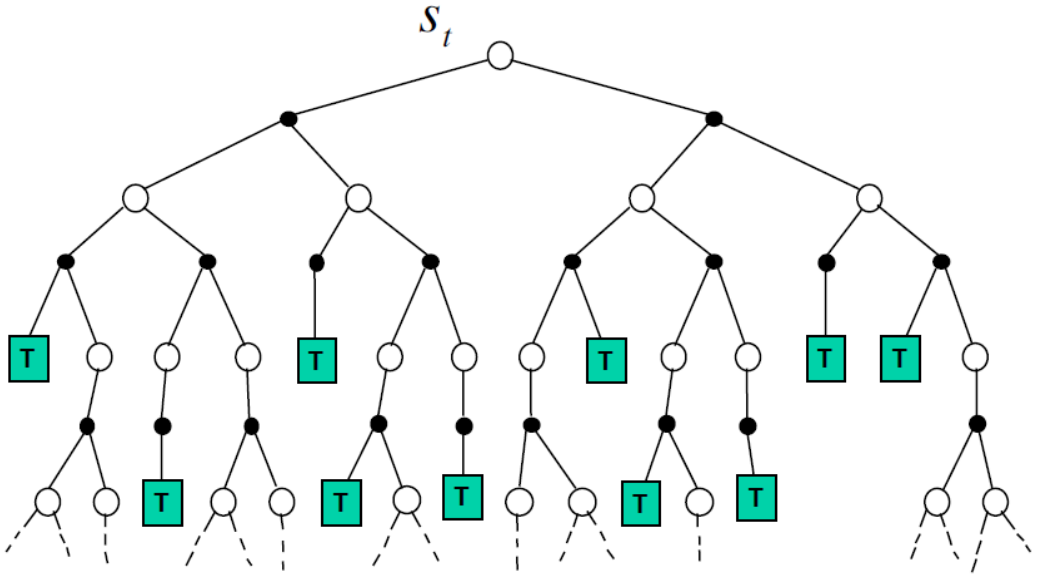
强化学习基础篇（三十四）基于模拟的搜索算法

上一篇Dyna算法是基于真实经验数据和模拟经验数据来解决马尔科夫决策过程的问题。本篇将结合前向搜索和采样法，构建更加高效的搜索规划算法，即基于模拟的搜索算法。

1、前向搜索算法（Forward Search）

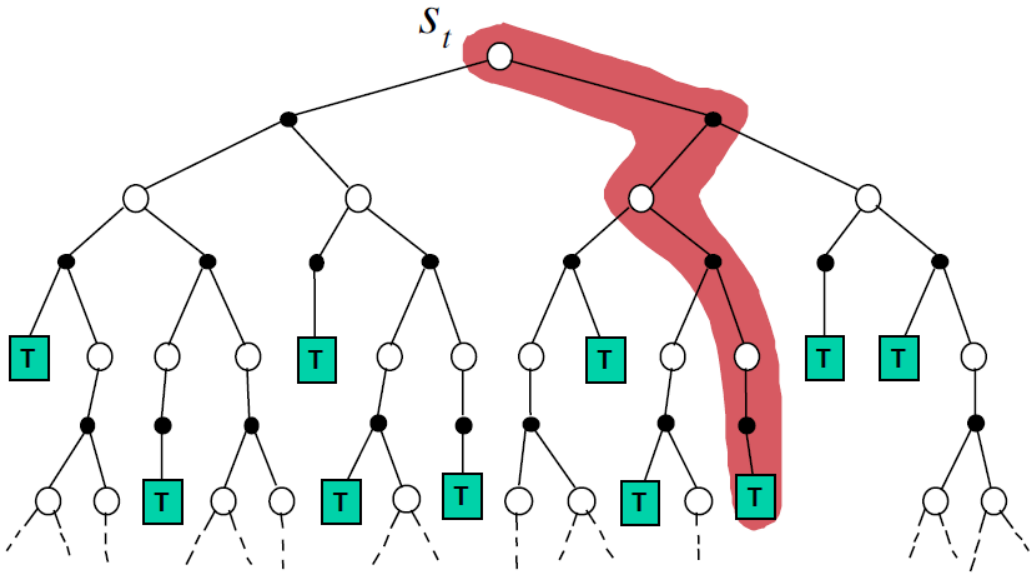
前向搜索算法将当前状态s_t作为根节点构建一个搜索树，并使用马尔科夫决策过程模型进行前向搜索。需要注意的是前向搜索主要关注的是从当前状态s_t开始构建的马尔科夫决策过程，而非整个马尔科夫决策过程。



2、基于模拟的搜索（Simulation-Based Search）

基于模拟的搜索算法从当前时间步t开始，在环境模型或者实际环境中进行采样，生成当前状态s_t到终止状态s_T的K条经验模拟轨迹：

$$\{s_t^k, A_t^k, R_{t+1}^k, \ldots, S_T^k\}_{k=1}^K \sim \mathcal{M}_\nu$$



在获得模拟经验轨迹数据后，使用model-free的强化学习算法，求解价值函数或者策略函数。基于蒙特卡洛控制算法的模拟搜索称为蒙特卡洛搜索，基于Sarsa算法或者Q-learning的模拟搜索称为时间差分搜索。

3、蒙特卡洛搜索

蒙特卡洛搜索是基于模拟的搜索中最为简单的一种形式。其实现形式简单，运行速度快。但由于该方法基于特定的模拟策略π，如果模拟策略π自身并非较优策略，基于模拟策略π下产生的动作很可能不是状态s下的较优动作。

