强化学习笔记

彭灵伟

2021 年 1 月 4 日

目录

1 马尔可夫决策过程

在日常生活中，人们在不断地做着各种各样的决策。而且这些决策通常包含着短期结果和长期结果。有很多决策是单独做出来的，只考虑到了当前能获得的信息，但是又会同时影响今天、明天和后天。如果不综合考虑当前和将来的决策，以及它们会带来的当前和将来的结果，我们可能很难获得一个好的结果。例如，对于一个长跑比赛，如果一开始就冲刺，我们获得一个很高的速度，但是也会消耗大量的能量，最终也只能取得一个比较差的成绩。我们希望研究如何进行最好的决策，来获得一个最好地综合结果。从这里来看，这个问题似乎是矛盾的：即只能依靠当前信息来做决策，又需要最大化一个综合结果。

幸运的是，马尔可夫决策过程给这个问题提供了一个解决思路。马尔可夫决策过程是一个关于随机序列的决策模型。马尔可夫决策过程本身形式上非常简单，并且有非常严谨且充分的数学理论作为基础，所以它的应用非常广泛。其中，强化学习就是基于马尔可夫决策过程发展出的一套学习理论，并且在近些年成功解决了一系列非常复杂的决策问题。本章接下来将介绍一下马尔可夫决策过程，以及在相关的重要结论。

本章节将先从马尔科夫链（Markov Chain， MC）切入，紧接着介绍马尔科夫决策过程（Markov Decision Processes， MDPs）的数学模型，同时揭示马尔科夫链和马尔科夫决策过程关系。然后本章节将介绍两个马尔科夫决策过程的两个重要的等式——贝尔曼等式（Bellman Equaiton）和最优贝尔曼等式（Optimal Bellman Equation），分别用于策略评估（Policy Evaluation）和策略提升（Policy Improvement）。

1.1 马尔科夫链

这一小节的目的是为了介绍一下马尔可夫决策过程使用的随机过程模型——马尔科夫链（Markov Chain, MC）。我们将介绍马尔可夫链的数学描述以及一些与马尔科夫决策过程以及强化学习息息相关的数学结论 [28]。

马尔可夫链是一个离散的随机过程模型。在离散的随机过程模型中，时间被分割成一个个离散的时间点，对应自然数集合 $\mathcal{I} = \{0, 1, 2, \ldots, t, \ldots\}$。每个时间点 $t \in \mathcal{I}$ 对应一个随机变量 X_t。由于时间维度的单向性，我们就得到了一个过程 $(X_0, X_1, X_2, \ldots, X_t, \ldots)$。 一个随机过程也可以看作是一个概率空间，它对应的空间中的样本为 $\tau = (x_0, x_1, x_2, \ldots, x_t, \ldots)$，我们称

τ 为一条**轨迹**（trajectory），同时轨迹 τ 服从某种由随机变量 $\{X_t\}$ 确定的概率分布 $\mathcal{D}$。

我们可以把每一个样本 τ 张成一个个无穷维的向量，这样一个随机过程对应的概率空间就变成了一个复杂的无穷维的概率空间。通常直接在这种无穷维的概率空间上求解问题非常困难，所以一类随机过程先验性地给这些数据赋予了**单向性**，来大大降低问题的复杂程度：在一个随机过程中，当 $(X_0 = x_0, X_1 = x_1, \ldots, X_t = x_t, \ldots)$，那么随机变量 X_{t+1} 由过去的信息唯一确定：

$$\mathbf{P}(X_{t+1}|X_t = x_t, \ldots, X_1 = x_1, X_0 = x_0). \tag{1}$$

这里的唯一确定性是由时间维度赋予的，也就是贝叶斯公式增加了一个单向性的限制条件。

马尔科夫链则是在单向性上再增加了一个历史无关性，即下一时刻的随机变量只与当前时刻的随机变量有关，与当前时刻以前的随机变量都无关。这样我们就可以引出马尔科夫链的定义：

定义 1.1 (马尔科夫链)**.** 对于一个离散随机过程 $(X_0, X_1, \ldots, X_t, \ldots)$，如果满足

$$\mathbb{P}(X_{t+1}|X_0 = x_0, \ldots, X_t = x_t, X_{t+2} = x_{t+2}, \ldots) = \mathbb{P}(X_{t+1}|X_t = x_t), \tag{2}$$

那么，我们称这个离散随机过程为马尔科夫链。

我们需要更精确的符号来描述一个马尔科夫链。首先我们假设随机变量 $\{X_t\}$ 都是定义在**状态集合** $\mathcal{X}$ 上的。接着，对于马尔可夫链的一条轨迹 $\tau = (x_0, x_1, \ldots, x_t, \ldots)$，它的概率应该满足：

$$\mathbb{P}[\tau = (x_0, x_1, \ldots, x_t, \ldots)] = \mathbf{p}_0(x_0) \prod_{t=0}^{\infty} \mathbf{P}(x_{t+1}|x_t). \tag{3}$$

也就是说，确定一条轨迹的概率需要两个概率：**初始概率分布** $\mathbf{p}_0$ 与**状态转移概率分布** $\mathbf{P}$。综上，任意一个马尔科夫链可以由三元组 $(\mathcal{X}, \mathbf{p}_0, \mathbf{P})$ 唯一确定。如果 $\mathcal{X} = \{x1, x2, \ldots, x_n\}$，马尔状态转移概率分布可以用矩阵来表示：

$$\mathbf{P} = \begin{bmatrix} P(x1|x1) & P(x2|x1) & \ldots & P(xn|x1) \\ P(x1|x2) & P(x2|x2) & \ldots & P(xn|x2) \\ \vdots & \vdots & \ddots & \vdots \\ P(x1|xn) & P(x2|xn) & \ldots & P(xn|xn) \end{bmatrix}. \tag{4}$$

注意到状态转移矩阵的每一行和为 1。对于任意一个定义在 $\mathcal{X}$ 上的分布 $\mathbf{p}$，$\mathbf{P}^T\mathbf{p}$ 则表示分布 $\mathbf{p}$ 经过一次马尔科夫链状态转移后的分布。

接着，我们需要介绍一个马尔科夫链中的重要概念——**稳态分布**。

定义 1.2 (稳态分布, Stationary Distribution)**.** 一个马尔科夫链 $(\mathcal{X}, \mathbf{p}_0, \mathbf{P})$，如果存在一个定义在 $\mathcal{X}$ 上的分布 $\mathbf{d}$ 满足

$$\mathbf{P}^T\mathbf{d} = \mathbf{d}, \tag{5}$$

那么，我们称 $\mathbf{d}$ 为这个马尔科夫链的稳态分布。

下面的讨论需要用到一个非常重要的矩阵论中的定理——佛罗贝尼乌斯—佩龙定理，所以这里先补充这个定理的相关知识。

定义 1.3 (可约方阵和不可约方阵)**.** 对于 n 阶方阵，如果下标集合 $\mathcal{I} = \{1, 2, \ldots, n\}$ 能够被划分成两个不想交的集合 J 和 K，使得任意的 $j \in J$ 和 $k \in K$ 都有 $a_{jk} = 0$，那么方阵 A 就是可约的；否则方阵 A 就是不可约的。

定理 1.1 (佛罗贝尼乌斯—佩龙定理, Perron-Frobenius Theorem)**.** 如果 $\mathbf{A}$ 是一个不可约的非负方阵时，那么有如下结论：

1. $\mathbf{A}$ 的最大的实数特征值 $\lambda_{\max}$ 和其他特征根 λ 满足 $|\lambda| < \lambda_{\max}$；
2. $\mathbf{A}$ 的特征值 $\lambda_{\max}$ 对应的特征向量 $\mathbf{x}$ 方向唯一，并且满足 $\mathbf{x} \succ \mathbf{0}$ 或者 $\mathbf{x} \prec \mathbf{0}$。

通常，我们会假设马尔科夫链是满足**不可约** (irreducible) 的和**非周期循环的** (aperiodical)。其中，不可约指的是马尔科夫链的任意一个状态 $x \in \mathcal{X}$ 被访问到的概率大于 0 (在离散情况下等价为状态转移矩阵 $\mathbf{P}$ 是不可约的)；而非周期循环，指的是所有概率不为零的轨迹 τ 都不存在循环周期大于 1 的循环节。这样，就引出了关于稳态分布的一个定理：

定理 1.2 (稳态分布的存在性定理)**.** 如果马尔可夫链是不可约的与非周期循环的，那么，这条马尔科夫链的稳态分布 $\mathbf{d}$ 一定存在，并且任意初始分布 $\mathbf{p}_0$ 经过马尔科夫链的状态转移，最终都会收敛到稳态分布 $\mathbf{d}$ 上。

证明. 首先，我们需要证明状态转移矩阵 $\mathbf{P}^T$ 存在特征值为 1。设 $\mathbf{1}$ 所有维度全部是 1 的向量，那么根据定义可得 $\mathbf{P1} = \mathbf{1}$。所以 $\mathbf{P}$ 存在特征值包含 1。又因为 $\mathbf{P}$ 和 $\mathbf{P}^T$ 的特征值相同，所以 $\mathbf{P}^T$ 的特征值包含 1。

接着，我们要证明 $\mathbf{P}^T$ 的最大特征值为 1。我们设 $\mathbf{P}$ 的特征值为 γ 以及对应的特征向量为 $\mathbf{z}$，并且我们设 $z_{\max} = |z_k| = \max_i |z_i|$。那么，我们有：

$$|\gamma z_{\max}| = \left| \sum_j \mathbf{P}_{jk} z_j \right| \leq \sum_j \mathbf{P}_{jk} |z_j| \leq \sum_j \mathbf{P}_{jk} z_{\max} \leq z_{\max}. \tag{6}$$

因此，我们可得 $|\gamma| \leq 1$，也就是 $\mathbf{P}$ 的最大特征值为 1。又因为 $\mathbf{P}$ 和 $\mathbf{P}^T$ 的特征值相同，所以我们可得 $\mathbf{P}^T$ 的最大特征值为 1，也就是说，状态转移矩阵的谱半径为 1。

这时，我们可以带入佛罗贝尼乌斯—佩龙定理可得，$\mathbf{P}^T$ 存在一个方向唯一且全部为正数的特征向量，它对应的特征值为 1。这里也就是说，任意一个不可约的状态转移矩阵都存在一个稳态分布。

又因为马尔科夫链是非周期循环的，所以我们可知，对于任意初始分布 $\mathbf{p}_0$，经过无限次状态转移后，存在收敛分布 $\mathbf{p}_\infty = \lim_{t\to\infty} (\mathbf{P}^T)^t \mathbf{p}_0$。根据极限的性质，我们可得 $\mathbf{p}_\infty = \mathbf{P}^T \mathbf{p}_\infty$。又因为状态转移方程特征值为 1 的特征向量的方向唯一性，我们可知 $\mathbf{p}_\infty = \mathbf{d}$。 □

注 1.1. 当马尔可夫链的状态集合不可数时，我们就不方便使用概率分布以及矩阵的形式来表达马尔可夫链，我们转而使用初始状态概率密度 $p_0(x_0)$ 以及状态转移概率密度 $p(x_{t+1}|x_t)$ 来替换对应的概率分布 $\mathbf{p}_0$ 与 $\mathbf{P}$。同样的，对应的稳态分布概率密度 d 应该满足：

$$d(x_{t+1}) = \int_{x_t \in \mathcal{X}} p(x_{t+1}|x_t) d(x_t) \mathrm{d}x_t. \tag{7}$$

也就是，在一个不可数的状态集合中，马尔科夫链的三元组变为 $(\mathcal{X}, p_0, p)$。

经过本小节关于马尔科夫链的介绍，接下来本文可以开始介绍马尔可夫决策过程了。

1.2 马尔可夫决策过程

本小节将介绍一下强化学习所使用的基本模型——马尔可夫决策过程 (Markov Decision Processes, MDPs)，主要涉及到马尔可夫决策过程的基本定义。在后边的小节将展开介绍马尔可夫决策过程相关算法，它们都与强化学习息息相关。

1.2.1 马尔科夫决策过程的发展历程

马尔可夫决策过程也被称为序列随机优化、离散随机控制、以及随机动态规划。它的基本目标是：面对一个时间离散的随机系统，我们可以控制它的状态转移，而我们希望能够学到一个针对这个随机系统最好的控制策略。马尔可夫决策过程假设这个系统的状态转移满足马尔可夫性，今后的状态只与当前的状态和动作有关。当前的动作会有一个短期的花销或者奖励，但是马尔可夫决策过程希望能够学到一个长期累计花销小或者奖励值大的控制策略 [16]。

由于马尔可夫决策过程的实用价值和巧妙的模型思想，吸引了大量的研究。它提供了大量的针对真实世界问题的解决方案，尤其是在商业与工程应用里。从马尔可夫决策过程引发了许多数学与计算的问题。

一部分研究人员从动态规划的角度来理解马尔可夫决策过程，其中贝尔曼等人做出了非常系统性的研究，出版了大量的论文与书籍 [8, 7, 35, 40]。整个工作发生在上世纪五十年代就有大量的关于随机动态规划的研究。自从上世纪五十年代开始引出马尔可夫决策过程，科研界产生了大量的深入的理论与应用 [34, 17, 24, 3, 10]。事实上，马尔可夫决策过程已经成为一个基本的分析工具，用于分析各种电力系统、控制、以及计算机科学领域的问题。

一直到上世纪八十年代，大部分的工作都集中在最优方程与相关求解算法：策略迭代与值迭代。它们都属于动态规划的范畴。 动态规划通常最大化一个累加的奖励函数，但是无法解决多奖励函数以及附加某些限制的马尔科夫决策过程问题。另外一些工作集中在非直接求解的算法上 [49, 5, 3]，这些方法中，凸分析、线性规划和凸规划等工具经常被使用的。在近些年，研究领域在马尔科夫决策过程中取得了非常大的成功 [31, 42, 37]，人们开始研究多目标马尔可夫决策过程 [5]，以及对抗与合作马尔科夫决策过程 [25]。

1.2.2 马尔可夫决策过程的定义

我们这里只介绍一种马尔可夫决策过程，使用折扣累加奖励函数的时间无限且离散的马尔可夫决策过程。这种决策过程模型是马尔可夫决策过程模型中使用最为广泛的模型。尤其是在强化学习领域中，我们默认算法所使用的就是这种模型。

所有的离散随机决策过程面对的是一个离散时间动力学系统。和马尔科夫链相同，我们用自然数集合将时间编码为 $\mathcal{I} = \{0, 1, 2, \ldots, t, \ldots\}$。在每

个时间节点 t 对应一个**状态**随机变量 S_t，和一个**动作**随机变量 A_t。由于时间的单向性，我们获得了一个随机过程 $(S_0, A_0, S_1, A_1, \ldots, S_t, A_t, \ldots)$。我们可以对一个随机过程进行采样就可以获得一条**轨迹**:

$$\tau = (s_0, a_0, s_1, a_1, \ldots, s_t, a_t, \ldots). \tag{8}$$

同样的，对于一个随机过程，我们可以简化它，即要求当前时刻的随机变量只与历史随机变量有关。即我们可以要求 S_{t+1} 和 A_{t+1} 由如下概率分布唯一确定:

$$\begin{cases} \mathbf{P}(S_{t+1}|S_t = s_t, A_t = a_t, \ldots, S_0 = s_0, A_0 = a_0); \\ \mathbf{P}(A_{t+1}|S_{t+1} = s_{t+1}, S_t = s_t, A_t = a_t, \ldots, S_0 = s_0, A_0 = a_0). \end{cases} \tag{9}$$

其中关于 S_{t+1} 的分布我们称为**状态转移分布**，是由决策过程唯一确定的；而关于 A_{t+1} 的分布我们称为**决策分布**或者叫做**策略函数**，是可以被人为主动选定的。人为选定不同的决策分布，就会产生不同的随机过程，而我们希望能够选到一个特定的决策分布，来产生我们希望获得的随机过程。

我们可以通过引入马尔可夫性质，我们可以进一步地简化一个随机过程，得到马尔科夫决策过程。

定义 1.4 (马尔可夫决策过程，Markov Decision Processes)**.** 对于一个随机决策过程

$$(S_0, A_0, S_1, A_1, \ldots, S_t, A_t, \ldots), \tag{10}$$

如果它的状态转移分布满足:

$$\begin{aligned} &\mathbf{P}(S_{t+1}|S_0 = s_0, A_0 = a_0, \ldots, S_t = s_t, A_t = a_t, \\ &\qquad S_{t+2} = s_{t+2}, A_{t+2} = a_{t+2}, \ldots) = \mathbf{P}(S_{t+1}|S_t = s_t, A_t = a_t), \end{aligned} \tag{11}$$

那么我们称它为马尔可夫决策过程。

同样的，我们也可以给策略增加一个马尔可夫性，来获得一个更加简化的随机决策过程:

定义 1.5 (马尔科夫策略，Markov Policy)**.** 如果一个策略函数满足

$$\begin{aligned} &\mathbf{P}(A_{t+1}|S_0 = s_0, A_0 = a_0, \ldots, S_t = s_t, A_t = a_t, S_{t+1} = s_{t+1}, \\ &\qquad S_{t+2} = s_{t+2}, A_{t+2} = a_{t+2} \ldots,) = \mathbf{P}(A_{t+1}|S_{t+1} = s_{t+1}), \end{aligned} \tag{12}$$

那么我们就称这个策略函数为马尔科夫策略。

同样的，我们需要更精细化的描述一个马尔可夫决策过程。首先我们假设状态随机变量 $\{S_t\}$ 都是定义在状态集合 $\mathcal{S}$ 上的，并且，我们假设动作随机变量 $\{A_t\}$ 都是定义在动作集合 $\mathcal{A}$ 上的。对于一条轨迹 $\tau = (s_0, a_0, s_1, a_1, \ldots, s_t, a_t, \ldots)$，如果使用马尔可夫策略，那么它对应的概率应该是：

$$\mathbb{P}(\tau) = \mathbf{p}_0(s_0) \prod_{t=0}^{\infty} \pi(a_t|s_t)\mathbf{P}(s_{t+1}|s_t, a_t). \tag{13}$$

其中 $\mathbf{p}_0$ 是初始状态分布，π 是马尔可夫策略，以及 $\mathbf{P}$ 是状态转移分布。这里策略函数使用符号 π 是为了和状态转移分布区别开来，在以后的章节中，我们使用 π 来专门指代策略函数。因为策略可以由人为任意选定，所以并不是一个马尔可夫决策过程的特征量。最终，我们只需要一个四元组就可以唯一确定一个马尔可夫决策过程：$(\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P})$。

值得注意的是，当使用的是某个确定的马尔可夫策略 π 时，马尔可夫决策过程就变成了一条马尔可夫链：马尔科夫链的状态集合 $\mathcal{X} = \mathcal{S} \times \mathcal{A}$（笛卡尔积），初始状态分布为 $\mathbf{p}_0[(s_0, a_0)] = \mathbf{p}_0(s_0)\pi(a_0|s_0)$, 以及状态转移分布为 $\mathbf{P}[(s_{t+1}, a_{t+1})|(s_t, a_t)] = \pi(a_{t+1}|s_{t+1})\mathbf{P}[s_{t+1}|s_t, a_t]$。也就是说，马尔可夫决策过程可以看成是将马尔可夫策略集合映射到某个马尔科夫链集合的一个映射函数。

1.2.3 马尔科夫决策问题的定义

在本文中，马尔可夫决策过程在英文中指的是 Markov Decision Processes，而马尔可夫决策问题在英文中指的是 Markov Decision Problems，它们的英文缩写都是 MDPs。而通常情况下，马尔科夫决策过程是泛指马尔可夫决策过程与马尔可夫决策问题。在本小节，我们将两个概念分开描述，用于强调不同的定义量各自的侧重点。而在本论文的其他部分，我们则使用马尔可夫决策过程来泛指马尔可夫决策过程以及马尔可夫决策问题。

马尔科夫决策过程通常建模的是一个序列问题的客观规律，它们是所面对的问题中不受控制的一部分，反映的是问题自身的特性。继续上面一个小节的讨论，如果马尔可夫决策过程模型是一个从马尔科夫策略集合映射到马尔科夫链集合的映射函数，虽然自变量马尔科夫策略可以随意选定，但是每个马尔科夫策略对应的马尔科夫链是不会变化的。

马尔科夫决策问题整体上是对马尔科夫链施加一个评价函数，因此马尔科夫链有了“优”与“劣”的差别。马尔科夫决策问题要求出一个策略函

数使得我们获得一个最优的马尔科夫链。

评价函数在最优控制的领域表现为损失函数，而在强化学习的领域则表现为奖励函数，我们这里就沿用奖励函数的形式。首先为了简化问题，实际上我们设计了一个马尔可夫奖励函数对马尔科夫链的状态进行编码，它只与马尔可夫链的当前状态有关。

定义 1.6 (马尔可夫奖励函数)**.** 对于马尔可夫链 $(\mathcal{S}\times\mathcal{A},\mathbf{p}_0,\mathbf{P})$，任意一个定义在 $\mathcal{S}\times\mathcal{A}$ 上因变量为一维实数的奖励函数 R 都可以作为为它的马尔可夫奖励函数。

马尔科夫链的任意一条轨迹 $(s_0,a_0,s_1,a_1,\ldots,s_t,a_t,\ldots)$，都会对应一条数列 $(R(s_0,a_0),R(s_1,a_1),\ldots,R(s_t,a_t),\ldots)$。然而，我们对于一条数列，也无法评价它的好坏，所以我们还需要一个针对数列的评价函数。这里我们就介绍一个使用最为广泛的函数——折扣累加函数。

定义 1.7 (折扣累加函数)**.** 对于一条数列 $(a_0,a_1,\ldots,a_t,\ldots)$，我们有折扣累加函数，

$$discounted[\gamma,(a_0,a_1,\ldots,a_t,\ldots)]==\frac{\sum_{t=0}^{\infty}\gamma^t a_t}{\sum_{t=0}^{\infty}\gamma^t}=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t a_t,\quad(14)$$

将它映射为一个实数，其中 γ 为折扣因子。

马尔可夫决策过程 $\{\mathcal{S},\mathcal{A},\mathbf{p}_0,\mathbf{P}\}$ 加上奖励函数 $R(s,a)$ 以及折扣累加函数 $discounted(\gamma,\tau)$，我们就能够获得一个马尔科夫决策问题了。

定义 1.8 (无限时间折扣马尔科夫决策问题)**.** 现在已知马尔可夫决策过程 $\{\mathcal{S},\mathcal{A},\mathbf{p}_0,\mathbf{P}\}$，奖励函数 $R(s,a)$，以及折扣累加函数 $discounted(\gamma,\tau)$。对于任何一个决策函数 π，我们标记它对应的马尔科夫链的轨迹为 $\tau=(s_0,a_0,s_1,a_1,\ldots,s_t,a_t,\ldots)$。那么，无限时间折扣马尔可夫决策问题就是要求解：

$$\max_{\pi}\rho(\pi)=(1-\gamma)\mathbb{E}_{\tau}\left[\sum_{t=1}^{\infty}\gamma^t r_t\right],\quad(15)$$

其中 $r_t=R(s_t,a_t)$。

从定义可知，要精确描述一个无限时间折扣马尔科夫决策问题，我们通常需要一个六元组 $(\mathcal{S},\mathcal{A},\mathbf{p}_0,\mathbf{P},R,\gamma)$。其中使用折扣因子 γ 来代指折扣累加函数 $discounted(\gamma,\cdot)$。同样，如果状态集合与动作集合是不可数的时

候，我们使用概率密度函数来替换概率分布函数。也就是说，无限时间折扣马尔可夫决策问题所使用的六元组变为 $(\mathcal{S}, \mathcal{A}, p_0, p, R, \gamma)$。

注 1.2. 对于不同的奖励函数以及不同的数列评价函数，则会对应完全不同的马尔可夫决策问题。我们这里介绍的是本文以及强化学习中广泛使用的模型，同时也为其他类型的马尔科夫决策问题保留了接口，例如：有限平均马尔科夫决策问题。

注 1.3. 在大部分的强化学习设定中，奖励函数是由随机过程提供的，也因此很容易被认为是随机过程中的某种不变量。但在作者实际的科研过程中发现，对于一个随机过程决策问题，往往是没有奖励函数的，反而是需要研究者通过个人大量的先验知识，设计出一个合适的奖励函数，存在一定的主观色彩。但是，从更高层的视角来看，这种关于奖励函数的先验知识也应该是一个人工智能自主学习的，这也是另一个研究热点，逆强化学习的问题设定。综上，作者认为，奖励函数应该是放在马尔可夫决策问题部分，使马尔科夫决策过程保持客观性。

在今后的章节，我们使用泛化的马尔可夫决策过程。并且，大部分情况下，文章中的马尔可夫决策过程就是特指无限时间折扣马尔可夫决策问题。同时，在强化学习部分，我们将使用强化学习的术语来替换本章节的一些名词，例如：我们会使用**环境**来替换马尔可夫决策过程。具体我们会在该章节再做说明。

1.3 策略评估

本节着重介绍两个内容**策略评估**（Policy Evaluation）以及**贝尔曼等式**（Bellman Equation）。要解决马尔科夫决策问题 $\max_\pi \rho(\pi)$，我们首先面对的就是一个子问题：给定任意一个马尔可夫策略函数 π，我们如何获得策略函数预计能得到的折扣累加奖励值 $\rho(\pi)$。这个问题也被叫做**策略评估**。通过对这个问题的研究，也能得到一个马尔科夫决策过程一个非常重要的概念**贝尔曼等式**，这个等式是强化学习最为重要的基础之一。

