Introduction to Gradient-based Direct Policy Search

Finite-Time Control

Adrien Bolland

10/03/2021

Finite-time MDP

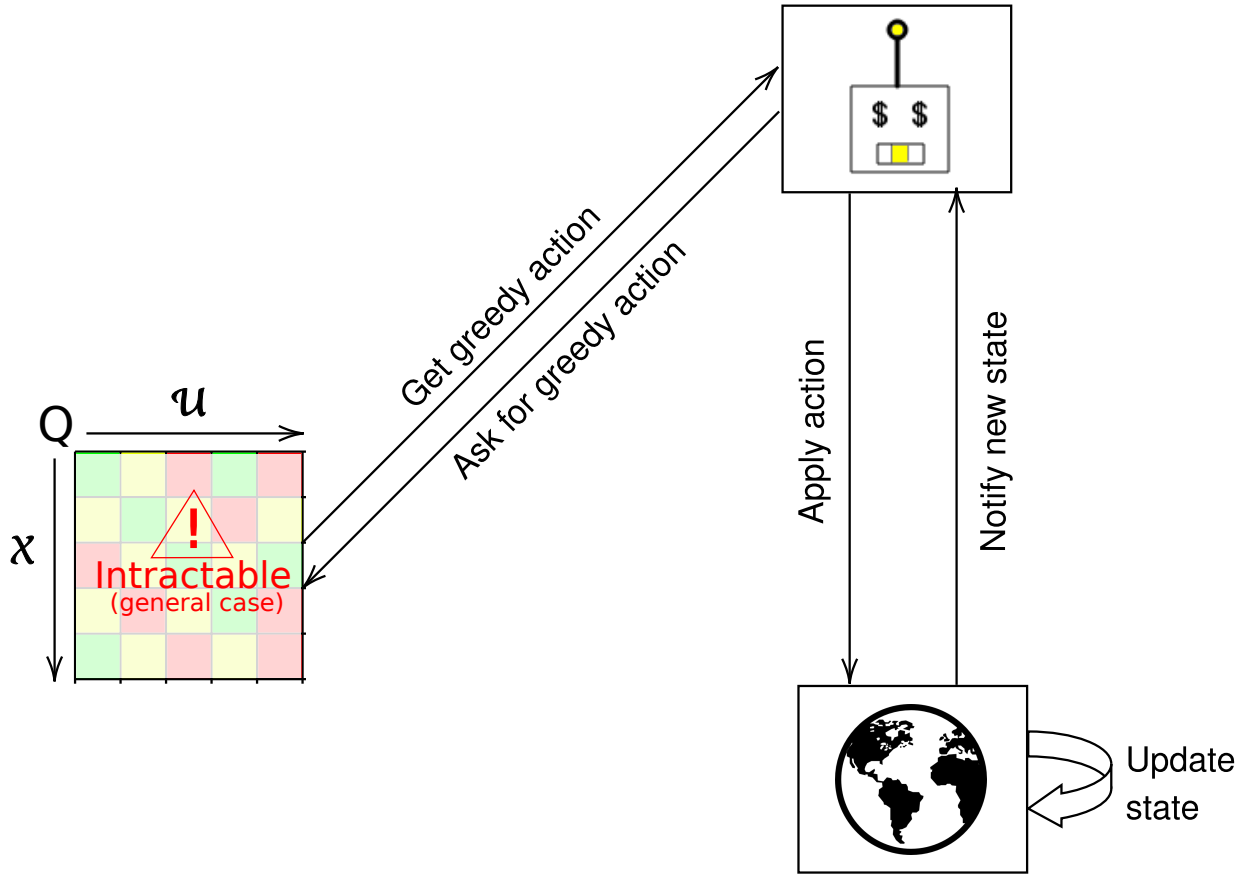
- State space $\mathcal{S}$
- Action space $\mathcal{A}$
- Probability distribution over the initial states $P_0(s_0)$
- Transition probability distribution $p(s_{t+1}|s_t, a_t)$
- Reward probability distribution $\rho(r_t|s_t, a_t)$
- Finite horizon T

Finite-time MDP - Optimal policies

- A policy is a sequence of T distributions $\pi_t(a_t|s_t)$
- Time-dependent policies $\pi_t \in \Pi$
- We look for policies maximizing the expected sum of rewards

$$\pi_t^* \in \underset{\pi_t \in \Pi}{\text{argmax}} \underset{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_t(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}}{\mathbb{E}} \{ \sum_{t=0}^{T-1} r_t \} \tag{1}$$

Optimal decision making with Q-functions (cont'd)



How to compute the maximal Q-value?