蒙特卡洛搜索的具体步骤如下：

- （1）给定环境模型M_v和模拟策略π。
- （2）针对动作空间A中每一个动作a，从当前状态s_t开始模拟出K条模拟经验轨迹：

$$\left\{s_t, a, R_{t+1}^k, S_{t+1}^k, A_{t+1}^k, \ldots, S_T^k\right\}_{k=1}^K \sim \mathcal{M}_\nu, \pi$$

- （3）使用平均奖励评估动作a的动作价值

$$Q\left(s_t, a\right)=\frac{1}{K} \sum_{k=1}^K G_t \stackrel{P}{\rightarrow} q_\pi\left(s_t, a\right)$$

- （4）选择动作值函数$Q(s_t, a)$的极大值，作为当前状态s_t下的最优动作a_t^*：

$$a_t^*=\underset{a \in \mathcal{A}}{\operatorname{argmax}} Q\left(s_t, a\right)$$

4、蒙特卡洛树搜索

简单蒙特卡洛搜索算法中，模拟策略π可保持不变，导致最终获得的动作不一定是针对当前状态的最优动作。本节介绍的蒙特卡洛树搜索法，通过评估基于当前模拟策略π构建的搜索树中的每一个动作值，并基于评估的动作值改进模拟策略π。随后不断重复上述评估与改进的过程，使得最终改进的模拟策略能够生成更优的动作。
蒙特卡洛树搜索算法分为选择、扩展、模拟和回溯4个步骤。而本节为了将蒙特卡洛树搜索和强化学习中的策略改进过程结合起来，将主要介绍评估和模拟两个阶段，以更好理解简单蒙特卡洛搜索和蒙特卡洛树搜索的差异。

评估

蒙特卡洛搜索树的评估，主要指衡量基于模拟策略π针对当前状态s_t所构建的搜索树中的每一个(状态，动作)对的价值，其具体步骤如下：

- （1）给定环境模型M_v。
- （2）使用模拟策略π，从当前状态s_t模拟出K条模拟经验轨迹：

$$\left\{s_t, A_t^k, R_{t+1}^k, S_{t+1}^k, \ldots, S_T^k\right\}_{k=1}^K \sim \mathcal{M}_\nu, \pi$$

- （3）基于上一步生成的模拟经验轨迹数据集，生成包括智能体所经历过（状态，动作）对的搜索树。
- （4）针对上一步生成的搜索树，计算搜索树中每个（状态，动作）对从开始到终止状态的一个完整经验轨迹的平均奖励，作为该（状态，动作）对的动作价值$Q(s, a)$：

$$Q(s, a)=\frac{1}{N(s, a)} \sum_{k=1}^K \sum_{u=t}^T \mathbf{1}\left(S_u, A_u=s, a\right) G_u \stackrel{P}{\rightarrow} q_\pi(s, a)$$

- （5）当所有（状态-动作）对的价值得到更新后，选择动作值函数$Q(s_t, a)$的极大值，作为当前状态s_t下的最优动作a_t^*：

$$a_t^*=\underset{a \in \mathcal{A}}{\operatorname{argmax}} Q\left(s_t, a\right)$$

模拟

由评估过程可知，在搜索树的构建过程中，其中所有的<状态-动作>对的价值都得到更新。而更新后的一状态-动作>对价值信息，可用于改进模拟策略π（类似于策略优化过程），即选取能够最大化动作值的动作。

需要注意的是，由于构建的搜索树并不包括所有<状态。动作>对空间的价值，所以每次模拟（从当前状态s到终止状态s_T）都包含了2个部分：搜索树内状态以及搜索树外状态。策略改进时要分情况进行处理：针对搜索树内状态采用树内确定性策略，针对搜索树外状态采用树外默认策略。

- 树内确定性策略：对于搜索树中已存在的（状态，动作）对，策略的更新倾向于选择使得Q值最大化的动作。随着模拟的进行，已存在的（状态，动作）对的策略会持续得到改进。
- 树外默认策略：对于搜索树中不包含的状态，可采用随机策略对状态进行选择。

在重复模拟中，搜索树的（状态，动作）对的价值将得到持续更新，并基于$\epsilon-greedy$可以使得搜索树不断进行扩展，使得模拟策略π得到持续改进。

5、时间差分搜索

相比于蒙特卡洛法，时间差分法无须等到一次经验轨迹采样结束之后才进行学习，可以在每一时间步进行学习，使得时间差分算法具有更高的学习效率。与此类似，相比于蒙特卡洛树搜索，时间差分搜索同样无须等到经验轨迹的终止状态，可以聚焦于特定节点的状态，使得节点价值的更新更加高效。

简而言之，时间差分搜索可看成采用Sarsa学习算法对从当前状态开始的子马尔可夫决策过程问题进行求解，主要求解过程如下。

- （1）从当前实际状态s_t开始模拟经验轨迹集，采样过程中，将（状态-动作）对作为节点录入搜索树.
- （2）估计搜索树内每一个节点（状态-动作对）的动作价值Q
- （3）在模拟过程的每一步，采用Sarsa算法更新动作值：

$$\Delta Q(S,A)=\alpha\left(R+\gamma Q\left(S',A'\right)-Q(S,A)\right)$$

- （4）基于步骤（3）获得的动作值函数$Q(s,a)$，使用$\epsilon-greedy$策略或其他策略获得执行动作。