Reinforcement Learning: An Introduction notebook

黎雷蕾

2017 年 12 月 22 日

目录

11 Off-policy Methods with Approximation **2**

11.1 Semi-gradient Methods 2

11.2 Examples of Off-policy Divergence 3

11.3 The Deadly Triad 4

11.4 Linear Value-function Geometry 5

Chapter 11

Off-policy Methods with Approximation

11.1 Semi-gradient Methods

在函数值近似中，每步重要性采样比例公式可以写为：

$$\rho_t \doteq \rho_{t:t} = \frac{\pi(A_t|S_t)}{b(A_t|S_t)} \tag{11.1}$$

比如说，semi-gradient off-policy TD(0) 算法的权重向量更新可以写为：

$$\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\rho_t\delta_t\nabla\hat{v}(S_t, \mathbf{w}) \tag{11.2}$$

其中 δ_t 是采用平均回报的 TD error：

$$\begin{aligned} \delta_t &\doteq R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t), \text{or} \\ \delta_t &\doteq R_{t+1} - \overline{R}_t + \hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t) \end{aligned} \tag{11.3}$$

对于动作-价值来说，上面的算法就变成了 one-step semi-gradient Expected Sarsa：

$$\begin{aligned} &\mathbf{w}_{t+1} \doteq \mathbf{w}_t + \alpha\rho_t\delta_t\nabla\hat{q}(S_t, A_t, \mathbf{w}_t), \text{with} \\ &\delta_t \doteq R_{t+1} + \gamma\sum_a \pi(a|S_t+1)\hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t), \text{or} \\ &\delta_t \doteq R_{t+1} - \overline{R}_t + \sum_a \pi(a|S_{t+1})\hat{q}(S_{t+1}, a, \mathbf{w}_t) - \hat{q}(S_t, A_t, \mathbf{w}_t) \end{aligned} \tag{11.4}$$

同理，n-step semi-gradient Expected Sarsa 可以写为：

$$\mathbf{w}_{t+n} \doteq \mathbf{w}_{t+n-1} + \alpha\rho_{t+1}\cdots\rho_{t+n-1}\left[G_{t:t+n} - \hat{q}(S_t, A_t, \mathbf{w}_{t+n-1})\right]\nabla\hat{q}(S_t, A_t, \mathbf{w}_{t+n-1}),$$

with

$$\begin{aligned} G_{t:t+n} &\doteq R_{t+1} + \cdots + \gamma^{n-1}R_{t+n} + \gamma^n\hat{q}(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}), \text{or} \\ G_{t:t+n} &\doteq R_{t+1} - \overline{R}_t + \cdots + R_{t+n} - \overline{R}_{t+n-1} + \hat{q}(S_{t+n}, A_{t+n}, \mathbf{w}_{t+n-1}) \end{aligned} \tag{11.5}$$

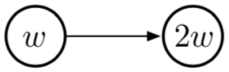
n-step semi-gradient tree-backup 算法可以写为：

$$\begin{aligned} &\mathbf{w}_{t+n} \doteq \mathbf{w}_{t+n-1} + \alpha[G_{t:t+n} - \hat{q}(S_t, A_t, \mathbf{w}_{t+n-1})]\nabla\hat{q}(S_t, A_t, \mathbf{w}_{t+n-1}), \text{with} \\ &G_{t:t+n} \doteq \hat{q}(S_t, A_t, \mathbf{w}_{t-1}) + \sum_{k=t}^{t+n-1}\delta_k\prod_{i=t+1}^{k}\gamma\pi(A_i|S_i) \end{aligned} \tag{11.6}$$

11.2 Examples of Off-policy Divergence

在近似值逼近中会遇到一个困难：行为策略的分布和目标策略的分布不一致 (the distribution of updates does not match the on-policy distribution)。

在下图中，有两个状态，一个动作，从左边状态到右边状态的 reward 为 0：



两个状态进行转换时产生的 TD error:

$$\begin{aligned} \delta_t &= R_{t+1} + \gamma\hat{v}(S_{t+1}, \mathbf{w}_t) - \hat{v}(S_t, \mathbf{w}_t) \\ &= 0 + \gamma 2w_t - w_t \\ &= (2\gamma - 1)w_t \end{aligned} \tag{11.7}$$

off-policy semi-gradient TD(0) 更新公式：

$$\begin{aligned} w_{t+1} &= w_t + \alpha\rho_t\delta_t\nabla\hat{v}(S_t, w_t) \\ &= w_t + \alpha\cdot 1\cdot(2\gamma - 1)w_t\cdot 1 \\ &= (1 + \alpha(2\gamma - 1))w_t \end{aligned} \tag{11.8}$$

其中重要性采样比率 $\rho_t = 1$。

这个例子的关键是，对于一个转化 (行为策略)，它重复发生时 w 在目标策略上没有发生更新。发生这样的原因是由于行为策略可能采样时会选择目标策略可能永远不会选择的行为。这说明了目标策略和行为策略会造成差异。

为了减小这种差异，有两种途径：

- 采用重要性采样，来平衡行为策略之中的权重。
- 采用一个不依赖于样本的随机梯度下降策略。

11.3 The Deadly Triad

我们结合下面三个要素容易造成系统的不稳定与差距，所以我们称之为 (The Deadly Triad):

- **Function approximation**: 一种强大、可扩展的方法，使用大量的空间和计算资源来生成状态空间，（如线性函数近似、人工神经网络）；
- **Bootstrapping**: 更新包括现有估计的目标时完全依靠实际回报，和完全的回报。(如 DP 算法和 TD 算法)
- **Off-policy training**: 离策略指的是关于目标策略之外的另外的训练，如 DP 算法一样，扫描所有的状态空间并更新所有的状态，这个行为不需要遵循目标策略。

这个误差是无法避免的，至于解决方法到目前为止也没有太好的办法，只能是采用上节所提到的途径:

- 采用重要性采样，来平衡行为策略之中的权重。
- 采用一个不依赖于样本的随机梯度下降策略。

11.4 Linear Value-function Geometry

我们要注意的是，在绝大多数情况下，大多数价值函数并不符合任何的策略。对我们而言，更多的并不是需要通过函数值进行逼近，而是通过设计比状态量更少的参数 (More important for our purposes is that most are not representable by the function approximator, which by design has far fewer parameters than there are states.)??

