第二章 概率论基础与蒙特卡洛

本章首先复习一些概率论基础知识，然后介绍蒙特卡洛。蒙特卡洛是一类随机算法的总称。蒙特卡洛是理解很多数强化学习算法的关键。请读者耐心读完本章内容，不要跳过。

2.1 概率论基础

强化学习中会经常用到两个概念：**随机变量**、**观测值**。随机变量是一个不确定量，它的值取决于一个随机事件的结果。比如抛一枚硬币，正面朝上记为 0，反面朝上记为 1。抛硬币是个随机事件，抛硬币的结果记为随机变量 X，用大写字母表示。随机变量 X 有两种可能的取值：可能是 0，也可能是 1。抛硬币之前，X 是未知的，而且带有随机性。抛硬币之后，我们会观测到硬币哪一面朝上，此时随机变量 X 就有了观测值，记作 x。举个例子，如果重复抛硬币 4 次，得到了 4 个观测值：

$$x_1 = 1, \quad x_2 = 1, \quad x_3 = 0, \quad x_4 = 1.$$

这四个观测值只是数字而已，没有随机性。本书用大写字母表示随机变量，小写字母表示观测值，避免造成混淆。

强化学习会反复用到**概率质量函数** (Probability Mass Function, PMF) 或**概率密度函数** (Probability Density Function, PDF)，意思是随机变量 X 在确定的取值点 x 附近的可能性。

- 概率质量函数 (PMF) 描述一个**离散概率分布**——即变量的取值范围 $\mathcal{X}$ 是个离散的集合。在抛硬币的例子中，随机变量 X 的取值范围是集合 $\mathcal{X} = \{0, 1\}$。X 的概率质量函数是

  $$p(0) = 0.5, \quad p(1) = 0.5.$$

  公式的意思随机变量取值 0 和 1 的概率都是 0.5。见图 2.1(左) 的例子。概率质量函数有这样的性质：

  $$\sum_{x \in \mathcal{X}} p(x) = 1.$$

- 当考虑连续随机变量，我们用概率密度函数 (PDF)。正态分布是最常见的一种**连续概率分布**。随机变量 X 的取值范围是所有实数 $\mathbb{R}$。X 的概率密度函数是

  $$p(x) = \frac{1}{\sqrt{2\pi}\,\sigma} \cdot \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right).$$

  此处的 μ 是均值，σ 是标准差。图 2.1(右) 的例子说明 X 在均值附近取值的可能性大，在远离均值的地方取值的可能性小。设 $\mathcal{X}$ 为变量 X 的取值范围。概率密度函数有这样的性质：

  $$\int_{\mathcal{X}} p(x)\, dx = 1.$$

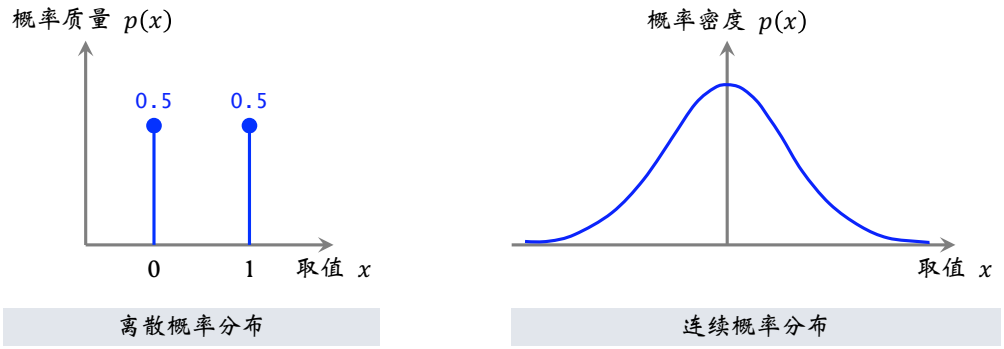

图 2.1: 左图是抛硬币的例子。右图是均值为零的正态分布。

函数 $f(X)$ 的**期望**是这样定义的。设 $p(X)$ 为 X 的概率密度函数（或概率质量函数）。对于离散概率分布，$f(X)$ 的期望是

$$\mathbb{E}_{X\sim p(\cdot)}\big[f(X)\big] = \sum_{x\in\mathcal{X}} p(x)\cdot f(x).$$

对于连续概率分布，$f(X)$ 的期望是

$$\mathbb{E}_{X\sim p(\cdot)}\big[f(X)\big] = \int_{\mathcal{X}} p(x)\cdot f(x)\,dx.$$