首先，引入一个非常重要的值函数——状态动作价值函数，在强化学习中，也被称为 Q-函数。

定义 1.9 (状态动作价值函数、Q 函数)**.** 首先我们来使用 $\mathcal{MDP}$ 来表示一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，其次使用 π 来表示一个马尔可夫策略

函数，那么马尔可夫决策过程将策略函数映射为一个马尔科夫链 $\mathcal{MDP}(\pi)$。令轨迹 $\tau = (s_0, a_0, s_1, a_1, \ldots, s_t, a_t, \ldots)$ 服从分布 $\mathcal{MDP}(\pi)$，那么，我们有状态动作价值函数

$$Q_\pi(s,a) = \mathbb{E}_{\tau \sim \mathcal{MDP}(\pi)} \left[\sum_{t=1}^{\infty} \gamma^t r_t \middle| s_0 = s, a_0 = a \right], \tag{16}$$

其中 $r_t = R(s_t, a_t)$。

Q 函数是来衡量马尔科夫链 $\mathcal{MDP}(\pi)$ 的轨迹集合中，轨迹开头确定的子集合的奖励值的期望。有了 Q 函数，我们可以简化策略评价函数 $\rho(\pi) = \mathbb{E}_{s_0 \sim \mathbf{p}_0, a_0 \sim \pi(s_0)}[Q_\pi(s,a)]$，也就是说，如果能求解一个马尔科夫链的 Q-函数，我们就能很方便地进行策略评估。

我们首先发现，Q_π 函数存在一个递归的自兼容的结构特性：

$$Q_\pi(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbf{P}(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') Q_\pi(s', a'). \tag{17}$$

这个关系揭示了马尔可夫决策过程的核心特点，是马尔可夫决策过程中的重要概念**贝尔曼等式**的基础。

要介绍贝尔曼等式，需要先引入一个操作——**贝尔曼操作**。

定义 1.10 (贝尔曼操作，Bellman Operator)**.** 对于一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，以及马尔可夫策略 π，我们可以定义一个基于此的贝尔曼操作：对于任意定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数 Q，我们可以对它进行一个转换操作，

$$T_\pi Q(s,a) = R(s,a) + \gamma \sum_{s' \in \mathcal{S}} \mathbf{P}(s'|s,a) \sum_{a' \in \mathcal{A}} \pi(a'|s') Q(s', a'). \tag{18}$$

其中 T_π 表示贝尔曼操作，而 $T_\pi Q$ 表示对 Q 进行贝尔曼操作后的结果，显然结果仍然是一个定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数。

要引出贝尔曼等式，我们还需要一个先验知识——巴拿赫不动点定理。

定理 1.3 (巴拿赫不动点定理，Banach Fixed-point Theorem)**.** 如果 U 是一个巴拿赫空间，并且 $T : U \to U$ 是一个 γ-收缩映射（即对于任意的 $\mathbf{u}, \mathbf{v} \in U$，满足 $\|T\mathbf{u} - T\mathbf{v}\| \leq \gamma \|\mathbf{u} - \mathbf{v}\|$。并且 $\|\cdot\|$ 指的是巴拿赫空间 U 使用的范数），其中 $\gamma \in (0,1)$ 是收缩因子。那么

- 在空间 U 中，存在唯一的不动点 $\mathbf{v}^*$ 满足 $T\mathbf{v}^* = \mathbf{v}^*$；

- *在空间 U 中，对于任意的点 $\mathbf{v}_0$，以及递推公式 $\mathbf{v}_{n+1} = T\mathbf{v}_n$ 所形成的数列 $(\mathbf{v}_n)$，存在极限 $\lim_{n\to\infty} \mathbf{v}_n = \mathbf{v}^*$。*

证明. 首先，我们证明 $(\mathbf{v}_n)$ 是一个柯西序列。对于任意的 $m \geq 1$,

$$\|\mathbf{v}_{n+m} - \mathbf{v}_n\| \leq \sum_{k=0}^{m-1} \|\mathbf{v}_{n+k+1} - \mathbf{v}_{n+k}\| = \sum_{k=0}^{m-1} \|T^{n+k}\mathbf{v}_1 - T^{n+k}\mathbf{v}_0\|$$
$$\leq \sum_{k=0}^{m-1} \gamma^{n+k}\|\mathbf{v}_1 - \mathbf{v}_0\| = \frac{\gamma^n(1-\gamma^m)}{1-\gamma}\|\mathbf{v}_1 - \mathbf{v}_0\|.$$

由上式可知，随着 n 增大，序列 $(\mathbf{v}_n)$ 的元素无限靠近，所以它是一个柯西序列，一定存在一个极限 $\mathbf{v}_\infty = \lim_{n\to\infty} \mathbf{v}_n$。又因为空间 U 是一个完全集（巴拿赫空间的性质），所以 $\mathbf{v}_\infty$ 一定在空间 U 中。

接着，我们来证明 $\mathbf{v}_\infty$ 是收缩映射 T 的不动点。

$$\begin{aligned}
&\|T\mathbf{v}_\infty - \mathbf{v}_\infty\| \\
\leq &\|T\mathbf{v}_\infty - \mathbf{v}_n\| + \|\mathbf{v}_n - \mathbf{v}_\infty\| \\
= &\|T\mathbf{v}_\infty - T\mathbf{v}_{n-1}\| + \|\mathbf{v}_n - \mathbf{v}_\infty\| \\
\leq &\gamma\|\mathbf{v}_\infty - \mathbf{v}_{n-1}\| + \|\mathbf{v}_n - \mathbf{v}_\infty\|
\end{aligned}$$

当 n 趋向于无穷时，我们可得 $\|T\mathbf{v}_\infty - \mathbf{v}_\infty\| \leq 0$。因为范数总是大于 0，最终我们可得：$\|T\mathbf{v}_\infty - \mathbf{v}_\infty\| = 0$，也就是说 ∞ 是收缩映射 T 的不动点。

最后，我们再证明收缩映射的不动点是唯一的。假设收缩映射 T 存在两个不动点 $\mathbf{u}$ 和 $\mathbf{v}$，那么我们可得：$\|\mathbf{u} - \mathbf{v}\| = \|T\mathbf{u} - T\mathbf{v}\| \leq \gamma\|\mathbf{u} - \mathbf{v}\|$。因为 $0 < \gamma < 1$，所以可得 $\mathbf{u} = \mathbf{v}$。 □

经过上面的准备，我们终于引出马尔可夫决策过程最关键的一个定理——**贝尔曼等式**。

定理 1.4 (贝尔曼等式，Bellman Equation)**.** *对于一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，马尔可夫策略 π，以及定义在 π 对应的马尔科夫链上的函数 Q_π 和贝尔曼操作 T_π：Q_π 是贝尔曼操作唯一的不动点。其中，不动点的关系就叫做贝尔曼等式:*

$$Q_\pi(s,a) = R(s,a) + \gamma \sum_{s'\in\mathcal{S}} \mathbf{P}(s'|s,a) \sum_{a'\in\mathcal{A}} \pi(a'|s')Q_\pi(s',a'). \tag{19}$$

如果我们定义 π 对应的马尔科夫链的状态转移矩阵为 $\mathbf{P}_\pi[(s',a')|(s,a)] = \mathbf{P}(s'|s,a)\pi(a'|s')$，那么贝尔曼等式也可以等价地写成：

$$Q_\pi = R + \gamma \mathbf{P}_\pi Q_\pi. \tag{20}$$

证明. 首先，根据 Q_π 函数的定义，我们可知 Q_π 满足上式，所以 Q_π 是贝尔曼操作 T_π 的不动点。其次，我们只要证明贝尔曼操作是一个收缩映射，我们就可以带入巴拿赫定理证明贝尔曼操作存在唯一的不动点。

设两个定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数 Q_1 和 Q_2，有关系：

$$\begin{aligned} &\|T_\pi Q_1 - T_\pi Q_2\| \\ =&\gamma \|\mathbf{P}_\pi Q_1 - \mathbf{P}_\pi Q_2\| \\ \leq&\gamma \|Q_1 - Q_2\|. \end{aligned}$$

上面这个等式利用了马尔科夫链地状态转移矩阵的谱不大于 1 的性质。因此，我们得到了贝尔曼操作是一个收缩映射。 □

注 1.4. 巴拿赫定理并没有要求映射函数对应的空间是离散的，所以当马尔可夫决策过程的状态集合 $\mathcal{S}$ 和动作集合 $\mathcal{A}$ 是不可数的紧致集时，贝尔曼操作与等式都换成了积分操作，但是依旧满足贝尔曼等式：Q_π 是贝尔曼操作 T_π 唯一的不动点。本文为了描述清楚概念，所以都是以状态集合和动作集合是离散并且有限的前提下进行论述的。这些结论很容易就能扩展到连续紧致的马尔可夫决策过程。

根据巴拿赫不动点定理，我们已经能够构造出两种算法来进行策略评估：

- 迭代法：首先任意构造一个定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数 Q_0。然后不断使用贝尔曼操作来将 Q_n 映射到 Q_{n+1}。经过多轮迭代后，Q_t 的变化会越来越小，并且会越来越接近 Q_π。在得到 Q_π 后，我们就能够评估出策略函数的价值了。
- 构造损失函数法：构造损失函数 $L(Q) = \|Q - T_\pi Q\|$ 来求解贝尔曼操作的不动点，获得的不动点就是要求的 Q_π。在得到 Q_π 后，我们就能评估出策略函数的价值了。

这两种方法的具体特性将在后面的章节中讨论到。

1.4 策略提升

当然，我们不能只局限于进行策略评估，我们最终还是想要获得最优策略 $\pi^* = \arg\max_\pi \rho(\pi)$。那么如何使策略 π 往 π^* 的方向提升呢？本节将主要介绍两个内容**策略提升**（Policy Improvement）和**最优贝尔曼等式**（Optimal Bellman Equation）。其中，最优贝尔曼等式是强化学习基本算法 Q-Learning 的理论基础，可以说是更加重要的一个定理。所以本节会侧重讲解最有贝尔曼等式的推导与理解。

首先我们先确定两个符号：使得奖励值最大的最优策略为 π^*，最优策略对应的值函数为 Q_{π^*}。最优策略以及对应的最优值函数应该满足的结构特性像上一节中一般策略函数与值函数的结构那样直观，我们需要一些迂回的研究策略。

首先，我们直觉性地定义一个贪婪的算子——**最优贝尔曼操作**。

定义 1.11 (最优贝尔曼操作，Optimal Bellman Operator)**.** *我们可以定义一个基于任意马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 的最优贝尔曼操作：对于任意定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数 Q，我们可以对它施加一个转换操作，*

$$\begin{aligned} TQ(s,a) =& \max_\pi T_\pi Q(s,a) \\ =& \max_\pi \left[R(s,a) + \gamma \sum_{s'} \mathbf{P}(s'|s,a) \sum_{a'} \pi(a'|s') Q(s',a') \right]. \end{aligned} \tag{21}$$

写成矩阵的形式就是 $TQ = \max_\pi[R + \gamma \mathbf{P}_\pi Q]$，其中 $\mathbf{P}_\pi[(s',a')|(s,a)] = \mathbf{P}(s'|s,a)\pi(a'|s')$。

最优贝尔曼操作就是在每一步贝尔曼操作中，取得对当前步骤最有益处的策略进行转换。这是一个贪婪的操作，只关注于如何将当前 Q 值最大化。接下来将介绍这个操作的特性，以及这个操作如何帮助我们学到最优的策略函数 π^*。

首先，最优贝尔曼操作也是一个收缩映射。

引理 1.1. *基于马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 的最优贝尔曼操作是一个 γ-收缩映射。*

证明. 首先设两个定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数 Q_1 和 Q_2，$\pi_1 \in \arg\max_\pi T_\pi Q_1$ 以及 $\pi_2 \in \arg\max_\pi T_\pi Q_2$。那么

$$TQ_1 - TQ_2 = T_{\pi_1} Q_1 - T_{\pi_2} Q_2$$

$$\begin{aligned}\preceq & T_{\pi_1}Q_1 - T_{\pi_1}Q_2\\ = & \gamma\mathbf{P}_{\pi_1}(Q_1 - Q_2),\end{aligned}$$

以及

$$\begin{aligned}TQ_2 - TQ_1 = & T_{\pi_2}Q_2 - T_{\pi_1}Q_1\\ \preceq & T_{\pi_2}Q_2 - T_{\pi_2}Q_1\\ = & \gamma\mathbf{P}_{\pi_2}(Q_2 - Q_1).\end{aligned}$$

当 $s \in \mathcal{M} = \{s : TQ_1(s) \geq TQ_2(s)\}$ 时，

$$0 \leq TQ_1(s) - TQ_2(s) \leq \gamma \sum_{s' \in \mathcal{S}} \mathbf{P}_{\pi_1}(s'|s)[Q_1(s') - Q_2(s')].$$

当 $s \in \mathcal{N} = \{s : TQ_1(s) < TQ_2(s)\}$ 时，

$$0 \leq TQ_2(s) - TQ_1(s) \leq \gamma \sum_{s' \in \mathcal{S}} \mathbf{P}_{\pi_2}(s'|s)[Q_2(s') - Q_1(s')].$$

我们人为构建一个状态转移矩阵 $\mathbf{P}_{\pi_1,\pi_2}$ 满足：当 $s \in \mathcal{M}$ 时 $\mathbf{P}_{\pi_1,\pi_2}(\cdot|s) = \mathbf{P}_{\pi_1}(\cdot|s)$；当 $s \in \mathcal{N}$ 时 $\mathbf{P}_{\pi_1,\pi_2}(\cdot|s) = \mathbf{P}_{\pi_2}(\cdot|s)$。那么

$$\begin{aligned}\|TQ_1 - TQ_2\| \leq & \gamma\|\mathbf{P}_{\pi_1,\pi_2}(Q_1 - Q_2)\|\\ \leq & \gamma\|Q_1 - Q_2\|.\end{aligned}$$

□

因为最优贝尔曼操作是一个 γ-收缩映射，所以我们可以根据巴拿赫不动点定理得知最优贝尔曼操作一定存在一个唯一的不动点 Q^*。那么，Q^* 和 Q_{π^*} 是什么关系呢？这就是最优贝尔曼等式要解决的问题。

定理 1.5 (最优贝尔曼等式，Optimal Bellman Equation)**.** *设一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，它的策略评价函数为 $\rho(\pi)$，它的最优策略为 $\pi^* = \arg\max_\pi \rho(\pi)$，以及最优策略对应的值函数为 Q_{π^*}。我们有*

$$Q_{\pi^*}(s,a) = \max_\pi \left[R(s,a) + \gamma \sum_{s'} \mathbf{P}(s'|s,a) \sum_{a'} \pi(a'|s') Q_{\pi^*}(s,a)\right]. \quad (22)$$

写成矩阵的形式就是 $Q_{\pi^*} = \max_\pi[R + \gamma\mathbf{P}_\pi Q_{\pi^*}]$。

证明. 根据前面的描述，我们仅需要证明的关键步骤就是最优贝尔曼操作的不动点 $Q^* = Q_{\pi^*}$。我们证明的整体步骤分为两步：第一步证明当 $Q \succeq TQ$ 时，$Q \succeq Q_{\pi^*}$，第二步证明当 $Q \preceq TQ$ 时，$Q \preceq Q_{\pi^*}$。

我们先论述上面两个步骤为什么能够证明 $Q^* = Q_{\pi^*}$。我们从定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数集合中找出两类特殊的子集，$\mathcal{M} = \{Q : Q \succeq TQ\}$ 和 $\mathcal{N} = \{Q : Q \preceq TQ\}$。我们将证明集合 $\mathcal{M}$ 中的元素一定都不大于 Q_{π^*}，而集合 $\mathcal{N}$ 中的元素一定都不小于 Q_{π^*}。又因为 $\mathcal{M} \cap \mathcal{N} = \{Q : Q = TQ\} = \{Q^*\}$，所以 $Q_{\pi^*} \preceq Q^* \preceq Q_{\pi^*}$，即 $Q_{\pi^*} = Q^*$。

首先证明：当 $Q \succeq TQ$ 时，$Q \succeq Q_{\pi^*}$。

$$
\begin{aligned}
Q \succeq & TQ \succeq T_\pi Q = R + \gamma \mathbf{P}_\pi Q \\
\succeq & R + \gamma \mathbf{P}_\pi (R + \gamma \mathbf{P}_\pi Q) \\
& \vdots \\
\succeq & \lim_{K \to \infty} \sum_{t=0}^{K} (\gamma \mathbf{P}_\pi)^t R + (\gamma \mathbf{P}_\pi)^{K+1} Q \\
= & Q_\pi .
\end{aligned}
$$

又因为上式对所有的 π 都成立，所以也包括最优的策略函数，即 $Q \succeq Q_{\pi^*}$。

接着证明：当 $Q \preceq TQ$ 时，$Q \preceq Q_{\pi^*}$。设 $\pi_Q = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q$。那么，

$$
\begin{aligned}
Q \preceq & TQ = R + \gamma \mathbf{P}_{\pi_Q} Q \\
\preceq & R + \gamma \mathbf{P}_{\pi_Q} (R + \gamma \mathbf{P}_{\pi_Q} Q) \\
& \vdots \\
\preceq & \lim_{K \to \infty} \sum_{t=0}^{K} (\gamma \mathbf{P}_{\pi_Q})^t R + (\gamma \mathbf{P}_{\pi_Q})^{K+1} Q = Q_{\pi_Q} \\
\preceq & Q_{\pi^*} .
\end{aligned}
$$

□

我们根据最优贝尔曼等式，就可以构造出两种求解最优策略的方法：

- 迭代法：构造一个定义在 $\mathcal{S} \times \mathcal{A}$ 上的策略函数 Q_0。然后，不断使用最优贝尔曼操作来将 Q_n 映射到 Q_{n+1}。经过多轮迭代后，Q_t 的变化会越

来越小，并且会越来越接近 Q^*。在得到 Q^* 后，再进行一步最优贝尔曼操作，求解 $\arg\max_\pi R + \gamma \mathbf{P}_\pi Q^*$，就能求出最优的策略来。

- 构造损失函数法：构造损失 $L(Q) = \|Q - TQ\|$ 来求解最优贝尔曼操作的不动点，最终获得的不动点就是 Q^*。在得到 Q^* 后，再进行 一步最优贝尔曼操作，求解 $\arg\max_\pi R + \gamma \mathbf{P}_\pi Q^*$，就能求出最优的策略来。

这两种方法存在着各自的优势与劣势，具体的特性分析将在后面的章节中讨论。

1.5 关于广义策略的讨论

在前面的章节中，本文默认在马尔科夫策略中讨论马尔科夫决策过程。本小节从两个方向来扩展策略：第一个方向是具有随机性的马尔科夫策略 $\pi_{MS}(a_t|s_t)$（Markovian Stochastic Policy）退化成确定性的马尔科夫策略 $a_t = \pi_{MD}(s_t)$（Markovian Deterministic Policy）；第二个方向是将马尔科夫策略 $\pi_{MS}(a_t|s_t)$ 扩展成历史随机策略 $\pi_{HS}(a_t|s_t, a_{t-1}, \dots a_0, s_0)$（Historical Stochastic Policy）。 我们将证明，在马尔科夫随机策略集合中找到的最优策略，在马尔科夫确定策略中找到的最优策略，以及在历史随机策略中找的最优策略所能获得的奖励值是一样的。

首先，在策略提升的方法中，本文介绍了使用最优贝尔曼操作来学习一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 的最优策略 $\pi^* = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q^*$。这是一个凸集合上的线性优化问题， 所以它的最优值可以取在凸集合的边界点上，也就是说最优策略可以是 $\pi^*(s) \in \arg\max_a Q^*(s, a)$ 的马尔科夫确定策略。所以马尔科夫随机策略集合中找到的最优策略和马尔科夫确定策略集合中找到的最优策略是等价的。

注 1.5. 最优策略的形式也可以是最优马尔可夫确定策略的凸组合形成的马尔可夫随机策略。

接着，我们来证明第二个结论。

对于任意的一个历史随机策略 $\pi_{HS}(a_t|s_t, a_{t-1}, \dots, a_0, s_0)$，我们可以构造一个与它相关的马尔科夫随机策略

$$\pi_{MS}(a_t|s_t) = \int_{\tau_{t-1}=(s_0,a_0,\dots,s_{t-1},a_{t-1})} \pi_{HS}(a_t|s_t, \tau_{t-1}) \mathrm{d}\tau_{t-1}, \tag{23}$$

并且 $\pi_{MS}(a_0|s_0) = \pi_{HS}(a_0|s_0)$。又因为

$$\rho(\pi_{HS}) = \sum_{s_0} \mathbf{p}_0(s_0) \sum_{a_0} \pi_{HS}(a_0|s_0) Q_{\pi_{HS}}(s_0, a_0), \tag{24}$$

和

$$\rho(\pi_{MS}) = \sum_{s_0} \mathbf{p}_0(s_0) \sum_{a_0} \pi_{MS}(a_0|s_0) Q_{\pi_{MS}}(s_0, a_0), \tag{25}$$

所以如果对于所有的 s_0, a_0 满足 $Q_{\pi_{HS}}(s_0, a_0) = Q_{\pi_{MS}}(s_0, a_0)$ 的话，那么我们就能证明 $\rho(\pi_{HS}) = \rho(\pi_{MS})$。又因为

$$Q_\pi(s_0, a_0) = \sum_{t=0}^{\infty} \sum_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}} \gamma^t R(s, a) \mathbb{P}_\pi(S_t = s, A_t = a | S_0 = s_0, A_0 = a_0), \tag{26}$$

所以我们只需要证明：

$$\begin{aligned} &\mathbb{P}_{\pi_{HS}}(S_t = s, A_t = a | S_0 = s_0, A_0 = a_0) \\ =& \mathbb{P}_{\pi_{MS}}(S_t = s, A_t = a | S_0 = s_0, A_0 = a_0). \end{aligned} \tag{27}$$

我们使用数学归纳法来证明上式。首先 $t = 0$ 必然成立，其次我们假设 t 时成立，那么

$$\begin{aligned} &\mathbb{P}_{\pi_{MS}}(S_{t+1} = s, A_{t+1} = a | S_0 = s_0, A_0 = a_0) \\ =& \sum_{s', a'} \mathbb{P}_{\pi_{MS}}(S_{t+1} = s, A_{t+1} = a | S_t = s', A_t = a', S_0 = s_0, A_0 = a_0) \\ &\cdot \mathbb{P}_{\pi_{MS}}(S_t = s', A_t = a' | S_0 = s_0, A_0 = a_0) \\ =& \sum_{s', a'} \mathbb{P}_{\pi_{MS}}(S_{t+1} = s, A_{t+1} = a | S_t = s', A_t = a') \\ &\cdot \mathbb{P}_{\pi_{MS}}(S_t = s', A_t = a' | S_0 = s_0, A_0 = a_0) \\ =& \sum_{s', a'} \mathbb{P}_{\pi_{HS}}(S_{t+1} = s, A_{t+1} = a | S_t = s', A_t = a') \\ &\cdot \mathbb{P}_{\pi_{HS}}(S_t = s', A_t = a' | S_0 = s_0, A_0 = a_0) \\ =& \mathbb{P}_{\pi_{HS}}(S_{t+1} = s, A_{t+1} = a | S_0 = s_0, A_0 = a_0). \end{aligned} \tag{28}$$

综上，我们得到了结论：任意一个历史随机策略总能对应一个等效的马尔科夫随机策略，所以

$$\max_{\pi_{HS}} \rho(\pi_{HS}) \leq \max_{\pi_{MS}} \rho(\pi_{MS}). \tag{29}$$

又因为历史随机策略集合包含马尔科夫随机策略集合，所以

$$\max_{\pi_{HS}} \rho(\pi_{HS}) \geq \max_{\pi_{MS}} \rho(\pi_{MS}). \tag{30}$$

综上所述，可以得到 $\max_{\pi_{HS}} \rho(\pi_{HS}) = \max_{\pi_{MS}} \rho(\pi_{MS})$.

