Introduction to Reinforcement Learning

A mini course @ HCMUS, Vietnam

Lectures 4-6

Long Tran-Thanh

Long.tran-thanh@warwick.ac.uk

Last lectures

Function approximation (Lectures 3&4):

- In large scale RL, we approximate V(s) and Q(s,a) with v(s, w) and q(s,a,w)
- We implicitly derive a policy from there: main goal is to learn the values. Policy is just a helper to do this learning efficiently (e.g., we use epsilon-greedy)
- This is called **value-based RL**

Direct optimisation on policy space:

- More compact representation of policies
- Can do efficient search directly on the policy space (using gradient descent)
- This is called **policy-based RL**

Policy gradient (Lecture 5):

- REINFORCE (Williams, 1992)
- Actor-Critic (Crites & Barto, 1994)
- A2C, A3C (Mnih *et al.*, 2016)

Advanced RL methods

Natural policy gradient (Kakade, 2001)

Standard PG revisited: $\Delta\theta = \alpha \nabla_\theta J(\theta)$

- Gradient descent is on the parameter space Θ
- Depends on the choice of feature vectors. HOW can we get rid of this dependency?

Idea: policy = distribution $P(\pi(s) = a) := P_\pi(a|s)$ (independent from choice of parameters)

- Would be much more "natural" if we do gradient descent on the probability space of policies
- What we need is the "distance" definition for distributions (for parameter space we used e.g., Euclidean)
- Solution: KL-divergence: $\bar{D}_{KL}(\pi_{\theta_{t+1}} \| \pi_{\theta_t}) := \mathbb{E}_{s \sim \pi_{\theta_t}} \left[D_{KL}(\pi_{\theta_{t+1}}(\cdot|s) \| \pi_{\theta_t}(\cdot|s)) \right]$
- When KL-divergence is very small: $\bar{D}_{KL}(\pi_{\theta_{t+1}} \| \pi_{\theta_t}) \approx \frac{1}{2}(\theta_{t+1} - \theta_t)^T F(\theta_t)(\theta_{t+1} - \theta_t)$
  - Fischer information matrix (curvature of J on the distribution space)

$$F(\theta) = \mathbb{E}_{s,a \sim \pi_\theta} \left[\nabla_\theta \ln \pi_\theta(a|s) (\nabla_\theta \ln \pi_\theta(a|s))^T \right]$$

Natural policy gradient

Update rule (Kakade, 2001): $\theta_{t+1} = \theta_t + \alpha F(\theta_t)^{-1} \nabla_\theta J(\theta_t)$

- Much more robust and natural than PG
- BUT: computationally very expensive (inverting F)
- Approximation of F inverse is needed (ideas from the optimisation literature)

TRPO: Trust Region Policy Optimization (Schulman et al., 2015)

Update rule of PG: $\theta_{t+1} = \theta_t + \alpha \nabla_\theta J(\theta_t)$

- This is in fact a solution of the following optimisation problem

$$\min_{\theta_{t+1}} J(\theta_t) + (\theta_{t+1} - \theta_t)^T \nabla_\theta J(\theta_t)$$

$$||\theta_{t+1} - \theta_t|| \leq \alpha ||\nabla_\theta J(\theta_t)||$$

- For natural PG, the constraint becomes $\bar{D}_{KL}(\pi_{\theta_{t+1}} \| \pi_{\theta_t}) \leq \epsilon$

- TRPO uses a surrogate objective function: $\max_\theta L(\theta, \theta_t)$

$$L(\theta, \theta_t) = \mathbb{E}_{s,a \sim \pi_{\theta_t}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} A^{\pi_{\theta_t}}(s, a) \right]$$

TRPO: Trust Region Policy Optimization (Schulman et al., 2015)

$$\max_\theta L(\theta, \theta_t)$$
$$\bar{D}_{KL}(\pi_\theta \| \pi_{\theta_t}) \leq \epsilon \qquad L(\theta, \theta_t) = \mathbb{E}_{s,a\sim\pi_{\theta_t}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

Why this is good?

Answer: Using Taylor expansion as approximator, we can get

$$\theta_{t+1} = \theta_t + \sqrt{\frac{2\epsilon}{g^T F^{-1} g}} F^{-1} g$$

$$g = \nabla_\theta L(\theta, \theta_t)\big|_{\theta=\theta_t}$$ (policy gradient)

Issue of inverting F -> use conjugate gradient method for Fx = g (to calculate x)

PPO: Proximal Policy Optimization (Schulman *et al.,* 2017)

$$\max_\theta L(\theta, \theta_t)$$
$$\bar{D}_{KL}(\pi_\theta \| \pi_{\theta_t}) \le \epsilon \qquad L(\theta, \theta_t) = \mathbb{E}_{s,a\sim\pi_{\theta_t}}\left[\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} A^{\pi_{\theta_t}}(s,a)\right]$$

It’s a challenge to handle the KL divergence constraint

- Idea: use clipping

$$\max_\theta \mathbb{E}_{s,a\sim\pi_{\theta_t}}\left[\begin{cases} \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)}, 1+\epsilon\right) A^{\pi_{\theta_t}}(s,a) & \text{if } A^{\pi_{\theta_t}}(s,a) > 0 \\ \max\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)}, 1-\epsilon\right) A^{\pi_{\theta_t}}(s,a) & \text{if } A^{\pi_{\theta_t}}(s,a) < 0 \end{cases}\right]$$

- Then use SGD/Adam etc. to optimise this clipped objective function

GRPO: Group Relative Policy Optimization (DeepSeek, 2025 – Shao *et al.*, 2025)

$$\max_\theta L(\theta, \theta_t)$$
$$\bar{D}_{KL}(\pi_\theta \| \pi_{\theta_t}) \le \epsilon$$

$$L(\theta, \theta_t) = \mathbb{E}_{s,a\sim\pi_{\theta_t}} \left[\frac{\pi_\theta(a|s)}{\pi_{\theta_t}(a|s)} A^{\pi_{\theta_t}}(s,a) \right]$$

$$A(s,a) = Q(s,a) - V(s)$$

Advantage value

Instead of calculating V(s) as PPO does, GRPO uses a relative group advantage estimate:

- Samples G actions for each state s: $a_1, a_2, \ldots, a_G$
- Calculate: $A(s,a_j) = \dfrac{r(s,a_j) - \mu}{\sigma}$ where (μ, σ) is the empirical mean-std of $r(s,a_i)$
- Use PPO to solve

$$\max_\theta \frac{1}{G} \sum_{i=1}^{G} \mathbb{E}_{(s,a_1,\ldots,a_G)\sim\pi_{\theta_t}} \left[\begin{cases} \min\left(\dfrac{\pi_\theta(a_i|s)}{\pi_{\theta_t}(a_i|s)}, 1+\epsilon\right) A^{\pi_{\theta_t}}(s,a_i) & \text{if } A^{\pi_{\theta_t}}(s,a_i) > 0 \\ \max\left(\dfrac{\pi_\theta(a_i|s)}{\pi_{\theta_t}(a_i|s)}, 1-\epsilon\right) A^{\pi_{\theta_t}}(s,a_i) & \text{if } A^{\pi_{\theta_t}}(s,a_i) < 0 \end{cases} \right]$$