

# 步步深入TRPO

论文《Trust Region Policy Optimization》[1]提出了鼎鼎大名的 TRPO 算法，这是 policy gradient 系列强化学习（RL）算法的里程碑之作，但原论文包含大量晦涩难懂的公式和定理，对于入门者并不友好。本文将详细讲解 TRPO 中关键公式的推导过程，希望能够理清 TRPO 作者想解决的问题以及采用的方法。

## 引言

先讲一下 TRPO 的**优化目标**，TRPO 和大多数 RL 算法一样，希望提升策略  $\pi$  的**期望累积回报**  $\eta(\pi)$ ：

$$\eta(\pi) = E_{s_0, a_0, \dots} \left[ \sum_{t=0}^{\infty} \gamma^t r_t \right]$$

其中，每一步的动作  $a_t \sim \pi(a_t | s_t)$ ，服从策略  $\pi$  所决定的动作概率分布。

在进一步分析  $\eta(\pi)$  的性质之前，我们需要定义三个函数，动作值函数  $Q(s_t, a_t)$ ，状态值函数  $V(s_t)$  和优势函数  $A(s_t, a_t)$ 。

$Q(s_t, a_t) = E_{s_{t+1}, a_{t+1}, \dots} \left[ \sum_{l=0}^{\infty} \gamma^l r^{t+l} \right]$ ，即在状态  $s_t$  下采用动作  $a_t$  后，后续动作服从策略  $\pi$  的情况下的累积期望回报。

$V(s_t) = E_{a_t, s_{t+1}, a_{t+1}, \dots} \left[ \sum_{l=0}^{\infty} \gamma^l r^{t+l} \right]$ ，即在状态  $s_t$  下，后续动作服从策略  $\pi$  的情况下的累积期望回报。

$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$ ，表示在状态  $s_t$  下，直接采用动作  $a_t$  相比于按照  $a_t \sim \pi(a_t | s_t)$  采样动作的优势。

如何提升  $\eta(\pi)$  呢，或者换个问题，**如何找到一个新的策略  $\tilde{\pi}$  使得  $\eta(\tilde{\pi})$  高于  $\eta(\pi)$  呢**？这就需要分析  $\eta(\tilde{\pi})$  和  $\eta(\pi)$  的定量关系了。这里作者引用了一个 RL 领域的经典结论[2]：

$$\eta(\tilde{\pi}) = \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t), \text{ Eq.1}$$

这里的  $A_{\pi}(s_t, a_t)$  表示策略  $\pi$  下的优势函数  $A(s_t, a_t)$ ，也就是说

$A_{\pi}(s_t, a_t) = Q_{\pi}(s_t, a_t) - V_{\pi}(s_t)$ ，其中动作价值函数和状态价值函数对应了策略  $\pi$ 。

这个式子可以这么理解， $\eta(\tilde{\pi})$  与  $\eta(\pi)$  的差等于  $E_{s_0, a_0, \dots \sim \tilde{\pi}} \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t)$ ，它的含义是：按照策略  $\tilde{\pi}$  采样动作  $a_t$ ，在走出的轨迹中，每一步的策略  $\pi$  下的优势函数  $A_{\pi}(s_t, a_t)$  的累积和。

这里给大家简单证明一下：

$$\begin{aligned} \eta(\tilde{\pi}) &= E_{s_0, a_0, \dots \sim \tilde{\pi}} \sum_{t=0}^{\infty} \gamma^t r^t \\ &= \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \sum_{t=0}^{\infty} \gamma^t r^t - \eta(\pi) \\ &= \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t r^t - V_{\pi}(s_0) \right] \\ &= \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ r^0 + \gamma V_{\pi}(s_1) - V_{\pi}(s_0) + \gamma \left( \sum_{t=0}^{\infty} \gamma^t r^{t+1} - V_{\pi}(s_1) \right) \right] \\ &= \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ r^0 + \gamma V_{\pi}(s_1) - V_{\pi}(s_0) + \gamma (r^1 + \gamma V_{\pi}(s_2) - V_{\pi}(s_1)) + \gamma^2 \left( \sum_{t=0}^{\infty} \gamma^t r^{t+2} - V_{\pi}(s_2) \right) \right] \\ &= \dots\dots\dots \\ &= \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t (r^t + \gamma V_{\pi}(s_{t+1}) - V_{\pi}(s_t)) \right] \\ &= \eta(\pi) + E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right] \end{aligned}$$