1.6 本章小结

本章节主要介绍了强化学习算法所使用的基本数学模型——马尔科夫决策模型。本章节从马尔科夫链切入，介绍了马尔科夫链的基本定义与特性，然后从概率测度的角度阐释了马尔科夫链的本质。接着本文引入了马尔可夫决策过程，将其分成了狭义的马尔科夫决策过程与马尔科夫决策问题两个部分。马尔可夫决策过程描述了一个环境的基本动力学特征，从抽象角度来看，它是将马尔可夫策略函数映射成一个马尔科夫链的泛函映射。接着，本章节介绍了贝尔曼等式与最优贝尔曼等式，分别用于进行策略评估和策略提升。本章节基本奠定了强化学习的理论基础，是非常重要的一个章节。本章节同时补充讨论了马尔可夫随机策略、马尔可夫确定策略和历史随机策略三个集合对应的马尔可夫决策问题的最优价值是等价的，从而证明了在马尔可夫随机策略集合中求解马尔可夫决策问题的合理性，以及揭示了使用马尔可夫确定策略集合来简化马尔可夫决策问题的可能。

2 基于最优贝尔曼等式的算法

本章内容继续探讨马尔科夫决策过程，主要侧重于介绍一类求解最优策略的具体算法——基于最优贝尔曼等式的算法。基于最优贝尔曼等式的算法主体上分为两大类：一类是基于模型的算法，另一类是无模型的算法。而无模型的算法通常也被称为强化学习算法。基于模型的算法又分为：值迭代算法（Value Iteration）、策略迭代算法（Policy Iteration）以及改进策略迭代算法（Modifited Policy Iteration）[34]，它们都使用了最优贝尔曼等式来提升策略的性能。而无模型的算法本章节介绍的则是重要的 Q-Learning 算法 [48, 31, 32]。本论文通过对这些算法的介绍，来完善整个马尔科夫决策过程理论，为本论文的算法打下坚实的基石。

2.1 值迭代算法

在马尔科夫决策过程领域，**值迭代算法**（Policy Iteration）是使用最广，研究最为充分的算法。它有着非常多的别名：逐次逼近法，超松弛法，逆向归纳法，以及动态规划法 [16]。它应用这么广，可能是因为它的概念非常简单，代码也非常容易实现。这个算法背后蕴含的思想同时也应用在很多其他的数学领域 [12]。我们就从介绍值迭代法的最基本形式开始，它虽然并不是最高效的形式，但是十分简单并且容易分析。

值函数算法一个求解最优值函数 Q^* 的数值估计方法，所以它求解到的是 Q^* 某个精度的近似解。这里有一个假设：当值函数 Q 非常接近的时候，对应的策略函数 $\pi_Q = \arg\max_\pi T_\pi Q$ 可能是完全相同的。也就是说，我们只要求到最优值函数 Q^* 某个精度范围内的近似解，我们可能就已经能获得最优的策略函数了，所以这个值函数算法是能够完全精确地求解一个马尔科夫决策问题的。

从上面值迭代法来看，值迭代法实际上就是对一个初始化的值函数 Q 不断进行最优贝尔曼操作。所以，值迭代法其实是构造了一个收敛到最优值函数的柯西序列 $\{Q_n\}$。接下来的引理就揭示了这个柯西序列收敛到最优值函数的速度。

定理 2.1 (值迭代法的收敛率)**.** *使用值迭代法来求解任意一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 的最优策略时构造的值函数序列 $\{Q_n\}$ 满足：*

$$\|Q_{n+1} - Q^*\| \leq \gamma \|Q_n - Q^*\|, \tag{31a}$$

算法 1: 值迭代算法

Input: 要求精度 ϵ, 以及一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$

Output: 最优策略值函数 Q^*，以及最优策略 π^*

随机初始化一个值函数 Q, 一个策略函数 π;
while *True* **do**
  $Q' = Q$;
  $\pi = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q$;
  $Q = R + \gamma \mathbf{P}_\pi Q'$;
  if $\|Q' - Q\| \leq \epsilon(1-\gamma)/(2\gamma)$ **then**
    break;
  end
end
return Q, π

$$\|Q_{n+1} - Q^*\| \leq \frac{\gamma}{1-\gamma}\|Q_{n+1} - Q_n\|, \tag{31b}$$

$$\|Q_{n+1} - Q^*\| \leq \frac{\gamma^n}{1-\gamma}\|Q_1 - Q_0\|. \tag{31c}$$

证明. 关于于第一个不等式的证明:

$$\|Q_{n+1} - Q^*\| = \|TQ_n - TQ^*\| \leq \gamma\|Q_n - Q^*\|.$$

关于第二个不等式的证明:

$$\begin{aligned}
&\|Q_{n+1} - Q^*\| \\
=&\|Q_{n+1} - TQ_{n+1} + TQ_{n+1} - Q^*\| \\
\leq&\|Q_{n+1} - TQ_{n+1}\| + \|TQ_{n+1} - Q^*\| \\
=&\|TQ_n - TQ_{n+1}\| + \|TQ_{n+1} - TQ^*\| \\
\leq&\gamma\|Q_n - Q_{n+1}\| + \gamma\|Q_{n+1} - Q^*\|,
\end{aligned}$$

将不等式左右两边整理可得:

$$(1-\gamma)\|Q_{n+1} - Q^*\| \leq \gamma\|Q_n - Q_{n+1}\|.$$

关于第三个不等式的证明：

$$
\begin{aligned}
&\|Q_{n+1} - Q^*\| \\
\leq &\frac{\gamma}{1-\gamma}\|Q_n - Q_{n+1}\| \\
= &\frac{\gamma}{1-\gamma}\|TQ_{n-1} - TQ_n\| \\
\leq &\frac{\gamma^2}{1-\gamma}\|Q_{n-1} - Q_n\| \\
&\vdots \\
\leq &\frac{\gamma^n}{1-\gamma}\|Q_0 - Q_1\|.
\end{aligned}
$$

□

从上面的引理可知，值迭代法的终止条件 $\|Q_{n+1} - Q_n\| \leq \epsilon(1-\gamma)/(2\gamma)$ 可以保证最终求得的值函数的精度满足要求 ϵ。

2.2 策略迭代算法

策略迭代算法（Policy Iteration）是在策略空间上进行优化的算法。它的收敛率远大于值迭代算法，但是单步的计算复杂度也远大与值迭代算法，它提供了一个很好的提升值迭代算法收敛率的方向。下一小节将介绍一个把两者结合的平衡了计算复杂度与收敛速度的新的算法——改进策略迭代算法。

从策略迭代算法的流程可以看到，它并不需要接收一个精度。一旦策略满足最优贝尔曼等式时，策略迭代算法将返回这个策略。因此策略迭代算法返回的策略值一定是输入的马尔科夫决策过程的最优策略。其中，最令人疑惑的问题是：这个算法能够终止吗？我们将在接下来进行严格的数学论述。

我们先来详细解释一下策略迭代算法的具体流程。首先，策略迭代算法生成了两个序列 $\{\pi_n\}$ 和 $\{Q_n\}$。我们添加上序列下标，可以得到策略迭代算法的两个关键步骤：$Q_n = (\mathbf{I}-\gamma\mathbf{P}_{\pi_n})^{-1}R$，以及 $\pi_{n+1} = \arg\max_\pi R+\gamma\mathbf{P}_\pi Q_n$。其中第一个操作就是对策略进行估计： 根据贝尔曼等式 $Q_\pi = R + \gamma\mathbf{P}_\pi Q_\pi$ 直接可得 $Q_\pi = (\mathbf{I} - \gamma\mathbf{P}_\pi)^{-1}R$。也就是说在策略迭代算法中 $Q_n = Q_{\pi_n}$。而第二个操作就是对 Q_n 进行一步最优贝尔曼操作，使策略 π 往最优策略的方向移动一步。

算法 2: 策略迭代算法

Input: 一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$

Output: 最优策略值函数 Q^*，以及最优策略 π^*

随机初始化一个值函数 Q, 一个策略函数 π;
while *True* **do**
  $\pi' = \pi$;
  $Q = (\mathbf{I} - \gamma \mathbf{P}_\pi)^{-1} R$;
  $\pi = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q$;
  if $\pi == \pi'$ **then**
    break;
  end
end
return Q, π

对于策略迭代算法的分析比值迭代算法要复杂一些，本文会介绍两个关于策略迭代算法的性质，来帮助对策略迭代算法进行分析。

定义一个与最优贝尔曼操作有关的新算子 U 满足 $UQ = TQ - Q = \max_\pi R + \gamma \mathbf{P}_\pi Q - Q$。那么，

引理 2.1. *任意的 Q_1 和 Q_2 以及 π_{Q_1} 和 π_{Q_2}，如果满足 $\pi_{Q_i} = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q_i$。我们有*

$$UQ_1 \succeq UQ_2 + (\gamma \mathbf{P}_{\pi_{Q_2}} - \mathbf{I})(Q_1 - Q_2). \tag{32}$$

证明. 已知

$$\begin{aligned} UQ_1 =& \max_\pi R + \gamma \mathbf{P}_{\pi_{Q_1}} Q_1 - Q_1 \\ =& R + \gamma \mathbf{P}_{\pi_{Q_1}} Q_1 - Q_1 \\ \succeq& R + \gamma \mathbf{P}_{\pi_{Q_2}} Q_1 - Q_1, \end{aligned}$$

以及

$$UQ_2 = R + \gamma \mathbf{P}_{\pi_{Q_2}} Q_2 - Q_2,$$

我们将两式相减可得引理结论。 □

根据上面这个引理，我们可以知道算子 U 的“次梯度”为 $\gamma \mathbf{P}_{\pi_Q} - \mathbf{I}$，其中 $\pi_Q = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q$ 。

引理 2.2. 关于一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，策略迭代算法可以产生两个序列：$\{\pi_n\}$，以及 $\{Q_n\}$。设最优值函数为 Q^*，那么 $\{Q_n\}$ 是绝对单调非减的以及极限为 Q^*，即 $Q_0 \preceq Q_1 \preceq \ldots Q_n \preceq \ldots \preceq Q^*$，并且 $\lim_{n\to\infty} Q_n = Q^*$。

证明. 定义序列 $\{a_n\}$ 满足：$a_0 = Q_0$，并且 $a_{n+1} = Ta_n = \max_\pi R + \gamma \mathbf{P}_\pi a_n$，也就是从 a_0 进行值迭代算法而获得的序列。然后我们用数学归纳法来证明 $a_n \preceq Q_n \preceq Q^*$。

当 $n = 0$ 时，满足条件 $a_0 = Q_0 \preceq Q^*$。当 $n \geq 0$ 时，假设满足 $a_n \preceq Q_n \preceq Q^*$。接着有如下关系：

$$\begin{aligned} Q_{n+1} =& (\mathbf{I} - \lambda \mathbf{P}_{\pi_{n+1}})^{-1} R \\ =& Q_n + (\mathbf{I} - \lambda \mathbf{P}_{\pi_{n+1}})^{-1}(R + \lambda \mathbf{P}_{\pi_{n+1}} Q_n - Q_n) \\ \succeq& Q_n + (R + \lambda \mathbf{P}_{\pi_{n+1}} Q_n - Q_n) \\ =& R + \lambda \mathbf{P}_{\pi_{n+1}} Q_n = TQ_n \succeq Ta_n = a_{n+1}. \end{aligned}$$

这里用到了两个结论。一个是当 $Q \succeq 0$ 时，$(\mathbf{I} - \lambda \mathbf{P}_\pi)^{-1} Q = Q + (\lambda \mathbf{P}_\pi) Q + \ldots + (\lambda \mathbf{P}_\pi)^n Q + \ldots \succeq Q$。另一个是当 $Q_n \succeq a_n$ 时，设 $\pi_{Q_n} = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q_n$ 以及 $\pi_{a_n} = \arg\max_\pi R + \gamma \mathbf{P}_\pi a_n$，那么

$$TQ_n \succeq R + \gamma \mathbf{P}_{\pi_{a_n}} Q_n \succeq Ta_n.$$

因为 $\{a_n\}$ 的极限为 Q^*，所以 $\{Q_n\}$ 的极限大于 Q^*。又因为 $Q_n \preceq Q^*$，所以 $\{Q_n\}$ 的极限为 Q^*。

接着我们证明 $\{Q_n\}$ 的单调性。首先，因为 $Q_n = Q_{\pi_n}$，我们有

$$Q_n = R + \gamma \mathbf{P}_{\pi_n} Q_n \preceq R + \gamma \mathbf{P}_{\pi_{n+1}} Q_n,$$

所以

$$Q_n \preceq (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})^{-1} R = Q_{n+1}.$$

□

定理 2.2 (策略迭代算法的收敛率)**.** 使用策略迭代法来求解任意一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 的最优策略时构造的值函数序列 $\{\pi_n\}$ 和 $\{Q_n\}$ 满足：

$$\|Q_{n+1} - Q^*\| \leq \frac{\gamma}{1-\gamma} \|(\mathbf{P}_{\pi_{n+1}} - \mathbf{P}_{\pi^*})(Q_n - Q^*)\|. \tag{33}$$

而且，如果存在 K 使得所使用的策略 π 满足 $\|\mathbf{P}_\pi - \mathbf{P}_{\pi^*}\| \leq K\|Q_\pi - Q^*\|$，那么

$$\|Q_{n+1} - Q^*\| \leq \frac{\gamma}{1-\gamma}\|Q_n - Q^*\|^2. \tag{34}$$

证明. 根据引理 2.1 可得

$$\begin{aligned}
UQ_n \succeq & UQ^* + (\gamma\mathbf{P}_{\pi^*} - \mathbf{I})(Q_n - Q^*) \\
= & TQ^* - Q^* + (\gamma\mathbf{P}_{\pi^*} - \mathbf{I})(Q_n - Q^*) \\
= & (\gamma\mathbf{P}_{\pi^*} - \mathbf{I})(Q_n - Q^*) \\
= & (\mathbf{I} - \gamma\mathbf{P}_{\pi^*})(Q^* - Q_n).
\end{aligned}$$

接着有

$$\begin{aligned}
& Q^* - Q_{n+1} \\
= & Q^* - (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}R \\
= & Q^* - Q_n + Q_n - (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}R \\
= & Q^* - Q_n + (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}[(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})Q_n - R] \\
= & Q^* - Q_n + (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}[Q_n - TQ_n] \\
= & Q^* - Q_n + (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}(-UQ_n) \\
\preceq & Q^* - Q_n - (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}(\mathbf{I} - \gamma\mathbf{P}_{\pi^*})(Q^* - Q_n) \\
= & (\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}(\gamma\mathbf{P}_{\pi^*} - \gamma\mathbf{P}_{\pi_{n+1}})(Q^* - Q_n)
\end{aligned}$$

有因为 $Q_{n+1} \preceq Q^*$，所以我们可得

$$\begin{aligned}
\|Q^* - Q_{n+1}\| \leq & \|(\mathbf{I} - \gamma\mathbf{P}_{\pi_{n+1}})^{-1}(\gamma\mathbf{P}_{\pi^*} - \gamma\mathbf{P}_{\pi_{n+1}})(Q^* - Q_n)\| \\
\leq & \frac{\gamma}{1-\gamma}\|(\mathbf{P}_{\pi^*} - \mathbf{P}_{\pi_{n+1}})(Q^* - Q_n)\|.
\end{aligned}$$

□

从上面的引理可得，如果满足假设：存在 K 使得使用到的策略 π 满足 $\|\mathbf{P}_\pi - \mathbf{P}_{\pi^*}\| \leq K\|Q_\pi - Q^*\|$，那么策略迭代算法的收敛率可以提升到二次收敛速度。在通常的实验中，却是能够发现策略迭代算法要快于值迭代算法。

2.3 改进策略迭代算法

本小节开始介绍一个结合了值迭代算法和策略迭代算法共同特点的算法——**改进策略迭代算法**（Modified Policy Iteration）。

从前文可知，值迭代算法计算简单但是收敛速度慢，而策略迭代算法计算复杂但是收敛速度快。在策略迭代算法中有一步 $Q_n = (\mathbf{I} - \gamma \mathbf{P}_{\pi_n})^{-1} R$ 需要求解矩阵的逆，通常的计算复杂度为 $O(N^3)$。

策略迭代算法还是有非常大的改进空间的。在策略迭代算法中 $Q_n = (\mathbf{I} - \gamma \mathbf{P}_{\pi_n})^{-1} R$ 实际上是用来求解 $\pi_{n+1} = \max_\pi R + \gamma \mathbf{P}_\pi Q_n$ 的，并且算法的最终目的也是求解一个最优的策略 π^*。而我们往往会发现，当 Q 值足够接近的时候，$\pi_Q = \max_\pi R + \gamma \mathbf{P}_\pi Q$ 会是相同的。也就是说，我们其实不需要完全精确地求解 Q_n，而只需要求解一个近似的结果。我们注意到

$$Q_n = (\mathbf{I} - \gamma \mathbf{P}_{\pi_n})^{-1} R = \sum_{m=0}^{\infty} (\gamma \mathbf{P}_{\pi_n})^m R, \tag{35}$$

所以很直接地，我们就能够想到一个 Q_t 的近似估计:

$$\tilde{Q}_n = \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_n})^m R + (\gamma \mathbf{P}_{\pi_n})^{M+1} \tilde{Q}_{n-1}, \tag{36}$$

其中 $M \geq 0$。其实我们也就得到了改进策略迭代算法。

算法 3: 改进策略迭代算法

Input: 要求精度 ϵ, 以及一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$

Output: 最优策略值函数 Q^*，以及最优策略 π^*

随机初始化一个值函数 Q, 一个策略函数 π;
while *True* **do**
    $Q' = Q$;
    $Q = \max_\pi R + \gamma \mathbf{P}_\pi Q$;
    if $\|Q' - Q\| \leq \epsilon(1-\gamma)/(2\gamma)$ **then**
        break;
    end
    $\pi = \arg\max_\pi R + \gamma \mathbf{P}_\pi Q'$;
    $Q = \sum_{m=0}^{M} (\gamma \mathbf{P}_\pi)^m R + (\gamma \mathbf{P}_\pi)^{M+1} Q'$;
end
return Q, π

首先，我们仍然需要一个引理来证明改进策略迭代法的一个性质，它有助于改进策略迭代法收敛率的证明。

引理 2.3. 输入一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 我们使用改进策略迭代法求解它的最优策略时, 会构造的一个序列 $\{Q_n\}$。当 $Q_0 \preceq Q^*$ 时, $\{Q_n\}$ 是严格单调非减的, 并且存在极限等于 Q^*。

证明. 定义序列 $\{a_n\}$ 满足 $a_0 = Q_0$ 并且 $a_{n+1} = \max_\pi R + \gamma \mathbf{P}_\pi a_n$。再定义序列 $\{b_n\}$ 满足 $b_0 = Q_0$ 并且 $b_{n+1} = \max_\pi \sum_{m=0}^{M} (\gamma \mathbf{P}_\pi)^m R + (\gamma \mathbf{P}_\pi)^{M+1} b_n$。

首先我们可知 $\{a_n\}$ 和 $\{b_n\}$ 都是由值迭代法产生的序列, 所以它们的极限都为 Q^*。接下来我们用数学归纳法来证明 $a_n \preceq Q_n \preceq b_n$。当 $n = 0$ 时满足 $a_0 = Q_0 = b_0$。假设 $a_n \preceq Q_n \preceq b_n$, 那么首先比较好证明 $b_{n+1} \succeq Q_{n+1}$:

$$\begin{aligned} b_{n+1} =& \max_\pi \sum_{m=0}^{M} (\gamma \mathbf{P}_\pi)^m R + (\gamma \mathbf{P}_\pi)^{M+1} b_n \\ \succeq& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} b_n \\ \succeq& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} Q_n = Q_{n+1}. \end{aligned}$$

另外

$$\begin{aligned} & Q_{n+1} - a_{n+1} \\ =& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} Q_n - T a_n \\ \succeq& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} Q_n - T Q_n \\ =& \sum_{m=1}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} Q_n - \gamma \mathbf{P}_{\pi_{n+1}} Q_n \\ =& \sum_{m=1}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m [R + \gamma \mathbf{P}_{\pi_{n+1}} Q_n - Q_n] \succeq 0. \end{aligned}$$

综上, 我们证明了 $a_{n+1} \succeq Q_{n+1} \succeq b_{n+1}$。所以 $\{Q_n\}$ 收敛到 Q^*。

接着我们证明当 $Q_0 \succeq Q^*$ 时, $\{Q_n\}$ 严格单调非减。

$$\begin{aligned} & Q_{n+1} - Q_n \\ =& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} Q_n - Q_n \end{aligned}$$

$$= \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (R + \gamma \mathbf{P}_{\pi_{n+1}} Q_n - Q_n) \succeq 0.$$

所以我们可得：当 $Q_0 \preceq Q^*$ 时，$Q_0 \preceq Q_1 \preceq \ldots \preceq Q_n \preceq \ldots \preceq Q^*$。 □

定理 2.3 (改进策略迭代法的收敛率)**.** 使用改进策略迭代法来求解任意一个马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 的最优策略时构造的序列 $\{Q_n\}$ 和 $\{\pi_n\}$ 满足：如果 $Q_0 \preceq Q^*$，那么

$$\|Q_{n+1} - Q^*\| \leq \left[\frac{\gamma(1-\gamma^{M+1})}{1-\gamma} \|\mathbf{P}_{\pi_{n+1}} - \mathbf{P}_{\pi^*}\| + \gamma^{M+1} \right] \|Q_n - Q^*\|. \quad (37)$$

证明.

$$\begin{aligned}
Q^* - Q_{n+1} =& Q^* - \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m R - (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} Q_n \\
=& Q^* - Q_n - \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m [R + \gamma \mathbf{P}_{\pi_{n+1}} Q_n - Q_n] \\
=& Q^* - Q_n - \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (U Q_n) \\
\preceq& Q^* - Q_n - \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (\mathbf{I} - \gamma \mathbf{P}_{\pi^*})(Q^* - Q_n) \\
=& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (\mathbf{I} - \gamma \mathbf{P}_{\pi_{n+1}})(Q^* - Q_n) \\
& + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} (Q^* - Q_n) \\
& - \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (\mathbf{I} - \gamma \mathbf{P}_{\pi^*})(Q^* - Q_n) \\
=& \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (\gamma \mathbf{P}_{\pi^*} - \gamma \mathbf{P}_{\pi_{n+1}})(Q^* - Q_n) \\
& + (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1} (Q^* - Q_n).
\end{aligned}$$

又因为 $Q^* \succeq Q_{n+1}$ 所以

$$\begin{aligned}
& \|Q^* - Q_{n+1}\| \\
\leq& \Big\| \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (\gamma \mathbf{P}_{\pi^*} - \gamma \mathbf{P}_{\pi_{n+1}})(Q^* - Q_n)
\end{aligned}$$

$$
\begin{aligned}
&+ (\gamma \mathbf{P}_{\pi_{n+1}})^{M+1}(Q^* - Q_n)\big\| \\
\leq &\big\| \sum_{m=0}^{M} (\gamma \mathbf{P}_{\pi_{n+1}})^m (\gamma \mathbf{P}_{\pi^*} - \gamma \mathbf{P}_{\pi_{n+1}})(Q^* - Q_n)\big\| \\
&+ \big\|(\gamma \mathbf{P}_{\pi_{n+1}})^{M+1}(Q^* - Q_n)\big\| \\
= &\left[\frac{\gamma(1-\gamma^{M+1})}{1-\gamma}\|\mathbf{P}_{\pi_{n+1}} - \mathbf{P}_{\pi^*}\| + \gamma^{M+1}\right]\|Q_n - Q^*\|.
\end{aligned}
$$

□

修正策略迭代法是介于值策略迭代与策略迭代算法之间的一个算法，当 $M = 0$ 时，修正策略迭代算法就退化为值迭代算法，收敛率也退化为值迭代算法的收敛率；当 $M \to \infty$ 时，修正策略迭代算法就变成了策略迭代算法，收敛率就变成了策略迭代算法的收敛率。它很好地平衡了计算复杂度与收敛速度。

2.4 Q-Learning 算法

前文所介绍的算法（值迭代算法、策略迭代算法和修正策略迭代算法）都是基于模型的算法，它们要求输入一个完整的马尔科夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，其中状态转移矩阵 $\mathbf{P}$ 最为关键，它既反映了马尔科夫决策过程的本质特征，又是上述基于模型算法更新公式的关键量。

但是，在现实生活中，我们遇到的决策问题很少能够提供一个状态转移矩阵。在控制领域，通常有大量的研究关注于如何人为地进行状态转移矩阵的建模，这个过程需要相关人员对控制系统有深刻认识，也就是需要大量的物理与数学相关的知识。而在强化学习领域中，研究人员希望能够通过一个智能的算法，来自动化地完成从对环境的认识到学习出一个最有效的策略这整个过程。

也就是说，我们希望算法只需要接收一个 “黑箱”——满足马尔科夫决策过程模型的环境。我们可以观测这个环境当前状态 s，能够向环境执行一个动作 a，然后环境能够返回一个奖励值 r 和下一个状态 s'。因为如此，我们只能够对马尔科夫决策过程进行采样，获取若干条轨迹 $\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_t, a_t, r_t, \ldots)$。强化学习希望能够从这些轨迹样本中学习出针对这个环境的最优策略 π^* 来最大化累计奖励值。

那么，本节就介绍一个强化学习中非常重要的基于样本的随机优化算法——**Q-Learning**[48]。它可以看成一个近似值迭代算法，在值空间直接优

化求解最优值函数 Q^*。因为大部分强化学习相关的资料在介绍 Q-Learning 算法时会直接跳到它的更新公式上，从而造成了理解上的困难以及隐藏了原始 Q-Learning 算法的问题。所以本文选择从更基本的优化的角度来介绍 Q-Learning。

算法 4: Q-Learning 算法

Input: 总步数 N，一个满足马尔科夫决策过程的环境。

Output: 最优策略值函数的估计值 Q_θ

随机初始化值函数 Q_{θ_1} 和 Q_{η_1}，使 $\eta_1 = \theta_1$;

for $n = [1..N]$ **do**

  使用策略 $\pi_{Q,\epsilon}$ 从环境中采集 k 个样本 $\{(s, a, r, s')_i; i = [1..k]\}$，其中 $\pi_{Q,\epsilon} = (1-\epsilon)\pi_Q + \epsilon u$, 并且 u 是关于动作的均匀分布;

  将这 k 个样本加入到记忆池 $\mathcal{D}$;

  从 $\mathcal{D}$ 中采集 m 个样本 $\{(s, a, r, s')_i : i = [1..m]\}$;