设 $g(X,Y)$ 为二元函数。如果对 $g(X,Y)$ 关于随机变量 X 求期望，那么会消掉 X，得到的结果是 Y 的函数。举个例子，设随机变量 X 的取值范围是 $\mathcal{X}=[0,10]$，概率密度函数是 $p(x)=\frac{1}{10}$。设 $g(X,Y)=2XY$，那么 $g(X,Y)$ 关于 X 的期望等于

$$\begin{aligned}\mathbb{E}_{X\sim p(\cdot)}\Big[g\big(X,Y\big)\Big] &= \int_{\mathcal{X}} g(x,Y)\cdot p(x)\,dx\\ &= \int_0^{10} 2xY\cdot\frac{1}{10}\,dx\\ &= 10Y.\end{aligned}$$

这个例子说明期望如何消掉函数 $g(X,Y)$ 中的变量 X。

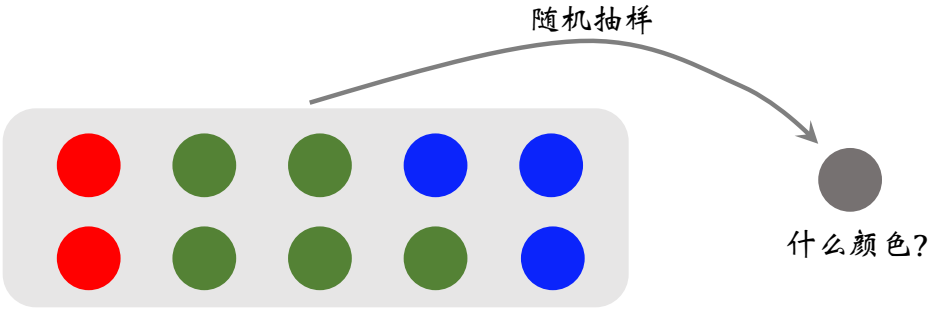

图 2.2: 箱子里有 10 个球。2 个是红色，5 个是绿色，3 个是蓝色。

强化学习中常用到**随机抽样**。举个例子，箱子里有 10 个球，其中 2 个是红色，5 个是绿色，3 个是蓝色；见图 2.2。我现在把箱子摇一摇，把手伸进箱子里，闭着眼睛摸出来一个球。这个球是什么颜色？答案很显然，三种颜色的球都有可能被摸到；摸到红色的概率是 0.2，摸到绿色的概率是 0.5，摸到蓝色的概率是 0.3。在我伸手摸球之前，摸到的颜色是随机变量 X，它有 3 可能的取值——红色、绿色、蓝色。现在我摸出来了一个

球，我睁开眼睛一看，是红色的。这时候随机变量 X 就有了观测值，观测值是红色。这个过程就叫随机抽样。我从箱子里随机摸出来了一个球，并且观测到了颜色，一次随机抽样就完成了。

现在换一种问法。箱子里有很多个球，不知道有多少个。做随机抽样，抽到红色球概率是 0.2，抽到绿色球概率是 0.5，抽到蓝色球概率是 0.3。我把手伸到箱子里，闭着眼睛摸个球。我摸到的是什么颜色的球？当然三种球都有可能摸到。根据

$$p(\text{红}) = 0.2, \qquad p(\text{绿}) = 0.5, \qquad p(\text{蓝}) = 0.3,$$

这个概率密度函数来随机选一个球，就是一次随机抽样。下面的 Python 代码按照上面的概率做随机抽样，重复 100 次，输出抽样的结果。

```
from numpy.random import choice

samples = choice(['R', 'G', 'B'],
                 size=100,
                 p=[0.2, 0.5, 0.3])

print(samples)
```

随机变量的取值范围是集合 {R, G, B}。

重复抽样100次，函数返回长度为100的数组。

R, G, B 三种颜色的球被选中的概率分别是 0.2, 0.5, 0.3。

```
['B' 'R' 'R' 'G' 'B' 'B' 'G' 'B' 'G' 'G' 'G' 'R' 'R' 'G' 'G' 'B' 'R' 'G' 'G' 'B'
'B' 'G' 'B' 'G' 'G' 'R' 'G' 'G' 'B' 'R' 'G' 'G' 'R' 'G' 'G' 'R' 'G' 'G' 'G' 'G'
'G' 'B' 'G' 'B' 'R' 'R' 'G' 'G' 'G' 'B' 'B' 'G' 'R' 'R' 'B' 'G' 'G' 'G' 'B' 'B'
'G' 'G' 'G' 'B' 'B' 'G' 'G' 'B' 'G' 'G' 'B' 'G' 'R' 'G' 'B' 'G' 'R' 'B' 'B' 'G'
'G' 'G' 'R' 'G' 'G' 'R' 'R' 'G' 'G' 'G' 'G' 'G' 'B' 'G' 'B' 'G' 'G' 'G' 'R' 'B']
```

