

# 强化学习基本原理及编程实现：TRPO

郭宪

2019.11.16

人工智能学院

College of Artificial Intelligence



南开大学  
Nankai University



# 策略梯度的缺点

1. 很难在整个优化过程选择一个时间步长，特别是由于状态和回报在改变统计特性。

2. 策略经常会过早地收敛到一个次优的几乎确定的策略。

步长的重要性！

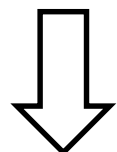
当步长不合适时，所学到的策略是一个坏的策略，由坏的策略所采集到的数据也是不好的数据，学习很可能会崩溃

# 替代回报函数

用  $\tau$  表示一组状态-行为序列  $s_0, u_0, \dots, s_H, u_H$

目标函数为:

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right]$$



等价的替代函数

$$\eta(\tilde{\pi}) = \underbrace{\eta(\pi)}_{\text{老的策略}} + \underbrace{E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]}_{\text{新旧策略回报差}}$$

老的策略

新旧策略  
回报差

优势函数的定义:

$$\begin{aligned} A_{\pi}(s, a) &= Q_{\pi}(s, a) - V_{\pi}(s) \\ &= E_{s' \sim P(s'|s, a)} [r(s) + \gamma V^{\pi}(s') - V^{\pi}(s)] \end{aligned}$$

证明:

$$\begin{aligned} &E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right] \\ &= E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t) + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)) \right] \\ &= E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) + \sum_{t=0}^{\infty} \gamma^t (\gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)) \right] \\ &= E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right] + E_{s_0} [-V^{\pi}(s_0)] \\ &= \eta(\tilde{\pi}) - \eta(\pi) \end{aligned}$$

# 替代回报函数

用  $\tau$  表示一组状态-行为序列  $s_0, u_0, \dots, s_H, u_H$

目标函数为:

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right]$$

等价的替代函数

$$\eta(\tilde{\pi}) = \underbrace{\eta(\pi)}_{\text{老的策略}} + \underbrace{E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]}_{\text{新旧策略回报差}}$$

定义:

$$\rho_{\pi}(s) = P(s_0 = s) + \gamma P(s_1 = s) + \gamma^2 P(s_2 = s) + \dots$$

第t步出现s的概率      状态s处动作进行加和

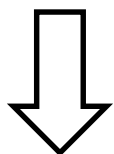
$$\begin{aligned} \eta(\tilde{\pi}) &= \eta(\pi) + \sum_{t=0}^{\infty} \sum_s \underbrace{P(s_t = s|\tilde{\pi})}_{\text{第t步出现s的概率}} \underbrace{\sum_a \tilde{\pi}(a|s)}_{\text{状态s处动作进行加和}} \gamma^t A_{\pi}(s, a) \\ &= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a) \end{aligned}$$

# 替代回报函数

用  $\tau$  表示一组状态-行为序列  $s_0, u_0, \dots, s_H, u_H$

目标函数为：

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right]$$



等价的替代函数

$$\eta(\tilde{\pi}) = \underbrace{\eta(\pi)}_{\text{老的策略}} + \underbrace{E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]}_{\text{新旧策略回报差}}$$

老的策略

新旧策略  
回报差

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a)$$

策略改善理论

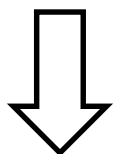
$$\begin{aligned} v_{\pi}(s) &\leq q_{\pi}(s, \pi'(s)) \\ &= \mathbb{E}[R_{t+1} + \gamma v_{\pi}(s_{t+1}) | s_t = s, a_t = \pi'(s)] \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma v_{\pi}(s_{t+1}) | s_t = s] \\ &\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma q_{\pi}(s_{t+1}, \pi'(s_{t+1})) | s_t = s] \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma \mathbb{E}_{\pi'}[R_{t+2} + \gamma v_{\pi}(s_{t+1}) | s_{t+1}, a_{t+1} = \pi'(s_{t+1})] | s_t = s] \\ &= \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 v_{\pi}(s_{t+2}) | s_t = s] \\ &\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 v_{\pi}(s_{t+2}) | s_t = s] \\ &\vdots \\ &\leq \mathbb{E}_{\pi'}[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \gamma^3 R_{t+4} + \dots | s_t = s] \\ &= v_{\pi'}(s) \end{aligned}$$

