第八章　基于模型的学习和规划

多数强化学习问题可以通过表格式或基于近似函数来直接学习状态价值或策略函数，在这些学习方法中，个体并不试图去理解环境动力学。如果能建议一个较为准确地模拟环境动力学特征的模型或者问题的模型本身就类似于一些棋类游戏是明确或者简单的，个体就可以通过构建这样的模型来模拟其与环境的交互，这种依靠模型模拟而不实际与环境交互的过程类似于“思考”过程。通过思考，个体可以对问题进行规划、在与环境实际交互时搜索交互可能产生的各种后果并从中选择对个体有利的结果。这种思想可以广泛应用于规则简单、状态或结果复杂的强化学习问题中。

8.1　环境的模型

模型是个体构建的对于环境动力学特征的表示。在解决强化学习问题时，个体可以不需要建立一个模型，通过与环境直接进行交互而学习得到状态的价值函数或策略函数。在某些情况下，例如环境的动力学比较简单或者个体不想与环境进行过多的实际交互，个体可以与环境进行直接交互学习得到一个模型，再根据这个模型去构建状态的价值函数或策略函数。当个体得到了一个较为准确的描述环境动力学的模型时，它在与环境交互的过程中，既可以通过实际交互来提高模型的准确程度，也可以在交互间隙利用构建的模型进行思考、规划，决策出对个体有力的行为。基于模型的强化学习流程可以用图 8.1 来表示。

理论上来说，模型 M 是一个马尔科夫决策过程 $MDP\ \langle S, A, P, R\rangle$ 的参数化的表现形式。假设状态和行为空间是已知的，那么模型 $M = \langle P_\eta, R_\eta\rangle$ 则描述了环境动力学中的状态转换 $P_\eta \approx P$ 和奖励函数 $R_\eta \approx R$:

$$S_{t+1} \sim P_\eta(S_{t+1}|S_t, A_t)$$

$$R_{t+1} = R_\eta(R_{t+1}|S_t, A_t)$$

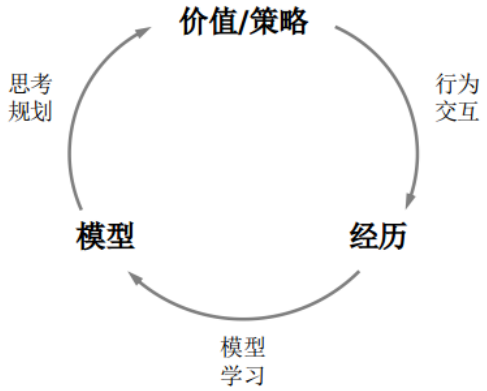

图 8.1: 基于模型的强化学习流程

并且假设状态转换和奖励之间是条件独立的:

$$\mathbb{P}[S_{t+1}, R_{t+1}|S_t, A_t] = \mathbb{P}[S_{t+1}|S_t, A_t]\ \mathbb{P}[R_{t+1}|S_t, A_t]$$

学习一个模型相当于从经历 $S_1, A_1, R_2, \ldots, S_T$ 中通过监督学习得到一个模型 M_η。其中:

1. 训练数据为:

$$S_1, A_1 \to R_2, S_2$$

$$S_2, A_2 \to R_3, S_3$$

$$\vdots$$

$$S_{T-1}, A_{T-1} \to R_T, S_T$$

2. 从 $s, a \to r$ 是一个回归问题; 从 $s, a \to s'$ 是一个概率密度估计问题。所有监督学习的相关算法都可以用来解决这两个问题。

根据具体使用的算法的不同和状态的特征表示模型可以有传统的查表式模型以及基于深度神经网络的模型等。各种模型的构建和学习其本质都是通过训练得到最符合经历数据的参数 η。下文仅通过查表式模型来解释模型的构建和学习。

查表式模型将经历得到的状态转移和概率存入一个表中, 需要时通过检索表格得到相关数据。其中状态转移概率和奖励的计算方法为:

$$\hat{P}^a_{ss'} = \frac{1}{N(s,a)} \sum_{t=1}^{T} 1(S_t, A_t, S_{t+1} = s, a, s')$$

$$\hat{R}_s^a = \frac{1}{N(s,a)} \sum_{t=1}^{T} (S_t, A_t = s, a) R_t$$

在实际使用模型虚拟一个经历时，并不直接使用上述公式，而是从符合当前状态和行为 (s,a) 的状态转换集合中随机的选择一个 $\langle s, a, \hat{r}, \hat{s}' \rangle$ 作为虚拟经历。这里的随机选择也就体现了状态 s 后续状态的概率分布。

模型的建立是为了解决问题，使用模型来解决问题是通过规划过程来进行的，规划的过程相当于解决一个 MDP 的过程，即给定一个模型 $M_\eta = \langle P_\eta, R_\eta \rangle$，求解基于该模型的 $MDP\ \langle S, A, P_\eta, R_\eta$，最终找到基于该模型的最优价值函数或最优策略。求解已知 MDP 的强化学习问题可以本书一开始介绍的价值迭代、策略迭代等方法来进行，对于状态和行为空间规模较大的 MDP 问题，可以使用基于模型的采样，在采样得到的虚拟经历基础上使用不基于模型的强化学习方法，例如 MC 学习、TD 学习等方法。由于实际经历的不足或者一些无法避免的缺陷，通过实际经历学习得到的模型不可能是完美的模型，即：

$$\langle P_\eta, R_\eta \rangle \neq \langle P, R \rangle$$

而从基于不完美模型的 MDP 中学习得到的最优策略通常也不是实际问题的最优策略，这就要求个体在环境实际交互的同时要不断的更新模型参数，基于更新模型来更新最优策略。这种使用近似的模型解决强化学习问题与使用价值函数或策略函数的近似表达来解决强化学习问题并不冲突，它们是从不同角度来近似求解一个强化学习问题，当构建一个模型比构建近似价值函数或近似策略函数更方便时，那么使用近似模型来求解会更加高效。使用模型来解决强化问题时要特别注意模型参数要随着个体与环境交互而不断地动态更新，即通过实际经历要与使用模型产生的虚拟经历相结合来解决问题，这就催生了一类整合了学习与规划的强化学习算法——Dyna 算法。

8.2 整合学习与规划——Dyna 算法

Dyna 算法从实际经历中学习得到模型，同时联合使用实际经历和基于模型采样得到的虚拟经历来学习和规划，更新价值和 (或) 策略函数 (图 8.2)。

基于行为价值的 Dyna-Q 算法的流程如算法 7 所述。

8.3 基于模拟的搜索

在强化学习中，基于模拟的搜索 (simulation-based search) 是一种前向搜索形式，它从当前时刻的状态开始，利用模型来模拟采样，构建一个关注短期未来的前向搜索树，将构建得到的搜

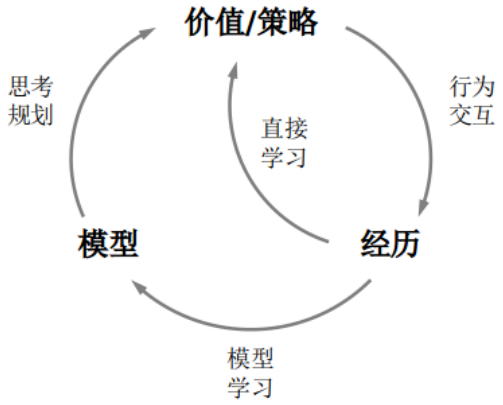

图 8.2: 基于模型的强化学习流程

算法 7: Dyna-Q 算法

输入: Q, γ, α
输出: optimized Q
initialize: $Q(s,a)$ and $Model(s,a)$ for all $s \in \mathbb{S}$ and $a \in \mathbb{A}(s)$
repeat for ever
  $S \leftarrow$ current(nonterminal) state
  $A \leftarrow \epsilon - greedy(S,Q)$
  execute action A; observe resultant reward R and next state S'
  $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma max_a Q(S',a) - Q(S,A)]$
  $Model(S,A) \leftarrow R, S'$ (assuming deterministic environment)
  repeat n times
    $S \leftarrow$ random previously observed state
    $A \leftarrow$ random action previously taken in S
    $R, S' \leftarrow Model(S,A)$
    $Q(S,A) \leftarrow Q(S,A) + \alpha[R + \gamma max_a Q(S',a) - Q(S,A)]$
  until;
until;

索树作为一个学习资源，使用不基于模型的强化学习方法来寻找当前状态下的最优策略 (图 8.3)。如果使用蒙特卡罗学习方法则称为蒙特卡罗搜索，如果使用 Sarsa 学习方法，则称为 TD 搜索。其中蒙特卡罗搜索又分为简单蒙特卡罗搜索和蒙特卡罗树搜索。

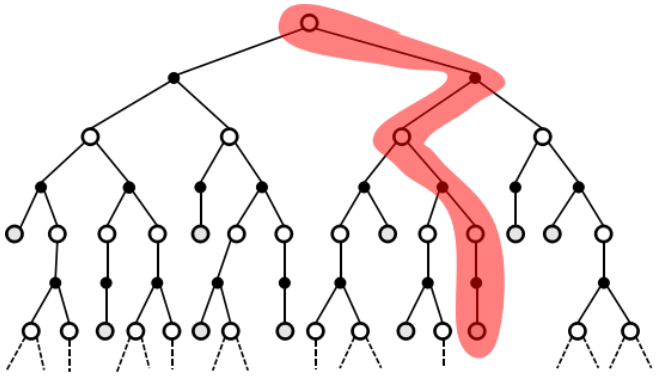

图 8.3: 从状态 S_t 开始的基于模拟的搜索

8.3.1 简单蒙特卡罗搜索

对于一个模型 M_v 和一个一致的模拟过程中使用的策略 π，简单蒙特卡罗搜索在当前实际状态 s_t 时会针对行为空间中的每一个行为 $a \in \mathbb{A}$ 进行 K 次的模拟采样：

$$\{s_t, a, R_{t+1}^k, S_{t+1}^k, A_{t+1}^k, \ldots, S_T^k\}_{k=1}^K \sim M_v, \pi$$

通过计算模拟采样得到的 k 个状态 s_t 时采取行为 s 的收获的平均值来估算该状态行为对的价值：

$$Q(s_t, a) = \frac{1}{K} \sum_{k=1}^{K} G_t$$

比较行为空间中所有行为 a 的价值，确定当前状态 s_t 下与环境发生实际交互的行为 a_t：

$$a_t = \underset{a \in \mathbb{A}}{argmax} Q(s_t, a)$$

简单蒙特卡罗搜索可以使用基于模拟的采样对当前模拟采样的策略进行评估，得到基于模拟采样的某状态行为对的价值，这个价值的估计同时还与每次采样的 K 值大小有关。在估算行为价值时，关注点在于从当前状态和行为对应的收获，并不关注模拟采样得到的一些中间状态和对应行为的价值。如果同时考虑模拟得到的中间状态和行为的价值，则可以考虑蒙特卡罗树搜索。

8.3.2 蒙特卡罗树搜索

蒙特卡罗树搜索 (Monte-Carlo tree search, MCTS) 在构建当前状态 s_t 的基于模拟的前向搜索时，关注模拟采样中所经历的所有状态及对应的行为，以此构建一个搜索树。利用这颗搜索树不仅可以对当前模拟策略进行评估，还可以改善模拟策略。在使用蒙特卡罗树搜索进行模拟策略评估时，对于个体构建的模型 M_v 和当前的模拟策略 π，在实际当前状态 s_t 时模拟采样出 K 个完整状态序列：

$$\{s_t, A_t^k, R_{t+1}^k, S_{t+1}^k, \ldots, S_T^k\}_{k=1}^K \sim M_v, \pi$$

构建一颗以状态 s_t 为根节点包括所有已访问的状态和行为的搜索树，对树内的每一个状态行为对 (s, a) 使用该状态行为对的平均收获来估算其价值：

$$Q(s,a) = \frac{1}{N(s,a)} \sum_{k=1}^{K} \sum_{u=t}^{T} 1(S_u, A_u = s, a) G_u$$

当搜索结束时，比较当前状态 s_t 下行为空间 $\mathbb{A}$ 内的每一个行为的价值，从中选择最大价值对应的行为 a_t 作为当前状态 s_t 时个体与环境实际交互的行为。

比较简单蒙特卡罗搜索和蒙特卡罗树搜索，可以看出两者之间的区别在于前者针对当前状态 s_t 时每一个可能的行为都进行相同数量的采样，而后者则是根据模拟策略进行一定次数的采样。此外，蒙特卡罗树搜索会对模拟采样产生的状态行为对进行计数，并计算其收获，根据这两个数据来计算模拟采样对应的状态行为对价值。比较两者之间的差别可以看出，如果问题的行为空间规模很大，那么使用蒙特卡罗树搜索比简单蒙特卡罗搜索要更实际可行。在蒙特卡罗树搜索中，搜索树的广度和深度是伴随着模拟采样的增多而逐渐增多的。在构建这个搜索树的过程中，搜索树内状态行为对的价值也在不停的更新，利用这些更新的价值信息可以使得在每模拟采样得到一个完整的状态序列后都可以一定程度地改进模拟策略。通常蒙特卡罗树搜索的策略分为两个阶段：

1. 树内策略 (tree policy)：为当模拟采样得到的状态存在于当前的搜索树中时适用的策略，该策略。树内策略可以使 ϵ-贪婪策略，随着模拟的进行可以得到持续改善；

2. 默认策略 (default policy)：当前状态不在搜索树内时，使用默认策略来完成整个状态序列的采样，并把当前状态纳入到搜索树中。默认策略可以使随机策略或基于某目标价值函数的策略。

随着不断地重复模拟，状态行为对的价值将得到持续地得到评估。同时搜索树的深度和广度将得到扩展，策略也不断得到改善。蒙特卡罗树搜索较为抽象，本章暂时介绍到这里，在第十章介绍 AlphaZero 算法时会利用五子棋实例详细讲解蒙特卡罗树搜索的过程细节。