2.2 蒙特卡洛

蒙特卡洛 (Monte Carlo) 是一大类随机算法 (Randomized Algorithms) 的总称，它们通过随机样本来估算真实值。本节用几个例子讲解蒙特卡洛算法。

2.2.1 例一：近似 π 值

我们都知道 π 约等于 3.1415927。现在假装我们不知道 π，而是要想办法近似估算 π 值。假设我们有（伪）随机数生成器，我们能不能用随机样本来近似 π 呢？这一小节使用蒙特卡洛近似 π 值。

假设我们有一个（伪）随机数生成器，可以均匀生成 -1 到 $+1$ 之间的数。每次生成两个随机数，一个作为 x，另一个作为 y。于是每次就生成了一个平面坐标系中的点 (x, y)；见图 2.3(左)。因为 x 和 y 都是在 $[-1, 1]$ 区间上均匀分布，所以 $[-1, 1] \times [-1, 1]$ 这个正方形内的点被抽到的概率是相同的。我们重复抽样 n 次，得到了 n 个正方形内的点。

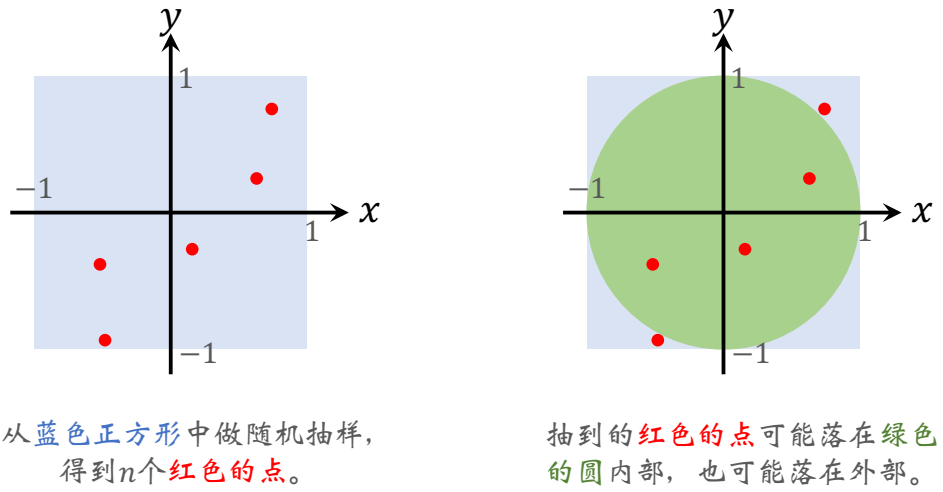

图 2.3: 通过抽样来近似 π 值。

如图 2.3(右) 所示，蓝色正方形里面包含一个绿色的圆，圆心是 $(0, 0)$，半径等于 1。刚才随机生成的 n 个点有些落在圆外面，有些落在圆里面。请问一个点落在圆里面的概率有多大呢? 由于抽样是均匀的，因此这个概率显然是圆的面积与正方形面积之比。正方形的面积是边长的平方，即 $a_1 = 2^2 = 4$。圆的面积是 π 乘以半径的平方，即 $a_2 = \pi \cdot 1^2 = \pi$。那么一个点落在圆里面的概率就是

$$p = \frac{a_2}{a_1} = \frac{\pi}{4}.$$

设我们随机抽样了 n 个点，设圆内的点的数量为随机变量 M。很显然，M 的期望等于

$$\mathbb{E}[M] = pn = \frac{\pi n}{4}.$$

注意，这只是期望，并不是实际发生的结果。如果你抽 $n = 5$ 个点，那么期望有 $\mathbb{E}[M] = \frac{5\pi}{4}$ 个点落在园内。但实际观测值 m 可能等于 0、1、2、3、4、5 中的任何一个。

