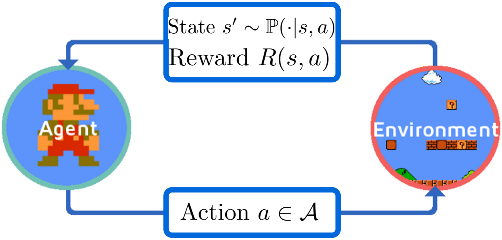
Deep Reinforcement Learning with Smooth Policy

Presenter: Yan Li

Georgia Institute of Technology

Reinforcement Learning

Markov Decision Process: $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, \gamma)$



Goal: maximize expected (discounted) reward

$$\max_{\pi} V(\pi) = \mathbb{E}_{s_0,a_0,\dots}\left[\sum_{t\geq 0}\gamma^t r(s_t,a_t)\right],$$

$$s_0 \sim p_0, a_t \sim \pi(s_t), s_{t+1} \sim \mathbb{P}(s_{t+1}|s_t,a_t).$$

Function Approximation in RL

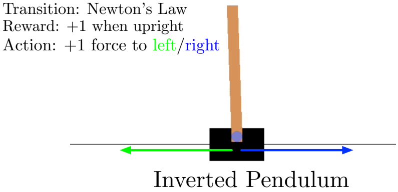
Policy gradient algorithms:

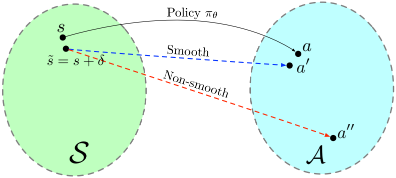
- Parameterizes policy π with function approximation (DNNs).
- Estimates the gradient of the $V(\pi)$ through trajectory samples: $\widehat{g}_t$, update: $\pi_{t+1} = \pi_t + \eta \widehat{g}_t$.
- Large variance (coming from environment noise and large search space) leads training instability.
- Improved variants (actor-critic): TRPO (Schulman et al. 2015), DDPG (Lillicrap et al. 2015). Controls local search space.

Still... the global search space of DNNs is prohibitively large, current RL algorithms are sample inefficient.

RL with smooth environments

Smooth reward function (w.r.t. state), smooth transition (w.r.t. state) ⇒ Exists an optimal policy that is smooth (w.r.t. state).





Smoothness-inducing regularization

Promoting smoothness in policy: adversarially defined regularization

$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \mathop{\mathbb{E}}_{s \sim \rho^{\pi_\theta}} \max_{\widetilde{s} \in \mathbb{B}_d(s,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(\widetilde{s})).$$

$\mathcal{D}(\cdot,\cdot)$ appropriate metric, $\mathbb{B}_d(s,\epsilon) = \{s', \|s - s'\| \leq \epsilon\}$, ρ^{π_θ} the stationary state distribution induced by π_θ.

Choices of metric $\mathcal{D}$

- Stochastic policy (Jeffrey's divergence):
  $\mathcal{D}_{\mathrm{J}}(\pi(s), \pi(\widetilde{s})) = \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(\pi(s)\|\pi(\widetilde{s})) + \frac{1}{2}\mathcal{D}_{\mathrm{KL}}(\pi(\widetilde{s})\|\pi(s)).$
- Deterministic policy (squared difference):
  $\mathcal{D}(\mu(s), \mu(\widetilde{s})) = \|\mu(s) - \mu(\widetilde{s})\|_2^2.$

Smoothness-inducing regularization

Promoting smoothness in policy: adversarially defined regularization

$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \mathop{\mathbb{E}}_{s\sim\rho^{\pi_\theta}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(\widetilde{s})).$$

$\mathcal{D}(\cdot,\cdot)$ appropriate metric, $\mathbb{B}_d(s,\epsilon) = \{s', \|s - s'\|_\infty \leq \epsilon\}$, ρ^{π_θ} the stationary state distribution induced by π_θ.