注意论文原文的证明也是一样的裂项相消，区别只是这里作者写成了连等式的过程，就像川菜的一锅成菜，一个等号从头连到尾。

实际上这个等式给了我们很强的指引：即满足  $E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right] \geq 0$  的策略  $\tilde{\pi}$  可以使得  $\eta(\tilde{\pi}) \geq \eta(\pi)$ 。但是策略  $\tilde{\pi}$  并没有**显式出现**在式中，我们并不清楚

$$E_{s_0, a_0, \dots \sim \tilde{\pi}} \left[ \sum_{t=0}^{\infty} \gamma^t A_{\pi}(s_t, a_t) \right]$$

和策略  $\tilde{\pi}$  的具体关系，因此接下来需要变形。

首先定义累积频率函数  $\rho_{\pi}(s)$ ：

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

这个函数的含义是：在策略  $\pi$  下，每一步状态  $s_t$  等于  $s$  的概率在折扣系数下的累积和。基于  $\rho_{\pi}(s)$  的定义，我们对 Eq.1 变形得到 Eq.2:

$$\begin{aligned} \eta(\tilde{\pi}) &= \eta(\pi) + \sum_{t=0}^{\infty} \sum_s P(s_t = s | \tilde{\pi}) \sum_a \tilde{\pi}(a | s) \gamma^t A_{\pi}(s, a) \\ &= \eta(\pi) + \sum_s \sum_{t=0}^{\infty} \gamma^t P(s_t = s | \tilde{\pi}) \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) \quad \text{Eq.2} \\ &= \eta(\pi) + \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) \end{aligned}$$

通过Eq.2不难看出，只要新策略  $\tilde{\pi}$  满足：

对于每个状态  $s$ ， $\sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) \geq 0$ ，即可保证  $\eta(\tilde{\pi}) \geq \eta(\pi)$ 。

很完美对不对，只要最大化  $\sum_a \tilde{\pi}(a | s) A_{\pi}(s, a)$  就可以得到更优的策略  $\tilde{\pi}$ ，看起来问题到这里就熟了。其实不然，如果在这里完美画上句号，就没 TRPO 什么事了。下面，真正的 TRPO 即将开始。

## 前菜

在 RL 算法的实际应用中，我们通常通过神经网络来学习一个策略  $\pi$ ，即输入状态  $s$ ，输出这个状态下选择每个动作  $a$  的概率  $\pi(a | s)$ 。既然是参数化的神经网络，就难免有误差，换句话说，“对于每

个状态  $s$ ， $\sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) \geq 0$ ” 这个完美条件难以成立，总有一些隐藏的坏点状态  $s$ ，使得  $\sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) < 0$ 。

怎么办？其实也好办也不好办。

好办的是，根据Eq.2， $\eta(\tilde{\pi}) - \eta(\pi) = \sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a)$ ，那么只要让  $\sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a)$  整体  $\geq 0$  就行了，中间每个状态上的  $\sum_a \tilde{\pi}(a | s) A_{\pi}(s, a)$  是正是负我们并不需要考虑。

不好办的是，想求  $\sum_s \rho_{\tilde{\pi}}(s) \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a)$  对策略  $\tilde{\pi}$  的导数，是万分困难的，因为  $\rho_{\tilde{\pi}}(s)$  的导数我们搞不到。

## 正篇1：替代函数

这该怎么办？把难搞的东西给 ban 掉，换成相应的替代。一个很自然的想法就是把  $\rho_{\tilde{\pi}}(s)$  换成  $\rho_{\pi}(s)$ ，于是，我们定义一个近似的替代函数：

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

$L_{\pi}(\tilde{\pi})$  比  $\eta(\tilde{\pi})$  要好处理多了， $\rho_{\pi}(s)$  中不包含策略  $\tilde{\pi}$ ，因此对策略  $\tilde{\pi}$  的导数为零。但是， $L_{\pi}(\tilde{\pi})$  既然是近似，必然有误差。那么问题来了，这种近似的效果如何呢？