给定一个点的坐标 (x, y)，该如何判断该点是否在圆内呢？已知圆心在原点，半径等于 1，我们用一下圆的方程。如果 (x, y) 满足：

$$x^2 + y^2 \le 1,$$

则说明 (x, y) 落在圆里面；反之，点就在圆外面。

我们均匀随机抽样得到 n 个点，通过圆的方程对每个点做判别，发现有 m 个点落在圆里面。假如 n 非常大，那么随机变量 M 的真实观测值 m 就会非常接近期望 $\mathbb{E}[M] = \frac{\pi n}{4}$：

$$m \approx \frac{\pi n}{4}.$$

对公式做个变换，得到：

$$\pi \approx \frac{4m}{n}.$$

我们可以依据这个公式做编程实现。下面是伪代码：

1. 初始化 $m = 0$。用户指定样本数量 n 的大小。n 越大，精度越高，但是计算量越大。
2. 把下面的步骤重复 n 次：
   (a). 从区间 $[-1, 1]$ 上均匀随机抽样得到 x；再做一次均匀随机抽样，得到 y。
   (b). 如果 $x^2 + y^2 \le 1$，那么 $m \leftarrow m + 1$。
3. 返回 $\frac{4m}{n}$ 作为对 π 的估计。

大数定律保证了蒙特卡洛的正确性：当 n 趋于无穷，$\frac{4m}{n}$ 趋于 π。其实还能进一步用概率不等式分析误差的上界。使用 Bernstein 不等式，可以证明出这个结论：

$$\left| \frac{4m}{n} - \pi \right| = O\left(\frac{1}{\sqrt{n}}\right).$$

这个不等式说明 $\frac{4m}{n}$（即对 π 的估计）会收敛到 π，收敛率是 $\frac{1}{\sqrt{n}}$。然而这个收敛率并不快：样本数量 n 增加一万倍，精度才能提高一百倍。

2.2.2 例二：估算阴影部分面积

图 2.4 中有正方形、圆、扇形，几个形状相交。请估算阴影部分面积。这个问题常见于初中数学竞赛。假如你不会微积分，也不会几何的奇技淫巧，你是否有办法近似估算阴影部分面积呢？用蒙特卡洛可以很容易解决这个问题。

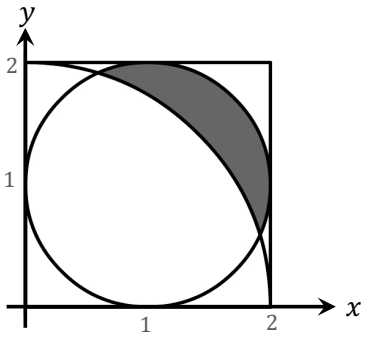

图 2.4: 估算阴影部分面积。

图 2.5 中绿色圆的圆心是 $(1, 1)$，半径等于 1；蓝色扇形的圆心是 $(0, 0)$，半径等于 2。阴影区域内的点 (x, y) 在绿色的圆中，而不在蓝色的扇形中。

- 利用圆的方程可以判定点 (x, y) 是否在绿色圆里面。如果 (x, y) 满足方程

$$(x - 1)^2 + (y - 1)^2 \le 1, \tag{2.1}$$

则说明 (x, y) 在绿色圆里面。

- 利用扇形的方程可以判定点 (x,y) 是否在蓝色扇形外面。如果点 (x,y) 满足方程

$$x^2 + y^2 > 2^2, \tag{2.2}$$

则说明 (x,y) 在蓝色扇形外面。

如果一个点同时满足方程 (2.1) 和 (2.2)，那么这个点一定在阴影区域内。从 $[0,2]\times[0,2]$ 这个正方形中做随机抽样，得到 n 个点。然后用两个方程筛选落在阴影部分的点。