- The inner max inspired by local-shift sensitivity in robust statistics (Hampel, 1974).
- Measures local smoothness of policy under metric $\mathcal{D}$.
- Take expectation w.r.t. state-visitation distribution: smoothness along trajectory.
- Wide applicability: can be applied to both on-policy and off-policy algorithms.

Beyond directly smoothing policy

Actor-critic framework:

- Actor: policy network.
- Critic: network to approximate Q-function (expected future reward given initial state-action pair (s, a)).

Idea: use critic to help update the policy (actor), reduce variance.

Smoothness inducing regularization for critic:

Smooth critic (Q-function) can also be used to induce a smooth policy.

$$\mathcal{R}_s^Q(\phi) = \mathbb{E}_{\substack{s \sim \rho^\beta, \\ a \sim \beta}} \max_{\widetilde{s} \in \mathbb{B}_d(s, \epsilon)} (Q_\phi(s, a) - Q_\phi(\widetilde{s}, a))^2.$$

Application: TRPO (stochastic policy)

Smooth policy (TRPO–SR):

$$\theta_{k+1} = \arg\min_{\theta} - \underbrace{\mathbb{E}_{\substack{s\sim\rho^{\pi_{\theta_k}},\\ a\sim\pi_{\theta_k}}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)} A^{\pi_{\theta_k}}(s,a)\right]}_{\text{linearization of value function}}$$

$$+ \lambda_{\mathrm{s}} \underbrace{\mathbb{E}_{s\sim\rho^{\pi_{\theta_k}}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} \mathcal{D}_{\mathrm{J}}(\pi_\theta(\cdot|s) \mid\mid \pi_\theta(\cdot|\widetilde{s}))}_{\text{adversarial regularization with Jefferey's divergence}},$$

$$\text{s.t.} \quad \underbrace{\mathbb{E}_{s\sim\rho^{\pi_{\theta_k}}} \left[\mathcal{D}_{\mathrm{KL}}(\pi_{\theta_k}(\cdot|s) \| \pi_\theta(\cdot|s_{)}\right] \le \delta}_{\text{trust-region}}.$$

Solving the min-max problem:

Projected gradient ascent for the inner max, gradient descent for the outer min.

Application: DDPG (deterministic policy)

Smooth critic (DDPG-SR-C):

$$\phi_{t+1} = \arg\min_{\phi} \underbrace{\sum_{i \in B} \left(y_t^i - Q_\phi(s_t^i, a_t^i)\right)^2}_{\text{approximate Bellman error}}$$

$$+ \lambda_{\mathrm{s}} \underbrace{\sum_{i \in B} \max_{\widetilde{s}_t^i \sim \mathbb{B}_d(s_t^i, \epsilon)} \left(Q_\phi(s_t^i, a_t^i) - Q_\phi(\widetilde{s}_t^i, a_t^i)\right)^2}_{\text{smoothness inducing regularization for Q-function}},$$

with $y_t^i = r_t^i + \gamma Q_{\phi_t'}(s_{t+1}^i, \mu_{\theta_t'}(s_{t+1}^i)), \forall i \in B$, where B denotes the mini-batch sampled from the replay buffer.

Application: DDPG (deterministic policy)

Smooth actor (DDPG-SR-A):

$$\mu_{\theta_{t+1}} = \mu_{\theta_t} - \eta \mathop{\mathbb{E}}_{s\sim\rho^\beta} \Big[- \underbrace{\nabla_a Q_\phi(s,a)\big|_{a=\mu_{\theta_t}(s)} \nabla_\theta \mu_{\theta_t}(s)}_{\text{deterministic policy gradient}}$$

$$+\lambda_{\mathrm{s}} \underbrace{\nabla_\theta \left\| \mu_{\theta_t}(s) - \mu_{\theta_t}(\widetilde{s}) \right\|_2^2}_{\text{gradient of smoothness inducing regularization}} \Big],$$

with $\widetilde{s} = \mathop{\arg\max}_{\widetilde{s}\sim\mathbb{B}_d(s,\epsilon)} \left\| \mu_{\theta_t}(s) - \mu_{\theta_t}(\widetilde{s}) \right\|_2^2$ for $s \sim \rho^\beta$.

