Reinforcement Learning

Policy Gradient

袁路展

北京邮电大学

2021 年 10 月 29 日

- 什么是策略梯度？
- 为什么要策略梯度？
- 策略梯度的关键问题是？
- 一些重要的概念
  - REINFORCE
  - High variance
  - Score function
  - Actor-Critic

Review on Reinforcement Learning

What is Reinforcement Learning?

- Learn an optimal policy from the interact messages.

Why Reinforcement Learning?

- High calculation complexity of classical method like DP ($O(n^2)$), Optimal Control(Complex equations).
- Complex state represetation.

How to achieve the goal of RL?

- Value-based
- Policy-based
- Actor-critic

1 Target

2 Review

- Review on Reinforcement Learning
- Review on Value-Based Methods

3 Policy Gradient

4 Summary

Review on Value-Based Methods

- In Value-Based methods, the agent approximate the value or action-value function using parameter θ,

$$V_\theta \approx V^\pi(s)$$
$$Q_\theta(s,a) \approx Q^\pi(s,a)$$

- Then a policy is generated from the value function by various methods like greedy method, $\epsilon - greedy$ and so on.
- Disadvantages of Value-based methods
  - How to select the action to take especially in continous action space.
  - Cannot handle the stochastic policy well.

What is Policy Gradient

- 强化学习的目标是找到最优的策略 π, 最大化收益 $J(\pi)$.
- 基于价值函数的方法通过估计每个动作对应的价值，然后通过选取价值最大的动作来得到最优的策略。
- 但是选取价值最大的动作在高维动作空间或者连续动作空间是非常困难的。
- 所以我们为什么不直接优化 $J(\pi)$
- 最简单的优化方法: 梯度下降

What is Policy Gradient

- The original form of Policy Gradient:

$$\theta := \theta + \alpha \nabla J(\theta) \tag{1}$$

- Basic framework of Policy Gradient

  while θ *is not optimal* **do**

  Evaluate the policy

  Update θ by gradient descent

  end

- **Key problem: How to evaluate the policy and the policy gradient?**

How to evaluate the policy

- 在强化学习中，我们的目标是让策略 π_θ 与环境交互产生的序列 $\tau=(s_0,a_0,s_1,a_1,\cdots,s_T,a_T)$ 的回报值最大.

$$J(\theta)=\mathbb{E}_{\tau\sim\rho_\theta}\left[\sum_{t=0}^{T}\gamma^t r\left(s_t,a_t,s_{t+1}\right)\right]=\int_\tau \rho_\theta(\tau)R(\tau)d\tau \quad (2)$$

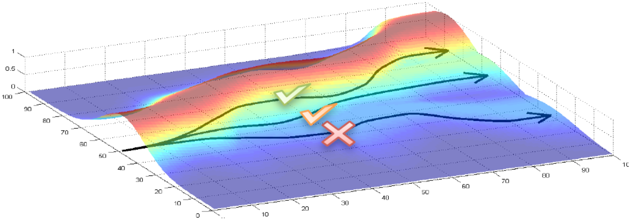
策略梯度:

$$\nabla_\theta J(\theta)=\nabla_\theta\int_\tau \rho_\theta(\tau)R(\tau)d\tau=\int_\tau\left(\nabla_\theta\rho_\theta(\tau)\right)R(\tau)d\tau \quad (3)$$

score function

In equation(3), the gradient tries to

- Increase the probability of trajectories with higer return.
- Decrease the probability of trajectories with lower return.