  $\theta_{n+1} = \theta_n + \frac{\alpha}{m}\sum_{i=0}^{m}[r_i + \gamma \max_{a'_i} Q_{\eta_n}(s'_i, a'_i) - Q_{\theta_n}(s_i, a_i)]\nabla_{\theta_n} Q_{\theta_n}(s_i, a_i)$;

  $\eta_{n+1} = \eta_n + \beta(\theta_n - \eta_n)$;

end

return $Q_{\theta_{N+1}}$

首先要说明一下值函数 Q 的表达方式。在前文中，我们默认直接使用符号 Q 来表示值函数。这种表达方式在状态集合 $\mathcal{S}$ 和动作集合 $\mathcal{A}$ 是有限集的时候比较方便，这时 Q 既可以看成一个定义在 $\mathcal{S} \times \mathcal{A}$ 上的值函数也可以看成一个 $|\mathcal{S} \times \mathcal{A}|$ 大小的向量。如果 Q 函数是一个 $|\mathcal{S} \times \mathcal{A}|$ 维度的向量，那么这种表达方式下 Q 函数可以表示任何一个定义在 $\mathcal{S} \times \mathcal{A}$ 上的函数，它的表达形式是最丰富的。我们也称这种形式为**表格**（tabular）。

这里，本文使用一种更一般的表达形式——参数化的值函数。本文使用符号 θ 来表示值函数的参数，它可以是一个线性函数的系数，也可以是一个神经网络的参数。同时，本文使用 Q_θ 来表示值函数被 θ 参数化。首先这种表达形式可以表达状态集合 $\mathcal{S}$ 和动作集合 $\mathcal{A}$ 很大或者是或者无限集的情况。其次，当 θ 的维度等于 $|\mathcal{S} \times \mathcal{A}|$ 时，一个线性函数就能够等价表达一个表格。所以这种表达方式更加广义。

我们希望使用一个参数化的 Q_θ 来逼近马尔科夫决策过程的最优值函数 Q^*。如果状态集合和动作集合有限时，根据最优贝尔曼公式，我们可以构

造一个损失函数来衡量 Q_θ 与 Q^* 的接近程度：

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{s,a}[Q_\theta(s,a) - TQ_\theta(s,a)]^2. \tag{38}$$

带入最优贝尔曼操作公式可得

$$\begin{aligned}\mathcal{L}(\theta) = \frac{1}{2}\sum_{s,a}\Bigg\{Q_\theta(s,a) - \max_\pi \Bigg[R(s,a) + \gamma\sum_{s'}\mathbf{P}(s'|s,a) \\ \sum_{a'}\pi(a'|s')Q_\theta(s',a')\Bigg]\Bigg\}^2,\end{aligned} \tag{39}$$

将继续化简 max 操作可得

$$\mathcal{L}(\theta) = \frac{1}{2}\sum_{s,a}\left\{Q_\theta(s,a) - \left[R(s,a) + \gamma\sum_{s'}\mathbf{P}(s'|s,a)\max_{a'}Q_\theta(s',a')\right]\right\}^2. \tag{40}$$

上式隐式构造了一个确定性策略 $\pi_\theta(s) \in \arg\max_a Q_\theta(s,a)$。由贝尔曼等式可得：当 $\mathcal{L}(\theta) = 0$ 时，$Q_\theta = Q^*$。损失函数 $\mathcal{L}_\theta$ 虽然能够保证全局最优解对应马尔科夫决策过程的最优值函数。同时也能保证，$\mathcal{L}(\theta)$ 越小，Q_θ 越接近 Q^*，π_θ 越接近 π^*。

在本小节的设定下，我们只能从环境中采集轨迹样本

$$\{\tau_i\}_{i=1}^m = \{(s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}, a_1^{(i)}, r_1^{(i)}, \ldots, s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, \ldots)_{i=1}^m\}. \tag{41}$$

我们将轨迹样本 $\{\tau_i\}$ 拆分成一段段的状态转移

$$\{(s_i, a_i, r_i, s_i')\}_{i=1}^{m'}. \tag{42}$$

这时，我们就能用这些状态转移样本来近似地估计 (40) 的梯度：

$$\frac{\mathrm{d}\mathcal{L}(\theta)}{\mathrm{d}\theta} = \sum_{i=1}^{m'}\left\{Q_\theta(s_i,a_i) - \left[R(s_i,a_i) + \gamma\max_{a'}Q_\theta(s_i',a')\right]\right\}\nabla_\theta Q(s_i,a_i). \tag{43}$$

在只能获得轨迹样本的前提条件下，(43) 式是损失函数 (40) 梯度的有偏估计值。我们可以换一个角度，不再用近似的角度，而是对损失函数做进一步的改变。损失函数 $\mathcal{L}(\theta)$ 使用了二范数距离来衡量 Q_θ 与 TQ_θ 之间的差异。定义一个在 $\mathcal{S}\times\mathcal{A}$ 上的均匀分布 $u(s,a)$，那么可以将损失函数等价修改为

$$\mathcal{L}(\theta)=\frac{1}{2}\mathbb{E}_{(s,a)\sim u}\left\{Q_\theta(s,a)-\left[R(s,a)+\gamma\sum_{s'}\mathbf{P}(s'|s,a)\max_{a'}Q_\theta(s',a')\right]\right\}^2. \tag{44}$$

那么，对于一个只能采样的马尔科夫决策过程，我们无法获得满足 $u(s,a)$ 分布的样本，因此我们考虑使用一个泛化的任意分布 $p(s,a)$ 来替换均匀分布 $p(s,a)$。那么损失函数就变成了

$$\mathcal{L}(\theta)=\frac{1}{2}\mathbb{E}_{(s,a)\sim p}\left\{Q_\theta(s,a)-\left[R(s,a)+\gamma\sum_{s'}\mathbf{P}(s'|s,a)\max_{a'}Q_\theta(s',a')\right]\right\}^2. \tag{45}$$

我们通常直接将轨迹 $\tau=(s_0,a_0,r_0,s_1,a_1,r_1\ldots,s_t,a_t,r_t,\ldots)$ 拆分成状态转移样本 $((s_0,a_0,r_0,s_1),(s_1,a_1,r_1,s_2),\ldots,(s_t,a_t,r_t,s_{t+1}),\ldots)$ 用于损失函数的优化。也就是说，在强化学习中最常用的分布 $p(s,a)$ 就是一个马尔科夫决策过程中 (s,a) 会访问到的概率。通常我们将状态转移样本称为**经验池**，通常它对应的分布 $p(s,a)$ 和当前策略无关。用与当前策略无关的样本来优化求解的马尔科夫决策过程算法称为**异策略算法**，而 Q-Learning 算法就是一种**异策略算法**。

如果使用梯度下降法来对上面的损失函数进行求导可得：

$$\begin{aligned}\nabla_\theta\mathcal{L}(\theta)=\mathbb{E}_{(s,a)\sim p}\Bigg\{&\left[Q_\theta(s,a)-\left[R(s,a)+\gamma\sum_{s'}\mathbf{P}(s'|s,a)\max_{a'}Q_\theta(s',a')\right]\right]\\&\cdot\left[\nabla_\theta Q_\theta(s,a)-\gamma\sum_{s'}\mathbf{P}(s'|s,a)\nabla_\theta\max_{a'}Q_\theta(s',a')\right]\Bigg\}.\end{aligned} \tag{46}$$

这个导数面对着两个问题：一个问题是需要在固定 (s,a) 后对 s' 进行一次双采样，这是非常困难的；另一方面因为导数中的两项乘积都存在由 $\mathbf{P}(s'|s,a)$ 引起的随机性，导致如果使用样本来估计这个梯度时会产生很大的方差。因此，我们需要进一步改进损失函数，来解决这两个问题。

在过去的 Q-Learning 中，研究人员提出直接忽略 max 项的梯度，使用如下公式来近似 $\mathcal{L}(\theta)$ 的梯度：

$$\mathbb{E}_{(s,a)\sim p}\left\{\left[Q_\theta(s,a)-\left[R(s,a)+\gamma\sum_{s'}\mathbf{P}(s'|s,a)\max_{a'}Q_\theta(s',a')\right]\right]\cdot\nabla_\theta Q_\theta(s,a)\right\}. \tag{47}$$

这种近似的梯度仅在值函数使用表格形式或者线性函数形式时，被证明是收敛的。在 [31] 中提出了一种更加有效的方法：使用第二个值函数来解耦和化简损失函数。原本的一个损失函数变成了两个损失函数：

$$\begin{cases} \mathcal{L}(\eta) = \dfrac{1}{2}\|\eta - \theta\|_2^2 & (48) \\ \mathcal{L}(\theta) = \dfrac{1}{2}\mathbb{E}_{(s,a)\sim p}\Big\{ Q_\theta(s,a) - \Big[R(s,a) \\ \qquad\qquad + \gamma \sum\limits_{s'} \mathbf{P}(s'|s,a) \max\limits_{a'} Q_\eta(s',a') \Big] \Big\}^2 . & (49) \end{cases}$$

显然，新的损失函数的最优解依旧满足最优贝尔曼公式，至此，我们获得了 Q-Learning 算法的核心流程 [31]。需要注意的是，我们可以直接求解出损失函数 $\mathcal{L}(\eta)$ 的最优解 $\eta = \theta$，同时算法也退化成使用式 (47) 的原始的 Q-Learning 算法。但是，在实际实践过程中，在使用深度神经网络等复杂的函数来表达值函数时，过快地求解 $\mathcal{L}(\eta)$ 会造成算法的不稳定性。由于 Q-Learning 算法并没有完善的收敛性方面的理论研究，研究人员还无法确定 $\mathcal{L}(\theta)$ 和 $\mathcal{L}(\eta)$ 的最佳优化方案。

2.5 本章小结

本章介绍了两类基于最优贝尔曼等式的算法：一类是基于模型的算法，包含计算简单但是收敛较慢的值迭代算法、计算复杂但是收敛较快的策略迭代算法和兼具前两种算法优势的改进策略迭代算法；另一类是无模型的算法，即强化学习中非常重要的 Q-Learning 算法，它在实际实验中效果非常好，但是目前没有完善的理论来有效地分析它的本质机理。基于最优贝尔曼等式的算法在单次迭代时必须要求解一个子问题 $\max_a Q(s,a)$，导致我们比较难直接使用它来求解连续动作空间的马尔科夫决策问题。接下来本文将介绍一些新的算法框架来更好地解决连续动作空间的马尔科夫决策问题。

3 基于策略梯度下降的算法

在前面的章节中，介绍了马尔科夫决策过程以及两个重要的定理：贝尔曼等式和最优贝尔曼等式。可以根据贝尔曼等式和最优贝尔曼等式直接构造出一系列求解马尔科夫决策过程的算法：值迭代算法、策略迭代算法、修正策略迭代算法和 Q-Learning 算法，也就是说它们都是基于值函数空间的不动点特性构造的算法。在本小节中，本文将介绍一个全新的思路，通过研究策略空间的特性来构造新的算法。本文将从优化的角度来分析马尔科夫决策问题：对于一个策略 π 本文将它的价值函数 $\rho(\pi)$ 看成一个自变量为 π 的非凸损失函数。算法先求解出 $\rho(\pi)$ 关于 π 的导数，然后使用优化算法（如：梯度下降算法）来优化求解 $\rho(\pi)$。

3.1 随机策略梯度

使用函数参数化策略 π 在强化学习中是非常重要的。在 [45, 44] 中提出了一种非常重要的对任意参数化的随机策略 π 直接进行优化的方法，它的关键是求出了参数化策略的值函数的梯度。

首先我们声明一些符号声明。针对策略 π，我们可以使用线性函数或者神经网络来参数化它，我们用符号 θ_π 表示函数的参数。那么，对于一个马尔科夫决策过程 $\mathcal{MDP} = \{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，我们求解的问题变成了：

$$\max_{\theta_\pi} \rho(\theta_\pi) = (1-\gamma) \sum_{s\in\mathcal{S}} \mathbf{p}_0(s) \sum_{a\in\mathcal{A}} \pi(a|s;\theta_\pi) Q(s,a;\theta_\pi), \tag{50}$$

其中 $\tau = (s_0, a_0, r_1, \ldots, s_t, a_t, r_t, \ldots)$ 表示马尔科夫决策过程的一条轨迹，并且

$$Q(s,a;\theta_\pi) = \mathbb{E}_{\tau\sim\mathcal{MDP}(\theta_\pi)}\left[\sum_{t=0}^{\infty} \gamma^t r_t \middle| s_0 = s, a_0 = a\right]. \tag{51}$$

这里需要强调 $\rho(\theta_\pi)$ 只跟参数 θ_π 有关，Q_{θ_π} 也是由策略函数 π 确定的值函数。这一点和上一章基于最优贝尔曼等式的值函数空间的优化算法有非常大的不同。

接着我们需要对策略的值函数求导。随机策略梯度定理提供了一个它经验估计式，使我们可以通过使用当前策略和环境交互获得的样本，来估计当前的梯度。

定理 3.1 (随机策略梯度，Stochastic Policy Gradient Theorem)**.** 首先，声明一个马尔科夫随机过程 $\mathcal{MDP} = \{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$ 和一个参数化的马尔

科夫策略 θ_π。那么，这个策略的值函数 $\rho(\theta_\pi)$ 的梯度是

$$\begin{aligned}\frac{\mathrm{d}\rho(\theta_\pi)}{\mathrm{d}\theta_\pi} =& \sum_{s\in\mathcal{S}} \mathbf{p}_\gamma(s;\theta_\pi) \sum_{a\in\mathcal{A}} \frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi} Q(s,a;\theta_\pi) \\ =& \mathbb{E}_{s\sim\mathbf{p}_\gamma(\theta_\pi), a\sim\pi(\cdot|s;\theta_\pi)} \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi} Q(s,a;\theta_\pi) \right]\end{aligned} \tag{52}$$

其中 $\mathbf{p}_\gamma(s;\theta_\pi) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t \mathbf{p}_t(s;\theta_\pi)$，以及 $\mathbf{p}_t(s;\theta_\pi)$ 表示轨迹 $\tau \sim \mathcal{MDP}(\theta_\pi)$ 中 $s_t = s$ 出现的可能。

证明.

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}\rho(\theta_\pi) \\ =&\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}(1-\gamma)\left[\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta_\pi)Q(s,a;\theta_\pi)\right] \\ =&(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}[\pi(a|s;\theta_\pi)Q(s,a;\theta_\pi)] \\ =&(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\ &+(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta_\pi)\frac{\mathrm{d}Q(s,a;\theta_\pi)}{\mathrm{d}\theta_\pi} \\ =&(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\ &+(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta_\pi) \\ &\cdot\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}\left[r(s,a)+\gamma\sum_{s'\in\mathcal{S}}\mathbf{P}(s'|s,a)\sum_{a'\in\mathcal{A}}\pi(a'|s';\theta_\pi)Q(s',a';\theta_\pi)\right] \\ =&(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\ &+(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\pi(a|s;\theta_\pi) \\ &\cdot\gamma\sum_{s'\in\mathcal{S}}\mathbf{P}(s'|s,a)\sum_{a'\in\mathcal{A}}\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}[\pi(a'|s';\theta_\pi)Q(s',a';\theta_\pi)] \\ =&(1-\gamma)\sum_{s\in\mathcal{S}}\mathbf{p}_0(s)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\ &+(1-\gamma)\gamma\sum_{s'\in\mathcal{S}}\mathbf{p}_1(s')\sum_{a'\in\mathcal{A}}\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}[\pi(a'|s';\theta_\pi)Q(s',a';\theta_\pi)]\end{aligned}$$

$$
\begin{aligned}
&\vdots \\
=&(1-\gamma)\sum_{s\in\mathcal{S}}\sum_{t=0}^{\infty}\gamma^t\mathbf{p}_t(s;\theta_\pi)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\
=&\sum_{s\in\mathcal{S}}\mathbf{p}_\gamma(s;\theta_\pi)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\
=&\sum_{s\in\mathcal{S}}\mathbf{p}_\gamma(s;\theta_\pi)\sum_{a\in\mathcal{A}}\pi(a|s;\theta_\pi)\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi) \\
=&\mathbb{E}_{s\sim\mathbf{p}_\gamma,a\sim\pi(\cdot|s;\theta_\pi)}\left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi)\right].
\end{aligned}
$$

□

在马尔科夫决策过程模型未知时，我们只能通过采集的轨迹来探知环境。随机策略梯度公式暗示了一个经验估计式，来使我们能够在模型未知的设定下，使用随机梯度下降等算法来求解最优策略。当我们能够使用策略 π_{θ_π} 从环境中采集一系列样本 $\{(s_i,a_i)\}_{i=1}^m$，那么策略梯度的无偏估计为：

$$
\frac{\mathrm{d}\rho(\theta_\pi)}{\mathrm{d}\theta_\pi}\approx\frac{1}{m}\sum_{i=1}^{m}\frac{\mathrm{d}\log\pi(a_i|s_i;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s_i,a_i;\theta_\pi). \tag{53}
$$

我们暂时遇到了两个问题：如何获得值函数 $Q(s,a;\theta_\pi)$, 以及怎么从分布 $\mathbf{p}_\gamma$ 中采集样本 s。后面我们将解决这两个问题，不过本文会先介绍一个修正的策略梯度公式，有助于后面的讨论。

3.2 优势策略梯度公式

使用原本的策略梯度的无偏估计时，我们往往会遇到梯度估计的方差过大的问题。要解决策略梯度的方差过大的问题，我们可以增大每一次采样的数量，但是也增加了算法的样本复杂度。本节将介绍一种策略梯度的变形——优势随机策略梯度（Advantage Stochastic Policy Gradient）[19, 30, 20]，实现在相同的样本数量的前提下，大大降低梯度估计值的方差。

因为对于任意的状态 s，满足 $\sum_{a\in\mathcal{A}}\pi(a|s;\theta_\pi)=1$，因此

$$
\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}=0. \tag{54}
$$

那么，对于任意一个定义在 $\mathcal{S}$ 上的函数 $b(s)$，满足

$$
\sum_{s\in\mathcal{S}}\mathbf{p}_\gamma(s;\theta_\pi)\sum_{a\in\mathcal{A}}\frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}b(s)=0. \tag{55}
$$

将上面的结论带入到策略梯度公式中可得：

$$\begin{aligned}\frac{\mathrm{d}\rho(\theta_\pi)}{\mathrm{d}\theta_\pi} =& \sum_s \mathbf{p}_\gamma(s;\theta_\pi)\sum_a \frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)] \\ =& \mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left\{\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)]\right\}.\end{aligned} \tag{56}$$

在策略梯度公式中，我们称 $b(s)$ 为**基准**函数（Baseline Function）。

接下来我们来研究一下这个修改过后的策略梯度公式在什么情况下能够降低方差。首先这个策略梯度的方差为：

$$\begin{aligned}\sigma(b) =& Var_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left\{\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)]\right\} \\ =& \mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left\{\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)]\right\}^2 \\ & -\left\{\mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)]\right]\right\}^2 \\ =& \mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left\{\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)]\right\}^2 \\ & -\left\{\mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi)\right]\right\}^2.\end{aligned}$$

接下来我们来研究不同的 $b(s)$ 对策略梯度的方差的影响。首先定义：

$$\sigma_{\theta_\pi}(b) = \mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_\pi),a\sim\pi(\cdot|s;\theta_\pi)}\left\{\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}[Q(s,a;\theta_\pi)-b(s)]\right\}^2. \tag{57}$$

接着，我们求解问题 $\min_b \sigma_{\theta_\pi}(b)$，来获取最优的基准函数 $b^*(s;\theta_\pi)$。因为这是一个简单的二次优化问题，很容易可得：

$$b^*(s;\theta_\pi) = \frac{\sum_a \pi(a|s;\theta_\pi)\left(\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right)^2 Q(s,a;\theta_\pi)}{\sum_a \pi(a|s;\theta_\pi)\left(\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right)^2}. \tag{58}$$

到此，我们可以看到 $b^*(s;\theta_\pi)$ 非常复杂，接下来我们使用一个相对简单好求的等价的函数来替换 $b^*(s;\theta_\pi)$。

这里需要介绍马尔科夫决策过程中另一个相对重要的值函数——**状态值函数**（Value Function）：

$$V(s;\theta_\pi) = \sum_{a\in\mathcal{A}} \pi(a|s;\theta_\pi)Q(s,a;\theta_\pi). \tag{59}$$

在形式上，状态值函数 $V(s;\theta_\pi)$ 比最优基准函数 $b^*(s;\theta_\pi)$ 要简单得多。接着，我们研究一下使用 $V(s;\theta_\pi)$ 作为基准函数时策略梯度的方差和最优方差的关系:

$$
\begin{aligned}
&\sigma_{\theta_\pi}(V_{\theta_\pi}) - \sigma_{\theta_\pi}(b^*_{\theta_\pi}) \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi) \sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 \\
&(V(s;\theta_\pi) - b^*(s;\theta_\pi))(V(s;\theta_\pi) + b^*(s;\theta_\pi) - 2Q(s,a;\theta_\pi)) \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi)(V(s;\theta_\pi) - b^*(s;\theta_\pi)) \\
&\Bigg(\sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 V(s;\theta_\pi) \\
&+ \sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 b^*(s;\theta_\pi) \\
&- 2\sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 Q(s,a;\theta_\pi)\Bigg) \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi)(V(s;\theta_\pi) - b^*(s;\theta_\pi)) \\
&\Bigg(\sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 V(s;\theta_\pi) \\
&- \sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 b^*(s;\theta_\pi)\Bigg) \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi) \sum_a \pi(a|s;\theta_\pi) \left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi}\right]^2 (V(s;\theta_\pi) - b^*(s;\theta_\pi))^2 \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi) \mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[\nabla_{\theta_\pi}\ln\pi(a|s;\theta_\pi)]^2 (V(s;\theta_\pi) - b^*(s;\theta_\pi))^2 \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi) \frac{1}{\mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[\nabla_{\theta_\pi}\ln\pi(a|s;\theta_\pi)]^2} \\
&\cdot \{\mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[\nabla_{\theta_\pi}\ln\pi(a|s;\theta_\pi)]^2 V(s;\theta_\pi) \\
&\quad - \mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[[\nabla_{\theta_\pi}\ln\pi(a|s;\theta_\pi)]^2 b^*(s;\theta_\pi)\}^2 \\
=&\sum_s \mathbf{p}_\gamma(s;\theta_\pi) \frac{1}{\mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[\nabla_{\theta_\pi}\ln\pi(a|s;\theta_\pi)]^2} \\
&\cdot \{\mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[\nabla_{\theta_\pi}\ln\pi(a|s;\theta_\pi)]^2 \cdot \mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[Q(s,a;\theta_\pi)]
\end{aligned}
$$

$$- \mathbb{E}_{a\sim\pi(\cdot|s;\theta_\pi)}[[\nabla_{\theta_\pi} \ln \pi(a|s;\theta_\pi)]^2 Q(s,a;\theta_\pi)]\}^2.$$

那么，如果随机变量 $[\nabla_{\theta_\pi} \ln \pi(a|s;\theta_\pi)]^2$ 和随机变量 $Q(s,a;\theta_\pi)$ 独立时，可知 $\sigma_{\theta_\pi}(V_{\theta_\pi}) - \sigma_{\theta_\pi}(b^*_{\theta_\pi}) \approx 0$。

我们定义**优势函数**（Advantage Function）为:

$$A(s,a;\theta_\pi) = Q(s,a;\theta_\pi) - V(s;\theta_\pi). \tag{60}$$

那么，带基准函数的策略梯度公式就变成了**优势策略梯度公式**:

$$\begin{aligned}\frac{\mathrm{d}\rho(\theta_\pi)}{\mathrm{d}\theta_\pi} =& \sum_{s\in\mathcal{S}} \mathbf{p}_\gamma(s) \sum_{a\in\mathcal{A}} \frac{\mathrm{d}\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi} A(s,a;\theta_\pi) \\ =& \mathbb{E}_{s\sim\mathbf{p}_\gamma, a\sim\pi(\cdot|s;\theta_\pi)}\left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi} A(s,a;\theta_\pi)\right].\end{aligned} \tag{61}$$

相比于原始的策略梯度公式，优势策略梯度公式的方差大大降低，在使用样本估计策略梯度的优化算法中有显著的优势。

3.3 优势函数的估计

本节我们将解决 3.1 小节中提出的第一个问题：如何求解当前策略对应的状态动作值函数 $Q(s,a;\theta_\pi)$？等进一步，如何求解当前策略对应的优势函数 $A(s,a;\theta_\pi)$?