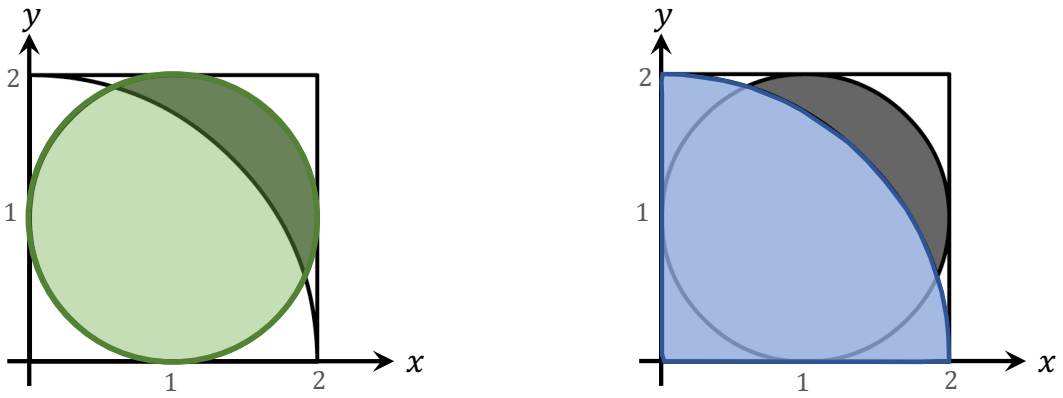

图 2.5: 如果一个点在阴影部分，那么它在左边绿的的圆中，而不在右边蓝色的扇形中。

我们在正方形 $[0,2]\times[0,2]$ 中随机均匀抽样，得到的点有一定概率会落在阴影部分。我们来计算这个概率。正方形的边长等于 2，所以面积 $a_1=4$。设阴影部分面积为 a_2。那么点落在阴影部分概率是

$$p = \frac{a_2}{a_1} = \frac{a_2}{4}.$$

我们从正方形中随机抽 n 个点，设有 M 个点落在阴影部分内（M 是个随机变量）。每个点落在阴影部分的概率是 p，所以 M 的期望等于

$$\mathbb{E}[M] = np = \frac{na_2}{4}.$$

用方程 (2.1) 和 (2.2) 对 n 个点做筛选，发现实际上有 m 个点落在阴影部分内（m 是随机变量 M 的观测值）。如果 n 很大，那么 m 会比较接近期望 $\mathbb{E}[M]=\frac{na_2}{4}$，即

$$m \approx \frac{na_2}{4}.$$

把等式变换一下，得到：

$$a_2 \approx \frac{4m}{n}.$$

这个公式就是对阴影部分面积的估计。我们可以依据这个公式做编程实现。下面是伪代码：

1. 初始化 $m=0$。用户指定样本数量 n 的大小。n 越大，精度越高，但是计算量越大。
2. 把下面的步骤重复 n 次：
   (a). 从区间 $[0,2]$ 上均匀随机抽样得到 x；再做一次均匀随机抽样，得到 y。
   (b). 如果 $(x-1)^2+(y-1)^2 \le 1$ 和 $x^2+y^2>4$ 两个不等式都成立，那么让 $m \leftarrow m+1$。

3. 返回 $\frac{4m}{n}$ 作为对阴影部分面积的估计。

2.2.3 例三：近似定积分

近似求积分是蒙特卡洛最重要的应用之一，在科学和工程中有广泛的应用。举个例子，给定一个函数：

$$f(x) \ = \ \frac{1}{1+(\sin x)\cdot(\ln x)^2},$$

要求计算 f 在区间 0.8 到 3 上的定积分：

$$I \ = \ \int_{0.8}^{3} f(x)\,d\,x.$$

有很多科学和工程问题需要计算定积分，而函数 $f(x)$ 可能很复杂，求定积分会很困难，甚至有可能不存在解析解。如果求解析解很困难，或者解析解不存在，则可以用蒙特卡洛近似计算数值解。

一元函数的定积分是相对比较简单的问题。一元函数的意思是变量 x 是个标量。给定一元函数 $f(x)$，求函数在 a 到 b 区间上的定积分：

$$I \ = \ \int_{a}^{b} f(x)\,d\,x.$$

蒙特卡洛方法通过下面的步骤近似定积分：

1. 在区间 $[a,b]$ 上做随机抽样，得到 n 个样本，记作：$x_1,\cdots,x_n$。样本数量 n 由用户自己定，n 越大，计算量越大，近似越准确。
2. 对函数值 $f(x_1),\cdots,f(x_n)$ 求平均，再乘以区间长度 $b-a$：

$$q_n \ = \ (b-a)\cdot\frac{1}{n}\sum_{i=1}^{n} f(x_i).$$

3. 返回 q_n 作为定积分 I 的估计值。

多元函数的定积分要复杂一些。设 $f:\mathbb{R}^d \mapsto \mathbb{R}$ 是一个多元函数，变量 $\boldsymbol{x}$ 是 d 维向量。要求计算 f 在集合 Ω 上的定积分：

$$I \ = \ \int_{\Omega} f(\boldsymbol{x})\,d\,\boldsymbol{x}.$$

蒙特卡洛方法通过下面的步骤近似定积分：

1. 在集合 Ω 上做均匀随机抽样，得到 n 个样本，记作向量 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n$。样本数量 n 由用户自己定，n 越大，计算量越大，近似越准确。
2. 计算集合 Ω 的体积：

$$v \ = \ \int_{\Omega} d\,\boldsymbol{x}.$$

3. 对函数值 $f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_n)$ 求平均，再乘以 Ω 体积 v：

$$q_n \ = \ v\cdot\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i). \tag{2.3}$$