Does not try to change the trajectories!

score function

What's worse, because $\rho_\theta(\tau)$ is always represented in the likelihood form:

$$\rho_\theta(\tau) = p_\theta\left(s_0, a_0, \ldots, s_T, a_T\right) = p_0\left(s_0\right) \prod_{t=0}^{T} \pi_\theta\left(a_t | s_t\right) p\left(s_{t+1} \mid s_t, a_t\right)$$

and the gradient of $\prod$ is not easy to calculate.
In real analysis, log trick a common method to solve this problem.

$$\log \rho_\theta(\tau) = \log p_0\left(s_0\right) + \sum_{t=0}^{T} \log \pi_\theta\left(a_t | s_t\right) + \sum_{t=0}^{T} \log p\left(s_{t+1} \mid s_t, a_t\right)$$

$$\nabla_\theta \rho_\theta(\tau) = \nabla_\theta \rho_\theta(\tau) \frac{\rho_\theta(\tau)}{\rho_\theta(\tau)} = \rho_\theta(\tau) \nabla_\theta \log \rho_\theta(\tau)$$

score function

Let's Decompose the tracjectories into states and actions ad the policy gradient becomes:

$$
\begin{aligned}
\nabla_\theta \log \rho_\theta &= \nabla_\theta \left[p_0\left(s_0\right) \prod_{t=0}^{T} \pi_\theta\left(a_t \middle| s_t\right) p\left(s_{t+1} \mid s_t, a_t\right) \right] \\
&= \nabla_\theta \left[\log p_0\left(s_0\right) + \sum_{t=0}^{T} \log \pi_\theta\left(a_t \middle| s_t\right) + \sum_{t=0}^{T} \log p\left(s_{t+1} \mid s_t, a_t\right) \right] \\
&= \nabla_\theta \sum_{t=0}^{T} \log \pi_\theta\left(a_t \middle| s_t\right) \\
&= \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta\left(a_t \middle| s_t\right)
\end{aligned} \tag{4}
$$

The score function is $\nabla_\theta \log \pi_\theta(a|s)$

score function

The policy gradient becomes:

$$\begin{aligned}
\nabla_\theta J(\theta) &= \int_\tau \rho_\theta(\tau) \nabla_\theta \log \rho_\theta(\tau) R(\tau) d\tau \\
&= \mathbb{E}_{\tau \sim \rho_\theta} \left[\nabla_\theta \log \rho_\theta R(\tau) \right] \\
&= \mathbb{E}_{\tau \sim \rho_\theta} \left[\sum_{t=0}^{T} \nabla_\theta \log \pi_\theta \left(a_t | s_t \right) R(\tau) \right] \\
&\approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_{i,t} \mid \mathbf{s}_{i,t} \right) \right) \left(\sum_{t=1}^{T} r \left(\mathbf{s}_{i,t}, \mathbf{a}_{i,t} \right) \right)
\end{aligned} \tag{5}$$

advantage of score function

- No dynamics in the objective function.
- Focus on the policy instead of the trajectory.
- Better computing feathers.
  - Most stochastic policies are represented in exponential form.

Monte-Carlo Policy(REINFORCE)

- REINFORCE
  Initialise θ
  for *each episode* **do**
  Generate an episode τ^i by $\pi(\theta)$
  $$\nabla_\theta J(\theta) \approx \sum_i \left(\sum_t \nabla_\theta \log \pi_\theta \left(\mathbf{a}_t^i \mid \mathbf{s}_t^i \right) \right) \left(\sum_t r \left(\mathbf{s}_t^i, \mathbf{a}_t^i \right) \right)$$
  $$\theta \leftarrow \theta + \alpha \nabla_\theta J(\theta)$$
  end
  return θ
- **pay attention to** v_t

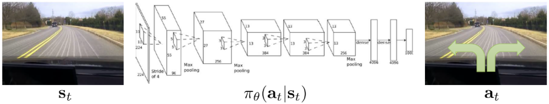
Compaison to maximum likelihood

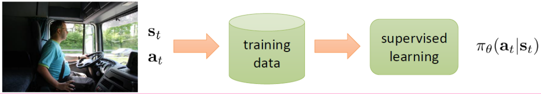
policy gradient

$$\nabla_\theta J(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_{i,t} \mid \mathbf{s}_{i,t} \right) \right) \left(\sum_{t=1}^{T} r \left(\mathbf{s}_{i,t}, \mathbf{a}_{i,t} \right) \right) \tag{6}$$

maximum likelihood

$$\nabla_\theta J_{\mathrm{ML}}(\theta) \approx \frac{1}{N} \sum_{i=1}^{N} \left(\sum_{t=1}^{T} \nabla_\theta \log \pi_\theta \left(\mathbf{a}_{i,t} \mid \mathbf{s}_{i,t} \right) \right) \tag{7}$$





Restriction on REINFORCE

While very simple, REINFORCE does not work well in practice:

- The return $R(\tau)$ have a very high variance.
- Sensitive to the reward.
- It requires a lot of episodes to converge.
- It only works with online learning.
- Restricted in episodic environments.