Robustness against measurement error

Measurement error is prevalent in practice: state information is obtained from (noisy) sensor data (e.g., robotic motion control).

Previous approach: POMDP (Astrom, 1965), requires i.i.d. noise with known distribution.

Our regularization improves robustness: Smooth environments requires similar actions for similar states, our regularization

$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \mathop{\mathbb{E}}_{s \sim \rho^{\pi_\theta}} \max_{\widetilde{s} \in \mathbb{B}_d(s,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(\widetilde{s}))$$

naturally induces robustness and avoids overfitting to noise.

Robust against random, and even adversarial perturbation to the state.

Extension to distributionally robust optimization

Perturbing state-visitation distribution:

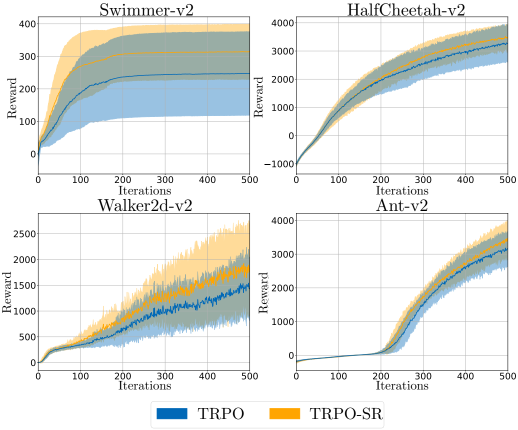
$$\mathcal{R}_{\mathrm{s}}^{\pi}(\theta) = \max_{\mathcal{F}(\mathbb{P},\mathbb{P}') \leq \epsilon} \mathbb{E}_{s \sim \mathbb{P}, s' \sim \mathbb{P}'} \mathcal{D}\left(\pi_\theta(s), \pi_\theta(s')\right),$$

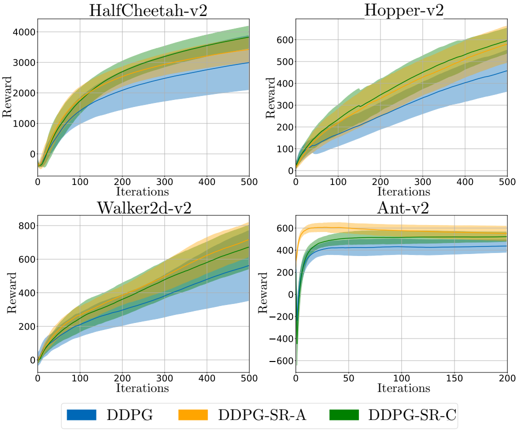
where $\mathcal{F}(\cdot,\cdot)$ denotes some discrepancy measure between a pair of visitation probability distributions (e.g., Wasserstein distance, f-divergence). Inner problem can be solved via duality (Gao and Kleywegt, 2016).

Experiments

Environments: OpenAI gym (Brockman et al., 2016): Swimmer, HalfCheetah, Walker, Ant, Hopper.

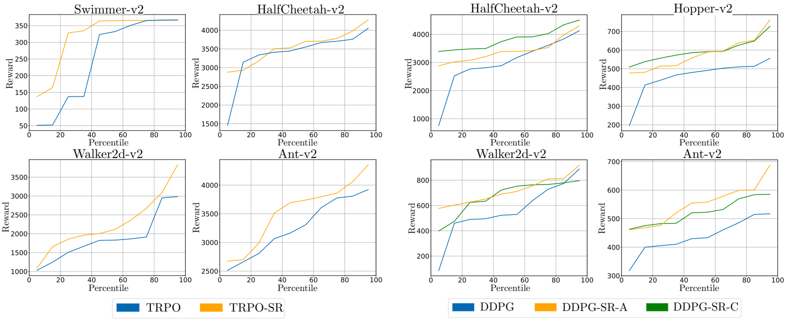
Learning Curves:





Smoothness regularization promotes faster learning compared to strong implementation of baseline.