# 替代回报函数

用  $\tau$  表示一组状态-行为序列  $s_0, u_0, \dots, s_H, u_H$

目标函数为：

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right]$$



等价的替代函数

$$\eta(\tilde{\pi}) = \underbrace{\eta(\pi)}_{\text{老的策略}} + \underbrace{E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]}_{\text{新旧策略回报差}}$$

老的策略

新旧策略  
回报差

$$= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a)$$

策略改善理论

一个更好的策略应该满足：

在每个状态  $s$  处：

$$\sum_a \tilde{\pi}(a|s) A^{\pi}(s, a) \geq 0$$

然而，在函数逼近的情况下：

存在估计和逼近误差，因此导致在一些状态处

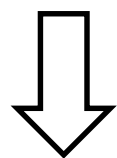
$$\sum_a \tilde{\pi}(a|s) A^{\pi}(s, a) < 0$$

# 替代回报函数

用  $\tau$  表示一组状态-行为序列  $s_0, u_0, \dots, s_H, u_H$

目标函数为：

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right]$$



等价的替代函数

$$\eta(\tilde{\pi}) = \underbrace{\eta(\pi)}_{\text{老的策略}} + \underbrace{E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]}_{\text{新旧策略回报差}}$$

老的策略

新旧策略  
回报差

$$\eta(\tilde{\pi}) = \eta(\pi) + \underbrace{\sum_s \rho_{\tilde{\pi}}(s)}_{\text{注意：这时状态分布由新的策略产生，对新的策略严重依赖}} \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a)$$

注意：这时状态分布由新的策略产生，对新的策略严重依赖

**第一个技巧：**对状态分布的处理，忽略状态分布的变化，依然用旧的策略所对应的状态分布

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\pi}(s) \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a)$$

**第二个技巧：**对动作分布的处理，重要性采样

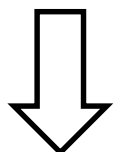
$$\sum_a \tilde{\pi}_{\theta}(a|s_n) A_{\theta_{old}}(s_n, a) = E_{a \sim q} \left[ \frac{\tilde{\pi}_{\theta}(a|s_n)}{q(a|s_n)} A_{\theta_{old}}(s_n, a) \right]$$

# 替代回报函数

用  $\tau$  表示一组状态-行为序列  $s_0, u_0, \dots, s_H, u_H$

目标函数为：

$$\eta(\tilde{\pi}) = E_{\tau|\tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r(s_t)) \right]$$



等价的替代函数

$$\eta(\tilde{\pi}) = \underbrace{\eta(\pi)}_{\text{老的策略}} + \underbrace{E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]}_{\text{新旧策略回报差}}$$

老的策略

新旧策略  
回报差

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\pi}(s) \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a)$$

$$\sum_a \tilde{\pi}_{\theta}(a|s_n) A_{\theta_{old}}(s_n, a) = E_{a \sim q} \left[ \frac{\tilde{\pi}_{\theta}(a|s_n)}{q(a|s_n)} A_{\theta_{old}}(s_n, a) \right]$$

$$\frac{1}{1-\gamma} E_{s \sim \rho_{\theta_{old}}} [\dots] \quad \text{代替} \quad \sum_s \rho_{\theta_{old}}(s) [\dots]$$

$$q(a|s_n) = \pi_{\theta_{old}}(a|s_n)$$

$$L_{\pi}(\tilde{\pi}) = \eta(\pi) + E_{s \sim \rho_{\theta_{old}}, a \sim \pi_{\theta_{old}}} \left[ \frac{\tilde{\pi}_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right]$$



# 原回报函数与替代回报函数比较

原回报函数：

$$\eta(\tilde{\pi}) = \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a|s) A^{\pi}(s, a)$$

替代回报函数：

$$L_{\pi_{\theta_{old}}}(\pi_{\theta}) = \eta(\pi_{\theta_{old}}) + E_{s \sim \rho_{\theta_{old}}, a \sim \pi_{\theta}} \left[ \frac{\tilde{\pi}_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right]$$