综合前文的随机策略梯度，我们可得：

$$\frac{\mathrm{d}\rho(\theta_\pi)}{\mathrm{d}\theta_\pi} = \mathbb{E}_{s\sim\mathbf{p}_\gamma, a\sim\pi(\cdot|s;\theta_\pi)}\left[\frac{\mathrm{d}\ln\pi(a|s;\theta_\pi)}{\mathrm{d}\theta_\pi} \Phi(s,a;\theta_\pi)\right]. \tag{62}$$

其中 Φ 代指当前策略函数对应的状态动作值函数 $Q(s,a;\theta_\pi)$ 或者优势函数 $A(s,a;\theta_\pi)$。

假设我们使用当前策略与环境交互获得了 m 条轨迹样本

$$\{(s_0^{(i)}, a_0^{(i)}, r_0^{(i)}, s_1^{(i)}, a_1^{(i)}, r_1^{(i)}, \ldots, s_t^{(i)}, a_t^{(i)}, r_t^{(i)}, \ldots)\}_{i=1}^m。 \tag{63}$$

通常我们使用如下方式来估计 $\Phi(s,a;\theta_\pi)$：

1. 蒙特卡洛估计（Monto Carlo，MC）：$\Phi(s_t^{(i)}, a_t^{(i)}) = \sum_{t'=t}^{\infty} \gamma^{t'-t} r_{t'}^{(i)}$；
2. 状态动作值函数的差分估计（Temporal-difference，TD）：初始化一个估计函数 $Q(s,a;\theta_Q)$，然后求解贝尔曼等式构造的优化目标

$$\min_{\theta_Q} \sum_{i=1}^{m} \sum_{t=0}^{\infty} [r_t^{(i)} + \gamma Q(s_{t+1}^{(i)}, a_{t+1}^{(i)}; \theta_Q) - Q^{(i)}(s_t^{(i)}, a_t^{(i)}; \theta_Q)]^2, \tag{64}$$

最后的估计值为 $\Phi(s_t^{(i)}, a_t^{(i)}) = Q(s_t^{(i)}, a_t^{(i)}; \theta_Q)$;

3. 优势函数的差分估计：初始化一个估计函数 $V(s;\theta_V)$，并使用如下优化目标

$$\min_{\theta_V} \sum_{i=1}^{m} \sum_{t=0}^{\infty} [r_t^{(i)} + \gamma V(s_{t+1}^{(i)}; \theta_V) - V(s_t^{(i)}; \theta_V)]^2, \tag{65}$$

最后使用如下式来估计优势函数：

$$\Phi(s_t^{(i)}, a_t^{(i)}) = r_t^{(i)} + \gamma V(s_{t+1}^{(i)}; \theta_V) - V(s_t^{(i)}; \theta_V); \tag{66}$$

4. 泛化优势函数估计（Generalized Advantage Estimation，GAE）[38]：初始化一个估计函数 $V(s;\theta_V)$，并使用 (65) 式来估计它，最后使用如下式来估计泛化优势函数

$$\Phi(s_t^{(i)}, a_t^{(i)}) = \sum_{t'=t}^{\infty} \lambda^{t'-t} [r_{t'}^{(i)} + \gamma V(s_{t'+1}^{(i)}; \theta_V) - V(s_{t'}^{(i)}; \theta_V)], \tag{67}$$

其中 $\lambda \in (0, \gamma)$ 是一个选定的超参数。

前面三个估计器比较直观，不需要做进一步的阐述。最后一个估计器用到了奖励值重塑技术，本文将做进一步的阐述。

首先介绍一下**奖励值重塑技术**（Reward Shaping, RS）[33]。对于一个马尔科夫决策问题，我们非常关心轨迹样本

$$(s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_t, a_t, r_t, \ldots) \tag{68}$$

的累计奖励函数

$$\sum_{t=0}^{\infty} \gamma^t r_t. \tag{69}$$

奖励值重塑技术提供了一种修改奖励值 r 但是策略值保持不变的泛式：给定任意一个定义在状态集合 $\mathcal{S}$ 上的函数 $F : \mathcal{S} \to \mathbb{R}$，奖励值重塑技术将奖励函数变为：

$$rs(s, a, s') = r(s, a) + \gamma F(s') - F(s). \tag{70}$$

使用重塑奖励函数后，累计奖励函数变为

$$\begin{aligned}\rho_{rs}(\theta_\pi) =& \sum_{t=0}^{\infty} \gamma^t rs(s_t, a_t, s_{t+1}) \\ =& \lim_{K\to\infty} \sum_{t=0}^{K} \gamma^t rs(s_t, a_t, s_{t+1}) \\ =& \lim_{K\to\infty} \sum_{t=0}^{K} \gamma^t [r_t + \gamma F(s_{t+1}) - F(s_t)] \\ =& \lim_{K\to\infty} \sum_{t=0}^{K} \gamma^t r_t + \gamma^{K+1} F(s_{K+1}) - F(s_0) \\ =& \sum_{t=0}^{\infty} \gamma^t r_t - F(s_0) = \rho(\theta_\pi) - F(s_0).\end{aligned} \tag{71}$$

从 (71) 式可知，对于同一条轨迹，重塑奖励函数对应的累计奖励值只是原奖励函数对应的累计奖励值减去一个与策略无关的项 $F(s_0)$。那么重塑奖励函数对应的马尔科夫决策问题的最优策略和原奖励函数对应的马尔科夫决策问题的最优策略是相同的。

泛化优势函数选择 $F(s) = V(s;\theta_\pi)$，因此重塑奖励函数就变成了

$$rs(s, a, s';\theta_\pi) = r(s, a) + \gamma V(s';\theta_\pi) - V(s;\theta_\pi). \tag{72}$$

而这个重塑奖励函数关于 s' 的期望就是优势函数：

$$A(s, a;\theta_\pi) = \mathbb{E}_{s'\sim\mathbf{P}(\cdot|s,a)}[rs(s, a, s;\theta_\pi)]. \tag{73}$$

那么泛化优势函数估计使用的模型其实是将原马尔科夫决策问题的优势函数作为奖励函数的**重塑马尔科夫决策问题**

$$\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, A(s, a;\theta_\pi), \lambda\}. \tag{74}$$

如果 $\lambda = \gamma$ 时，由 (71) 式可知，两个问题的最优策略相同。而当 $\lambda = 0$ 时，重塑马尔科夫决策问题的值函数 $Q_{rs}(s, a;\theta_\pi)$ 就等于原马尔科夫决策问题的优势函数 $A(s, a;\theta_\pi)$。那么两个马尔科夫决策问题对应的策略梯度唯一的不同之处就是分布 $\mathbf{p}_\gamma(s;\theta_\pi)$ 和 $\mathbf{p}_\lambda(s;\theta_\pi)$，如果在求解重塑马尔科夫决策问题的策略梯度时是从 $\mathbf{p}_\gamma$ 中采样，那么其实得到的就是原马尔科夫决策问题的策略梯度，即等价于求解如下马尔可夫决策问题

$$\{\mathcal{S}, \mathcal{A}, \mathbf{p}_\gamma, \mathbf{P}, A(s, a;\theta_\pi), 0\}. \tag{75}$$

在实践中研究者发现，$\lambda = \gamma$ 会导致估计出来的策略梯度的方差过大，而 $\lambda = 0$ 过于“短视”，所以人们通常取 $0 < \lambda < \gamma$。

注 3.1. 泛化优势函数是一个启发式的估计算法，它的好处是降低了梯度策略估计值的方差（*variance*），但它同时也引进了偏差（*bias*），而超参数 λ 则是起到了平衡方差和偏差的作用。它在实践中取到了非常好的效果，通常被默认使用在策略梯度优化的算法中。

3.4 解决采样问题

本小节将解决 3.1 小节中提出的第二个问题：如何从分布 $\mathbf{p}_\gamma(s;\pi)$ 中采集状态样本 s?解决这个问题的同时，我们也能解决另一个问题：如何采集一条无限长度的轨迹样本? 在本节中，本文提出了一种新的采样方法——**随机截断法**（Stochastic Truncate）。

首先介绍一下传统的解决方法——**固定截断法**。通常，我们会设定一个最大长度 L（通常会取 $L = 1000$）。当轨迹的长度达到最大长度时，我们就人为截断它为：

$$\tau = (s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_L, a_L, r_L). \tag{76}$$

接着我们人为地将轨迹截断为状态转移片断样本集 $\{(s, a, r, s')\}$ 用于估计策略梯度。这种方式获得的状态样本是近似是从分布

$$\mathbf{p}_{avg}(s;\theta_\pi) = \frac{1}{L}\sum_{t=0}^{L-1}\mathbf{p}_t(s;\theta_\pi) \tag{77}$$

中采样获得。因为马尔科夫链会将分布逐渐收敛到稳态分布 $\mathbf{d}$，所以有：

$$\lim_{L\to\infty}\frac{1}{L}\sum_{t=0}^{L-1}\mathbf{p}_t = \mathbf{d}. \tag{78}$$

也就是说，固定截断法可以近似地看做从马尔科夫链的稳态分布中采集状态样本。

接下来本文将提出随机截断法，也进一步对固定截断法做更进一步的理解。首先对于一个马尔科夫决策问题 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，本文构造一个和它对应的马尔科夫决策问题：

$$\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, (1-\gamma)\mathbf{1}\mathbf{p}_0^T + \gamma\mathbf{P}, R, \gamma\}. \tag{79}$$

其中，$\mathbf{1}$ 是每个维度都是 1 的向量，$(1-\gamma)\mathbf{1}\mathbf{p}_0^T + \gamma\mathbf{P}$ 指的是在 (s,a) 状态转移到下一状态 s' 的概率为 $(1-\gamma)\mathbf{p}_0(s') + \gamma\mathbf{P}(s'|s,a)$。

接着，我们介绍原马尔科夫决策过程和新的马尔科夫决策过程的关系。在原马尔科夫决策过程未知，只能对它进行采样的设定下，我们很容易转而对新的马尔科夫决策过程进行采样。从新的马尔科夫决策过程的状态转移概率可知：对原马尔科夫决策过程进行一步状态转移的采样后，我们以 $(1-\gamma)$ 的可能性截断当前轨迹，重启一条新的轨迹采样；并且以 γ 的可能性对当前轨迹继续采样。根据这种随机截断采样的方式，我们就能从原马尔科夫决策过程转而向新的马尔科夫决策过程采样，所以我们称新的马尔科夫决策过程为**随机截断马尔科夫决策过程**（Stochastic Truncate Markov Decision Processes）。

其次，我们介绍随机截断马尔科夫决策过程和目标 $\mathbf{p}_\gamma(s;\pi)$ 分布的关系。本小节我们只关心状态，所以我们来研究只和状态有关的马尔科夫链以及它对应的随机截断马尔科夫链：$\{\mathcal{S}, \mathbf{p}_0, \mathbf{P}_\pi\}$ 和 $\{\mathcal{S}, \mathbf{p}_0, (1-\gamma)\mathbf{1}\mathbf{p}_0^T + \gamma\mathbf{P}_\pi\}$。其中

$$\mathbf{P}_\pi(s'|s) = \sum_a \mathbf{P}(s'|s,a)\pi(a|s), \tag{80}$$

以及

$$[(1-\gamma)\mathbf{1}\mathbf{p}_0^T + \gamma\mathbf{P}_\pi](s'|s) = (1-\gamma)\mathbf{p}_0(s') + \gamma\sum_a \mathbf{P}(s'|s,a)\pi(a|s). \tag{81}$$

下面这个定理揭示了两个马尔科夫链的关系。

定理 3.2. *对于任意的马尔科夫链 $\{\mathcal{X}, \mathbf{p}_0, \mathbf{P}\}$，它对应的随机截断马尔科夫链 $\{\mathcal{X}, \mathbf{p}_0, (1-\gamma)\mathbf{1}\mathbf{p}_0^T + \gamma\mathbf{P}\}$ 的稳态分布为：*

$$\mathbf{d}_{st}(s) = \left[(1-\gamma)\sum_{t=0}^{\infty}\gamma^t\mathbf{p}_0^T\mathbf{P}^t\right](s) = \mathbf{p}_\gamma(s). \tag{82}$$

证明. 由稳态分布的定义可得：

$$\begin{aligned}
\mathbf{d}_{st}^T &= \mathbf{d}_{st}^T[(1-\gamma)\mathbf{1}\mathbf{p}_0^T + \gamma\mathbf{P}] \\
&= (1-\gamma)\mathbf{p}_0^T + \gamma\mathbf{d}_{st}^T\mathbf{P} \\
\mathbf{d}_{st}^T(\mathbf{I} - \gamma\mathbf{P}) &= (1-\gamma)\mathbf{p}_0^T \\
\mathbf{d}_{st}^T &= (1-\gamma)\mathbf{p}_0^T(\mathbf{I} - \gamma\mathbf{P})^{-1}
\end{aligned}$$

$$
\begin{aligned}
&= (1-\gamma)\mathbf{p}_0^T \sum_{t=0}^{\infty} (\gamma \mathbf{P})^t \\
&= (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \mathbf{p}_0^T \mathbf{P}^t.
\end{aligned}
$$

□

根据定理 3.2 可知，我们只需要从随机截断马尔科夫决策过程采集一条轨迹，这条轨迹的状态对应的分布会逐渐收敛到随机截断马尔科夫决策过程的稳态分布，也就是我们成功从目标分布 $\mathbf{p}_\gamma(s;\pi)$ 中采集到了状态样本。

我们现在再来回顾固定截断法。在每一步的状态转移时，随机截断法会以 $(1-\gamma)$ 的可能性截断轨迹，那么轨迹的长度 L 服从几何分布 $p(L) = \gamma^{L-1}(1-\gamma)$，轨迹的长度 L 的期望为 $1/(1-\gamma)$。反过来理解，当我们取值 $\gamma = 1 - 1/L$ 时，使用随机截断法采集的轨迹的期望长度为 L。

注 3.2. 在实际实验中，使用随机截断法进行采样过程中，当环境处于 (s_L, a_L) 需要被截断时，算法可以在延续采样 K 个长度的样本，用于降低 $Q(s_L, a_L; \theta_\pi)$ 或者 $A(s_L, a_L; \theta_\pi)$ 的估计误差。即样本轨迹为：

$$
(s_0, a_0, r_0, s_1, a_1, r_1, \ldots, s_L, a_L, r_L, \ldots, s_{L+K}, a_{L+K}, r_{L+K}). \tag{83}
$$

最后 K 个样本用于估计 $Q(s_L, a_L; \theta_\pi)$ 或者 $A(s_L, a_L; \theta_\pi)$。

3.5 本章小结

本章介绍了强化学习中一个非常重要的定理——随机策略梯度定理。它将原本的序列决策优化问题转化为了一个优化问题，并且提供了使用基于梯度的优化算法来求解强化学习最优策略的算法的基础。本章也介绍了使用优势函数版本的随机策略梯度定理，它相比于原本的随机策略梯度而言，在理论上具有更小的方差。接着本章解决了两个估计策略梯度的难点：值函数的估计问题和状态采样问题。关于值函数的估计问题，本章介绍了在实践中非常有效地降低方差的泛化优势函数。关于状态采样问题，本章首次提出了随机截断法，并且从理论上证明了随机截断法比传统的固定截断法具有更小的偏差。

4 置信域策略梯度优化算法

本章将继续上一章的内容，继续深入讨论策略梯度算法。当使用深度神经网络来参数化策略时，马尔科夫决策问题就变成一个典型的非凸优化问题。当使用基于梯度的优化算法来求解这个问题时，我们很难确定最合适的迭代步长。本章将进一步分析马尔科夫决策问题的更精细的结构，从而引出一个保证策略单调上升的定理——**置信域定理** [26]（Trust Region Theorem），它通过置信域的形式来指导算法使用合适的更新步长。接着本章节将介绍两个基于置信域定理的两个算法：**置信域策略优化算法** [37]（Trust Region Policy Optimization，TRPO）和**近邻策略优化算法** [39]（Proximal Policy Optimization，PPO）。这两个算法在理论和实践中都取得了显著的成功。

4.1 置信域定理

本节将进一步研究马尔科夫决策问题，并且介绍一个重要的定理——**置信域定理** [26]（Trust Region Theorem）。在前一章节中介绍的策略梯度是从微分的角度来研究马尔科夫决策问题，而微分的角度通常因为极限的特性而隐藏了一些信息，本小节将从差分的角度来研究这个问题，进一步研究之前被隐藏了的信息。

4.1.1 策略的差分

首先，我们介绍两个策略的差分关系。

引理 4.1. *设一个马尔可夫决策过程 $\mathcal{MDP} = \{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，它的策略评价函数为 $\rho(\pi)$。那么，任意的两个策略 π_1 和 π_2 有如下关系：*

$$\begin{aligned}\rho(\pi_1) - \rho(\pi_2) =& (1-\gamma)\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t A(s_t, a_t; \pi_2)\right] \\ =& \sum_s \mathbf{p}_\gamma(s;\pi_1)\sum_a \pi_1(a|s)A(s,a;\pi_2).\end{aligned} \tag{84}$$

证明. 首先是第一个等式的证明

$$\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t A(s_t, a_t; \pi_2)\right]$$

$$
\begin{aligned}
=&\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t(Q(s_t,a_t;\pi_2)-V(s_t;\pi_2))\right]\\
=&\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t(r(s_t,a_t)+\gamma V(s_{t+1};\pi_2)-V(s_t;\pi_2))\right]\\
=&\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)-V(s_0;\pi_2)\right]\\
=&\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t r(s_t,a_t)\right]-\mathbb{E}_{s_0\sim\mathbf{p}_0}\left[V(s_0;\pi_2)\right]\\
=&\frac{1}{1-\gamma}[\rho(\pi_1)-\rho(\pi_2)].
\end{aligned}
$$

关于第二个等式，本质上只是求和顺序的更改。

$$
\begin{aligned}
&(1-\gamma)\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi_1)}\left[\sum_{t=0}^{\infty}\gamma^t A(s_t,a_t;\pi_2)\right]\\
=&(1-\gamma)\sum_{t=0}^{\infty}\sum_{s}\mathbf{p}_t(s;\pi_1)\sum_{a}\pi_1(a|s)\gamma^t A(s,a;\pi_2)\\
=&\sum_{s}\sum_{t=0}^{\infty}(1-\gamma)\gamma^t\mathbf{p}_t(s;\pi_1)\sum_{a}\pi_1(a|s)A(s,a;\pi_2)\\
=&\sum_{s}\mathbf{p}_\gamma(s;\pi_1)\sum_{a}\pi_1(a|s)A(s,a;\pi_2).
\end{aligned}
$$

□

当使用策略梯度优化算法时，会产生一系列策略 $\{\pi_n\}$。在理想情况下，我们希望在每一步求解优化问题：

$$
\begin{aligned}
\pi_{n+1}=&\arg\max_\pi\rho(\pi)\\
=&\arg\max_\pi\rho(\pi_n)+\sum_{s}p_\gamma(s;\pi)\sum_{a}\pi(a|s)A(s,a;\pi_n).
\end{aligned}
\tag{85}
$$

由于 (85) 式表示的原问题比较难解，普通的策略梯度优化算法将优化目标简化为

$$
\begin{aligned}
\pi_{n+1}=&\arg\max_\pi L(\pi,\pi_n)\\
=&\arg\max_\pi\rho(\pi_n)+\sum_{s}p_\gamma(s;\pi_n)\sum_{a}\pi(a|s)A(s,a;\pi_n),
\end{aligned}
\tag{86}
$$

其中我们称 $L(\pi, \pi_n)$ 为**相关策略评价函数**，它是一个关于策略 π 的线性函数。当使用参数 θ_π 来参数化策略时，对应的更新公式为

$$\theta_{\pi_{n+1}} = \theta_{\pi_n} + \alpha \sum_s p_\gamma(s; \theta_{\pi_n}) \sum_a \nabla_{\theta_{\pi_n}} \pi(a|s; \theta_{\pi_n}) \cdot A(s, a; \pi_n). \tag{87}$$

(87) 式最大的问题是它并不能保证 $\rho(\pi_n) \leq \rho(\pi_{n+1})$。因为 (85) 式和 (86) 式仅在 π_n 和 π_{n+1} 非常接近时两者才比较近似，策略梯度优化算法通常通过限制每一步更新参数的步长来保证 π_n 和 π_{n+1} 的距离不要太远。事实上，这个步长非常难以确定：当步长过小时，算法的学习收敛地非常慢；当步长过大时，又无法保证策略往好的方向更新。置信域定理的目标就是要构造一个显式的置信域，保证在这个置信域内 $\rho(\pi)$ 和 $L(\pi, \pi_n)$ 的差异不要过大。

4.1.2 全变分差异

在介绍置信域定理前，先补充一个概率论的先验知识——全变分差异 [11]。

定义 4.1 (全变分差异, Total Variation Divergence)**.** *设 $\mathcal{F}$ 是样本空间 Ω 的幂集合的 σ-代数 (测度论中的一种可测集合)。两个定义在 $\mathcal{F}$ 上的概率测度 P 和 Q 的全变分差异为*

$$D_{TV}(P\|Q) = \sup_{A \in \mathcal{F}} |P(A) - Q(A)|. \tag{88}$$

全变分差异衡量的是两个分布在同一事件中的最大差异。

引理 4.2. *设 $\mathcal{F}$ 是样本空间 Ω 的幂集合的 σ-代数 (测度论中的一种可测集合)。如果 Ω 是可数集合，那么两个定义在 $\mathcal{F}$ 上的概率测度 P 和 Q 的全变分差异有一个等价形式*

$$D_{TV}(P\|Q) = \frac{1}{2} \sum_{\omega \in \Omega} |P(\omega) - Q(\omega)|. \tag{89}$$

证明. 设集合 $A_P = \{\omega \in \Omega : P(\omega) \geq Q(\omega)\}$ 以及 $A_Q = \{\omega \in \Omega : P(\omega) < Q(\omega)\}$。对于任意的 $A \in \mathcal{F}$ 我们有

$$P(A) - Q(A) \leq P(A \cup A_P) - Q(A \cup A_P) \leq P(A_P) - Q(A_P),$$

以及

$$Q(A) - P(A) \leq Q(A \cup A_Q) - P(A \cup A_Q) \leq Q(A_Q) - P(A_Q),$$

所以

$$|P(A)-Q(A)| \le \min[P(A_P)-Q(A_P), Q(A_Q)-P(A_Q)].$$

因为 $A_P, A_Q \in \mathcal{F}$ 所以

$$D_{TV}(P\|Q) = \min[P(A_P)-Q(A_P), Q(A_Q)-P(A_Q)].$$

因为 $P(A_P)+P(A_Q)=Q(A_P)+Q(A_Q)=1$ 所以

$$P(A_P)-Q(A_P) = Q(A_Q)-P(A_Q),$$

所以

$$\begin{aligned} D_{TV}(P\|Q) =& \frac{1}{2}[P(A_P)-Q(A_P)+Q(A_Q)-P(A_Q)] \\ =& \frac{1}{2}\sum_{\omega\in\Omega}|P(\omega)-Q(\omega)|. \end{aligned}$$

□

引理 4.3. 设 $\mathcal{F}$ 是样本空间 Ω 的幂集合的 σ-代数（测度论中的一种可测集合）。如果 Ω 是可数集合，两个定义在 $\mathcal{F}$ 上的概率测度 P 和 Q 的全变分差异有一个等价形式

$$D_{TV}(P\|Q) = \inf_{\mathcal{D}} \mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}}(\{\omega_1\in\Omega, \omega_2\in\Omega : \omega_1\neq\omega_2\}), \tag{90}$$

其中 $\mathcal{D}$ 是定义在 $\Omega\times\Omega$ 上的联合分布，它的边缘分布分别是 P 和 Q。

证明. 对于任意的 $A\in\mathcal{F}$ 和任意的联合分布 $\mathcal{D}$，有

$$\begin{aligned} & P(A)-Q(A) \\ =& P(\{\omega_1\in A\}) + Q(\{\omega_2\notin A\}) - 1 \\ \le& \mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}}(\{\omega_1\in A, \omega_2\notin A\}) \\ \le& \mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}}(\{\omega_1\neq\omega_2\}). \end{aligned}$$

所以 $D_{TV}(P\|Q) \le \mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}}(\{\omega_1\neq\omega_2\})$。$(P(\{\omega_1\in A\}) + Q(\{\omega_2\notin A\}) - \mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}}(\{\omega_1\in A, \omega_2\notin A\}) \le 1)$

设集合 $A_P = \{\omega \in \Omega : P(\omega) \geq Q(\omega)\}$ 以及 $A_Q = \{\omega \in \Omega : P(\omega) < Q(\omega)\}$。我们构造一个特殊的联合分布 $\mathcal{D}^*$ 满足

$$\mathbb{P}_{\mathcal{D}^*}(\omega_1, \omega_2) = \begin{cases} Q(\omega_1) \cdot \mathbf{1}\{\omega_1 = \omega_2\}, & \omega_1 \in A_P, \omega_2 \in A_P; \\ P(\omega_1) \cdot \mathbf{1}\{\omega_1 = \omega_2\}, & \omega_1 \in A_Q, \omega_2 \in A_Q; \\ \frac{[P(\omega_1)-Q(\omega_1)][Q(\omega_2)-P(\omega_2)]}{P(A_P)-Q(A_P)}, & \omega_1 \in A_P, \omega_2 \in A_Q; \\ 0, & otherwise. \end{cases}$$

我们先证明它是一个分布：

$$\begin{aligned} &\sum_{\omega_1}\sum_{\omega_2} \mathbb{P}_{\mathcal{D}^*}(\omega_1, \omega_2) \\ =&\sum_{\omega_1 \in A_P, \omega_1 \in A_P} Q(\omega_1) \cdot \mathbf{1}\{\omega_1 = \omega_2\} \\ &+ \sum_{\omega_1 \in A_Q, \omega_1 \in A_Q} P(\omega_1) \cdot \mathbf{1}\{\omega_1 = \omega_2\}, \\ &+ \sum_{\omega_1 \in A_P} \sum_{\omega_2 \in A_Q} \frac{[P(\omega_1) - Q(\omega_1)][Q(\omega_2) - P(\omega_2)]}{P(A_p) - Q(A_p)} \\ =&Q(A_P) + P(A_Q) + \frac{[P(A_P) - Q(A_P)][Q(A_Q) - P(A_Q)]}{P(A_P) - Q(A_P)} = 1. \end{aligned}$$