Direct policy search

- Direct policy search consists in optimizing the policy directly
- Let $\pi_{t,\theta} \in \Pi$ be a policy parametrized by $\theta \in \Theta \subsetneq \mathbb{R}^{d_\Theta}$
- We look for policies maximizing the expected sum of reward in function of θ

$$\theta^* \in \underset{\theta \in \Theta}{\operatorname{argmax}}\, J(\theta) \tag{2}$$

$$J(\theta) = \underset{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_{t,\theta}(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}}{\mathbb{E}} \left\{ \sum_{t=0}^{T-1} r_t \right\}. \tag{3}$$

- Policy gradient consists in computing the gradient and doing gradient ascent on J

Direct policy search

$$\nabla_\theta J(\theta) = \nabla_\theta \mathop{\mathbb{E}}_{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_{t,\theta}(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}} \{\sum_{t=0}^{T-1} r_t\} \tag{4}$$

- *How to compute the gradient?*
- Gradient of a Monte-Carlo estimates of J will not work... Why?

Likelihood ratio policy gradient

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_{t,\theta}(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}} \{ \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \} \tag{5}$$

It can be estimated from Monte-Carlo samples of $\pi_{t,\theta}$ in the environment!

Likelihood ratio policy gradient - proof

- Let $\tau = (s_0, a_0, r_0, \ldots, s_T)$ be a trajectory
- Let $P_\theta(\tau)$ be the likelihood of the trajectory τ

$$P_\theta(\tau) = P_0(s_0) \prod_{t=0}^{T-1} \pi_{t,\theta}(a_t|s_t) p(s_{t+1}|s_t, a_t) \rho(r_t|s_t, a_t) \tag{6}$$

- Let $R(\tau)$ bet the cumulative reward collected in the trajectory τ:

$$R(\tau) = \sum_{t=0}^{T-1} r_t \tag{7}$$

Likelihood ratio policy gradient - proof

$$\nabla_\theta J(\theta) = \nabla_\theta \mathop{\mathbb{E}}_{\tau \sim P_\theta(\cdot)} \{R(\tau)\} \tag{8}$$

$$= \nabla_\theta \int P_\theta(\tau) R(\tau) d\tau \tag{9}$$

$$= \int (\nabla_\theta P_\theta(\tau)) R(\tau) d\tau \qquad \nabla f(x) = f(x) \nabla \log f(x) \tag{10}$$

$$= \int P_\theta(\tau) (\nabla_\theta \log P_\theta(\tau)) R(\tau) d\tau \tag{11}$$

$$= \mathop{\mathbb{E}}_{\tau \sim P_\theta(\cdot)} \{(\nabla_\theta \log P_\theta(\tau)) R(\tau)\} \tag{12}$$

Can we simplify $\nabla_\theta \log P_\theta(\tau)$... *Yes*

Likelihood ratio policy gradient - proof

$$\nabla_\theta \log P_\theta(\tau) = \nabla_\theta \log \left(P_0(s_0) \prod_{t=0}^{T-1} \pi_{t,\theta}(a_t|s_t) p(s_{t+1}|s_t, a_t) \rho(r_t|s_t, a_t) \right) \tag{13}$$

$$= \nabla_\theta \log P_0(s_0) + \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t)$$

$$+ \sum_{t=0}^{T-1} \nabla_\theta \log p(s_{t+1}|s_t, a_t) + \sum_{t=0}^{T-1} \nabla_\theta \log \rho(r_t|s_t, a_t) \tag{14}$$

$$= \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \, . \tag{15}$$

Likelihood ratio policy gradient

- Likelihood ratio policy gradient

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_{t,\theta}(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}} \{ \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \} \tag{16}$$

