1.马尔科夫过程(MP)

(一).马尔科夫过程(Markov Process)是一**种随机过程**，其**未来的状态仅依赖于当前状态**，而与**过去的状态无关**，这个性质称为马尔科夫性质，MP常用于描述系统的转移和决策过程.

(二).随机过程:它是概率论和统计学中的重要概念，用于描述和分析**随时间变化**的**随机现象.**

2.马尔科夫决策过程(MDP)

(一).通过当前**状态、动作、转移概率**与**奖励**帮助决策者在**不确定的环境**中做出**最优选择**;

(二).马尔科夫决策过程的核心是**马尔科夫性质**，即**当前状态**包含了预测未来状态所需要的所有信息，**过去的状态**和**动作**对**未来没有影响**;

(三).MDP由五元素组成:

1>.状态空间S: 所有可能得状态集合;

2>.动作空间A: **在给定状态下**可采取的所有动作集合;

3>.状态转移概率P: **给定当状态和所选动作a**，转移到具体下一个转态的概率;

4>.奖励函数R: **在给定状态和所选动作**后获得的**即时奖励**;

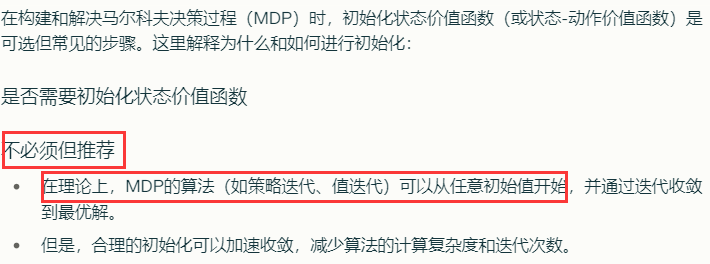
5>.折扣因子gamma: 用于权衡未来奖励的重要性，通常0~1之间.

(四).MDP的优化目标: 找到一个**最优的行动策略pi\***,该策略能够**最大化从当前状态开始的期望累积奖励；**

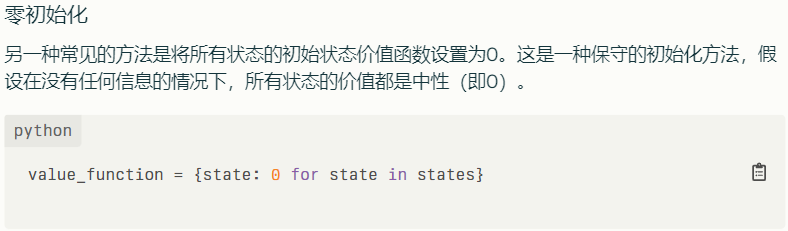
**(五).策略迭代法**优化MDP求解最优策略pi\*包含以下过程:

<5.1>.形式化状态空间S;

<5.2>.初始化状态价值函数;



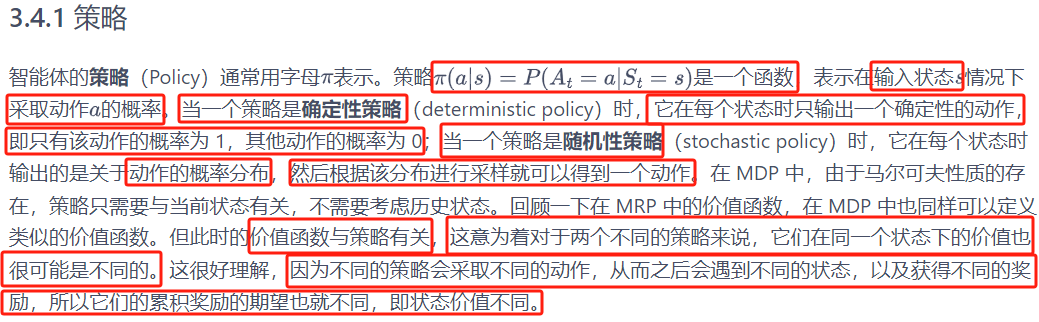






<5.3>.形式化动作空间A;

<5.4>.形式化策略;



<5.5>.定义折扣因子gamma;

<5.6>.根据先验知识，构建奖励函数，该函数输入目前的状态s与执行的动作a，输出及时奖励值;

<5.7>.根据先验知识，已形式化的状态空间S，已形式化的动作空间A，构建状态转移函数，该函数基于输入的当前状态s与执行的动作a输出下一个状态s’与状态转移概率p(s’|s,a)，p(s’|s,a)表示在状态s执行动作a后转移至状态s’的概率;