那么

$$\begin{aligned} &\mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}^*}(\omega_1 \neq \omega_2) \\ =&1 - \mathbb{P}_{(\omega_1,\omega_2)\sim\mathcal{D}^*}(\omega_1 = \omega_2) \\ =&1 - Q(A_P) - P(A_Q) \\ =&Q(A_Q) - P(A_Q) = D_{TV}(P\|Q). \end{aligned}$$

综上，得证。 □

4.1.3 置信域定理及其证明

本小节我们将介绍置信域定理 [26]（Trust Region Theorem）。本小节首先定义了两个策略之间的全变分差异。接着为了给置信域定理的证明做准备，本小节将先介绍两个相关引理。最后本小节正式讨论置信域定理和它的证明。

首先定义两个策略之间的最大变分差异：

定义 4.2 (最大变分差异)**.** 设一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，任意两个定义在 $\mathcal{S} \times \mathcal{A}$ 上的策略 π_1 和 π_2 之间的最大变分差异为

$$D_{TV}^{\max}(\pi_1 \| \pi_2) = \max_s D_{TV}(\pi_1(\cdot|s) \| \pi_2(\cdot\|s)). \tag{91}$$

引理 4.4. 设一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，它的优势函数为 $A(s, a; \pi)$。那么，任意的两个策略 π_1 和 π_2 有如下关系：

$$|\mathbb{E}_{a \sim \pi_2(s)}[A(s, a; \pi_1)]| \leq 2 D_{TV}^{\max}(\pi_1 \| \pi_2) \max_{s,a} |A(s, a; \pi_1)|. \tag{92}$$

证明. 设任意一个定义在 $\mathcal{A} \times \mathcal{A}$ 上的分布 $\mathcal{D}$，满足 $\mathcal{D}(\cdot|s)$ 的两个边缘分布分别为 $\pi_1(\cdot|s)$ 和 $\pi_2(\cdot|s)$。

$$\begin{aligned}
&\mathbb{E}_{a \sim \pi_2(\cdot|s)}[A(s, a; \pi_1)] \\
=&\mathbb{E}_{a \sim \pi_2(\cdot|s)}[A(s, a; \pi_1)] - \mathbb{E}_{a \sim \pi_1(\cdot|s)}[A(s, a; \pi_1)] \\
=&\mathbb{E}_{(a1,a2) \sim \mathcal{D}(\cdot|s)}[A(s, a_2; \pi_1) - A(s, a_1; \pi_1)] \\
=&\mathbb{P}_{(a1,a2) \sim \mathcal{D}(\cdot|s)}[a1 \neq a2] \\
&\cdot \mathbb{E}_{(a1,a2) \sim \mathcal{D}(\cdot|s)}[A(s, a_2; \pi_1) - A(s, a_1; \pi_1) | a1 \neq a2].
\end{aligned}$$

因此，我们可得

$$|\mathbb{E}_{a \sim \pi_2(\cdot|s)}[A(s, a; \pi_1)]| \leq 2 D_{TV}^{\max}(\pi_1 \| \pi_2) \max_{s,a} |A(s, a; \pi_1)|.$$

□

引理 4.5. 设一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, \mathbf{p}_0, \mathbf{P}, R, \gamma\}$，分布 $\mathbf{p}_t(s; \pi)$ 表示策略 π 对应的马尔科夫链的轨迹中第 t 时刻状态的分布。那么，任意的两个策略 π_1 和 π_2 有如下关系：

$$D_{TV}(\mathbf{p}_t(\cdot; \pi_1) \| \mathbf{p}_t(\cdot; \pi_2)) \leq t D_{TV}^{\max}(\pi_1 \| \pi_2). \tag{93}$$

证明. 我们将使用数学归纳法来证明。

第一步，当 $t = 0$ 时，初始状态分布 $\mathbf{p}_0(\cdot; \pi_1)$ 和 $\mathbf{p}_0(\cdot; \pi_2))$ 都等于马尔科夫决策过程的状态分布 $\mathbf{p}_0$，所以 (93) 式两边都为 0，满足引理。

第二步，我们证明

$$D_{TV}(\mathbf{p}_{t+1}(\cdot; \pi_1) \| \mathbf{p}_{t+1}(\cdot; \pi_2)) \leq D_{TV}^{\max}(\pi_1 \| \pi_2) + D_{TV}(\mathbf{p}_t(\cdot; \pi_1) \| \mathbf{p}_t(\cdot; \pi_2)).$$

具体过程如下

$$
\begin{aligned}
& D_{TV}(\mathbf{p}_{t+1}(\cdot;\pi_1)\|\mathbf{p}_{t+1}(\cdot;\pi_2)) \\
=&\frac{1}{2}\sum_{s_{t+1}}|\mathbf{p}_{t+1}(s_{t+1};\pi_1)-\mathbf{p}_{t+1}(s_{t+1};\pi_2)| \\
=&\frac{1}{2}\sum_{s_{t+1}}\Bigg|\sum_{s_t,a_t}\mathbf{p}_t(s_t;\pi_1)\pi_1(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t) \\
&-\sum_{s_t,a_t}\mathbf{p}_t(s_t;\pi_2)\pi_2(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t)\Bigg| \\
\leq&\frac{1}{2}\sum_{s_{t+1}}\Bigg|\sum_{s_t,a_t}\mathbf{p}_t(s_t;\pi_1)\pi_1(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t) \\
&-\sum_{s_t,a_t}\mathbf{p}_t(s_t;\pi_1)\pi_2(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t)\Bigg| \\
&+\frac{1}{2}\sum_{s_{t+1}}\Bigg|\sum_{s_t,a_t}\mathbf{p}_t(s_t;\pi_1)\pi_2(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t) \\
&-\sum_{s_t,a_t}\mathbf{p}_t(s_t;\pi_2)\pi_2(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t)\Bigg| \\
\leq&\frac{1}{2}\sum_{s_{t+1},s_t,a_t}\mathbf{p}_t(s_t;\pi_1)|\pi_1(a_t|s_t)-\pi_2(a_t|s_t)|\mathbf{P}(s_{t+1}|s_t,a_t) \\
&+\frac{1}{2}\sum_{s_{t+1},s_t,a_t}|\mathbf{p}_t(s_t;\pi_1)-\mathbf{p}_t(s_t;\pi_2)|\pi_2(a_t|s_t)\mathbf{P}(s_{t+1}|s_t,a_t) \\
\leq&D_{TV}^{\max}(\pi_1\|\pi_2)+D_{TV}(\mathbf{p}_{t,\pi_1}\|\mathbf{p}_{t,\pi_2})
\end{aligned}
$$

综合第一步和第二步，引理得证。 □

至此，我们已经完成了置信域定理所需要的所有准备工作，终于进入置信域定理的介绍和证明。

定理 4.1 (置信域定理，Trust Region Theorem)**.** 设一个马尔可夫决策过程 $\mathcal{MDP}=\{\mathcal{S},\mathcal{A},\mathbf{p}_0,\mathbf{P},R,\gamma\}$，它的策略评价函数为 $\rho(\pi)$，它的关于策略 π' 的相关策略评价函数为 $L(\pi_1,\pi_2)$。那么 $\rho(\pi_1)$ 和 $L(\pi_1,\pi_2)$ 的关系为

$$
|\rho(\pi_1)-L(\pi_1,\pi_2)|\leq\frac{4\gamma\max_{s,a}|A(s,a;\pi_2)|}{1-\gamma}[D_{TV}^{\max}(\pi_1\|\pi_2)]^2. \tag{94}
$$

证明. 首先根据引理 4.4

$$
|\rho(\pi_1)-L(\pi_1,\pi_2)|
$$

$$
\begin{aligned}
=&\Bigg| \sum_s \mathbf{p}_\gamma(s;\pi_1) \sum_a \pi_1(a|s) A(s,a,\pi_2) \\
&- \sum_s \mathbf{p}_\gamma(s;\pi_2) \sum_a \pi_1(a|s) A(s,a,\pi_2) \Bigg| \\
\leq& \Bigg| \sum_s [\mathbf{p}_\gamma(s;\pi_1) - \mathbf{p}_\gamma(s;\pi_2)] \sum_a \pi_1(a|s) A(s,a;\pi_1) \Bigg| \\
\leq& \Bigg| \sum_s [\mathbf{p}_\gamma(s;\pi_1) - \mathbf{p}_\gamma(s;\pi_2)] \Bigg| \cdot 2 D_{TV}^{\max}(\pi_1 \| \pi_2) \max_{s,a} |A(s,a;\pi_2)|.
\end{aligned}
$$

接着，我们继续研究 $\mathbf{p}_\gamma(\cdot;\pi_1)$ 和 $\mathbf{p}_\gamma(\cdot;\pi_2)$ 之间的关系。

$$
\begin{aligned}
&\Bigg| \sum_s [\mathbf{p}_\gamma(s;\pi_1) - \mathbf{p}_\gamma(s;\pi_2)] \Bigg| \\
=& \Bigg| (1-\gamma) \sum_{t=0}^{\infty} \gamma^t [\mathbf{p}_t(s;\pi_1) - \mathbf{p}_t(s;\pi_2)] \Bigg| \\
\leq& (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \sum_s |\mathbf{p}_t(s;\pi_1) - \mathbf{p}_t(s;\pi_2)| \\
=& (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \cdot 2 D_{TV}(\mathbf{p}_t(\cdot;\pi_1) \| \mathbf{p}_t(\cdot;\pi_2)) \\
\leq& (1-\gamma) \sum_{t=0}^{\infty} \gamma^t \cdot 2t D_{TV}^{\max}(\pi_1 \| \pi_2) \\
=& \frac{2\gamma}{1-\gamma} D_{TV}^{\max}(\pi_1 \| \pi_2).
\end{aligned}
$$

其中，最后一步用到了引理 4.5。综上我们可得

$$
|\rho(\pi_1) - L(\pi_1,\pi_2)| \leq \frac{4\gamma \max_{s,a} |A(s,a;\pi_2)|}{1-\gamma} [D_{TV}^{\max}(\pi_1 \| \pi_2)]^2.
$$

□

通常，马尔可夫决策过程的奖励函数是一个有界函数，所以它的优势函数也是有界的。所以，只要能够保证 π_1 和 π_2 之间的最大变分差异限制在阈值内，就能保证 $\rho(\pi_1)$ 和 $L(\pi_1,\pi_2)$ 的差异限制在某个范围内。

置信域定理暗示了一个新的策略优化算法：在策略 π_n 处优化如下子问题

$$
\max_{\pi} L(\pi,\pi_n) - \frac{4\gamma \max_{s,a} |A(s,a;\pi_n)|}{1-\gamma} [D_{TV}^{\max}(\pi \| \pi_n)]^2. \tag{95}
$$

因为

$$L(\pi_n, \pi_n) - \frac{4\gamma \max_{s,a} |A(s,a;\pi_n)|}{1-\gamma}[D_{TV}^{\max}(\pi_n \| \pi_n)]^2 = \rho(\pi_n), \tag{96}$$

所以 (95) 式对应的子问题的解 π_{n+1} 一定能够满足

$$\rho(\pi_{n+1}) \geq L(\pi_{n+1}, \pi_n) - \frac{4\gamma \max_{s,a} |A(s,a;\pi_n)|}{1-\gamma}[D_{TV}^{\max}(\pi_{n+1} \| \pi_n)]^2 \geq \rho(\pi_n), \tag{97}$$

即策略一定会越来越好。

由于系数 $\frac{4\gamma \max_{s,a} |A(s,a;\pi_n)|}{1-\gamma}$ 并不好确定，通常将 (95) 式简化为一个置信域受限优化

$$\max_{\pi} L(\pi, \pi_n), \quad s.t. \quad D_{TV}^{\max}(\pi \| \pi_n) \leq \delta. \tag{98}$$

其中 δ 是人为设定的超参数。当 δ 足够小时，依旧可以保证策略是单调上升的。

4.2 置信域策略优化算法

在上一小节中，本文介绍了置信域定理，并提出了一个迭代优化算法。但是，当我们使用参数化的策略时，这个迭代优化算法不再高效。[37] 等人在此基础上，构造了一个更加实用的策略优化算法——**置信域策略优化算法** [37]（Trust Region Policy Optimization, TRPO）。本文将介绍它的推导，以及它和梯度优化算法中的**自然梯度优化算法** [4] 的关系，最后将揭示置信域策略优化算法本质上是在一个概率空间上对一个参数化的策略函数进行优化。

定义 4.3 (相对熵，Kullback-Leibler Divergence)**.** 任意两个分布 $P(x)$ 和 $Q(x)$，它们的相对熵是

$$D_{KL}(P\|Q) = \sum_x P(x) \ln \frac{P(x)}{Q(x)}. \tag{99}$$

对于连续分布 $p(x)$ 和 $q(x)$，它们的相对熵是

$$D_{KL}(p\|q) = \int_x p(x) \ln \frac{p(x)}{q(x)} \mathrm{d}x. \tag{100}$$

相对熵是衡量两个分布之间差异性的另一个距离。平斯科尔不等式揭示了相对熵和全变分差异之间的关系。

定理 4.2 (平斯科尔不等式, Pinsker Inequality)**.** 对于两个分布 P 和 Q, 他们的相对熵和全变分差异之间的关系是

$$D_{TV}(P\|Q) \leq \sqrt{\frac{1}{2}D_{KL}(P\|Q)}. \tag{101}$$

根据平斯科尔不等式可知, 我们可以通过限制两个分布之间的相对熵来间接限制两个分布之间的全变分差异。

我们定义两个策略 π_1 和 π_2 之间**最大相对熵**为:

$$D_{KL}^{\max}(\pi_1\|\pi_2) = \max_s D_{KL}(\pi_1(\cdot|s)\|\pi_2(\cdot\|s)). \tag{102}$$

平斯科尔不等式保证了 $[D_{TV}^{\max}(\pi_1\|\pi_2)]^2 \leq D_{KL}^{\max}(\pi_1\|\pi_2)/2$。我们将 (95) 式中的全变分差异替换为相对熵就可以得到一个新的算法:

$$\max_\pi L(\pi, \pi_n) - \frac{2\gamma \max_{s,a}|A(s,a;\pi_n)|}{1-\gamma} D_{KL}^{\max}(\pi_n\|\pi). \tag{103}$$

由于相对熵前的系数在无模型的设定下是无法求解的, 所以我们选择将其转化为一个受限优化问题:

$$\max_\pi L(\pi, \pi_n), \quad s.t. \quad D_{KL}^{\max}(\pi_n\|\pi) \leq \delta, \tag{104}$$

其中 δ 是一个人为设定的限制域半径。

这个受限优化问题依旧非常难求解, 置信域策略优化算法提出使用一系列近似技术来进一步构造真正实用的算法。 继续讨论接下来的简化技术前, 需要做出一些说明。我们使用参数 θ 来参数化策略, 所以 (104) 式可以改写为

$$\max_\theta L(\theta, \theta_n), \quad s.t. \quad D_{KL}^{\max}(\theta_n\|\theta) \leq \delta. \tag{105}$$

第一个问题是在无模型的设定下, $D_{KL}^{\max}(\theta_n\|\theta)$ 求解困难。这里人为构造一个简化的平均相对熵距离:

$$D_{KL}^{p}(\theta_n\|\theta) = \mathbb{E}_{s\sim p}[D_{KL}(\pi(\cdot|s;\theta_n)\|\pi(\cdot|s;\theta))]. \tag{106}$$

我们可以通过采样的方式来估计平均相对熵距离将受限优化问题改写为期望的形式可得

$$\begin{aligned} &\max_\theta \mathbb{E}_{s\sim \mathbf{p}_\gamma(\cdot;\theta_n), a\sim\pi(\cdot|s;\theta_n)}\left[\frac{\pi(a|s;\theta)}{\pi(a|s;\theta_n)} A(s,a;\theta_n)\right], \\ &s.t. \mathbb{E}_{s\sim \mathbf{p}_\gamma(\cdot;\theta_n), a\sim\pi(\cdot|s;\theta_n)}[\ln \pi(a|s;\theta_n) - \ln \pi(a|s;\theta)] \leq \delta. \end{aligned} \tag{107}$$

至此，每一步的受限优化问题依旧非常难解，需要进一步简化。

首先使用一阶泰勒展开式将原问题简化：

$$L(\theta,\theta_n)\approx L(\theta_n,\theta_n)+\nabla_{\theta=\theta_n}L(\theta,\theta_n)(\theta-\theta_n). \tag{108}$$

接着使用二阶泰勒展开式将限制条件简化：

$$\begin{aligned} D^p_{KL}(\theta_n\|\theta)\approx & D^p_{KL}(\theta_n\|\theta_n)+\nabla_{\theta=\theta_n}D^p_{KL}(\theta_n\|\theta)(\theta-\theta_n)\\ & +\frac{1}{2}(\theta-\theta_n)^T\nabla^2_{\theta=\theta_n}D^p_{KL}(\theta_n\|\theta)(\theta-\theta_n), \end{aligned} \tag{109}$$

（因为 θ_n 是函数的 $D^p_{KL}(\theta_n\|\theta)$ 的最小值点，所以它的一阶导数恒等于 0）。简化后的受限约束问题就变成了

$$\begin{aligned} \max_{\theta}\quad & g_n^T(\theta-\theta_n),\\ s.t.\quad & (\theta-\theta_n)^T H_n(\theta-\theta_n)\le 2\delta. \end{aligned} \tag{110}$$

其中 $g_n=\nabla_{\theta=\theta_n}L(\theta,\theta_n)$ 并且 $H_n=\nabla^2_{\theta=\theta_n}D^p_{KL}(\theta_n\|\theta)$。我们使用拉格朗日算子来将约束优化问题转化为无约束优化问题：

$$\max_{\theta}\min_{\lambda\ge 0} g_n^T(\theta-\theta_n)+\lambda[2\delta-(\theta-\theta_n)^T H_n(\theta-\theta_n)], \tag{111}$$

根据强对偶性可以得到一个等价问题

$$\min_{\lambda\ge 0}\max_{\theta} g_n^T(\theta-\theta_n)+\lambda[2\delta-(\theta-\theta_n)^T H_n(\theta-\theta_n)]. \tag{112}$$

首先我们求解内部问题，可得 $g_n-2\lambda H_n(\theta^*-\theta_n)=0$，即

$$\theta^*=\theta_n+H_n^{-1}g_n/(2\lambda). \tag{113}$$

带入 θ^2 后，外部问题变成了

$$\min_{\lambda\ge 0} g_n^T H_n^{-1}g_n/(2\lambda)+\lambda[2\delta-g_n^T H_n^{-1}g_n/(4\lambda^2)], \tag{114}$$

并且这个问题的最优解是

$$\lambda^*=\sqrt{\frac{g_n^T H_n^{-1}g_n}{8\lambda}}. \tag{115}$$

最后，带入 λ^* 后我们可得

$$\theta_{n+1}=\theta^*=\theta_n+\sqrt{\frac{2\delta}{g_n^T H_n^{-1}g_n}}H_n^{-1}g_n. \tag{116}$$

至此，我们已经获得了置信域策略优化算法的核心更新步骤。这个核心更新规则完全等价于优化理论中的**自然梯度下降法** [4]。其实也比较好理解：我们求解的带约束的优化问题本质上是在交叉熵为距离的**概率空间**中，求解点 π_n 邻域中的最快上升方向。因为它本质就是一种梯度的定义，所以我们可以将这个更新方式看成是在概率空间上进行的一次梯度下降。这个下降方式修正了参数化策略函数带来的影响，提高了梯度优化算法的效率。

算法 5: 置信域策略优化

Input: 总的优化步数 N, 可以交互采样的马尔可夫决策过程的环境 env。

Output: 最优策略 π^*

随机初始化一个值函数 $V(s;\theta_V)$ 和一个策略函数 $\pi(a|s;\theta_\pi)$;
for $n = 1, 2, \ldots, N$ **do**
  执行 $\pi(a|s;\theta_\pi)$ 从环境 env 中采集 M 条长度为 T 的轨迹 $\mathcal{D}_M = \{\tau_i\}_{i=1}^M$;
  $\hat{V}(s_t^{(i)}) = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}^{(i)}$;
  $\theta_V = \arg\min_{\theta_V} \frac{1}{2M(T+1)} \sum_{i=1}^{M} \sum_{t=0}^{T} [V(s_t^{(i)};\theta_V) - \hat{V}(s_t^{(i)})]^2$;
  $\hat{A}(s_t^{(i)}, a_t^{(i)}) = \sum_{t'=t}^{T-1} \lambda^{t'-t} [r_{t'}^{(i)} + \gamma V(s_{t'+1}^{(i)};\theta_V) - V(s_{t'}^{(i)};\theta_V)]$;
  $\hat{g}_n = \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=0}^{T-1} \nabla_\theta \ln \pi(a_t^{(i)}|s_t^{(i)};\theta_\pi) \hat{A}(s_t^{(i)}, a_t^{(i)})$;
  for $j = 0, 1, \ldots, K$ **do**
    $\theta'_\pi = \theta_\pi + \alpha^j \sqrt{\frac{2\delta}{\hat{g}_n^T \hat{H}_n^{-1} \hat{g}_n}} \hat{H}_n^{-1} \hat{g}_n$;
    if $D_{KL}^p(\theta_\pi \| \theta'_\pi) \leq \delta$ **then**
      $\theta_\pi = \theta'_\pi$;
      break;
    end
  end
end
return $\pi(a|s;\theta_\pi)$

这个算法在实践中有一些需要注意的点。首先因为我们需要求解概率密度 $\ln \pi(a|s;\theta_\pi)$ 的梯度（连续动作马尔可夫决策过程），所以算法对参数化的策略函数的形式有一定的限制，所以通常这个算法使用高斯分布作为策略函数的基本形式，然后使用神经网络来参数化高斯分布的均值和方差，即

$\pi(a|s;\theta_\pi) = \mathcal{N}(\mu(\theta_\pi), \sigma(\theta_\pi))$。其次，由于我们的更新方式在推导过程中使用了大量的近似产生了误差，所以算法在更新参数前进行了一步检验操作，来检测更新后的参数是否满足交叉熵的置信域要求。如果不满足要求，算法就对更新步长进行缩放，直到满足要求（这一步也叫做**回溯**，backtrack）。最后，算法可以使用**共轭梯度法** [41]（Conjugate Gradient Algorithm）（引用）来直接求解 $H_n^{-1}g_n$ 从而降低算法的计算复杂度。

4.3 近邻策略优化算法

置信域策略优化算法最大的问题就是它的计算复杂度非常高。尽管从理论的角度来说，置信域策略优化算法具有非常好的数学意义，它是向以相对熵作为距离的概率空间中的概率梯度方向进行优化的算法。但是作为强化学习来说，我们更需要一个计算更加简单基于置信域定理的策略优化算法。本节将介绍一个非常易于理解并且在实践中和置信域策略优化算法同样高效的算法——**近邻策略优化算法** [39]（Proximal Policy Optimization，PPO）。近似策略优化算法有非常多的形式，本小节将介绍其中最基本的形式——截断近似策略优化算法（PPO-clip）。

基于策略梯度公式的算法则在理论上能够保证梯度的单调上升，算法实现简单，同时也能够处理连续动作空间的问题，所以经常被大型的工程中 [47, 2, 9] 用来作为首选算法。但是它们需要执行当前策略来采集的样本来估计策略梯度，所以它们的采样复杂度非常高，所以后面的章节将介绍一个新的框架来解决这个问题。

近邻策略优化算法和置信域策略优化算法最核心的不同就是它们所使用的置信域不同，近邻策略优化算法使用了一个更加简单的置信域——ϵ-置信域。

定义 4.4 (ϵ-置信域)**.** 对于一个策略 π，它的 ϵ-置信域定义如下

$$\epsilon(\pi) = \{(1-\epsilon)\pi + \epsilon\pi', \forall\pi\}, \tag{117}$$

其中 $\epsilon \in (0, 1)$。

策略 π 的 ϵ-置信域中的每一个策略都等价于以 $1-\epsilon$ 的概率执行策略 π，以 ϵ 的概率执行某一个其他策略 π'。以下引理揭示了它和最大全变分差异的关系：

算法 6: 近邻策略优化

Input: 总的优化步数 N, 可以交互采样的马尔可夫决策过程的环境 env。

Output: 最优策略 π^*

随机初始化一个值函数 $V(s;\theta_V)$ 和一个策略函数 $\pi(a|s;\theta_\pi)$;

for $n = 1, 2, \ldots, N$ **do**

  执行 $\pi(a|s;\theta_\pi)$ 从环境 env 中采集 M 条长度为 T 的轨迹 $\mathcal{D}_M = \{\tau_i\}_{i=1}^M$;

  $\hat{V}(s_t^{(i)}) = \sum_{t'=t}^{T} \gamma^{t'-t} r_{t'}^{(i)}$;

  $\theta_V = \arg\min_{\theta_V} \frac{1}{2M(T+1)} \sum_{i=1}^{M} \sum_{t=0}^{T} [V(s_t^{(i)};\theta_V) - \hat{V}(s_t^{(i)})]^2$;

  $\hat{A}(s_t^{(i)}, a_t^{(i)}) = \sum_{t'=t}^{T-1} \lambda^{t'-t} [r_{t'}^{(i)} + \gamma V(s_{t'+1}^{(i)}, \theta_V) - V(s_{t'}^{(i)};\theta_V)]$;