- Monte-Carlo approximation over n i.i.d. trajectories

$$\hat{\nabla}_\theta J(\theta) = \Big\langle \big(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \big) \big(\sum_{t=0}^{T-1} r_t \big) \Big\rangle_n \tag{17}$$

Policy gradient algorithm

Algorithm 1 On-policy stochastic gradient ascent

Inputs Environment
Inputs Number of iterations N
Inputs Learning rate α
Inputs Parametrized policy $\pi_{t,\theta}$
for all $n = 1, \ldots, N$ **do**
    Sample n trajectories
    $\theta \leftarrow \theta + \alpha \cdot \hat{\nabla}_\theta J(\theta)$
end for
return $\pi_{t,\theta}$

Likelihood ratio policy gradient - Limitations

VARIANCE OF THE ESTIMATE $\hat{\nabla}_\theta J(\theta)$

Likelihood ratio policy gradient - Baseline

- Subtracting a baseline from the cumulative reward can decrease the variance and is unbiased

$$\hat{\nabla}_\theta J(\theta) = \left\langle \left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \right) \left(\left(\sum_{t=0}^{T-1} r_t \right) - b \right) \right\rangle_n \tag{18}$$

- The optimal baseline can be derived

$$b^* = \frac{\mathop{\mathbb{E}}_{\tau \sim P_\theta(\cdot)} \left\{ (\nabla_\theta \log P_\theta(\tau))^2 R(\tau) \right\}}{\mathop{\mathbb{E}}_{\tau \sim P_\theta(\cdot)} \left\{ (\nabla_\theta \log P_\theta(\tau))^2 \right\}} \, . \tag{19}$$

- In practice, it is common to choose the mean cumulative rewards
- Using a baseline is equivalent to scaling the rewards

Likelihood ratio policy gradient - Baseline

- The proof behind the idea

$$\mathop{\mathbb{V}}_{\tau\sim P_\theta(\cdot)}\{\hat{\nabla}_\theta J(\theta)\} = \mathop{\mathbb{E}}_{\tau\sim P_\theta(\cdot)}\{(\hat{\nabla}_\theta J(\theta))^2\} - \mathop{\mathbb{E}}_{\tau\sim P_\theta(\cdot)}\{\hat{\nabla}_\theta J(\theta)\}^2 \tag{20}$$

$$= \mathop{\mathbb{E}}_{\tau\sim P_\theta(\cdot)}\left\{\left(\nabla_\theta \log P_\theta(\tau)\big(R(\tau) - b\big)\right)^2\right\} \tag{21}$$

$$- \mathop{\mathbb{E}}_{\tau\sim P_\theta(\cdot)}\left\{\nabla_\theta \log P_\theta(\tau)\big(R(\tau) \text{ - } \cancel{b}\big)\right\}^2 \tag{22}$$

- We can then compute the derivative w.r.t. b

Likelihood ratio policy gradient - Causality

- Likelihood ratio policy gradients weight the gradients of the log-likelihood by the cumulative rewards
- The actions resulting in high cumulative rewards are enforced
- But why do we estimate the quality of an action with the past rewards...
- The effect of an action is limited to the future rewards

$$\hat{\nabla}_\theta J(\theta) = \left\langle \sum_{t=0}^{T-1} \left(\nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \left(\left(\sum_{t'=t}^{T-1} r_{t'} \right) - b_t \right) \right) \right\rangle_n \tag{23}$$

Likelihood ratio policy gradient - Causality

- The proof behind the idea

$$\left(\sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t)\right)\left(\sum_{t=0}^{T-1} r_t\right) \tag{24}$$

$$= \underbrace{\sum_{t=1}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t)\left(\sum_{t'=0}^{t-1} r_{t'}\right)}_{\text{Zero on expectation but adds variance}} + \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t)\left(\sum_{t'=t}^{T-1} r_{t'}\right) \tag{25}$$

Likelihood ratio policy gradient - Limitations

- Likelihood ratio policy gradient is on-policy

- It is very sample inefficient... Trajectories are used to estimate ONCE the gradient

Is it possible to compute several gradient ascent steps with the same trajectories?