4. 返回 q_n 作为定积分 I 的估计值。

注意，算法第二步需要求 Ω 的体积。如果 Ω 是长方体、球体等规则形状，那么可以解析

地算出体积 v。可是如果 Ω 是不规则形状，那么就需要定积分求 Ω 的体积 v，这是比较困难的。可以用类似于上一小节“求阴影部分面积”的方法近似计算体积 v。

举例讲解多元函数的蒙特卡洛积分：这个例子中被积分的函数是二元函数：

$$f(x,y) \;=\; \begin{cases} 1, & \text{if } x^2+y^2 \le 1; \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

直观地说，如果点 (x,y) 落在右图的绿色圆内，那么函数值就是 1；否则函数值就是 0。定义集合 $\Omega = [-1,1]\times[-1,1]$，即右图中蓝色的正方形，它的面积是 $v=4$。定积分

$$I \;=\; \int_{\Omega} f(x,y)\,dx\,dy$$

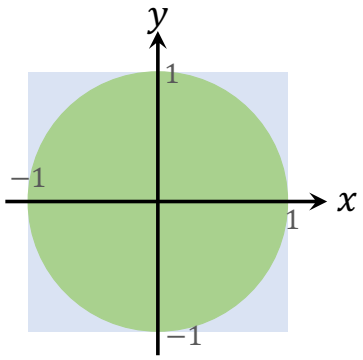

图 2.6: 用蒙特卡洛积分近似 π。

等于多少呢？很显然，定积分等于圆的面积，即 $\pi\cdot 1^2=\pi$。因此，定积分 $I=\pi$。用蒙特卡洛求出 I，就得到了 π。从集合 $\Omega=[-1,1]\times[-1,1]$ 上均匀随机抽样 n 个点，记作 $(x_1,y_1),\cdots,(x_n,y_n)$。应用公式 (2.3)，可得

$$q_n \;=\; v\cdot\frac{1}{n}\sum_{i=1}^{n} f(x_i,y_i) \;=\; \frac{4}{n}\sum_{i=1}^{n} f(x_i,y_i). \tag{2.5}$$

把 q_n 作为对定积分 $I=\pi$ 的近似。这与第 2.2.1 小节近似 π 的算法完全相同，区别在于此处的算法是从另一个角度推导出的。

2.2.4 例四：近似期望

蒙特卡洛还可以用来近似期望，这在整本书中会反复应用。定义 X 是 d 维随机变量，它的取值范围是集合 $\Omega\subset\mathbb{R}^d$。函数 $p(\boldsymbol{x})=\mathbb{P}(X=\boldsymbol{x})$ 是 X 的概率密度函数，它描述变量 X 在取值点 $\boldsymbol{x}$ 附近的可能性。设 $f:\Omega\mapsto\mathbb{R}$ 是任意的多元函数，它关于变量 X 的期望是：

$$\mathbb{E}_{X\sim p(\cdot)}\Big[f(X)\Big] \;=\; \int_{\Omega} p(\boldsymbol{x})\cdot f(\boldsymbol{x})\,d\boldsymbol{x}.$$

由于期望是定积分，所以可以按照上一小节的方法，用蒙特卡洛求定积分。上一小节在集合 Ω 上做**均匀抽样**，用得到的样本近似上面的定积分。

下面介绍一种更好的算法。既然我们知道概率密度函数 $p(\boldsymbol{x})$，我们最好是按照 $p(\boldsymbol{x})$ 做**非均匀抽样**，而不是均匀抽样。按照 $p(\boldsymbol{x})$ 做非均匀抽样，可以比均匀抽样有更快的收敛。具体步骤如下：