  $\theta'_\pi = \theta_\pi$;

  for $j = 0, 1, \ldots, K$ **do**

    $rt(a_t^{(i)}|s_t^{(i)};\theta'_\pi,\theta_\pi) = \frac{\pi(a_t^{(i)}|s_t^{(i)};\theta'_\pi)}{\pi(a_t^{(i)}|s_t^{(i)};\theta_\pi)}$;

    $L(\theta'_\pi) = \frac{1}{MT} \sum_{i=1}^{M} \sum_{t=0}^{T-1} \min\{rt(a_t^{(i)}|s_t^{(i)};\theta'_\pi,\theta_\pi)\hat{A}(s_t^{(i)}, a_t^{(i)}), clip[rt(a^{(i)}|s_t^{(i)};\theta'_\pi), 1-\epsilon, 1/(1-\epsilon)]\hat{A}(s_t^{(i)}, a_t^{(i)})\}$;

    $\theta'_\pi = \theta'_\pi + \alpha \nabla_{\theta'_\pi} L(\theta'_\pi)$;

    if $D_{KL}^p(\theta_\pi \| \theta'_\pi) \geq \delta$ **then**

      $\theta_\pi = \theta'_\pi$;

      break;

    end

  end

end

return $\pi(a|s;\theta_\pi)$

引理 4.6. 如果 $\pi_2 \in \epsilon(\pi_1)$，那么

$$D_{TV}^{\max}(\pi_1 \| \pi_2) \leq \epsilon. \tag{118}$$

证明.

$$\begin{aligned} & D_{TV}^{\max}(\pi_1 \| \pi_2) \\ =& \max_s \frac{1}{2} \sum_a |\pi_1(a|s) - \pi_2(a|s)| \\ =& \max_s \frac{1}{2} \sum_a \epsilon |\pi_1(a|s) - \pi(a|s)| \leq \epsilon. \end{aligned}$$

□

从 ϵ-置信域中得到启发，我们可以构造一个新的受限优化问题：

$$\begin{aligned} \max_{\pi} \quad & L(\pi, \pi_n), \\ s.t. \quad & \pi \in \epsilon(\pi_n), \pi_n \in \epsilon(\pi). \end{aligned} \tag{119}$$

本文在这里特意没有使用参数化的策略，因为从未参数化的策略的角度更容易分析出一些问题。从未参数化的 π 所在的 $\mathcal{S} \times \mathcal{A}$ 的集合来看，$\epsilon(\pi)$ 是一个凸集合。另一方面，$L(\pi, \pi_n)$ 是关于 π 的一个线性函数。所以，最优值通常取在 ϵ-置信域的边界点上。那么，接下来我们来研究以下 ϵ 的边界的特点。

从 ϵ-置信域的定义可得：对于任意的 $s \in \mathcal{S}$ 和 $a \in \mathcal{A}$，

$$\begin{cases} \pi(a|s) \geq (1-\epsilon)\pi_n(a|s), \\ \pi_n(a|s) \geq (1-\epsilon)\pi(a|s). \end{cases} \tag{120}$$

化简可得

$$1 - \epsilon \leq \frac{\pi(a|s)}{\pi_n(a|s)} \leq \frac{1}{1-\epsilon}. \tag{121}$$

为了简化计算，我们构造一个近似的无约束优化问题（参数化策略函数）：

$$\begin{aligned} & \min_{\theta} L^{CLIP}(\theta, \theta_n) = \\ & \quad \mathbb{E}_{s \sim \mathbf{p}_\gamma(\cdot;\theta_n), a \sim \pi(\cdot|s;\theta_n)} \left[clip\left(rt(a|s;\theta,\theta_n), 1-\epsilon, \frac{1}{1-\epsilon} \right) A(s,a;\theta_n) \right]. \end{aligned} \tag{122}$$

其中 $rt(a|s;\theta,\theta_n) = \frac{\pi(a|s;\theta)}{\pi(a|s;\theta_n)}$。

这个无约束的优化问题有一定的合理性同时也存在着一些参数化引发的问题。首先我们先来看它的合理性。因为起始优化时 $\theta = \theta_n$，所以 $rt(a|s;\theta,\theta_n) = 1$。当 $A(s,a;\theta_n) > 0$ 时，梯度优化算法会将 $rt(a|s;\theta,\theta_n)$ 往 $1/(1-\epsilon)$ 方向优化。当 $A(s,a;\theta_n) < 0$ 时，梯度优化算法会将 $rt(a|s;\theta,\theta_n)$ 往 $1-\epsilon$ 方向优化。当 $rt(a|s;\theta,\theta_n)$ 超出边界值 $1/(1-\epsilon)$ 或者 $1-\epsilon$ 时，损失函数对应的梯度为 0，也就保证了 $rt(a|s;\theta,\theta_n)$ 不再继续远离边界值。同时梯度优化算法使用一个小步长，以此保证 $rt(a|s;\theta,\theta_n)$ 不要超出边界值过多。综上，损失函数 $L^{CLIP}(\theta,\theta_n)$ 保证了梯度优化算法更新的 θ 保持在置信域的附近。

接下来，我们解释以下参数化策略会引发的问题。假设 $rt(s|a;\theta,\theta_n)$ 已经优化到边界点了，但是 $rt(s'|a';\theta,\theta_n)$ 还未优化到边界点。那么梯度优化算法对 $rt(s'|a';\theta,\theta_n)$ 进行优化时，参数化的策略函数值 $\pi(a|s;\theta)$ 也会不可避免地被修改，即 $rt(s|a;\theta,\theta_n)$ 会被修改。这就引发了两个问题：一是 $rt(s|a;\theta,\theta_n)$ 会继续远离边界值；二是 $rt(s|a;\theta,\theta_n)$ 可能反向远离边界值。我们假设 $A(s,a;\theta_n) > 0$，那么 $rt(s|a;\theta,\theta_n)$ 应该往 $1/(1-\epsilon)$ 方向优化，但是由于其他状态动作点的影响，梯度有可能使 $rt(s|a;\theta,\theta_n) < 1-\epsilon$，反之亦然，这种现象我们称为**反向远离边界值**。反向远离边界值后，损失函数 $L^{CLIP}(\theta,\theta_n)$ 在 (s, a) 点同样不会产生梯度，梯度优化算法也就不会弥补这个问题。

针对第一个问题，算法在求解子问题 (122) 时，会检测当前策略 $\pi(a|s;\theta)$ 和 $\pi(a|s;\theta_n)$ 的相对熵是否超出了某个阈值，如果超出了阈值，就停止求解子问题。针对第二个问题，算法会对 $L^{CLIP}(\theta,\theta_n)$ 做一些修改:

$$\begin{aligned}\min_{\theta} L^{CLIP}(\theta,\theta_n) =& \mathbb{E}_{s\sim\mathbf{p}_\gamma(\cdot;\theta_n),a\sim\pi(\cdot|s;\theta_n)}\Bigg\{\min\Bigg[rt(a|s;\theta,\theta_n)A(s,a;\theta_n),\\ & clip\bigg(rt(a|s;\theta,\theta_n),1-\epsilon,\frac{1}{1-\epsilon}\bigg)A(s,a;\theta_n)\Bigg]\Bigg\}.\end{aligned} \tag{123}$$

当 $A(s,a;\theta_n) > 0$ 并且 $rt(s,a;\theta_n) < 1-\epsilon$ 时，损失函数会在 (s,a) 点取值到 $rt(a|s;\theta,\theta_n)A(s,a;\theta_n)$，因此梯度优化算法使 $rt(a|s;\theta,\theta_n)$ 往 $1/(1-\epsilon)$ 方向优化。当 $A(s,a;\theta_n) < 0$ 并且 $rt(s,a;\theta) > 1/(1-\epsilon)$ 时，损失函数会在 (s,a) 点取值到 $rt(a|s;\theta,\theta_n)A(s,a;\theta_n)$，因此梯度优化算法会使 $rt(a|s;\theta,\theta_n)$ 往 $1-\epsilon$ 方向优化。从而，新的损失函数解决了反向远离边界值的问题。

4.4 本章小结

本章在策略梯度优化算法中引入了置信域，构造了带约束的优化问题，提高了算法的稳定性。本章首先介绍了策略的值函数的差分以及它的特性，然后引出并证明了置信域定理。本章介绍了两个基于置信域定理构建的算法：一个是基于相对熵的置信域构建的算法——置信域策略优化算法，它本质上是一个自然梯度优化算法；二是基于 ϵ-置信域构建的算法——近邻策略优化，它相比于置信域策略优化算法来说，计算简单很多，并且在实际实验中效果和置信域策略优化算法相同。

5 批判执行算法

在前面的章节中介绍了两类强化学习算法：基于最优贝尔曼等式的算法和基于策略梯度公式的算法。本章节将介绍一种新的强化学习框架——批判执行算法（Actor-Critic Algorithm），结合了前面两种算法共同的特性，既可以处理连续动作空间的强化学习问题，又大大降低了采样复杂度。在本节中将介绍三个算法：一是基于确定性策略梯度的**深度确定性策略优化算法** [13]（Deep Deterministic Policy Gradient, DDPG），二是基于深度确定性策略优化算法的改进算法**双延迟深度确定性策略优化算法** [18]（Twin Delayed Deep Deterministic Policy Gradient, TD3），三是基于最大熵理论的**最大熵批判执行算法** [22]（Soft Actor-Critic，SAC）。

5.1 深度确定性策略优化算法

本小节将介绍首个解决连续性强化学习问题的异策略强化学习算法——深度确定性策略优化算法 [13]（Deep Deterministic Policy Optimization，DDPG）。它的流程非常简单，代码易于实现，所以到现在为止，它在强化学习中的使用频率也非常高。本小节将先介绍它的理论基础——确定性策略梯度 [43]（Deterministic Policy Gradient），然后再介绍它的具体流程。

5.1.1 确定性策略梯度

本小节的内容将涉及确定性策略梯度及其定义。顾名思义，确定性策略梯度就是指策略是确定性策略时，强化学习的目标函数关于策略参数的导数。在确定性策略梯度被提出前，人们一度认为关于确定性策略的梯度时不存在的，而 [43] 则提出了这个确定性策略梯度，并且证明了它正是随机策略梯度在随机性策略收敛为确定性策略后的极限。

首先我们参数化一个确定性策略。一个确定性策略在任何状态只会做出某个确定的动作，即 $a = \pi(s)$，所以一个参数化的确定性策略的形式为

$$a = \pi(s; \theta_\pi). \tag{124}$$

马尔可夫决策问题的目标是

$$\max_{\theta_\pi} \rho(\theta_\pi) = \int_s p_0(s) Q(s, \pi(s; \theta_\pi), \theta_\pi) \mathrm{d}s. \tag{125}$$

确定性策略梯度定理告诉我们可以对函数 $\rho(\theta_\pi)$ 求梯度。

定理 5.1 (确定性策略梯度, Deterministic Policy Gradient)**.** 一个马尔可夫决策过程 $\{\mathcal{S}, \mathcal{A}, p_0, P, R, \gamma\}$ 以及它的奖励函数 $\rho(\theta_\pi)$。如果我们使用确定性策略 $\pi(s;\theta_\pi)$，那么，奖励函数关于参数的梯度为

$$\begin{aligned}\rho(\theta_\pi) =& \int_s p_\gamma(s;\theta_\pi)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\mathrm{d}s \\ =&\mathbb{E}_{s\sim p_\gamma(\cdot;\theta_\pi)}\Big[\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\Big],\end{aligned} \tag{126}$$

其中 $p_\gamma(s;\theta_\pi) = (1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t(s;\theta_\pi)$，并且 $p_t(s;\theta_\pi)$ 表示马尔科夫链 $\mathcal{MDP}(\theta_\pi)$ 中 t 时刻 S_t 状态的分布。

证明.

$$\begin{aligned}&\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}\rho(\theta_\pi) \\ =&(1-\gamma)\int_s p_0(s)\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}Q(s,\pi(s;\theta_\pi);\theta_\pi)\mathrm{d}s \\ =&(1-\gamma)\int_s p_0(s)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\mathrm{d}s \\ &+(1-\gamma)\int_s p_0(s)\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}Q(s,a;\theta_\pi)\big|_{a=\pi(s;\theta_\pi)}\mathrm{d}s \\ =&(1-\gamma)\int_s p_0(s)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\mathrm{d}s \\ &+(1-\gamma)\int_s p_0(s)\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}\Big[r(s,a) \\ &+\gamma\int_{s'}p(s'|s,a)\pi(s',\pi(s';\theta_\pi);\theta_\pi)\Big]\Big|_{a=\pi(s;\theta_\pi)}\mathrm{d}s'\mathrm{d}s \\ =&(1-\gamma)\int_s p_0(s)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\mathrm{d}s \\ &+(1-\gamma)\gamma\int_s p_0(s)\int_{s'}p(s'|s,a) \\ &\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}Q(s',\pi(s';\theta_\pi);\theta_\pi)\Big|_{a=\pi(s;\theta_\pi)}\mathrm{d}s'\mathrm{d}s \\ =&(1-\gamma)\int_s p_0(s)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_a Q(s,a;\theta_\pi)\big|_{a=\pi(s;\theta_\pi)}\mathrm{d}s \\ &+(1-\gamma)\int_s \gamma p_1(s;\theta_\pi)\frac{\mathrm{d}}{\mathrm{d}\theta_\pi}Q(s,\pi(s;\theta_\pi);\theta_\pi)\mathrm{d}s \\ =&\ \vdots\end{aligned}$$

$$
\begin{aligned}
=&(1-\gamma)\int_s\sum_{t=0}^{\infty}\gamma^t p_t(s;\theta_\pi)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\mathrm{d}s\\
=&\int_s p_\gamma(s;\theta_\pi)\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\mathrm{d}s\\
=&\mathbb{E}_{s\sim p_\gamma(\cdot;\theta_\pi)}\Big[\nabla_{\theta_\pi}\pi(s;\theta_\pi)\cdot\nabla_{a=\pi(s;\theta_\pi)}Q(s,a;\theta_\pi)\Big].
\end{aligned}
$$

□

确定性策略梯度定理要求奖励函数关于动作的梯度存在，因此仅在连续动作空间的强化学习算法中存在。同时，确定性策略梯度需要获得 Q 函数关于动作的一阶导数，它对估计 Q 函数的要求比随机策略梯度的要求高得多。更进一步地说，确定性策略梯度需要更多的样本，才能被准确地估计。所以目前没有同策略的确定性策略梯度优化算法，而本章将介绍一个基于确定性策略的重要算法——深度确定性策略优化算法。

5.1.2 深度确定性策略优化算法的原理及主要流程

深度确定性策略优化算法是一个结合了确定性策略梯度定理和批判执行算法框架 [29] 的异策略优化算法。由于它是第一个解决连续动作空间的强化学习算法的异策略算法，并且由于它的原理非常简单且易于实现，使得它在强化学习领域使用频率很高 [23, 6, 15]。

深度确定性策略优化算法的核心是构建了如下两个优化目标：

$$
\begin{cases}
\max\limits_{\theta_\pi}\mathcal{L}(\theta_\pi)=\mathbb{E}_{s\sim\mathcal{D}}[Q(s,\pi(s;\theta_\pi);\theta_Q)], & (127)\\
\min\limits_{\theta_Q}\mathcal{L}(\theta_Q)=\mathbb{E}_{(s,a,r)\sim\mathcal{D}}\{r+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a)} & \\
\qquad\qquad [Q(s',\pi(s';\theta_\pi);\theta_Q)]-Q(s,a;\theta_Q)\}^2, & (128)
\end{cases}
$$

其中 $\mathcal{D}=\{(s,a,r,s')\}$ 表示经验池。

接下来我们对算法的原理进行简单的解释，对 DDPG 可以有两个角度的理解。首先从确定性策略梯度的角度来理解：第一个优化目标来自确定性策略梯度定理。当我们对 $\mathcal{L}(\theta_\pi)$ 进行梯度上升优化时，近似于对 $\rho(\theta_\pi)$ 进行梯度上升优化；第二个优化目标来自贝尔曼等式定理。当 $\mathcal{L}(\theta_Q)$ 求解到最优时，我们可以认为 $Q(s,a;\theta_Q)\approx Q(s,a;\theta_\pi)$。

接着，深度确定策略优化算法也可以从最优贝尔曼等式的角度来理解。第一个优化目标实际上是求解子问题 $\max_a Q(s,a)$，它正对应 Q-Learning 算

算法 7: 深度确定性策略优化算法

Input: 总的优化步数 N, 可以交互采样的马尔可夫决策过程的环境 env。

Output: 最优策略 $\pi(s;\theta_\pi^*)$。

Function `Update`($\mathcal{D}=\{(s_i,a_i,r_i,s_i')\}_{i=1}^m$)**:**

  构造动作集合 $\{a_i'=\pi(s_i';\theta_{\pi 2})\}_{i=1}^m$;

  $\mathcal{L}(\theta_{Q1})=\frac{1}{2m}\sum_{i=1}^m\{r_i+\gamma Q(s_i',a_i';\theta_{Q2})-Q(s_i,a_1;\theta_{Q1})\}^2$;

  $\mathcal{L}(\theta_{Q2})=\frac{1}{2}\|\theta_{Q2}-\theta_{Q1}\|^2$;

  $\mathcal{L}(\theta_{\pi 1})=-\frac{1}{m}\sum_{i=1}^m Q(s_i,\pi(s;\theta_{\pi 1});\theta_{Q1})$;

  $\mathcal{L}(\theta_{\pi 2})=\frac{1}{2}\|\theta_{\pi 2}-\theta_{\pi 1}\|^2$;

  使用 Adam 梯度下降法来优化 $\mathcal{L}(\theta_{\pi 1})$ 和 $\mathcal{L}(\theta_{Q1})$;

  使用梯度下降法来优化 $\mathcal{L}(\theta_{\pi 2})$ 和 $\mathcal{L}(\theta_{Q2})$;

end

随机初始化一个空的经验池 $\mathcal{D}$，两个同构策略函数 $\pi(s;\theta_{\pi 1})$ 和 $\pi(s;\theta_{\pi 2})$，以及两个同构的值函数 $Q(s,a;\theta_{Q1})$ 和 $Q(s,a;\theta_{Q2})$;

for $n=1,2,\ldots,N$ **do**

  执行策略 $\pi_\epsilon(s;\theta_{\pi 1})$ 从环境 env 中采集 m 条状态转移样本 $\mathcal{D}_n=\{(s_i,a_i,r_i,s_i')\}_{i=1}^m$，其中 $\pi_\epsilon(s;\theta_{\pi 1})=\mathcal{N}(\pi(s;\theta_{\pi 1}),\epsilon)$;

  $\mathcal{D}=\mathcal{D}\cup\mathcal{D}_n$;

  for $n'=1,2,\ldots,N'$ **do**

    从 $\mathcal{D}$ 中采集 m' 条样本 $\mathcal{D}_{n'}=\{(s_i,a_i,r_i,s_i')\}_{i=1}^{m'}$;

    `Update`($\mathcal{D}_{n'}$);

  end

end

return $\pi(s;\theta_{\pi 1})$

法中的取最值操作。由于在连续动作空间的强化学习问题中，我们无法直接获得最优动作，所以我们构造了一个优化问题来求解它。当 $\mathcal{L}(\theta_\pi)$ 求解到最优时，第二个优化目标则对应着最优贝尔曼等式。根据最优贝尔曼等式定理，当第二个优化目标同时也求到最优时，我们所获得的最终策略就是马尔科夫决策问题的最优策略。

深度确定策略优化算法的整体流程如伪代码 7 所示。在实际的算法中，算法为了增加策略的探索性来充分探索环境，和环境交互的策略实际上是一个以 $\pi(s;\theta_\pi)$ 为均值，超参数 ϵ 为方差的高斯分布。

5.2 双延迟深度确定性策略优化算法

本小节将介绍一个重要的深度确定性策略优化算法的改进算法——双延迟深度确定性策略优化算法 [18]（Twin Delayed Deterministic Policy Optimization，TD3）。它提出了几个经验性的技术来降低深度确定性策略优化算法的误差，显著提高了算法的性能。

第一个技术使用截断 Q 函数来解决参数化的 Q 函数过高估计引发的一系列问题。在 Q-Learning 算法中，被错误地高估的 Q 函数对算法的负面影响远大于被错误地低估地 Q 函数。如果参数化的 Q 函数对某个状态动作 (s', a') 过高估计时，最大化策略 $\pi_Q(s') \in \max_{a'} Q(s', a')$ 也会高概率地执行动作 a'。通常我们使用时序差分误差

$$\delta(s, a) = r(s', a') + \gamma Q(s', a') - Q(s, a) \tag{129}$$

作为参考值修正 $Q(s, a)$。所以对 (s', a') 状态的 Q 值高估时，同时也会使得整条状态转移链的值都被高估，最后导致策略被错误引导向高估的动作，从而无法充分地探索环境，最终算法陷入到一个性能较差的局部最优策略附近。但是，当 $Q(s', a')$ 被错误地低估时，策略会在发现采取其他动作的效果都不理想的时候，开始尝试动作 a'。此时，策略已经对环境做出了充分地探索，相对来说更不容易陷入到一个局部最优策略。

因为深度确定性策略优化算法也是一个贪心地尝试当前 $Q(s, a)$ 值更大动作的算法，所以这个现象在深度确定性策略优化算法中也会出现。双延迟深度确定性策略优化算法则使用了一个 Double-Q 函数的技巧来解决这个问题。具体的做法是：随机初始化两个同构的 Q 函数 $Q(s, a; \theta_{QA})$ 和 $Q(s, a; \theta_{QB})$，实际使用到的参数化的 Q 函数的形式是

$$Q(s, a) = \min(Q(s, a; \theta_{QA}), Q(s, a; \theta_{QB})). \tag{130}$$

算法 8: 双延迟深度确定性策略优化算法

Input: 总的优化步数 N, 可以交互采样的马尔可夫决策过程的环境 env。

Output: 最优策略 $\pi(s;\theta_\pi^*)$。

Function `Update`($\mathcal{D}=\{(s_i,a_i,r_i,s_i')\}_{i=1}^m$)**:**

  构造动作集合 $\{a_i' \sim \pi_\epsilon(s_i';\theta_{\pi 2})\}_{i=1}^m$;

  $\{Q_1(s_i,a_i)=\min(Q(s_i,a_i;\theta_{Q1A}),Q(s_i,a_i;\theta_{Q1B}))\}_{i=1}^m$;

  $\{Q_2(s_i',a_i')=\min(Q(s_i',a_i';\theta_{Q2A}),Q(s_i',a_i';\theta_{Q2B}))\}_{i=1}^m$;

  $\mathcal{L}(\theta_{Q1A},\theta_{Q1B})=\frac{1}{2m}\sum_{i=1}^m\{r_i+\gamma Q_2(s_i',a_i')-Q_1(s_i,a_1)\}^2$;

  $\mathcal{L}(\theta_{Q2A},\theta_{Q2B})=\frac{1}{2}\|\theta_{Q2A}-\theta_{Q1A}\|^2+\frac{1}{2}\|\theta_{Q2B}-\theta_{Q2B}\|^2$;

  $\mathcal{L}(\theta_{\pi 1})=-\frac{1}{m}\sum_{i=1}^m Q(s_i,\pi(s;\theta_{\pi 1});\theta_{Q1})$;

  $\mathcal{L}(\theta_{\pi 2})=\frac{1}{2}\|\theta_{\pi 2}-\theta_{\pi 1}\|^2$;

  使用 Adam 梯度下降法来优化 $\mathcal{L}(\theta_{\pi 1})$ 和 $\mathcal{L}(\theta_{Q1A},\theta_{Q1B})$;

  使用梯度下降法来优化 $\mathcal{L}(\theta_{\pi 2})$ 和 $\mathcal{L}(\theta_{Q2A},\theta_{Q2B})$（$\theta_{\pi 2}$ 的步长相对小一些）;

end

随机初始化一个空的经验池 $\mathcal{D}$，两个同构策略函数 $\pi(s;\theta_{\pi 1})$ 和 $\pi(s;\theta_{\pi 2})$，以及四个同构的值函数 $Q(s,a;\theta_{Q1A})$、$Q(s,a;\theta_{Q1B})$、$Q(s,a;\theta_{Q2A})$ 和 $Q(s,a;\theta_{Q2B})$;

for $n=1,2,\ldots,N$ **do**

  执行策略 $\pi_\epsilon(s;\theta_{\pi 1})$ 从环境 env 中采集 m 条状态转移样本 $\mathcal{D}_n=\{(s_i,a_i,r_i,s_i')\}_{i=1}^m$，其中 $\pi_\epsilon(s;\theta_{\pi 1})=\mathcal{N}(\pi(s;\theta_{\pi 1}),\epsilon)$;

  $\mathcal{D}=\mathcal{D}\cup\mathcal{D}_n$;

  for $n'=1,2,\ldots,N'$ **do**

    从 $\mathcal{D}$ 中采集 m' 条样本 $\mathcal{D}_{n'}=\{(s_i,a_i,r_i,s_i')\}_{i=1}^{m'}$;

    `Update`($\mathcal{D}_{n'}$);

  end

end

return $\pi(s;\theta_{\pi 1})$

从概率的角度来看，$Q(s,a;\theta_{QA})$ 和 $Q(s,a;\theta_{QB})$ 同时出现过高估计的概率小于单个 Q 函数，所以它能够降低过高估计 Q 值的风险。

第二个技巧是对目标策略光滑化，用于降低算法的方差。在深度确定性策略优化算法中，使用高斯噪声来提高策略的探索性，即使用策略

$$\pi_\epsilon(s;\theta_\pi) = \mathcal{N}(\pi(s;\theta_\pi),\epsilon) \tag{131}$$