我们给出三个状态 $\mathcal{S} = \{s_1, s_2, s_3\}$ 和两个参数 $\mathbf{w} = (w_1, w_2)^T$。在一个三维空间中，对于任意一组数 (x, y)，我们设 $w_1 = x, w_2 = y$ 那么可以得到下图：

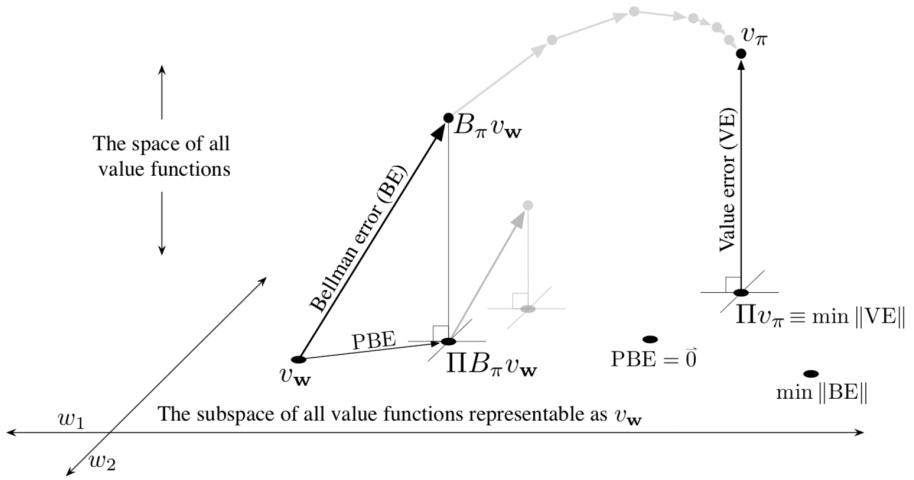


Figure 11.3: The geometry of linear value-function approximation. Shown as a plane is the subspace of all functions representable by the function approximator. The three-dimensional space above and below it is the much larger space of all value functions (functions from $\mathcal{S}$ to $\mathbb{R}$). The true value function v_π is in this larger space and projects down to its best approximation in the value error (VE) sense. The best approximators in the Bellman error (BE) and projected Bellman error (PBE) senses are different and are also shown in the lower right. (VE, BE, and PBE are all treated as the corresponding vectors in this figure.) The Bellman operator takes a value function in the plane to one outside, which can then be projected back. If you could iteratively apply the Bellman operator outside the space (shown in gray above) you would reach the true value function, as in conventional DP.

假设一个固定的额策略 π，假设它的真实状态值函数是 v_π。为了度量两个不同的价值函数 (目标策略和行为策略)，有 $v = v_1 - v_2$。我们可以用权

重值 $\mu : \mathcal{S} \to \mathbb{R}$ 权衡我们关心的不同状态的精确量化。综上，我们可以定义如下公式用来衡量状态值函数之间的差异：

$$\|v\|_\mu^2 \doteq \sum_{s \in \mathcal{S}} \mu(s) v(s)^2 \tag{11.9}$$

引入均方误差 (MSVE) 有:

$$\text{MSVE}(\mathbf{w}) = \|v_\mathbf{w} - v_\pi\|_\mu^2 \tag{11.10}$$

我们可以定义一个操作 Π 来表示最接近我们要求的状态值函数:

$$\begin{aligned} \Pi v &\doteq v_\mathbf{w} \\ &\text{where,} \\ \mathbf{w} &= \arg\min_\mathbf{w} \|v - v_\mathbf{w}\|_\mu^2 \end{aligned} \tag{11.11}$$

最接近真实值函数 v_π 是 Π 的投影，如下面所示：

The projection matrix

对于一个线性函数逼近，它的投影也是线性的，可以写为一个 $|\mathcal{S}| \times |\mathcal{S}|$ 的矩阵：

$$\Pi \doteq \mathbf{X} \left(\mathbf{X}^T D \mathbf{X} \right)^{-1} \mathbf{X}^T D \tag{11.12}$$

其中:

$$D \doteq \begin{bmatrix} \mu(1) & & & 0 \\ & \mu(2) & & \\ & & \ddots & \\ 0 & & & \mu(|\mathcal{S}|) \end{bmatrix} \tag{11.13}$$

$$\mathbf{X} \doteq \begin{bmatrix} -\mathbf{x}(1)^T - \\ -\mathbf{x}(2)^T - \\ \vdots \\ -\mathbf{x}(|\mathcal{S}|)^T - \end{bmatrix} \tag{11.14}$$

若上述存在广义逆矩阵，那么标准向量可以写为:

$$\|v\|_\mu^2 = v^T D v \tag{11.15}$$

线性逼近价值函数可以重写为:

$$v_{\mathbf{w}} = \mathbf{X}\mathbf{w} \tag{11.16}$$