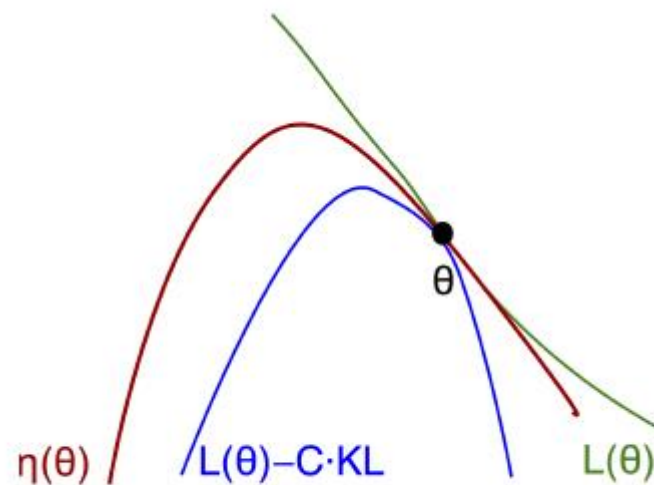
在  $\theta_{old}$  处一阶近似：

$$L_{\pi_{\theta_{old}}}(\pi_{\theta_{old}}) = \eta(\pi_{\theta_{old}})$$

$$\nabla_{\theta} L_{\pi_{\theta_{old}}}(\pi_{\theta})|_{\theta=\theta_{old}} = \nabla_{\theta} \eta(\pi_{\theta})|_{\theta=\theta_{old}}$$

在  $\theta_{old}$  附近，能改善L的策略也能改善

原回报函数。问题是步长多大呢？



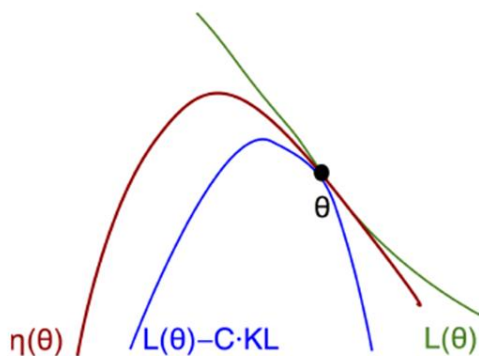
# 原回报函数与替代回报函数比较

## 定理1:

令  $\alpha = D_{TV}^{\max}(\pi_{old}, \pi_{new})$  , 则存在下面的界:

$$\eta(\pi_{new}) \geq L_{\pi_{old}}(\pi_{new}) - \frac{4\varepsilon}{(1-\gamma)^2} \alpha^2$$

Where:  $\varepsilon = \max_{s,a} |A_{\pi}(s,a)|$




---

**Algorithm 1** Policy iteration algorithm guaranteeing non-decreasing expected return  $\eta$

---

Initialize  $\pi_0$ .

**for**  $i = 0, 1, 2, \dots$  until convergence **do**

    Compute all advantage values  $A_{\pi_i}(s, a)$ .

    Solve the constrained optimization problem

$$\pi_{i+1} = \arg \max_{\pi} [L_{\pi_i}(\pi) - C D_{KL}^{\max}(\pi_i, \pi)]$$

    where  $C = 4\varepsilon\gamma/(1-\gamma)^2$

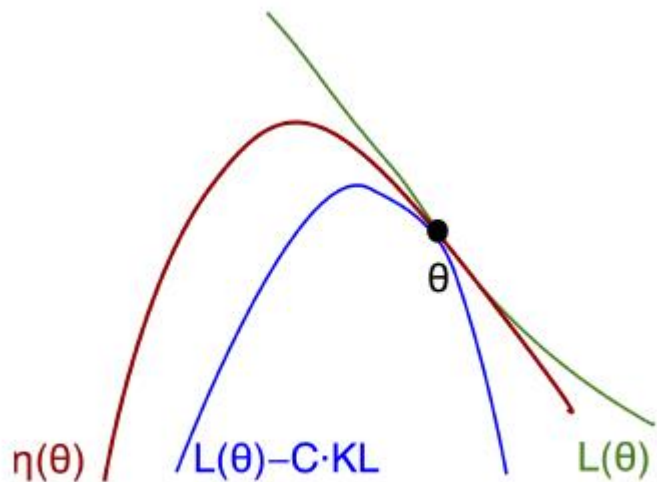
$$\text{and } L_{\pi_i}(\pi) = \eta(\pi_i) + \sum_s \rho_{\pi_i}(s) \sum_a \pi(a|s) A_{\pi_i}(s, a)$$