来和环境交互。本算法将它扩展到了 Q 函数的优化目标中，本算法不再求解 $\pi(s;\theta_\pi)$ 对应的值函数 $Q(s,a;\theta_\pi)$，而是求解 $\pi_\epsilon(s;\theta_\pi)$ 对应的值函数 $Q(s,a;\theta_\pi,\epsilon)$。也就是说，Q 函数的优化目标就变成了

$$\begin{aligned}\min_{\theta_Q}\mathcal{L}(\theta_Q) = \mathbb{E}_{(s,a,r)\sim\mathcal{D}}\{r + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a),\epsilon'\sim\mathcal{N}(0,\epsilon)} \\ [Q(s',\pi(s';\theta_\pi)+\epsilon';\theta_Q)] - Q(s,a;\theta_Q)\}^2.\end{aligned} \tag{132}$$

从损失函数的角度来看，本双延迟深度确定性策略优化算法中，Q 函数不再逼近某个确定性策略的 $Q(s',\pi(s';\theta_\pi);\theta_Q)$ 而是它周围的邻域的 Q 值的平均值 $\mathbb{E}_{\epsilon'}[Q(s',\pi(s';\theta_\pi+\epsilon';\theta_Q)]$。所以 $Q(s,a;\theta_\pi,\epsilon)$ 相比于 $Q(s,a;\theta_\pi)$ 更加光滑。

第三个技巧是延迟更新目标策略函数。在深度确定性优化算法中，算法等速率地更新目标策略函数和目标 Q 函数（$\mathcal{L}(\theta_{\pi 2})$ 和 $\mathcal{L}(\theta_{Q2})$）。双延迟深度确定性策略优化算法建议：目标策略函数的更新速度应该比目标 Q 函数的更新速度要慢一些。

综上，双延迟深度确定性策略优化算法的整体流程如伪代码 8 所示。

5.3 最大熵批判执行算法

本小节将介绍一个目前在实验环境表现最好的基准算法——最大熵批判执行算法 [22]（Soft Actor-Critic，SAC）。本小节将先讲述最大熵批判执行算法的理论基础——最大熵马尔可夫决策过程 [21, 46, 27]（Maximum Entropy Markov Decision Processes），然后再介绍最大熵批判执行算法的基本原理和具体流程。

5.3.1 最大熵马尔可夫决策过程

本小节将介绍最大熵马尔可夫决策过程，它是一个结合了最大熵理论和马尔可夫决策过程的数学模型。

首先回顾对于一个马尔可夫决策过程 $\mathcal{MDP}=\{\mathcal{S},\mathcal{A},p,p_0,R,\gamma\}$，我们的优化目标是

$$\begin{aligned}\max_{\pi}\rho(\pi)=&(1-\gamma)\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi)}\left[\sum_{t=0}^{\infty}\gamma^t R(s_t,a_t)\right]\\=&\mathbb{E}_{s\sim p_\gamma(\cdot;\pi),a\sim\pi(\cdot|s)}[R(s,a)].\end{aligned}\tag{133}$$

其中 $p_\gamma(s;\pi)=(1-\gamma)\sum_{t=0}^{\infty}\gamma^t p_t(s;\pi)$，并且 $p_t(s;\pi)$ 表示在策略 π 对应的马尔科夫链 $\mathcal{MDP}(\pi)$ 中 t 时刻 S_t 状态的分布。而最大熵马尔可夫决策过程则是使用一个和策略的动作分布的熵（$\mathcal{H}(p)=-\int_x p(x)\ln p(x)\mathrm{d}x$）有关的正则项来构造的新的优化目标

$$\begin{aligned}&\max_{\pi}\rho_{\mathcal{H}}(\pi)\\&=\mathbb{E}_{s\sim p_\gamma(\cdot;\pi),a\sim\pi(\cdot|s)}[R(s,a)]+\alpha\mathbb{E}_{s\sim p_\gamma(\cdot;\pi)}[\mathcal{H}(\pi(\cdot|s))]\\&=\mathbb{E}_{s\sim p_\gamma(\cdot;\pi),a\sim\pi(\cdot|s)}\Big[R(s,a)+\alpha\mathcal{H}(\pi(\cdot|s))\Big]\\&=(1-\gamma)\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi)}\left[\sum_{t=0}^{\infty}\gamma^t\Big(R(s_t,a_t)+\alpha\mathcal{H}(\pi(\cdot|s))\Big)\right].\end{aligned}\tag{134}$$

这是一个牵一发而动全身的改变。当我们获得了最大熵马尔可夫决策过程优化目标 $\rho_{\mathcal{H}}(\pi)$，同时改变了其他相关函数（Q 函数、V 函数、贝尔曼操作和最优贝尔曼等式等）。在最大熵马尔可夫决策过程框架中，我们需要定义新的函数。

我们先来研究一种最符合直觉的定义。首先是状态动作值函数定义为

$$Q_\pi(s,a)=\mathbb{E}_{\tau\sim\mathcal{MDP}(\pi)}\left[\sum_{t=0}^{\infty}\gamma^t\Big(R(s_t,a_t)+\alpha\mathcal{H}(\pi(\cdot|s_t))\Big)\middle|s_0=s,a_0=a\right].\tag{135}$$

那么状态值函数的定义可以是

$$V_\pi(s)=\mathbb{E}_{a\sim\pi(\cdot|s)}[Q_\pi(s,a)].\tag{136}$$

对于任意的函数 $Q\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$，对应的最大熵贝尔曼操作为

$$\begin{aligned}T_\pi Q(s,a)=&R(s,a)+\alpha\mathcal{H}(\pi(\cdot|s))\\&+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')}[Q(s',a')],\end{aligned}\tag{137}$$

以及对应的最大熵最优贝尔曼操作为

$$\begin{aligned}TQ(s,a)=&\max_{\pi}R(s,a)+\alpha\mathcal{H}(\pi(\cdot|s))\\&+\gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')}[Q(s',a')].\end{aligned}\tag{138}$$

问题就出现在最优贝尔曼操作上。在这个版本的定义中，最优贝尔曼操作计算上比较困难，并且比较难继续分析。

为了解决上面版本中最优贝尔曼操作计算困难的问题，最大熵马尔可夫决策过程 [21, 36] 定义了另一个版本的符号。首先是**最大熵的状态动作值函数**定义为

$$\begin{aligned} Q_{\pi,\mathcal{H}}(s,a) =& \mathbb{E}_{\tau\sim\mathcal{MDP}(\pi)}\Bigg[R(s_0,a_0) \\ &+\sum_{t=1}^{\infty}\gamma^t\Big(R(s_t,a_t)+\alpha\mathcal{H}(\pi(\cdot|s_t))\Big)\Big|s_0=s,a_0=a\Bigg]. \end{aligned} \tag{139}$$

那么**最大熵状态值函数**的定义是

$$V_{\pi,\mathcal{H}}(s)=\mathbb{E}_{a\sim\pi(\cdot|s)}[Q_{\pi,\mathcal{H}}(s,a)]+\alpha\mathcal{H}(\pi(\cdot|s)). \tag{140}$$

对于任意的函数 $Q\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$，对应的**最大熵贝尔曼操作**是

$$\begin{aligned} T_{\pi,\mathcal{H}}Q(s,a) = R(s,a)+&\gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')} \\ &[Q(s',a')+\alpha\mathcal{H}(\pi(\cdot|s'))], \end{aligned} \tag{141}$$

以及对应的**最大熵最优贝尔曼操作**为

$$\begin{aligned} T_{\mathcal{H}}Q(s,a) = \max_{\pi} R(s,a)+&\gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')} \\ &[Q(s',a')+\alpha\mathcal{H}(\pi(\cdot|s'))]. \end{aligned} \tag{142}$$

引理 5.1. *对于任意的函数 $Q\in\mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$，对它做最大熵最优贝尔曼操作时，设*

$$\begin{aligned} \pi_Q = \arg\max_{\pi} R(s,a)+&\gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s')} \\ &[Q(s',a')+\alpha\mathcal{H}(\pi(\cdot|s'))], \end{aligned} \tag{143}$$

那么

$$\pi_Q(a|s)=\frac{\exp[Q(s,a)/\alpha]}{\int_{a''}\exp[Q(s,a'')/\alpha]\mathrm{d}a''}. \tag{144}$$

证明. 要求解分布 $\pi(\cdot|s)$，我们要求解带约束的子问题

$$\begin{aligned} &\max_{\pi(\cdot|s)}\int\pi(a|s)[Q(s,a)-\alpha\ln\pi(a|s)]\mathrm{d}a, \\ &s.t.\int\pi(a|s)\mathrm{d}a=1. \end{aligned} \tag{145}$$

我们使用拉格朗日乘子法将带约束的优化问题转化为

$$\max_{\pi(\cdot|s)} \min_{\lambda\neq 0} \int \pi(a|s)[Q(s,a) - \alpha \ln \pi(a|s)]\mathrm{d}a + \lambda\bigg(\int_a \pi(a|s)\mathrm{d}a - 1\bigg). \quad (146)$$

根据强对偶性，我们得到等价的问题

$$\min_{\lambda\neq 0} \max_{\pi(\cdot|s)} \int \pi(a|s)[Q(s,a) - \alpha \ln \pi(a|s)]\mathrm{d}a + \lambda\bigg(\int_a \pi(a|s)\mathrm{d}a - 1\bigg). \quad (147)$$

我们先求解内部子问题 $\max_{\pi(\cdot|s)}$，对 $\pi(a|s)$ 求偏导可得它的最优值满足

$$Q(s,a) - \alpha - \alpha \ln \pi^*(a|s) + \lambda = 0, \quad (148)$$

即

$$\pi^*(a|s) = \exp[Q(s,a)/\alpha - 1 + \lambda/\alpha]. \quad (149)$$

接下来，我们不显式求解 λ^*，而是利用 λ^* 一定满足的性质。因为 $\pi^*(a|s)$ 必须是一个分布，所以

$$\int_{a''} \pi^*(a''|s)\mathrm{d}a'' = \int_{a''} \exp[Q(s,a'')/\alpha - 1 + \lambda^*/\alpha]\mathrm{d}a'' = 1, \quad (150)$$

转化可得

$$\exp(1 - \lambda^*/\alpha) = \int_{a''} \exp[Q(s,a'')/\alpha]\mathrm{d}a''. \quad (151)$$

带入 (149) 式可得

$$\pi^*(a|s) = \frac{\exp[Q(s,a)/\alpha]}{\int_{a''} \exp[Q(s,a'')/\alpha]\mathrm{d}a''}. \quad (152)$$

□

定理 5.2 (最大熵贝尔曼等式)**.** *最大熵贝尔曼操作 $T_{\pi,\mathcal{H}}$ 是收缩映射，并且存在唯一的不动点 $Q_{\mathcal{H}}$，满足*

$$Q_{\pi,\mathcal{H}} = T_{\pi,\mathcal{H}} Q_{\pi,\mathcal{H}}. \quad (153)$$

证明. 首先，根据 $Q_{\pi,\mathcal{H}}$ 的定义可知它是 $T_{\pi,\mathcal{H}}$ 的不动点。所以，我们只需要证明最大熵贝尔曼操作时收缩映射，那么就可以证明最大熵贝尔曼操作存在唯一的不动点。

设任意两个向量 $Q_1, Q_2 \in \mathbb{R}^{|\mathcal{S}\times\mathcal{A}|}$，那么

$$\begin{aligned} &\|T_{\pi,\mathcal{H}} Q_1 - T_{\pi,\mathcal{H}} Q_2\| \\ =&\gamma \|\mathbb{E}_{s'\sim p(\cdot|s,a), a'\sim\pi(\cdot|s')}[Q_1 - Q_2]\| \\ \leq&\gamma \|Q_1 - Q_2\|. \end{aligned} \quad (154)$$

□

定理 5.3 (最大熵最优贝尔曼等式). 最大熵最优贝尔曼操作 $T_{\mathcal{H}}$ 是收缩映射，并且存在唯一的不动点 $Q^*_{\mathcal{H}}$，满足

$$Q^*_{\mathcal{H}} = T_{\mathcal{H}} Q^*_{\mathcal{H}}, \tag{155}$$

其中 $Q^*_{\mathcal{H}} = Q^*_{\pi^*_{\mathcal{H}},\mathcal{H}}$，并且 $\pi^*_{\mathcal{H}} = \arg\max_\pi \rho_{\mathcal{H}}(\pi)$。

至此，我们可以进入到最大熵生成批判算法的介绍。

5.3.2 最大熵批判执行算法的原理及主要流程

本小节将介绍一个结合最大熵马尔可夫决策过程和生成批判算法框架的异策略优化算法——最大熵批判执行算法 [22] (Soft Actor-Critic, SAC)。本算法是目前在连续动作空间的实验环境中效果最好的异策略优化算法 [14, 1]。

最大熵批判执行算法的核心是构建了如下两个优化目标

$$\begin{cases} \min\limits_{\theta_\pi} \mathcal{L}(\theta_\pi) = \mathbb{E}_{s\sim\mathcal{D}}[D_{KL}(\pi(\cdot|s;\theta_\pi)\|\pi_{\mathcal{H}}(\cdot|s;\theta_Q))], & (156) \\ \min\limits_{\theta_Q} \mathcal{L}(\theta_Q) = \mathbb{E}_{(s,a,r)\sim\mathcal{D}}\{r + \gamma\mathbb{E}_{s'\sim p(\cdot|s,a),a'\sim\pi(\cdot|s';\theta_\pi)}[Q(s',a';\theta_Q) \\ \qquad\qquad - \alpha\ln\pi(a'|s';\theta_\pi)] - Q(s,a;\theta_Q)\}^2, & (157) \end{cases}$$

其中 $\mathcal{D} = \{(s,a,r,s')\}$ 表示经验池，并且

$$\pi_{\mathcal{H}}(s,a;\theta_Q) = \frac{\exp[Q(s,a;\theta_Q)/\alpha]}{\int_{a''}\exp\{\exp[Q(s,a;\theta_Q)/\alpha]\}\mathrm{d}a''}. \tag{158}$$

我们解释一下最大熵批判执行算法的核心是求解最大熵最优贝尔曼等式的解，而 (157) 式正来自最大熵最优贝尔曼等式。但是在连续动作空间的强化学习问题设定下，$\pi_{\mathcal{H}}(s,a;\theta_Q)$ 无法直接求解，所以最大熵批判执行算法通过构造一个优化问题 (157) 式来求解 $\pi_{\mathcal{H}}(s,a;\theta_Q)$。

我们无法直接求解损失函数 (156) 式，所以我们需要对它进行进一步地变换

$$\begin{aligned} \mathcal{L}(\theta_\pi) =& \mathbb{E}_{s\sim\mathcal{D}}[D_{KL}(\pi(\cdot|s;\theta_\pi)\|\pi_{\mathcal{H}}(\cdot|s;\theta_Q))] \\ =& \mathbb{E}_{s\sim\mathcal{D},a\sim\pi(a|s;\theta_\pi)}\left[\ln\pi(a|s;\theta_\pi) - \ln\frac{\exp[Q(s,a;\theta_Q)/\alpha]}{\int_{a''}\exp[Q(s,a'';\theta_Q)/\alpha]\mathrm{d}a''}\right] \\ =& \mathbb{E}_{s\sim\mathcal{D},\epsilon\sim p}\left[\ln\pi[f(s,\epsilon;\theta_\pi)|s;\theta_\pi] - \ln\frac{\exp[Q(f(s,\epsilon;\theta_\pi),a;\theta_Q)/\alpha]}{\int_{a''}\exp[Q(s,a'';\theta_Q)/\alpha]\mathrm{d}a''}\right] \\ =& \mathbb{E}_{s\sim\mathcal{D},\epsilon\sim p}\left[\ln\pi[f(s,\epsilon;\theta_\pi)|s;\theta_\pi] - Q[f(s,\epsilon;\theta_\pi),a;\theta_Q]/\alpha\right] + F(\theta_Q) \end{aligned} \tag{159}$$

算法 9: 最大熵批判执行算法

Input: 总的优化步数 N, 可以交互采样的马尔可夫决策过程的环境 env。

Output: 最优策略 $\pi(s;\theta_\pi^*)$。

Function `Update`($\mathcal{D}=\{(s_i,a_i,r_i,s_i')\}_{i=1}^m$)**:**

  构造动作集合 $\{a_i' \sim \pi(s_i';\theta_\pi)\}_{i=1}^m$;

  $\{Q_1(s_i,a_i)=\min(Q(s_i,a_i;\theta_{Q1A}),Q(s_i,a_i;\theta_{Q1B}))\}_{i=1}^m$;

  $\{Q_2(s_i',a_i')=\min(Q(s_i',a_i';\theta_{Q2A}),Q(s_i',a_i';\theta_{Q2B}))\}_{i=1}^m$;

  $\mathcal{L}(\theta_{Q1A},\theta_{Q1B}) =$
  $\frac{1}{2m}\sum_{i=1}^m\{r_i+\gamma[Q_2(s_i',a_i')-\alpha\ln\pi(a_i'|s_i';\theta_\pi)]-Q_1(s_i,a_1)\}^2$;

  $\mathcal{L}(\theta_{Q2A},\theta_{Q2B})=\frac{1}{2}\|\theta_{Q2A}-\theta_{Q1A}\|^2+\frac{1}{2}\|\theta_{Q2B}-\theta_{Q2B}\|^2$;

  采样 $\{\epsilon_i\sim p\}_{i=1}^m$;

  $\mathcal{L}(\theta_\pi)=\frac{1}{m}\sum_{i=1}^m\{\alpha\ln\pi(f(s_i,\epsilon_i;\theta_\pi)|s;\theta_\pi)-$
  $\min[Q(s_i,f(s_i,\epsilon_i;\theta_\pi);\theta_{Q1}),Q(s_i,f(s_i,\epsilon_i;\theta_\pi);\theta_{Q2})]\}$;

  使用 Adam 梯度下降法来优化 $\mathcal{L}(\theta_\pi)$ 和 $\mathcal{L}(\theta_{Q1A},\theta_{Q1B})$;

  使用梯度下降法来优化 $\mathcal{L}(\theta_{Q2A},\theta_{Q2B})$;

end

随机初始化一个空的经验池 $\mathcal{D}$，两个同构策略函数 $\pi(s;\theta_\pi)$，以及四个同构的值函数 $Q(s,a;\theta_{Q1A})$、$Q(s,a;\theta_{Q1B})$、$Q(s,a;\theta_{Q2A})$ 和 $Q(s,a;\theta_{Q2B})$;

for $n=1,2,\ldots,N$ **do**

  执行策略 $\pi(a|s;\theta_\pi)$ 从环境 env 中采集 m 条状态转移样本 $\mathcal{D}_n=\{(s_i,a_i,r_i,s_i')\}_{i=1}^m$;

  $\mathcal{D}=\mathcal{D}\cup\mathcal{D}_n$;

  for $n'=1,2,\ldots,N'$ **do**

    从 $\mathcal{D}$ 中采集 m' 条样本 $\mathcal{D}_{n'}=\{(s_i,a_i,r_i,s_i')\}_{i=1}^{m'}$;

    `Update`($\mathcal{D}_{n'}$);

  end

end

return $\pi(s;\theta_\pi)$

其中对策略函数 $\pi(a|s;\theta_\pi)$ 使用了重参数化技术，具体是将原本的参数化的策略重参数化为 $a = f(s,\epsilon;\theta_\pi)$，其中 ϵ 服从某个先验分布 p。这时，我们就可以对策略损失函数使用随机梯度优化算法进行优化。最大熵批判执行算法的整体流程如伪代码 9 所示。

5.4 本章小结

本章介绍了三个基于批判执行框架的算法：一是提出最早、使用范围最广的异策略优化算法——深度确定性策略优化算法；二是深度确定性策略优化算法的改进算法——双延迟深度确定性策略优化算法；三是当前解决连续动作强化学习问题性能最好的异策略优化算法——最大熵批判执行算法。三个算法本质上将强化学习问题建模为一个两目标的优化问题。相比于基于策略梯度的同策略优化算法，本章的批判执行算法大大降低了采样复杂度，在仿真环境中取得了巨大的成功，但是由于异策略算法缺乏相应的理论保障，还未在现实世界中得到应用。

参考文献

[1] Joshua Achiam. Spinning Up in Deep Reinforcement Learning. 2018.

[2] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al. Solving rubik's cube with a robot hand. *arXiv preprint arXiv:1910.07113*, 2019.

[3] Eitan Altman. *Constrained Markov decision processes*, volume 7. CRC Press, 1999.

[4] Shun-Ichi Amari and Scott C Douglas. Why natural gradient? In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'98 (Cat. No. 98CH36181)*, volume 2, pages 1213–1216. IEEE, 1998.

[5] Jacob Andreas, Dan Klein, and Sergey Levine. Modular multitask reinforcement learning with policy sketches. In *International Conference on Machine Learning*, pages 166–175. PMLR, 2017.

[6] Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob McGrew, Josh Tobin, OpenAI Pieter Abbeel, and Wojciech Zaremba. Hindsight experience replay. In *Advances in neural information processing systems*, pages 5048–5058, 2017.

[7] Kenneth J Arrow, David Blackwell, and Meyer A Girshick. Bayes and minimax solutions of sequential decision problems. *Econometrica, Journal of the Econometric Society*, pages 213–244, 1949.

[8] Richard Bellman. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767, 1956.

[9] Christopher Berner, Greg Brockman, Brooke Chan, Vicki Cheung, Przemysław Dębiak, Christy Dennison, David Farhi, Quirin Fischer,

Shariq Hashme, Chris Hesse, et al. Dota 2 with large scale deep reinforcement learning. *arXiv preprint arXiv:1912.06680*, 2019.

[10] Dimitri P Bertsekas and John N Tsitsiklis. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE Conference on Decision and Control*, volume 1, pages 560–564. IEEE, 1995.

[11] Patrick Billingsley. *Probability and measure*. John Wiley & Sons, 2008.

[12] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[13] Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*, pages 4299–4307, 2017.

[14] Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, Yuhuai Wu, and Peter Zhokhov. Openai baselines, 2017.

[15] Árpád Fehér, Szilárd Aradi, Ferenc Hegedüs, Tamás Bécsi, and Péter Gáspár. Hybrid ddpg approach for vehicle motion planning. 2019.

[16] Eugene A Feinberg and Adam Shwartz. *Handbook of Markov decision processes: methods and applications*, volume 40. Springer Science & Business Media, 2012.

[17] Wendell H Fleming and Halil Mete Soner. *Controlled Markov processes and viscosity solutions*, volume 25. Springer Science & Business Media, 2006.

[18] Scott Fujimoto, Herke Van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. *arXiv preprint arXiv:1802.09477*, 2018.

[19] Evan Greensmith, Peter L Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5(Nov):1471–1530, 2004.

[20] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning*, pages 573–582, 2016.

[21] Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. *arXiv preprint arXiv:1702.08165*, 2017.

[22] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *arXiv preprint arXiv:1801.01290*, 2018.

[23] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

[24] Ronald A Howard. Dynamic programming and markov processes. 1960.

[25] Junling Hu, Michael P Wellman, et al. Multiagent reinforcement learning: theoretical framework and an algorithm. In *ICML*, volume 98, pages 242–250. Citeseer, 1998.

[26] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *ICML*, volume 2, pages 267–274, 2002.

[27] Hilbert J Kappen. Path integrals and symmetry breaking for optimal control theory. *Journal of statistical mechanics: theory and experiment*, 2005(11):P11011, 2005.

[28] Robert E Kass, Bradley P Carlin, Andrew Gelman, and Radford M Neal. Markov chain monte carlo in practice: a roundtable discussion. *The American Statistician*, 52(2):93–100, 1998.

[29] Vijay R Konda and John N Tsitsiklis. Actor-critic algorithms. In *Advances in neural information processing systems*, pages 1008–1014, 2000.

[30] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pages 1928–1937, 2016.

[31] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[32] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

[33] Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *ICML*, volume 99, pages 278–287, 1999.

[34] Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

[35] Kaj Rosling. Optimal inventory policies for assembly systems under random demands. *Operations Research*, 37(4):565–579, 1989.

[36] John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017.

[37] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897, 2015.

[38] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015.

[39] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[40] Lloyd S Shapley. Stochastic games. *Proceedings of the national academy of sciences*, 39(10):1095–1100, 1953.

[41] Jonathan Richard Shewchuk et al. An introduction to the conjugate gradient method without the agonizing pain, 1994.

[42] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484, 2016.

[43] David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. 2014.

[44] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[45] Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12:1057–1063, 1999.

[46] Emanuel Todorov. General duality between optimal control and estimation. In *2008 47th IEEE Conference on Decision and Control*, pages 4286–4292. IEEE, 2008.

[47] Oriol Vinyals, Igor Babuschkin, Junyoung Chung, Michael Mathieu, Max Jaderberg, Wojciech M Czarnecki, Andrew Dudzik, Aja Huang, Petko Georgiev, Richard Powell, et al. Alphastar: Mastering the real-time strategy game starcraft ii. *DeepMind blog*, page 2, 2019.

[48] C Watkins. Learning from delayed rewards. phd thesis, university of cambridge, cambridge, england, 1989.

[49] Aaron Wilson, Alan Fern, Soumya Ray, and Prasad Tadepalli. Multi-task reinforcement learning: a hierarchical bayesian approach. In *Proceedings of the 24th international conference on Machine learning*, pages 1015–1022, 2007.