1. 按照概率密度函数 $p(\boldsymbol{x})$，在集合 Ω 上做非均匀随机抽样，得到 n 个样本，记作向量 $\boldsymbol{x}_1,\cdots,\boldsymbol{x}_n\sim p(\cdot)$。样本数量 n 由用户自己定，n 越大，计算量越大，近似越准确。
2. 对函数值 $f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_n)$ 求平均：

$$q_n \;=\; \frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i).$$

3. 返回 q_n 作为期望 $\mathbb{E}_{X\sim p(\cdot)}[f(X)]$ 的估计值。

注 如果按照上述方式做编程实现，需要储存函数值 $f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_n)$。用如下的方式做编程实现，可以减小内存开销。初始化 $q_0=0$。从 $t=1$ 到 n，依次计算

$$q_t \;=\; \big(1-\tfrac{1}{t}\big)\cdot q_{t-1} \;+\; \tfrac{1}{t}\cdot f\big(\boldsymbol{x}_t\big). \tag{2.6}$$

不难证明，这样得到的 q_n 等于 $\frac{1}{n}\sum_{i=1}^{n} f(\boldsymbol{x}_i)$。这样无需存储所有的 $f(\boldsymbol{x}_1),\cdots,f(\boldsymbol{x}_n)$。可以进一步把公式 (2.6) 中的 $\frac{1}{t}$ 替换成 α_t，得到公式：

$$q_t \;=\; \big(1-\alpha_t\big)\cdot q_{t-1} \;+\; \alpha_t\cdot f\big(\boldsymbol{x}_t\big).$$

这个公式叫做 Robbins-Monro 算法，其中 α_n 称为学习步长或学习率。只要 α_t 满足下面的性质，就能保证算法的正确性：

$$\lim_{n\to\infty}\sum_{t=1}^{n}\alpha_t \;=\; \infty \qquad \text{和} \qquad \lim_{n\to\infty}\sum_{t=1}^{n}\alpha_t^2 \;<\; \infty.$$

很显然，$\alpha_t=\frac{1}{t}$ 满足上述性质。Robbins-Monro 算法可以应用在 Q 学习算法中。

2.2.5 例五：随机梯度

蒙特卡洛近似期望在机器学习中的一个应用是**随机梯度**。设随机变量 X 为一个数据点，设 $\boldsymbol{w}$ 为神经网络的参数。设 $p(\boldsymbol{x})=\mathbb{P}(X=\boldsymbol{x})$ 为随机变量 X 的概率密度函数。定义损失函数 $L(X;\boldsymbol{w})$。它的值越小，意味着模型对 X 的预测越准确；反之，它的值越大，则意味着模型对 X 的预测越离谱。因此，我们希望调整神经网络的参数 $\boldsymbol{w}$，使得损失函数的期望尽量小。神经网络的训练可以定义为这样的优化问题：

$$\min_{\boldsymbol{w}}\; \mathbb{E}_{X\sim p(\cdot)}\Big[L\big(X;\boldsymbol{w}\big)\Big]. \tag{2.7}$$

目标函数 $\mathbb{E}_X[L(X;\boldsymbol{w})]$ 关于 $\boldsymbol{w}$ 的梯度是：

$$\boldsymbol{g} \;\triangleq\; \nabla_{\boldsymbol{w}}\,\mathbb{E}_{X\sim p(\cdot)}\Big[\,L(X;\boldsymbol{w})\,\Big] \;=\; \mathbb{E}_{X\sim p(\cdot)}\Big[\,\nabla_{\boldsymbol{w}}\,L(X;\boldsymbol{w})\,\Big].$$

可以做梯度下降更新 $\boldsymbol{w}$，以减小目标函数 $\mathbb{E}_X[L(X;\boldsymbol{w})]$：

$$\boldsymbol{w} \;\leftarrow\; \boldsymbol{w} \;-\; \alpha\cdot\boldsymbol{g}.$$

此处的 α 被称作学习率 (Learning Rate)。直接计算梯度 $\boldsymbol{g}$ 通常会比较慢。为了加速计算，可以对期望

$$\boldsymbol{g} \;=\; \mathbb{E}_{X\sim p(\cdot)}\Big[\,\nabla_{\boldsymbol{w}}\,L(X;\boldsymbol{w})\,\Big]$$

做蒙特卡洛近似，把得到的近似梯度 $\tilde{\boldsymbol{g}}$ 称作随机梯度 (Stochastic Gradient)，用 $\tilde{\boldsymbol{g}}$ 代替 $\boldsymbol{g}$ 来更新 $\boldsymbol{w}$。

