为什么 A2C 中减去 baseline 函数可以减小方差

在 PPO x Family 课程第一讲中，已经介绍了策略梯度（policy gradient）的基本原理，但是如果直接使用最朴素的策略梯度方法，会发现实际训练中梯度的方差很大，进而导致训练的效果不佳。为了解决这一问题，我们需要引入基线（baseline）函数来尽可能减小梯度的方差，进而提升算法的性能。

首先我们先来回顾一下 policy gradient 中标准的优化公式：

$$\nabla_\theta R_\theta = \mathbb{E}_\tau [\sum_{t=1}^{T} G_t(\tau)\nabla_\theta \log p_\theta(a_t|s_t)]$$

$$\Delta\theta = \eta\nabla_\theta R_\theta$$

在算法实践中，我们使用采样来获得 $\nabla_\theta R_\theta$ 的**无偏估计** $\nabla_\theta \bar{R}_\theta$：

$$\nabla_\theta \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} G_t(\tau^n)\nabla_\theta \log p_\theta(a_t^n|s_t^n)$$

可以看出，上式将期望改成了对N组轨迹进行采样和平均的形式。

在实践中，网络的训练会因为 $\nabla_\theta \bar{R}_\theta$ 的方差而出现不稳定，即虽然均值相同，但每次采样数据的 $\nabla_\theta \bar{R}_\theta$ 与真实的期望值有较大偏离，而环境本身奖励函数的随机性也会进一步加重这个问题，因此我们希望降低上述优化形式带来的梯度方差。

Baseline 函数就是为了这一点而提出的方法，具体来说，我们将上述公式改写为包含 baseline 的版本：

$$\nabla \bar{R}_\theta = \frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_n} (G_t(\tau^n) - b(s_t^n))\nabla \log p_\theta(a_t^n|s_t^n)$$

其中 $b(s_t)$ 是关于 t 时刻状态的一个函数，被称作基线（baseline）函数。这里要注意：其实我们没有限制 baseline 函数必须是某一特定的形式，只要它是 s_t 的函数就可以保证更新公式的正确性。但是一般我们认为 $b(s_t) = V(s_t)$ 时是最优的，因为此时能使得 $\nabla \bar{R}_\theta$ 的方差是最小的。

接下来我们要论证：

1. 为什么添加了这一项 baseline 函数之后，仍然能够保持估计的无偏性；
2. 为什么添加了这一项 baseline 函数之后，可以减小**方差**？

为什么添加 baseline 函数之后仍然能够保持梯度估计的无偏性

要证明这一点，本质是要证明：

$$\mathbb{E}_\tau [\sum_{t=1}^{T} G_t(\tau)\nabla \log p_\theta(a_t|s_t)] = \mathbb{E}_\tau [\sum_{t=1}^{T} (G_t(\tau) - b(s_t))\nabla \log p_\theta(a_t|s_t)]$$

移项化简，即证：

$$\mathbb{E}_{\tau}[\sum_{t=1}^{T} b(s_t)\nabla\log p_{\theta}(a_t|s_t)] = 0$$

考虑到对于任何一个 t 时刻的 transition，都有：

$$\begin{aligned}\mathbb{E}_{a_t}[b(s_t)\nabla_{\theta}\log\ p_{\theta}(a_t|s_t)] &= \int \frac{\nabla_{\theta}\ p_{\theta}(a_t|s_t)}{p_{\theta}(a_t|s_t)} p_{\theta}(a_t|s_t) b(s_t) da_t \\ &= b(s_t)\nabla_{\theta}\int p_{\theta}(a_t|s_t) da_t \\ &= b(s_t)\nabla_{\theta} 1 \\ &= 0\end{aligned}$$

因此对于整体的轨迹而言，自然对每个时刻 t 都成立，因此也就有：

$$\mathbb{E}_{\tau}[\sum_{t=1}^{T} b(s_t)\nabla\log p_{\theta}(a_t|s_t)] = 0$$

至此，我们就证明了添加 baseline 函数之后，梯度的无偏性仍然能够得以保持。

为什么添加 baseline 函数之后，梯度估计的方差会降低

在这一部分，我们将详细介绍添加 baseline 函数为何能减小方差。其实严格来说，并不是添加任何一种形式的 baseline 函数都能够减小方差。相反，不合适的 baseline 函数甚至会增大方差。因此我们在这部分要说明的其实是：