Quantile Plots: Repeated 10 runs with random initializations, plot the quantiles of the final cumulative rewards.

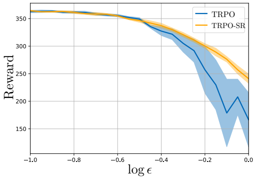


Smoothness regularization improves both worst cast and best case performance compared to baseline.

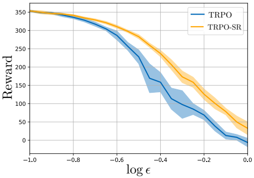
Evaluation of robustness:

Random error: $\delta \sim \mathbb{B}_d(0, \epsilon) = \{\delta : \|\delta\|_\infty \leq \epsilon\}$.

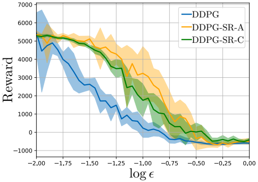
Adversarial error: $\widetilde{\delta} = \arg\max_{\delta \in \mathbb{B}_d(0,\epsilon)} \mathcal{D}(\pi_\theta(s), \pi_\theta(s + \delta))$.



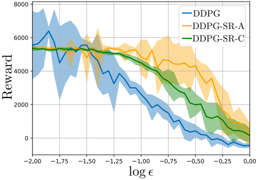
(a) Swimmer - Random Disturbed Rollout



(b) Swimmer - Adversarially Disturbed Rollout



(a) HalfCheetah - Adversarially Disturbed Rollout



(b) HalfCheetah - Random Disturbed Rollout

Improved robustness against random and adversarial measurement error to the states.

Doubly Robust Off-policy RL:

Challenges in off-policy training. In off-policy setting, policy evaluation often suffers high variance in reward and transition. This leads to unstable update for actor-critic.

Doubly robust estimation (Islam et al. 2019). Reduce the variance in critic estimation.

$$Q_\phi^{DR} = \widehat{Q}(s, \pi_\theta(s)) + \left[r(s,a) + \gamma Q_\phi^{DR}(s', \pi_\theta(s')) - \widehat{V}(s) \right]$$

where the $\widehat{Q}$ and $\widehat{V}$ are learned separately, together with an approximated reward function $\widehat{R}$.

$$\widehat{R} \leftarrow \min_{\psi} \mathbb{E}_{s,a,r \sim \text{Buffer}} (R_\phi(s,a) - R(s,a))^2$$

$$\widehat{Q} \leftarrow \min_{\phi} \mathbb{E}_{s,a,r,s' \sim \text{Buffer}} \left[(\widehat{R}(s,a) + \gamma \widehat{Q}(s', \pi_\theta(s'))) - \widehat{Q}(s,a) \right]^2$$

Doubly robust RL with smooth environments

We can further incorporate smoothness to further reduce variance for smooth environments.

Smooth reward and critic

$$\widehat{R} \leftarrow \min_{\psi} \mathbb{E}_{s,a,r\sim\text{Buffer}}(R_\psi(s,a) - R(s,a))^2 + \lambda_{\text{s}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} (R_\psi(s,a) - R_\psi(\widetilde{s},a))^2$$

$$\widehat{Q} \leftarrow \min_{\phi} \mathbb{E}_{s,a,r,s'\sim\text{Buffer}}\left[(\widehat{R}(s,a) + \gamma\widehat{Q}(s',\pi_\theta(s'))) - \widehat{Q}(s,a)\right]^2 + \lambda_{\text{s}} \max_{\widetilde{s}\in\mathbb{B}_d(s,\epsilon)} (Q_\phi(s,a) - Q_\phi(\widetilde{s},a))^2$$

Conclusion

Take-home message:

- Smooth environment advocates smooth policy.
- Smooth policy for smooth environments leads to robustness and better sample complexity.

Thank You!