效果还真不错，观察发现， $L_{\pi}(\tilde{\pi})$  和  $\eta(\tilde{\pi})$  在  $\tilde{\pi} = \pi$  处的值和对  $\tilde{\pi}$  的导数都是一样的，也就是：

$$L_{\pi}(\pi) = \eta(\pi)，且 \nabla_{\tilde{\pi}} L_{\pi}(\tilde{\pi})|_{\tilde{\pi}=\pi} = \nabla_{\tilde{\pi}} \eta(\tilde{\pi})|_{\tilde{\pi}=\pi}$$

第一个等式（值相等）一眼就可以看出来，第二个需要稍微证明一下，具体如下：

$$\nabla_{\tilde{\pi}} \eta(\tilde{\pi})|_{\tilde{\pi}=\pi} = \sum_s \rho_{\pi}(s) \nabla_{\tilde{\pi}} \left| \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) + \sum_s \nabla_{\tilde{\pi}} \left| \rho_{\tilde{\pi}}(s) \right. \right|_{\tilde{\pi}=\pi} \sum_a \pi(a | s) A_{\pi}(s, a)$$

注意，(突然冒出来的)  $\sum_a \pi(a | s) A_{\pi}(s, a) = 0$ ，代进去可得：

$$\begin{aligned} \nabla_{\tilde{\pi}} \eta(\tilde{\pi})|_{\tilde{\pi}=\pi} &= \sum_s \rho_{\pi}(s) \nabla_{\tilde{\pi}} \left| \sum_a \tilde{\pi}(a | s) A_{\pi}(s, a) + 0 \right|_{\tilde{\pi}=\pi} \\ &= \nabla_{\tilde{\pi}} L_{\pi}(\tilde{\pi})|_{\tilde{\pi}=\pi} \end{aligned}$$

这种在  $\tilde{\pi} = \pi$  处的值和梯度都相等的情况，叫做  $L_{\pi}(\tilde{\pi})$  是对  $\eta(\tilde{\pi})$  一阶近似。结合数学分析的知识，我们可以知道，这个性质说明  $\tilde{\pi} = \pi$  处必然存在一个邻域，域内的  $\tilde{\pi}$  满足：若  $L_{\pi}(\tilde{\pi})$  增大，则  $\eta(\tilde{\pi})$  也增大。这说明，在一定步长内优化  $L_{\pi}(\tilde{\pi})$ ，会使得  $\eta(\tilde{\pi})$  也得到优化。

## 正篇2：信赖域

仅凭替代函数对优化函数的一阶近似的性质，我们只能知道，在一定步长内提升  $L_{\pi}(\tilde{\pi})$ ，会使得  $\eta(\tilde{\pi})$  也提升。但我们还不知道**步长要选多大**。这时候我们回想一下文章标题，发现有一个关键词——trust region（信赖域），其实原文探讨的核心就是优化的步长要在什么范围（域）内选择，也就是这个信赖域。

沿着这个思路出发，文章的核心贡献点之一就是进一步量化了  $L_{\pi}(\tilde{\pi})$  和  $\eta(\tilde{\pi})$  的关系，也就是提出了下面这个不等式：

$$\begin{aligned} \eta(\tilde{\pi}) &\geq L_{\pi}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2} \alpha \\ \text{where } \alpha &= \max_s D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s)), \epsilon = \max_{s,a} |A_{\pi}(s, a)| \end{aligned}$$

这一步基本解决了步长的问题。因为我们得到了  $\eta(\tilde{\pi})$  和  $L_{\pi}(\tilde{\pi})$  以及步长（这里表现为  $\tilde{\pi}$  与  $\pi$  之间的KL散度）的定量关系，这个关系表现为  $\frac{4\epsilon\gamma}{(1-\gamma)^2} \alpha$  这个惩罚项，步长越大，惩罚就越大，此时  $\eta(\tilde{\pi})$  越难以享受到提升  $L_{\pi}(\tilde{\pi})$  所带来的优化效果。

有了定量关系就好办了，我们可以直接把优化目标从  $L_\pi(\tilde{\pi})$  改为：

$$M_\pi(\tilde{\pi}) = L_\pi(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha$$