Likelihood ratio policy gradient via importance sampling

- Let θ' be the parameters of the policy from which we sampled trajectories
- Let θ be the parameters of the current policy
- Off-policy likelihood ratio policy gradient

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_{t,\theta'}(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}} \left\{ \prod_{t=0}^{T-1} \frac{\pi_{t,\theta}(a_t|s_t)}{\pi_{t,\theta'}(a_t|s_t)} \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \right\} \quad (26)$$

Likelihood ratio policy gradient via importance sampling - Proof

$$\nabla_\theta J(\theta) = \int P_\theta(\tau) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \; d\tau \tag{27}$$

$$= \int \frac{P_{\theta'}(\tau)}{P_{\theta'}(\tau)} P_\theta(\tau) \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \; d\tau \tag{28}$$

$$= \mathop{\mathbb{E}}_{\tau \sim P_{\theta'}(\cdot)} \left\{ \frac{P_\theta(\tau)}{P_{\theta'}(\tau)} \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \right\} \tag{29}$$

Can we simplify $P_\theta(\tau)/P_{\theta'}(\tau)$... Yes

Likelihood ratio policy gradient via importance sampling - Proof

$$\frac{P_\theta(\tau)}{P_{\theta'}(\tau)} = \frac{P_0(s_0)\prod_{t=0}^{T-1}\pi_{t,\theta}(a_t|s_t)p(s_{t+1}|s_t,a_t)\rho(r_t|s_t,a_t)}{P_0(s_0)\prod_{t=0}^{T-1}\pi_{t,\theta'}(a_t|s_t)p(s_{t+1}|s_t,a_t)\rho(r_t|s_t,a_t)} \tag{30}$$

$$= \frac{\prod_{t=0}^{T-1}\pi_{t,\theta}(a_t|s_t)}{\prod_{t=0}^{T-1}\pi_{t,\theta'}(a_t|s_t)} \tag{31}$$

$$= \prod_{t=0}^{T-1}\frac{\pi_{t,\theta}(a_t|s_t)}{\pi_{t,\theta'}(a_t|s_t)} \tag{32}$$

Likelihood ratio policy gradient via importance sampling

- Exact gradient

$$\nabla_\theta J(\theta) = \mathop{\mathbb{E}}_{\substack{s_0 \sim P_0(\cdot) \\ a_t \sim \pi_{t,\theta'}(\cdot|s_t) \\ s_{t+1} \sim p(\cdot|s_t, a_t) \\ r_t \sim \rho(\cdot|s_t, a_t)}} \left\{ \prod_{t=0}^{T-1} \frac{\pi_{t,\theta}(a_t|s_t)}{\pi_{t,\theta'}(a_t|s_t)} \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \right\} \tag{33}$$

- Monte-Carlo approximation

$$\hat{\nabla}_\theta J(\theta) = \left\langle \prod_{t=0}^{T-1} \frac{\pi_{t,\theta}(a_t|s_t)}{\pi_{t,\theta'}(a_t|s_t)} \sum_{t=0}^{T-1} \nabla_\theta \log \pi_{t,\theta}(a_t|s_t) \sum_{t=0}^{T-1} r_t \right\rangle_n \tag{34}$$

Likelihood ratio policy gradient via importance sampling

Algorithm 2 Off-policy stochastic gradient ascent

Inputs Environment
Inputs Number of iterations N and M
Inputs Learning rate α
Inputs Parametrized policy $\pi_{t,\theta}$
for all $n = 1, \ldots, N$ **do**
    Sample a set of trajectories trajectories with $\pi_{t,\theta}$
    $\theta' \leftarrow \theta$
    for all $m = 1, \ldots, M$ **do**
        From the trajectories samples, compute $J(\theta)$
        $\theta \leftarrow \theta + \alpha \cdot \hat{\nabla}_\theta J(\theta)$
    end for
end for
return $\pi_{t,\theta}$

Finite-time MDPs

In practice, it is common to work with stationary policies (for large T...)

Next week

- Infinite-time MDPs and the policy gradient theorem
- Policy gradients in Riemannian spaces...