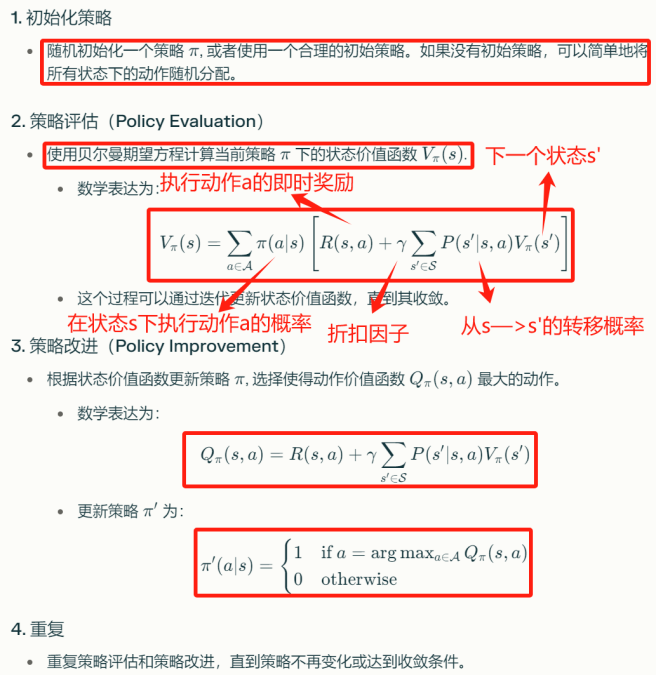
<5.8>.构建策略迭代:

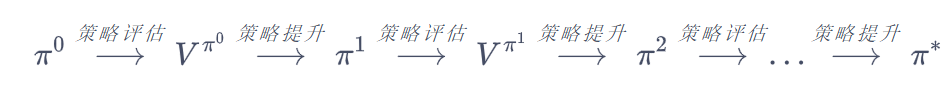
5.8.1>.策略评估:使用贝尔曼期望方程构建一个基于当前策略更新状态价值函数并使其收敛;

5.8.2>.策略改进:为每个一个状态的所有可能执行动作基于已更新的状态价值函数计算动作价值函数值，并基于最大的动作价值为每一个状态更新最优动作,更新状态执行动作的过程就是策略改进过程;

5.8.3>.基于策略评估与策略改进构建策略迭代过程来获得最优策略.

5.8.4>.策略迭代过程如下:





**(六).值迭代法**优化MDP求解最优策略pi\*包含以下过程:

<6.1>.形式化状态空间S;

<6.2>.初始化状态价值函数;

<6.3>.形式化动作空间A;

<6.4>.定义折扣因子gamma;

<6.5>.定义最大迭代;

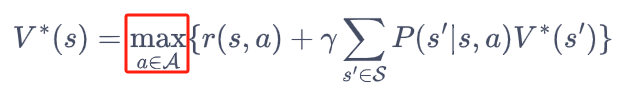
<6.6>.定义状态价值函数V的收敛阈值threshold，收敛阈值是为了检查本轮迭代后的每个状态的价值函数值是否都达到了预期的收敛程度，如果有一个状态的价值函数值没有达到预期，则继续迭代优化直至满足收敛预期或者满足最大迭代次数后才能结束优化;

<6.7>.根据先验知识，构建奖励函数;

<6.8>.根据先验知识，构建状态转移函数;

<6.9>.构建值迭代:

6.9.1>.使用贝尔曼最优方程为更新每一个状态价值函数值，其值为此状态可执行动作的最大动作价值;

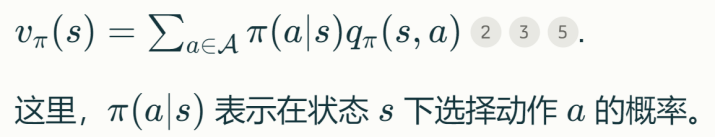
贝尔曼最优方程——>

6.9.2>.计算**每一个更新后的状态价值函数值**与对应更新前的状态价值函数值的**差值**,并提取其中的最大值max\_delta，使用max\_delta与threshold进行比较判断此时更新后的状态价值函数是否收敛，若收敛则退出值迭代否则继续值迭代直至满足收敛条件或满足最大迭代次数.