注意到  $M_\pi(\tilde{\pi})$  是  $\eta(\tilde{\pi})$  的下界，我们希望优化  $M_\pi(\tilde{\pi})$  来提升  $\eta(\tilde{\pi})$ ，这里证明一下，优化  $M_\pi(\tilde{\pi})$  得到的最优解  $\bar{\pi}$  一定是更好的策略，即  $\eta(\bar{\pi}) \geq \eta(\pi)$ 。

首先注意到两个事实：

1.  $M_\pi(\pi) = L_\pi(\pi) - 0 = \eta(\pi) - 0 = \eta(\pi)$
2.  $M_\pi(\bar{\pi}) \geq M_\pi(\pi)$

第一个由于  $\tilde{\pi} = \pi$  时惩罚项为零，第二个则利用了  $\bar{\pi}$  为最优解这个特性。那么接下来证明就一目了然：

$$\eta(\bar{\pi}) \geq M_\pi(\bar{\pi}) \geq M_\pi(\pi) = \eta(\pi)。$$

到此，我们证明了，直接优化  $M_\pi(\tilde{\pi})$  就可以得到更优的策略。

## 正篇3：优化

$M_\pi(\tilde{\pi})$  的优化其实是需要单独抽出来讲一下，它面临两个难点：

1.  $\max_s D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s))$  难以计算。
2.  $\max_{s,a} |A_\pi(s, a)|$  也难以计算。

对于第一个难点，TRPO 作者们将  $\max_s D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s))$  用  $E_{s \sim \rho_\pi} D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s))$  进行了近似替换。因为  $s \sim \rho_\pi$  这个分布上的期望是可以用**蒙特卡洛法**解决的。

第二个难点其实是惩罚项系数的问题，这个系数中的  $\max_{s,a} |A_\pi(s, a)|$  也是个难确定的东西。所以 TRPO 作者们直接就把最大化  $M_\pi(\tilde{\pi})$  换成了它的**对偶问题**：

也就是从：

$$\text{maximize } M_{\pi}(\tilde{\pi}) = L_{\pi}(\tilde{\pi}) - \frac{4\epsilon\gamma}{(1-\gamma)^2}\alpha$$

变成：

$$\begin{aligned} & \text{maximize } L_{\pi}(\tilde{\pi}) \\ & \text{subject to } E_{s \sim \rho_{\pi}} D_{\text{KL}}(\pi(\cdot | s) \| \tilde{\pi}(\cdot | s)) \leq \delta \end{aligned}$$

这是一种很好的偷懒方式，不管  $\max_{s,a} |A_{\pi}(s,a)|$  是多大，总存在对应的  $\delta$ ，使得对偶问题和原问题完全一致。那么  $\delta$  具体取多大？那就是调参的事了，哪个值性能好用哪个。

实际训练时，策略是用参数化的网络来实现的，我们用  $\theta$  和  $\tilde{\theta}$  来表示更新前后策略  $\pi$  和  $\tilde{\pi}$  的参数，因此原优化目标可以表示为  $L_{\theta}(\tilde{\theta})$ 。文章中优化  $L_{\theta}(\tilde{\theta})$  时，采用了通过 Fisher information matrix 来计算自然梯度（natural gradient）的方法，具体建模为：

$$\begin{aligned} & \text{maximize } L_{\theta}(\tilde{\theta}) \\ & \text{subject to } \frac{1}{2}(\tilde{\theta} - \theta)^T A(\tilde{\theta}, \theta)(\tilde{\theta} - \theta) \leq \delta \end{aligned}$$

其中， $A(\tilde{\theta}, \theta)$  是 KL divergence 关于  $\tilde{\theta}$  的 Hessian 矩阵，也就是 Fisher information matrix。求解这个问题可以直接套用自然梯度的公式，令  $g = \nabla_{\theta} L_{\theta}(\tilde{\theta})$ ，则自然梯度的方向为  $A^{-1}g$ ，考虑到  $\delta$  对步长的约束，实际更新使用的自然梯度为  $\sqrt{\frac{2\delta}{g^T A^{-1}g}} A^{-1}g$ 。

1. Schulman, J., Levine, S., Abbeel, P., Jordan, M. and Moritz, P., 2015, June. Trust region policy optimization. In *International conference on machine learning* (pp. 1889-1897). PMLR.
2. Kakade, S. and Langford, J., 2002. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*.