1. 根据概率密度函数 $p(\boldsymbol{x})$ 做随机抽样，得到 b 个样本，记作 $\tilde{\boldsymbol{x}}_1,\cdots,\tilde{\boldsymbol{x}}_b$。
2. 计算梯度 $\nabla_{\boldsymbol{w}}\,L(\tilde{\boldsymbol{x}}_j;\boldsymbol{w})$，$\forall\, j=1,\cdots,b$。对它们求平均：

$$\tilde{\boldsymbol{g}} \;=\; \frac{1}{b}\sum_{j=1}^{b}\nabla_{\boldsymbol{w}}\,L(\tilde{\boldsymbol{x}}_j;\boldsymbol{w}).$$

$\tilde{\boldsymbol{g}}$ 被称作随机梯度，它是 $\boldsymbol{g}$ 的蒙特卡洛近似。

3. 做随机梯度下降更新 $\boldsymbol{w}$：

$$\boldsymbol{w} \leftarrow \boldsymbol{w} - \alpha \cdot \tilde{\boldsymbol{g}}.$$

蒙特卡洛用的样本数量 b 称作批量大小 (Batch Size)，通常是一个比较小的整数，比如 1、8、16、32。

在机器学习中，随机变量 X 通常服从下面这个离散概率分布。已经收集到一个数据集 $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$，定义概率质量函数为：

$$p(\boldsymbol{x}_i) = \mathbb{P}\big(X = \boldsymbol{x}_i\big) = \frac{1}{n}, \qquad \forall\, i = 1, \cdots, n.$$

这个概率分布的意思是随机变量 X 的取值是 n 个数据点中的一个，概率都是 $\frac{1}{n}$。那么随机梯度下降每一轮都从集合 $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n\}$ 中均匀随机抽取 b 个样本。

第二章习题

1. 设 X 是离散随机变量，取值范围是集合 $\mathcal{X}=\{1,2,3\}$。定义概率密度函数：

$$\begin{aligned} p(1) &= \mathbb{P}\big(X=1\big) = 0.4, \\ p(2) &= \mathbb{P}\big(X=2\big) = 0.1, \\ p(3) &= \mathbb{P}\big(X=3\big) = 0.5. \end{aligned}$$

   定义函数 $f(x)=2x^2+3$。请计算 $\mathbb{E}_{X\sim p(\cdot)}[f(X)]$。

2. 设 X 服从均值为 $\mu=1$、标准差 $\sigma=2$ 的一元正态分布。定义函数 $f(x)=2x+10\ln|x|+3$。请设计蒙特卡洛算法，并编程计算 $\mathbb{E}_X[f(X)]$。

3. Bernstein 概率不等式是这样定义的。设 $Z_1,\cdots,Z_n$ 为独立的随机变量，它们的概率密度函数是任意的，但是它们必须满足三个条件：
   - 变量的期望为零：$\mathbb{E}[Z_1]=\cdots=\mathbb{E}[Z_n]=0$。
   - 变量是有界的：存在 $b>0$，使得 $|Z_i|\le b,\ \forall i=1,\cdots,n$。
   - 变量的方差是有界的：存在 $v>0$，使得 $\mathbb{E}[Z_i^2]\le v,\ \forall i=1,\cdots,n$。

   那么有这样的概率不等式：

$$\mathbb{P}\left(\left|\frac{1}{n}\sum_{i=1}^{n}Z_i\right| \ge \epsilon\right) \le \exp\left(-\frac{\epsilon^2 n/2}{v+\epsilon b/3}\right).$$

   公式 (2.5) 算出的 q_n 是 π 的蒙特卡洛近似。请用 Bernstein 不等式证明：

$$\Big|q_n-\pi\Big| = O\Big(\frac{1}{\sqrt{n}}\Big) \qquad \text{以很高的概率成立。}$$

   （**提示：** 设 (X_i,Y_i) 是从正方形 $[-1,1]\times[-1,1]$ 中随机抽取的点。二元函数 f 在公式 (2.4) 中定义。设 $Z_i=4f(X_i,Y_i)-\pi$，它是个均值为零的随机变量。）

4. 初始化 $q_0=0$。让 t 从 1 增长到 n，依次计算

$$q_t = \big(1-\tfrac{1}{t}\big)\cdot q_{t-1} + \tfrac{1}{t}\cdot f\big(\boldsymbol{x}_t\big).$$

   请证明上述迭代得到的结果 q_n 等于 $\frac{1}{n}\sum_{i=1}^{n}f(\boldsymbol{x}_i)$。