

Instantiating Decision-aware Actor-Critic – Direct functional representation

Importance of the decision-aware critic loss

Consider a two-armed bandit example with deterministic rewards where arm 1 is optimal and has reward $r_1 = Q_1 = 2$, whereas arm 2 has reward $r_2 = Q_2 = 1$. Using a linear parameterization for the critic, Q function is estimated as: $\hat{Q}_i = x_i \, \omega$. Set $x_1 = -2$ and $x_2 = 1$ and let p_t be the probability of pulling the optimal arm at iteration t.

Consider minimizing two alternative objectives to estimate ω:

(1) Squared loss: $\omega_t^{(1)} := \arg\min \mathrm{TD}(\omega) := \arg\min \left\{ \frac{p_t}{2} [\hat{Q}_1(\omega) - Q_1]^2 + \frac{1-p_t}{2} [\hat{Q}_2(\omega) - Q_2]^2 \right\}$.

(2) Decision-aware critic loss: $\omega_t^{(2)} := \arg\min \mathcal{L}_t(\omega) := p_t \, [Q_1 - \hat{Q}_1(\omega)] + (1 - p_t) \, [Q_2 - \hat{Q}_2(\omega)] + \frac{1}{c} \log \left(p_t \exp\left(-c \, [Q_1 - \hat{Q}_1(\omega)] + (1 - p_t) \exp\left(-c \, [Q_2 - \hat{Q}_2(\omega)] \right) \right) \right) \Big]$.

Using the tabular parameterization for the actor, the policy update at iteration t is given by:

$p_{t+1} = \frac{p_t \ \exp(\eta \hat{Q}_1)}{p_t \ \exp(\eta \hat{Q}_1) + (1 - p_t) \ \exp(\eta \hat{Q}_2)}$.

For any η, for $p_0 < \frac{2}{5}$, minimizing the squared loss results in convergence to the sub-optimal action, while minimizing the decision-aware loss (for any $c, p_0 > 0$) results in convergence to the optimal action.

- Similar results for the softmax functional representation.

Theoretical Guarantees

Monotonic policy improvement for AC algorithm

For any policy representation and any policy or critic parameterization, there exists a (θ, c) pair that makes the RHS of **inequality (I)** strictly positive, and hence guarantees monotonic policy improvement $(J(\pi_{t+1}) > J(\pi_t))$, if and only if the critic error satisfies a technical condition that depends on the policy parameterization and the mirror map.

Theoretical Guarantees

Monotonic policy improvement for AC algorithm

For any policy representation and any policy or critic parameterization, there exists a (θ, c) pair that makes the RHS of **inequality (I)** strictly positive, and hence guarantees monotonic policy improvement $(J(\pi_{t+1}) > J(\pi_t))$, if and only if the critic error satisfies a technical condition that depends on the policy parameterization and the mirror map.

Special case: For the tabular policy parameterization with the Euclidean mirror map, this condition is equivalent to: $\|\nabla J(\pi_t) - \hat{g}_t\|_2^2 < \|\hat{g}_t\|_2^2$.

Theoretical Guarantees

Monotonic policy improvement for AC algorithm

For any policy representation and any policy or critic parameterization, there exists a (θ, c) pair that makes the RHS of **inequality (I)** strictly positive, and hence guarantees monotonic policy improvement ($J(\pi_{t+1}) > J(\pi_t)$), if and only if the critic error satisfies a technical condition that depends on the policy parameterization and the mirror map.

Special case: For the tabular policy parameterization with the Euclidean mirror map, this condition is equivalent to: $\|\nabla J(\pi_t) - \hat{g}_t\|_2^2 < \|\hat{g}_t\|_2^2$.

Convergence of AC algorithm

For any critic error, policy representation and mirror map Φ such that (i) $J + \frac{1}{\eta}\Phi$ is convex in π, any policy parameterization such that (ii) $\ell_t(\theta)$ is smooth w.r.t θ and satisfies a gradient domination condition, for $c > 0$, the AC algorithm converges to a neighbourhood of a stationary point at an $O(1/T)$ rate. The neighbourhood depends on the critic error and the number of off-policy actor updates.

Outline

- Problem Formulation
- Functional Mirror Ascent for Policy Gradient (FMA-PG) Framework
  - Theoretical Guarantees
  - Instantiating the FMA-PG Framework
- Decision-aware Actor-Critic
  - Instantiating the AC Framework
  - Theoretical Guarantees
- **Conclusions and Future Work**

Conclusions and Future Work

✓ Generalized FMA-PG to design a generic decision-aware actor-critic framework where the actor and critic are trained cooperatively to optimize a joint objective.

Conclusions and Future Work

- ✓ Generalized FMA-PG to design a generic decision-aware actor-critic framework where the actor and critic are trained cooperatively to optimize a joint objective.
- ✓ Tabular RL experiments with a linear parameterization for the actor/critic demonstrate that being decision-aware is important when the critic is not as expressive.

Conclusions and Future Work

- ✓ Generalized FMA-PG to design a generic decision-aware actor-critic framework where the actor and critic are trained cooperatively to optimize a joint objective.
- ✓ Tabular RL experiments with a linear parameterization for the actor/critic demonstrate that being decision-aware is important when the critic is not as expressive.

- Prove convergence rates to the (neighbourhood) of the optimal policy for the AC algorithm.
- Benchmark the AC framework for complex deep RL environments.

Questions?

Papers: https://arxiv.org/abs/2108.05828, https://arxiv.org/abs/2305.15249

Contact: vaswani.sharan@gmail.com, nicolas.le.roux@gmail.com

References i

Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. Optimality and approximation with policy gradient methods in Markov decision processes. In *Conference on Learning Theory (COLT)*, pages 64–66, 2020.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized Markov decision processes. In *International Conference on Machine Learning*, pages 2160–2169. PMLR, 2019.

Chloe Ching-Yun Hsu, Celestine Mendler-Dünner, and Moritz Hardt. Revisiting design choices in proximal policy optimization. *arXiv preprint arXiv:2009.10897*, 2020.

Sham Kakade. A natural policy gradient. In *NIPS*, volume 14, pages 1531–1538, 2001.

Jincheng Mei, Chenjun Xiao, Csaba Szepesvari, and Dale Schuurmans. On the global convergence rates of softmax policy gradient methods. In *International Conference on Machine Learning*, pages 6820–6829. PMLR, 2020.

Lior Shani, Yonathan Efroni, and Shie Mannor. Adaptive trust region policy optimization: Global convergence and faster rates for regularized mdps. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 5668–5675, 2020.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. *arXiv preprint arXiv:2005.09814*, 2020.

References ii

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

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Lin Xiao. On the convergence rates of policy gradient methods. *Journal of Machine Learning Research*, 23(282): 1–36, 2022.