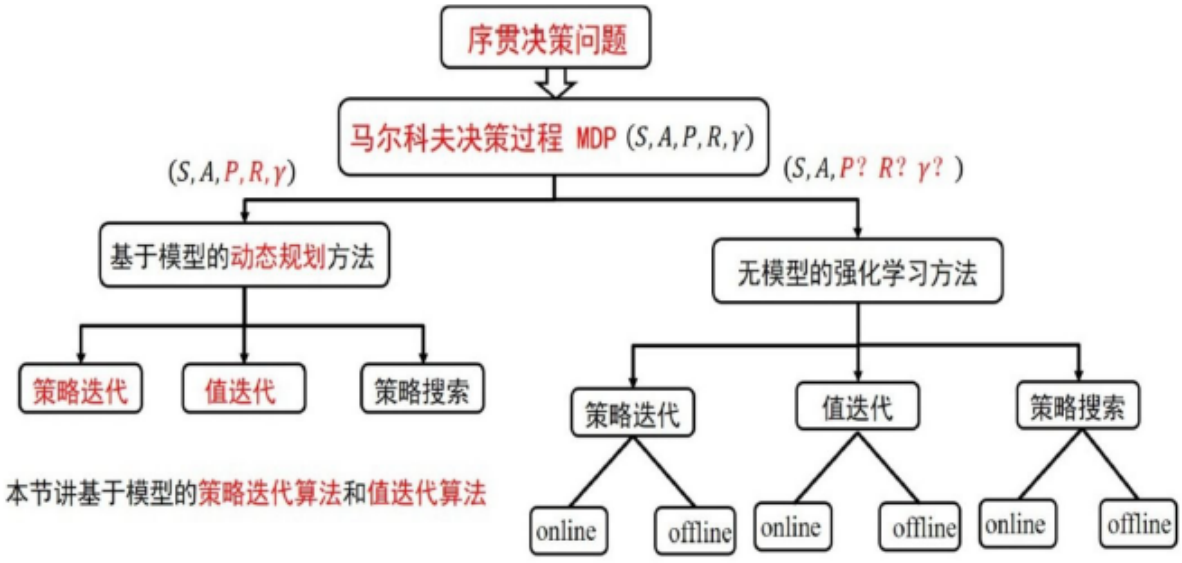
2 Model-Based Dynamic Programming Method

2.1 基于模型的动态规划方法理论

一个完整的已知模型的马尔可夫决策过程可以利用元组 (S, A, P, r, γ) 来表示。其中 S 为状态集， A 为动作集， P 为转移概率，也就是对应着环境和智能体的模型， r 为回报函数， γ 为折扣因子，用来计算累积回报 R 。累积回报公式为 $R=\sum_{t=0}^{T} \gamma^{t} r_{t}$ ，其中 $0 \leq \gamma \leq 1$ ，当 T 为有限值时，强化学习过程称为有限范围强化学习；当 $T=\infty$ 时，称为无限范围强化学习。

强化学习的目标是找到最优策略 π 使得累积回报的期望最大。所谓策略是指状态到动作的映射： $\pi: s \rightarrow a$ ，用 τ 表示从状态 s 到最终状态的一个序列 $\tau: s_{t}, s_{t+1}, \cdots, s_{T}$ ，则累积回报 $R(\tau)$ 是一个随机变量，随机变量无法进行优化，无法作为目标函数，我们采用随机变量的期望作为目标函数，即 $\int R(\tau) p_{\pi}(\tau) d \tau$ 作为目标函数。用公式来表示强化学习的目标： $\max _{\pi} \int R(\tau) p_{\pi}(\tau) d \tau$ 。强化学习的最终目标是找到最优策略为 $\pi^{*}: s \rightarrow u^{*}$ 。

从广义上来讲，强化学习可以归结为序贯决策问题，即找到一个决策序列，使得目标函数最优。这里目标函数是累积回报的期望值，累积回报的含义是评价策略完成任务的总回报，所以目标函数等价于任务。强化学习的直观目标是找到最优策略，目的是更好地完成任务。回报函数对应着具体的任务，所以强化学习所学到的最优策略是与具体的任务相对应的。从这个意义上来说，强化学习并不是万能的，它无法利用一个算法实现所有的任务。



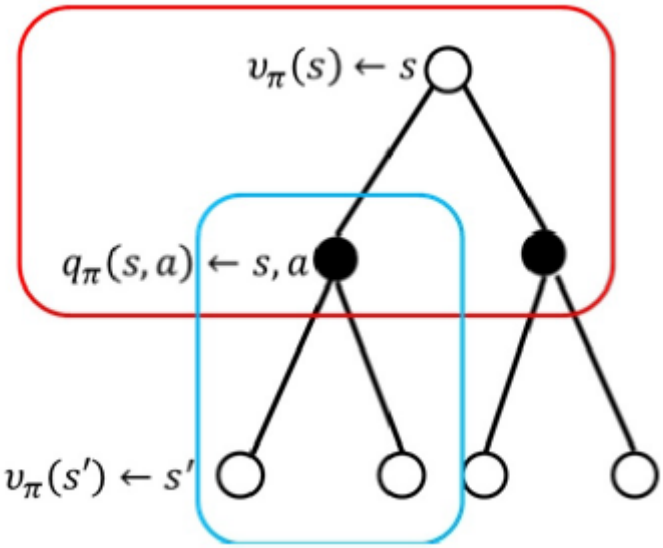
基于模型的强化学习可以利用动态规划的思想来解决。利用动态规划可以解决的问题需要满足两个条件：一是整个优化问题可以分解为多个子优化问题；二是子优化问题的解可以被存储和重复利用。强化学习可以利用马尔科夫决策过程来描述，利用贝尔曼最优性原理得到贝尔曼最优方程：

$$
\begin{aligned}
\nu^{*}(s) &= \max_{a} R_{s}^{a} + \gamma \sum_{s' \in S} P_{SS'}^{a} \nu^{*}(s') \\
q^{*}(s,a) &= R_{s}^{a} + \gamma \sum_{s' \in S} P_{SS'}^{a} \max_{a'} q^{*}(s',a')
\end{aligned}
\tag{1}
$$

从上式可以看出，马尔科夫决策问题符合使用动态规划的两个条件，因此可以用动态规划解决马尔科夫决策过程的问题。

2.1.1 求解在策略 π 下的值函数

贝尔曼方程指出，动态规划的核心是找到最优值函数。给定一个策略 π ，如何计算在策略 π 下的值函数?



图中上部大方框内的计算公式为

$$\nu_\pi(s) = \sum_{a \in A} \pi(a|s) q_\pi(s,a) \tag{2}$$

该方程表示，在状态 s 处的值函数等于采用策略 π 时，所有状态-行为值函数的总和。下面小方框的计算公式为状态-行为值函数的计算：

$$q_\pi(s,a) = R_s^a + \gamma \sum_{s' \in S} P_{SS'}^a \nu_\pi(s') \tag{3}$$

该方程表示，在状态 s 采用动作 a 的状态值函数等于回报加上后续状态值函数。

将方程（3）带入方程（2）便得到状态值函数的计算公式：

$$\nu_\pi(s) = \sum_{a \in A} \pi(a|s)(R_s^a + \gamma \sum_{s' \in S} P_{SS'}^a \nu_\pi(s')) \tag{4}$$

状态 s 处的值函数 $\nu_\pi(s)$ ，可以利用后继状态的值函数 $\nu_\pi(s')$ 来表示，即 **bootstrapping** 算法（自举算法）。

对于模型已知的强化学习算法，方程（4）中的 $P_{SS'}^a$ ， γ 和 R_s^a 都是已知数， $\pi(a|s)$ 为要评估的策略，是指定的。因此，方程（4）是关于值函数的线性方程组，其未知数的个数为状态的总数，用 $|S|$ 来表示。

此处，我们使用高斯-赛德尔迭代算法进行求解。即：

$$\nu_{k+1}(s) = \sum_{a \in A} \pi(a|s)(R_s^a + \gamma \sum_{s' \in S} P_{SS'}^a \nu_k(s')) \tag{5}$$

策略评估算法的伪代码如下图所示：

策略评估算法

[1] 输入：需要评估的策略 π 状态转移概率 $P_{ss'}^{a}$ 回报函数 R_s^a ， 折扣因子 γ

[2] 初始化值函数： $V(s)=0$

[3] Repeat k=0,1,…

一次状态扫描

[4] for every s do

[5] $$v_{k+1}(s)=\sum_{a\in A}\pi(a|s)\left(R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_k(s')\right)$$

[6] end for

[7] Until $v_{k+1}=v_k$

[8] 输出： $v(s)$

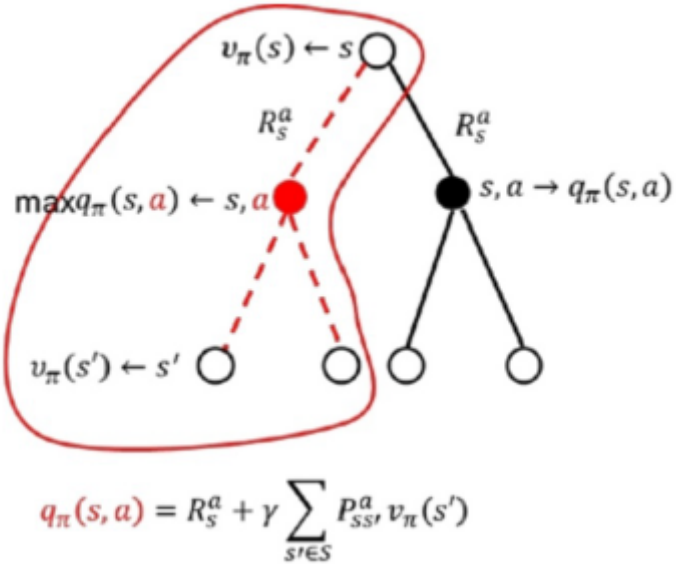
需要注意的是，每次迭代都需要对状态集进行一次遍历（扫描）以便评估每个状态的值函数。

2.1.2 求解最优策略

计算值函数的目的是利用值函数找到最优策略。如何利用值函数进行策略改善，从而得到最优策略?

一个很自然的方法是当已知当前策略的值函数时，在每个状态下采用贪婪策略对当前策略进行改善，即 $\pi_{l+1}(s)\in\arg\max_a q^{\pi_l}(s,a)$ 。

下图即贪婪策略示意图。



$$q_\pi(s,a)=R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_\pi(s')$$

2.1.3 策略迭代算法

将策略评估算法和策略改善算法组合起来便组成了策略迭代算法，如下图所示。

策略迭代算法

[1] 输入：状态转移概率 $P_{ss'}^{a}$ 回报函数 R_s^a ，折扣因子 γ

初始化值函数：$V(s)=0$ 初始化策略 π_0

[2] Repeat l=0,1,…

[3] find V^{π_l} Policy evaluation

[4] $\pi_{l+1}(s) \in \arg\max_a q^{\pi_l}(s,a)$ Policy improvement

[5] Until $\pi_{l+1}=\pi_l$

[6] 输出：$\pi^*=\pi_l$

策略迭代算法包括策略评估和策略改善两个步骤。在策略评估中，给定策略，通过数值迭代算法不断计算该策略下每个状态的值函数，利用该值函数和贪婪策略得到新的策略。如此循环下去，最终得到最优策略。这是一个策略收敛的过程。

如果在评估一次之后就进行策略改善，则称为值函数迭代算法。值函数迭代算法的伪代码为：

[1] 输入：状态转移概率 $P_{ss'}^{a}$ 回报函数 R_s^a ，折扣因子 γ

初始化值函数：$v(s)=0$ 初始化策略 π_0

[2] Repeat l=0,1,…

[3] for every s do

[4] $v_{l+1}(s)=\max_a R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_l(s')$

[5] Until $v_{l+1}=v_l$

[6] 输出：$\pi(s)=\arg\max_a R_s^a+\gamma\sum_{s'\in S}P_{ss'}^{a}v_l(s')$

在每次迭代过程中，需要对状态空间进行一次扫描，同时在每个状态对动作空间进行扫描以便得到贪婪的策略。值函数迭代是动态规划算法最一般的计算框架。

2.1.4 最优控制理论与值函数迭代的关系

解决最优控制的问题往往有三种思路：变分法原理、庞特里亚金最大值原理（极小值原理）和动态规划的方法。三种方法各有优缺点。基于庞特里亚金最大值原理的方法在变分法基础上进行发展，可以解决带约束的优化问题。相比于这两种经典的方法，动态规划的方法相对独立，主要是利用贝尔曼最优性原理。

对于一个连续系统，根据其状态方程和性能指标，由贝尔曼最优性原理可以得到哈密尔顿-雅可比-贝尔曼方程：

$$-\frac{\partial V}{\partial t}=\min_{u(t)\in U}\{L(x(t),u(t),t)+\frac{\partial V}{\partial X^T}f[x(t),u(t),t]\} \tag{6}$$

方程（6）是一个偏微分方程，一般不存在解析解。对于偏微分方程（6），往往有三种解决思路：第一种思路是将值函数进行离散，求解方程（6）的数值解，然后利用贪婪策略得到最优控制。这对应于求解微分方程的数值求解方法；第二种思路是利用变分法，将微分方程转化成变分代数方程，在标称轨迹展开，得到微分动态规划DDP；第三种思路是利用函数逼近理论，对方程中的回报函数和控制函数进行函数逼近，利用优化方法得到逼近系数，这类方法称为伪谱的方法。

前两种方法都是以值函数为中心，其思路与值函数迭代类似。

2.2 动态规划中的数学基础

2.2.1 线性方程组的迭代解法

利用方程（4）计算策略已知的状态值函数时，方程（4）为一个线性方程组。因此策略的评估变成了线性方程组的求解。线性方程组的数值求解包括直接法和迭代解法。策略评估中采用线性方程组的迭代解法。

方法一：雅可比（Jacobi）迭代法。

雅可比迭代法假设系数矩阵的对角元素 $a_{ii} \neq 0$，通过在第 i 个方程处分离出 x_i，构造迭代方程如下：

$$\begin{aligned} x_1 &= \frac{1}{a_{11}}(-a_{12}x_2 - a_{13}x_3 - \cdots - a_{1n}x_n + b_1) \\ x_2 &= \frac{1}{a_{22}}(-a_{21}x_1 - a_{23}x_3 - \cdots - a_{2n}x_n + b_2) \\ &\cdots \\ x_n &= \frac{1}{a_{nn}}(-a_{n1}x_1 - a_{n2}x_2 - \cdots - a_{n,n-1}x_{n-1} + b_n) \end{aligned} \tag{7}$$

写成矩阵形式为

$$X = -D^{-1}(L+U)X + D^{-1}b \tag{8}$$

其中 $B = -D^{-1}(L+U)$ 为迭代矩阵，迭代方程如下：

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}(-a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)} + b_1) \\ x_2^{(k+1)} &= \frac{1}{a_{22}}(-a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)} + b_2) \\ &\cdots \\ x_n^{(k+1)} &= \frac{1}{a_{nn}}(-a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \cdots - a_{n,n-1}x_{n-1}^{(k)} + b_n) \end{aligned} \tag{9}$$

方法二：高斯-赛德尔迭代法。

当求得新的分量之后，马上用来计算的迭代算法称为高斯-赛德尔迭代法。对于线性方程组，对应于雅克比迭代过程（9）的高斯-赛德尔迭代过程为：

$$\begin{aligned} x_1^{(k+1)} &= \frac{1}{a_{11}}(-a_{12}x_2^{(k)} - a_{13}x_3^{(k)} - \cdots - a_{1n}x_n^{(k)} + b_1) \\ x_2^{(k+1)} &= \frac{1}{a_{22}}(-a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \cdots - a_{2n}x_n^{(k)} + b_2) \\ &\cdots \\ x_n^{(k+1)} &= \frac{1}{a_{nn}}(-a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \cdots - a_{n,n-1}x_{n-1}^{(k+1)} + b_n) \end{aligned} \tag{10}$$

用矩阵的形式可表示为

$$(D+L)X^{(k+1)} = -UX^{(k)} \tag{11}$$

写成迭代方程为

$$X^{(k+1)} = GX^{(k)} + d_1 \tag{12}$$

其中 $G = -(D+L)^{-1}U$ ， $d_1 = (D+L)^{-1}b$ 。

2.2.2 压缩映射证明策略评估的收敛性

Contraction Mapping 中文可以译为压缩映射或压缩映像。

定义：设 X 是度量空间，其度量用 ρ 表示。映射 $T: X \to X$ ，若存在 a ， $0 \le a < 1$ 使得 $\rho(Tx, Ty) \le a\rho(x, y)$ ， $\forall x, y \in X$ ，则称 T 是 X 上的一个压缩映射；

若存在 $x_0 \in X$ 使得 $Tx_0 = x_0$ ，则称 x_0 是 T 的不动点。

定理1：完备度量空间上的压缩映射具有唯一的不动点。

定理1是说，从度量空间任何一点出发，只要满足压缩映射，压缩映射的序列必定会收敛到唯一的不动点。因此证明一个迭代序列是不是收敛，只要证明该序列所对应的映射是不是压缩映射。

对于值函数的求解，使用高斯-赛德尔迭代：

$$\nu_{k+1}(s) = \sum_{a \in A} \pi(a|s)(R_s^a + \gamma \sum_{s' \in S} P_{SS'}^a \nu_k(s')) \tag{13}$$

高斯-赛德尔解线性方程组迭代收敛条件是右面的未知数矩阵的谱半径小于1，这个条件也是由压缩映射定理得来的。在这里，我们不去求系数矩阵的谱，而是找一个特殊的度量 ρ 以便简化证明，这个度量我们选为无穷范数，即

$$||\nu||_\infty = \max_{s \in S} ||\nu(s)|| \tag{14}$$

从当前值函数到下一个迭代值函数的映射可以表示为

$$T^\pi(v) = R^\pi + \gamma P^\pi \nu \tag{15}$$

下面，我们证明该映射是一个压缩映射。

证明

$$\begin{aligned} \rho(T^\pi(u), T^\pi(\nu)) &= ||T^\pi(u) - T^\pi(\nu)||_\infty \\ &= ||(R^\pi + \gamma P^\pi u) - (R^\pi + \gamma P^\pi \nu)||_\infty \\ &= ||\gamma P^\pi (u - \nu)||_\infty \\ &\le ||\gamma P^\pi||(u - \nu)||_\infty||_\infty \\ &\le \gamma||(u - \nu)||_\infty \end{aligned} \tag{16}$$

因为 $0 \le \gamma < 1$ ，所以 $T^\pi(\nu)$ 是一个压缩映射，该迭代序列最终收敛到相应于策略 π 的值函数。