High Variance in REINFORCE

- 想象一下，你在玩英雄联盟，你刚开始进入游戏采取的动作都是一样的，但是有的局你赢了，有的局你输了，那么你最开始的动作，应该向哪个方向优化？
- 考虑两个环境 A 和 B，他们有相同的 dynamics 和相同的任务，但是 reward 设置不一样，在 A 中，执行动作 1 的奖励是 1000，执行动作 2 的奖励是 1001，在 B 中执行动作 1 的奖励是 0，执行动作 2 的奖励是 1，这两种情况下，显然 A 环境下收敛速度要缓慢。
- 在有监督学习中，high variance 的问题很少，因为训练集是固定的，但是在强化学习中，你不可能进行足够多的采样来覆盖整个动作-序列空间。

High Variance in REINFORCE

The problem is even worse in the following conditions:

- High-dimensional action spaces: it becomes difficult to sample the environment densely enough if many actions are possible.
- Long horizons: the longer the trajectory, the more likely it will be unique.
- Finite samples: if we cannot sample enough trajectories, the high variance can introduce a bias in the gradient, leading to poor convergence.

1. Target
2. Review
3. Policy Gradient
   - Intro to Policy Gradient
   - Score function
   - REINFORCE
   - Policy Gradient Theorem
4. Summary

Policy Grandient Theorem

定理

Policy Gradient Theorem

$$\nabla_\theta J(\theta) = \int_{\mathcal{S}} \rho^{\pi_\theta}(s) \int_{\mathcal{A}} \nabla_\theta \pi_\theta(a|s) Q^\pi(s,a) dads$$

- **Because the actual return $R(\tau)$ is replaced by its expectation Q(s,a), the policy gradient is now mathematical expectation over single transition instead of the complete trajectories.**
- $\rho^{\pi_\theta}(s)$ depends on θ, but there is no $\nabla_\theta \rho^{\pi_\theta}(s)$ term in $\nabla_\theta J(\theta)$

Why Policy Gradient

Advantages of PG

- Learn the final object of RL directly. (*learn the optimal policy directly*)
- Easy extend to **high-dimensional or continous** action sapce. (*no argmax operator*)
- Can learn **stochastic** policies.

Disadvantages of PG

- Get stuck in local optimal. (*common problem of gradient methods*)
- Evaluating a policy is typically inefficient and **high variance**.

Review on this class

- 什么是策略梯度?
  - 直接根据 $J(\pi)$ 的梯度更新策略的参数。
- 为什么要策略梯度?
  - 基于值函数的方法不能很好应对高维、连续动作空间。
- 策略梯度的关键问题是什么?
  - 策略的评估!!!!!

Review on this class

关键知识点：

- Score function
    - 为什么用 log？对 trajectory 的分解。
- REINFORCE
    - Monte-Carlo policy gradient.
- Policy Gradient Theorem
    - 不再是对整个路径的积分，只考虑单个 **transition**!! 为后面 **Actor-Critic** 结构奠定基础

Next Class

- Advantage Policy Gradient*(reduce the variance of policy gradient)*
- Actor-Critic *(Most common algorithm frame in RL)*
- off-policy Actor-Critic *(practical policy evaluation)*

Explore yourself

- TRPO,PPO. *(montonic improvenment based in policy gradient)*
- DPG, DDPG, TD3. *(introduce success tricks in DQN to PG)*
- soft-RL,SAC,entropy-based methods. *(Explore-Exploit in PG)*
- off-policy off-line policy evaluation.

Questions

Q&A