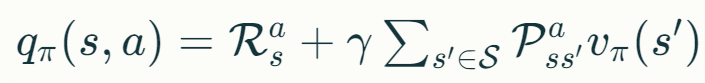
<6.10>.最优策略提取:**基于最优状态价值函数**，计算每个状态可能执行所有动作的动作价值，选取动作价值最大的具体动作为此状态的动作，并以此逻辑逐步为所有状态找到最优动作，最终，由状态与其对应的最优动作构成最优策略

(七).定义值函数

7.1>.**状态价值函数v\_pi(s):**表示在状态s下，按照策略pi采取动作后，预期能获得的总回报。

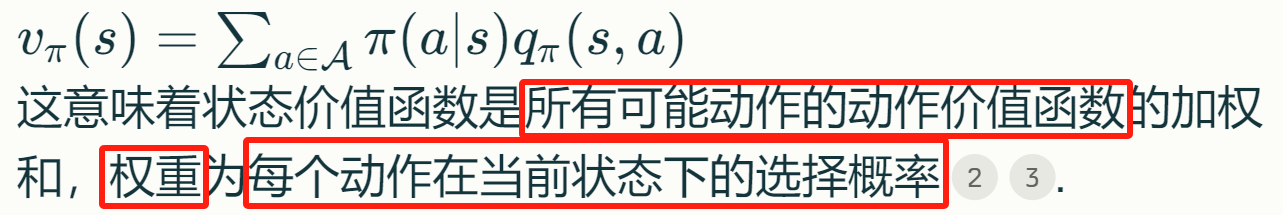


7.2>.**动作价值函数q\_pi(s,a):**表示在状态s下采取动作a后，按照策略pi采取后续动作，预期能获得的总回报。

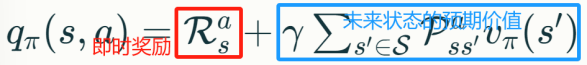


7.3>.两个值函数的进一步解释

7.3.1>.状态价值函数是**在状态s下**，**所有可能动作的动作价值函数的加权和**，**权重**为在**当前状态下**策略选择其动作的**概率**

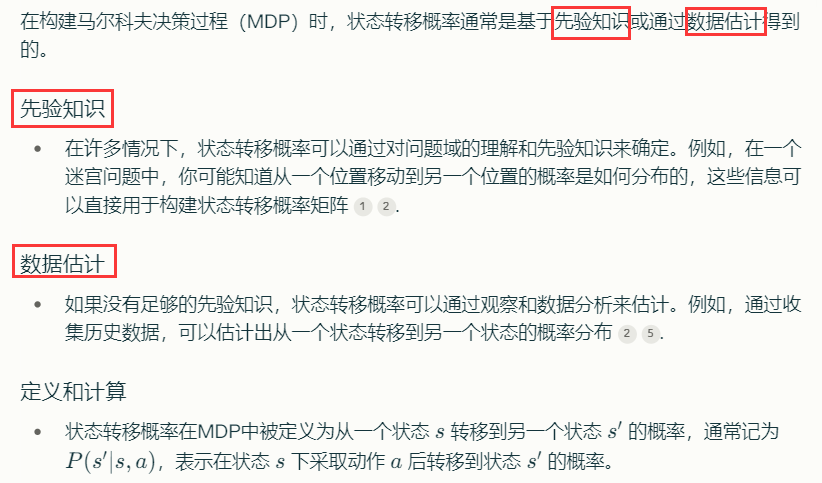


7.3.2>.动作价值函数包括了**即时奖励**和**未来状态的预期价值**，未来预期价值是根据当前策略计算的

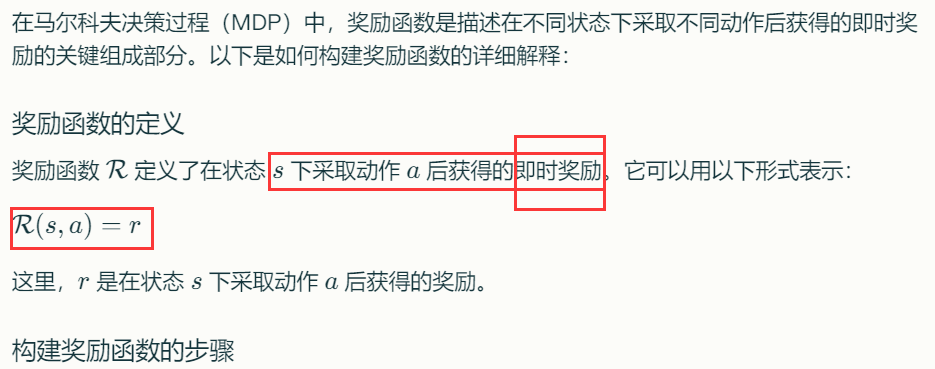


(八).相关知识点补充

<8.1>.状态转移概率函数

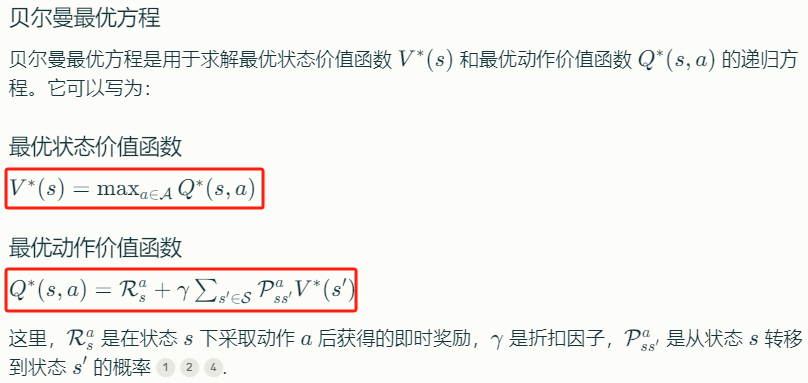


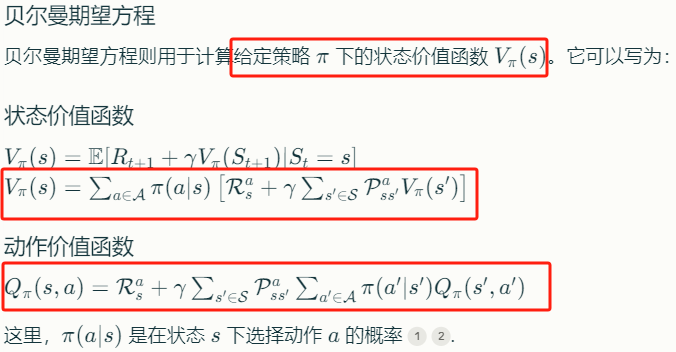
<8.2>.奖励函数构建



<8.3>.折扣因子gamma的物理含义:远期利益具有一定的不确定性，有时我们希望最优策略关注近期的奖励，所以引入折扣因子对远期的奖励大一些折扣，接近1的gamma更关注长期的累计奖励，接近0的gamma更关注短期奖励。

8.4>.贝尔曼最优方程与贝尔曼期望方程





8.5>.策略迭代与值迭代的收敛性证明



3.蒙特卡洛方法

(一).定义:一种基于**随机抽样**的**统计模拟**技术，其基本思想是通过**生成大量的随机样本**，利用这些样本的**统计特性**来估计目标值，特别适用于难以获得**解析解**的问题。