- 为什么添加**合适的** baseline 函数可以减小方差
- 为什么我们通常选用 $b(s_t) = V(s_t)$ 作为实践中的 baseline 函数。

首先，根据方差的计算公式：

$$Var(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$$

我们希望在添加了 baseline 函数之后，尽可能减小方差，即找到最优的 $b(s_t)$ ，最大化下面的优化目标，即：

$$\max_{b(.)}\ Var(\sum_{t=1}^{T} G_t(\tau)\nabla\log p_{\theta}(a_t|s_t)) - Var(\sum_{t=1}^{T}(G_t(\tau) - b(s_t))\nabla\log p_{\theta}(a_t|s_t))$$

代入方差的计算式化简，可以得到等价的优化目标：

$$\min_{b(.)}\ \mathbb{E}_{\tau}[(\sum_{t=1}^{T}(G_t(\tau) - b(s_t))\nabla\log p_{\theta}(a_t|s_t))^2]$$

但是遗憾的是，这个优化目标并不容易被直接求解，因为这里的 $\nabla\log p_{\theta}(a_t|s_t)$ 是前一项 $G_t(\tau) - b(s_t)$ 的加权项。于是这里利用两个假设：

1. 每个 t 时刻之间独立；
2. 对于每个 t 时刻， $G_t(\tau) - b(s_t)$ 和 $\nabla\log p_{\theta}(a_t|s_t)$ 独立。

那么，原优化问题即可转化为：

$$\min_{b(.)} \mathbb{E}_\tau\left[(G_i(\tau)-b(s_i))^T((\nabla \log p_\theta(a_i|s_i))(\nabla \log p_\theta(a_j|s_j))(G_i(\tau)-b(s_i))\right]$$

对上式求关于 b 的导数，并求导数的零点，可得：

$$b(s_i)=\mathbb{E}_\tau\left[G_i(\tau)^T(\nabla \log p_\theta(a_i|s_i)\nabla \log p_\theta(a_j|s_j))\right]\{\mathbb{E}_\tau\left[(\nabla \log p_\theta(a_i|s_i)\nabla \log p_\theta(a_j|s_j))\right]\}^{-1}$$

一般来说，$(\nabla \log p_\theta(a_i|s_i)\nabla \log p_\theta(a_j|s_j))$ 这个随机矩阵和 $G_i(\tau)$ 这个随机向量在给定了某个具体环境和策略时，它们是相关的。但对于广义上的各种环境和策略的随机组合而言，我们可以近似假定两者无关。**这样一来它们乘积的期望，就可以转化为它们期望的乘积**。此时，上述形式可进一步化简为：

$$b(s_i)=\mathbb{E}_\tau\left[G_i(\tau)^T\right]\mathbb{E}_\tau\left[(\nabla \log p_\theta(a_i|s_i)\nabla \log p_\theta(a_j|s_j))\right]\mathbb{E}_\tau\left[(\nabla \log p_\theta(a_i|s_i)\nabla \log p_\theta(a_j|s_j))\right]^{-1}$$

$$=\mathbb{E}_\tau\left[G_i(\tau)^T\right]$$

即最优的 baseline 函数可以写为：

$$b^*(s_t)=\mathbb{E}_\tau[G_t]=V(s_t)$$

此时我们发现，使得方差最小的最优 baseline 函数，恰巧就是价值函数。

至此，我们已经完成了证明，只要选取合适的 baseline 函数，就可以减小对梯度的估计方差；同时，在假定 $G_t(\tau)-b(s_t)$ 和 $\nabla \log p_\theta(a_t|s_t)$ 是相互独立的前提下，最优的 baseline 函数就是价值函数 $V(s_t)$ ，这也是我们在实践中一般使用价值函数作为 baseline 函数的理论依据所在。不过值得注意的是，如果在不满足上述假设的一些决策环境里，这种方法的效果可能会大打折扣。