## Deterministic Policy Gradient Algorithms

### Reminders

- Stochastic policy, usually denoted as $\pi(s, a)$ or $\mu(a|s)$, gives us the joint distribution of actions and policies, while the deterministic policy $\mu(s)$ gives us the actual action
- Stochastic policy natural for explorations

### Overview

- Continuous actions
- Deterministic policy gradient is the expected gradient of the action-value function
- Stochastic policy gradient integrates over both actions and states $\implies$ needs more samples to estimate
  - Necessary to explare full state and action space (unless environment super noisy)
- *compatible* function approximation ensures unbiasedness of the gradient
- Stochastic Policy Gradient Theorem [1]:

$$\nabla_\theta J(\pi_\theta) = \mathbb{E}_{s\sim d^\pi, a\sim\pi_\theta}[\nabla_\theta \log \pi_\theta(a|s) Q^\pi(s, a)]$$

  - Problem with this algorithms is estimating action-value function $Q^\pi(s, a)$
- Stochastic actor critic:
  - Actor adjusts parameters $\theta$ of the stochastic policy
  - Critic estimates the action-value function $Q^\omega(s, a) \approx Q^\pi(s, a)$
  - Critic is in general biased, unless certain conditions are satisfied (which are usually relaxed)
- Importance sampling in the stochastic case used in stochastic off-policy for both actor and critic, while in deterministic it is discarded for the actor (no integral over actions) and for the critic (e.g. by using Q-learning)

### Key ingredients

- Proves the existence of the deterministic policy gradient
- Deterministic policy is the limit of the stochastic policy as $\sigma \to 0$
- Deterministic policy gradient theorem:

$$\nabla_\theta J(\mu_\theta) = \int_{\mathcal{S}} d^\mu(s) \nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s, \mu_\theta(s)) \mathrm{d}s = \mathbb{E}_{d^\mu}[\nabla_\theta \mu_\theta(s) \nabla_a Q^\mu(s, \mu_\theta(s))]$$

- As in the stochastic case, the state-visitation distribution gradient does no go into the policy gradient update
- **How to calculate $\nabla_a Q^\mu(s, \mu_\theta(s))$ ?**
  - $Q^\mu(s, \mu_\theta(s)$ an be defined in the linear form from the compatibility function

### Comments

- IMO not that well written paper, dry and lacking important insights needed to follow the presented concepts
- After going through [1], it is a lot easier to understand
- Algorithm several orders of magnitude faster than the stochastic counterpart for higher dimensional action spaces

**Off topic** Can we make any process MDP? Use the whole previous history to represent the state?

[1] *Policy Gradient Methods for Reinforcement Learning with Function Approximation*, R. Sutton, et. al.