(二).蒙特卡洛方法的基本步骤:

<2.1>.确定随机变量:**识别与问题相关的随机变量**，并分析它们的性质和相关关系;

<2.2>.为随机变量构建概率分布模型:**为识别的随机变量构建适合的概率分布**，构建的概率分布越接近识别随机变量的真实概率分布，蒙特卡洛方法解决问题精度越高;

<2.3>.基于概率分布模型生成随机样本:使用概率分布模型生成随机样本;

<2.4>.构建解决问题的模型并进行模拟:基于随机样本构建解决问题数学模型，进行问题模拟输出模拟结果;

<2.5>.对模拟结果进行分析，例如计算期望，方差，并绘制相应的概率分布图等

4.GNAS蒙特卡洛树搜索(MCTS)案例

(一)GNAS搜索问题定义:

给定一个图神经网络的搜索空间GNN Search Space，搜索空间包含7个结构组件，每个结构组件有其候选项集合如下:

perturbation\_intensity\_candidate = [0.5, 0.1, 0.05, 0.01];

perturbation\_probability\_candidate = [0.1, 0.3, 0.7];

distance\_aggregator\_candidate = ["5", "10", "30"];

feature\_aggregator\_candidate = ["GCNConv", "SGConv", "None"];

updator\_dimension\_candidate = [128, 256, 512, 1024];

updator\_activation\_candidate = ["LeakyRelu", "Relu", "Relu6"];

fusion\_candidate = ["cat", "sum", "mean", "max", "weighted\_cat", "weighted\_sum"];

每个结构组件采样一个候选项可以组成一个GNN结构，例如:

GNN\_Architecture = [0.5,0.1,"5","SGConv",128,"Relu","cat"]

每个GNN\_Architecture送入评估函数后将返回一个验证评估反馈，例如:

Validation Score = estimator(GNN\_Architecture).