**end for**

---

策略迭代算法

# 单调改进策略



$$\eta(\tilde{\pi}) \geq L_{\pi}(\tilde{\pi}) - CD_{KL}^{\max}(\pi, \tilde{\pi}),$$

$$\text{where } C = \frac{2\varepsilon\gamma}{(1-\gamma)^2}$$

$$\text{令: } M_i(\pi) = L_{\pi_i}(\pi) - CD_{KL}^{\max}(\pi_i, \pi)$$

$$\text{则: } \eta(\pi_{i+1}) \geq M_i(\pi_{i+1})$$

$$\text{又因为: } \eta(\pi_i) = M_i(\pi_i)$$

$$\text{所以: } \eta(\pi_{i+1}) - \eta(\pi_i) \geq M_i(\pi_{i+1}) - M_i(\pi_i)$$

所以每次**最大化**  $M_i$  能够保证策略非递减。

如何利用理论来得到最优的策略  $\pi_{i+1}$ ?

参数化策略!

# TRPO实用算法：优化参数化的策略

问题形式化为：

$$\underset{\theta}{\text{maximize}} [L_{\theta_{old}}(\theta) - CD_{KL}^{\max}(\theta_{old}, \theta)]$$

如果利用惩罚因子C则每次迭代步长很小，  
因此问题可转化为：

$$\underset{\theta}{\text{maximize}} E_{s \sim \rho_{\theta_{old}}, a \sim \pi_{\theta_{old}}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right]$$

$$\text{subject to } D_{KL}^{\max}(\theta_{old}, \theta) \leq \delta$$

无穷多个状态，无穷多约束

第三个技巧：利用平均KL散度：

$$\text{subject to } \bar{D}_{KL}^{\rho_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$$

第四个技巧：利用当前策略状态空间分布

$$s \sim \rho_{\theta_{old}} \rightarrow s \sim \pi_{\theta_{old}}$$

TRPO的问题形式化为：

$$\underset{\theta}{\text{maximize}} E_{s \sim \pi_{\theta_{old}}, a \sim \pi_{\theta_{old}}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right]$$

$$\text{subject to } E_{s \sim \pi_{\theta_{old}}} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta$$

# TRPO实用算法

第三个技巧：利用平均KL散度：

$$\text{subject to } \bar{D}_{KL}^{\rho_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$$

第四个技巧：利用当前策略状态空间分布

$$s \sim \rho_{\theta_{old}} \rightarrow s \sim \pi_{\theta_{old}}$$

TRPO的问题形式化为：

$$\text{maximize}_{\theta} E_{s \sim \pi_{\theta_{old}}, a \sim \pi_{\theta}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right]$$

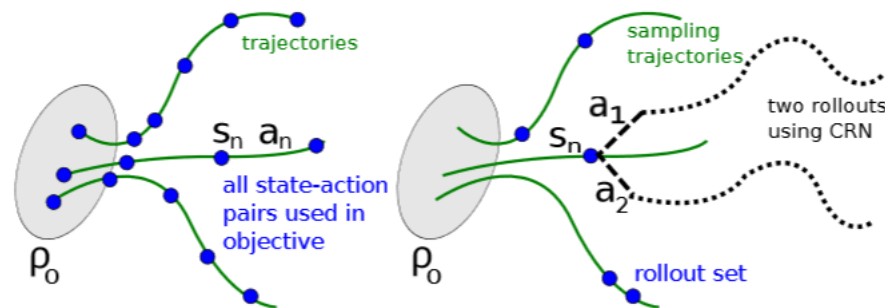
$$\text{subject to } E_{s \sim \pi_{\theta_{old}}} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta$$

如何得到目标函数和约束条件？

基于采样的方法对目标函数和约束进行估计

利用蒙特卡洛方法估计：

利用样本的平均值来代替期望



Single path

Vine

# TRPO实用算法

第三个技巧：利用平均KL散度：

$$\text{subject to } \bar{D}_{KL}^{\rho_{\theta_{old}}}(\theta_{old}, \theta) \leq \delta$$

第四个技巧：利用当前策略状态空间分布

$$s \sim \rho_{\theta_{old}} \rightarrow s \sim \pi_{\theta_{old}}$$

TRPO的问题形式化为：

$$\begin{aligned} & \underset{\theta}{\text{maximize}} E_{s \sim \pi_{\theta_{old}}, a \sim \pi_{\theta}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right] \\ & \text{subject to } E_{s \sim \pi_{\theta_{old}}} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta \end{aligned}$$

TRPO算法流程：

For 迭代 1,2, ... do

运行当前策略T步或N条轨迹

利用所有时间步数据估计优势函数

利用共轭梯度法和线性搜索计算参数更新

End for

# 共轭梯度法搜索可行方向

$$\begin{aligned} & \underset{\theta}{\text{maximize}} E_{s \sim \pi_{\theta_{old}}, a \sim \pi_{\theta_{old}}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right] \\ & \text{subject to } E_{s \sim \pi_{\theta_{old}}} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta \end{aligned}$$

将目标进行线性化逼近，将约束进行二次逼近后优化问题为：

$$\begin{aligned} & \underset{\theta}{\text{maximize}} [\nabla_{\theta} L_{\theta_{old}}(\theta)|_{\theta=\theta_{old}} \cdot (\theta - \theta_{old})] \\ & \text{subject to } \frac{1}{2} (\theta_{old} - \theta)^T A(\theta_{old}) (\theta_{old} - \theta) \leq \delta \end{aligned}$$

A为Fisher信息矩阵：

$$A_{ij} = \frac{\partial}{\partial \theta_i} \frac{\partial}{\partial \theta_j} \bar{D}_{KL}(\theta_{old}, \theta)$$

$$\underbrace{\frac{\partial \mu_a(x)}{\partial \theta_i} \text{kl}_{ab}''(\mu_{\theta}(x), \mu_{old}(x)) \frac{\partial \mu_b(x)}{\partial \theta_j}}_{J^T M J} + \underbrace{\frac{\partial^2 \mu_a(x)}{\partial \theta_i \partial \theta_j} \text{kl}'_a(\mu_{\theta}(x), \mu_{old}(x))}_{=0 \text{ at } \mu_{\theta}=\mu_{old}}$$



# 共轭梯度法搜索可行方向

$$\begin{aligned} & \underset{\theta}{\text{maximize}} E_{s \sim \pi_{\theta_{old}}, a \sim \pi_{\theta_{old}}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right] \\ & \text{subject to } E_{s \sim \pi_{\theta_{old}}} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta \end{aligned}$$

将目标进行线性化逼近，将约束进行二次逼近后优化问题为：

$$\begin{aligned} & \underset{\theta}{\text{maximize}} [\nabla_{\theta} L_{\theta_{old}}(\theta)|_{\theta=\theta_{old}} \cdot (\theta - \theta_{old})] \\ & \text{subject to } \frac{1}{2} (\theta_{old} - \theta)^T A(\theta_{old}) (\theta_{old} - \theta) \leq \delta \end{aligned}$$

令  $d = \theta - \theta_{old}$  为搜索方向，则搜索方向应该满足： $A(\theta_{old})d = \nabla_{\theta} L_{\theta_{old}}(\theta)|_{\theta=\theta_{old}}$

利用共轭梯度方法求解线性方程组  $AX=b$  的解方法：

构造目标函数  $f(x) = \frac{1}{2} x^T A x - b x$ ，则  $x$  是目标函数的最小值。

Step1: 给定初试迭代点  $x^{(1)}$ ，令  $k=1$

Step2: 计算梯度  $g_k = \nabla f(x^{(k)}) = A x^{(k)} - b$ ，若  $\|g_k\| = 0$  则停止计算，并令  $x^* = x^{(k)}$ ，否则转下一步

Step3: 构造搜索方向，首先计算步长  $\beta_{k-1} = \frac{(d^{(k-1)})^T A g_k}{(d^{(k-1)})^T A d^{(k-1)}}$ ，若  $k=1, \beta_{k-1}=0$

搜索方向为： $d^k = -g_k + \beta_{k-1} d^{k-1}$

Step4: 计算搜索步长  $\lambda_k = -\frac{g_k^T d^{(k)}}{(d^{(k)})^T A d^{(k)}}$ ，更新数据点  $x^{(k+1)} = x^{(k)} + \lambda_k d^{(k)}$

Step5: 若  $k=n$ ，则停止计算。得到  $x^* = x^{(k+1)}$ ，否则令  $k=k+1$ ，转步骤 step2

此处： $b = \nabla_{\theta} L_{\theta_{old}}(\theta)|_{\theta=\theta_{old}}$





## 线性搜索得到步长

第二步，利用线性搜索方法计算步长 $\beta$ ：

将第一步求得的搜索方向 $d$ 乘以步长，带入约束方程得到：

$$\delta \approx \frac{1}{2} (\beta d)^T A (\beta d) = \frac{1}{2} \beta^2 d^T A d$$

从而得到步长为：
$$\beta = \sqrt{\frac{2\delta}{d^T A d}}$$

将 $\beta$ 带入目标函数 $L_{\theta_{old}}(\theta) - \mathcal{X}[\bar{D}_{KL}(\theta_{old}, \theta) \leq \delta]$ ，其中 $\mathcal{X}[\bar{D}_{KL}(\theta_{old}, \theta) \leq \delta]$ 表示当满足不等式时为零，当不满足不等式时为无穷大。收缩步长 $\beta$ ，直到目标函数得到改善。

↵

$$\theta_{new} = \theta_{old} + \beta d$$

# 近端策略优化算法 (PPO)

策略梯度的目标函数：

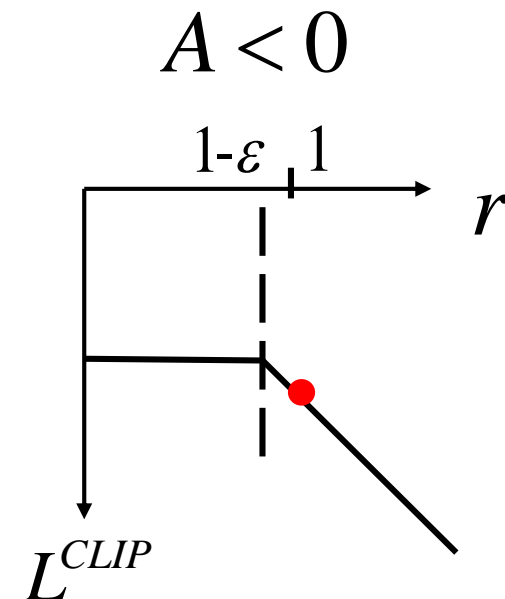
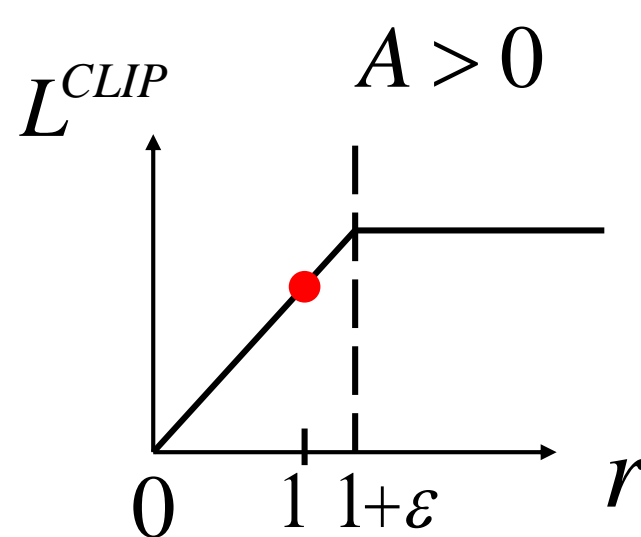
$$L^{PG}(\theta) = E_t \left[ \log \pi_{\theta}(a_t | s_t) \hat{A}_t \right]$$

TRPO目标函数：

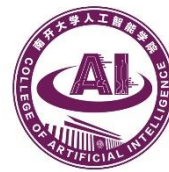
$$\begin{aligned} & \underset{\theta}{\text{maximize}} E_{s \sim \pi_{\theta_{old}}, a \sim \pi_{\theta}} \left[ \frac{\pi_{\theta}(a|s)}{\pi_{\theta_{old}}(a|s)} A_{\theta_{old}}(s, a) \right] \\ & \text{subject to } E_{s \sim \pi_{\theta_{old}}} [D_{KL}(\pi_{\theta_{old}}(\cdot|s) || \pi_{\theta}(\cdot|s))] \leq \delta \end{aligned}$$

CLIP目标函数：

$$L^{CLIP}(\theta) = E_t [\min(r_t(\theta) \hat{A}_t, \text{clip}(r_t(\theta), 1 - \varepsilon, 1 + \varepsilon) \hat{A}_t)]$$



优化更低的界



# 近端策略优化算法 (PP0)

PP0目标函数：

$$L_t^{CLIP+VF+S}(\theta) = \hat{E}_t[L_t^{CLIP}(\theta) - c_1 L_t^{VF}(\theta) + c_2 S[\pi_\theta](s_t)]$$

其中：  $V_t^{VF}$  为值函数损失函数，S为熵。

为减小方差，优势函数可写为：

$$\hat{A}_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1}$$

其中：  $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$



# 近端策略优化算法 (PPO)

---

## Algorithm 1 PPO, Actor-Critic Style

---

```
for iteration=1, 2, ... do
  for actor=1, 2, ..., N do
    Run policy  $\pi_{\theta_{\text{old}}}$  in environment for  $T$  timesteps
    Compute advantage estimates  $\hat{A}_1, \dots, \hat{A}_T$ 
  end for
  Optimize surrogate  $L$  wrt  $\theta$ , with  $K$  epochs and minibatch size  $M \leq NT$ 
   $\theta_{\text{old}} \leftarrow \theta$ 
end for
```

---

PPO伪代码

# TRPO用到的信息论

熵的概念：熵是信息多少的度量

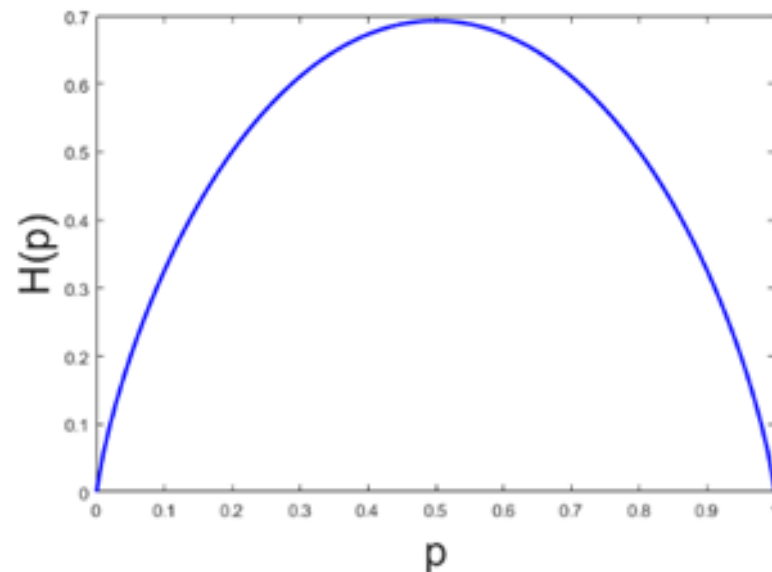
离散系统：  $H(X) = - \sum_i p_i \log p_i$

连续系统：  $H(x) = E_{x \sim P}[I(x)] = - E_{x \sim P}[\log P(x)]$

熵是不确定性的度量：不确定度越大，熵越大

二值熵定义：

$$H = -p \log(p) - (1-p) \log(1-p)$$





# TRPO用到的信息论

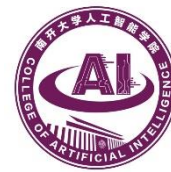
交叉熵:

$$H(P, Q) = -E_{P(x)}Q(x) = -\int P(x)\log Q(x)dx$$

交叉熵用来衡量编码方案不一定完美时, 平均编码的长度

KL散度: 衡量两个概率分布之间的距离

$$D_{KL}(P||Q) = E_{x \sim P}\left[\log \frac{P(x)}{Q(x)}\right] = \int P(x)\log P(x)dx - \int P(x)\log Q(x)dx$$



# 第七次作业

---

1. 阅读 Trust Region Policy Optimization 和 Proximal Policy Optimization Algorithms
2. 阅读代码