搜索任务:在GNN Search Space识别出最优的Validation Score对应的GNN\_Architecture

(二).蒙特卡洛树搜索:是一种用于决策过程的**启发式搜索算法**，特别适用于**大型**、**高复杂度**的搜索空间，如:棋类游戏、规划问题和强化学习等。MCTS的核心思想是通过**随机模型(蒙特卡洛方法)**来评估搜索树中的节点价值，从而指导搜索过程.

(三).GNAS场景下的蒙特卡洛树搜索流程

<1>创建节点类:

1>.**属性:**

(1)当前节点的父节点对象;

(2)当前节点的子节点列表;

(3)当前节点对应的组件索引;

(4)当前节点**被访问次数**;

(5)当前节点**累计评估值**;

(6)当前节点在本层树结构可扩展的子节点候选列表（**此候选项**对应的**下一层组件**还未探索可选择的候选项）

(7)当前节点是否扩展完毕标志位。

2>.**方法:**

(1)判断当前节点是否为终端节点,**返回**是否是终端节点标志位;

(2)判断当前节点是否已完全扩展完毕(对于当前候选项是否还有**下一层组件候选项**需要扩展),**返回**是否扩展完毕节点标志位;

(3)从当前节点可扩展子节点列表中扩展一个子节点，**如果本节点无可扩展子节点则将当前节点扩展完毕标志位置True,返回**扩展子节点对象;

(4)使用最优子节点选择策略(一般是UCB1策略)选择当前节点的最优子节点进行探索，如果有多个最优子节点则随机选择一个，**返回**当前节点的最优子节点对象;

<2>创建树策略:——>包含选子节点择与扩展操作，**优先扩展(expansion)未探索的子节点**，当所有子节点探索完毕后再选择(selection)最优子节点进行探索

(1)**输入**节点对象;

(2)判断节点是否是终止节点，若不是则基于输入节点对象递归访;

(3)判断节点是否有**可扩展的子节点**，如果没有则扩展子节点并**返回**子节点对象**退出**上层循环;

(4)如果当前节点全部子节点已扩展完毕则**使用最优节点选择策略探索子节点**，**返回**最优子节点对象**继续探索过程**;

(5)**返回**扩展子节点对象或终端节点对象.

<3>创建模拟(simulation)函数:

(1)**输入**节点对象;

(2)通过改节点回溯其父节点构建已探索的GNN结构组件;

(3)使用模拟策略获得剩余GNN结构组件候选项构建完整的GNN结构(一般可使用随机模拟策略（蒙特卡洛方法）);

(4)将完整的GNN结构送入GNN结构评估函数获得评估反馈构成此次树探索的奖励,**返回**奖励分数;

<4>回溯(backpropagation)本次树探索结果

(1)**输入**节点，奖励分数，折扣因子;

(2)判断节点对象是不是根节点的父节点None,不是给此节点对象被访问次数属性增加1，累计评估值属性增加奖励分数\*折扣因子，回溯其父节点;

(3)若回溯至None节点则退出回溯;

<5>构建蒙特卡洛搜索逻辑输出最优GNN结构

(1)输入根节点对象，迭代次数，折扣因子;

(2)构建循环进行树搜索:

<1>.执行树搜索策略**输入**根节点(包含子节点的扩展与选择)**返回**探索的叶子节点; <2>.对探索的叶子节点进行模拟,**输入**叶子节点**返回**奖励分数;

<3>.对探索的叶子节点进行回溯,**输入**叶子节点、奖励分数，折扣因子;

(3)对蒙特卡洛搜索过程生成的整个树结构从根节点开始进行**最优子节点选择**构建最优GNN结构，可以基于每个子节点的被访问次数、平均评估值（积累估值/被访问次数）等其他有意义的指标判断最优子节点。

(4)最优子节点构建最优GNN架构，计算最优子节点序列最优一个终端节点的平均奖励分数（积累估值/被访问次数）作为此最优GNN架构的最优验证评估性能指标

(5)**返回**最优GNN架构，最优GNN架构验证性能

(四).输出最优GNN结构过程中最优子节点选择知识点补充:

