# 应用题

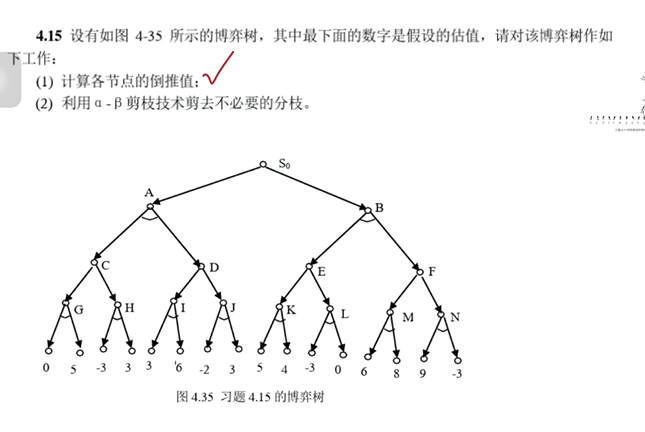
## 1.a b 剪枝

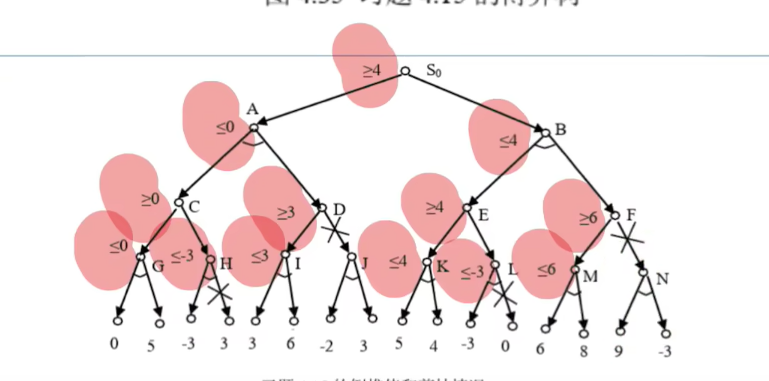
推荐视频网址：[人工智能α－β剪枝算法\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV16h411a7Kk/?spm_id_from=333.788.top_right_bar_window_history.content.click&vd_source=7adf5059039c9fd15984302e08b131c6)

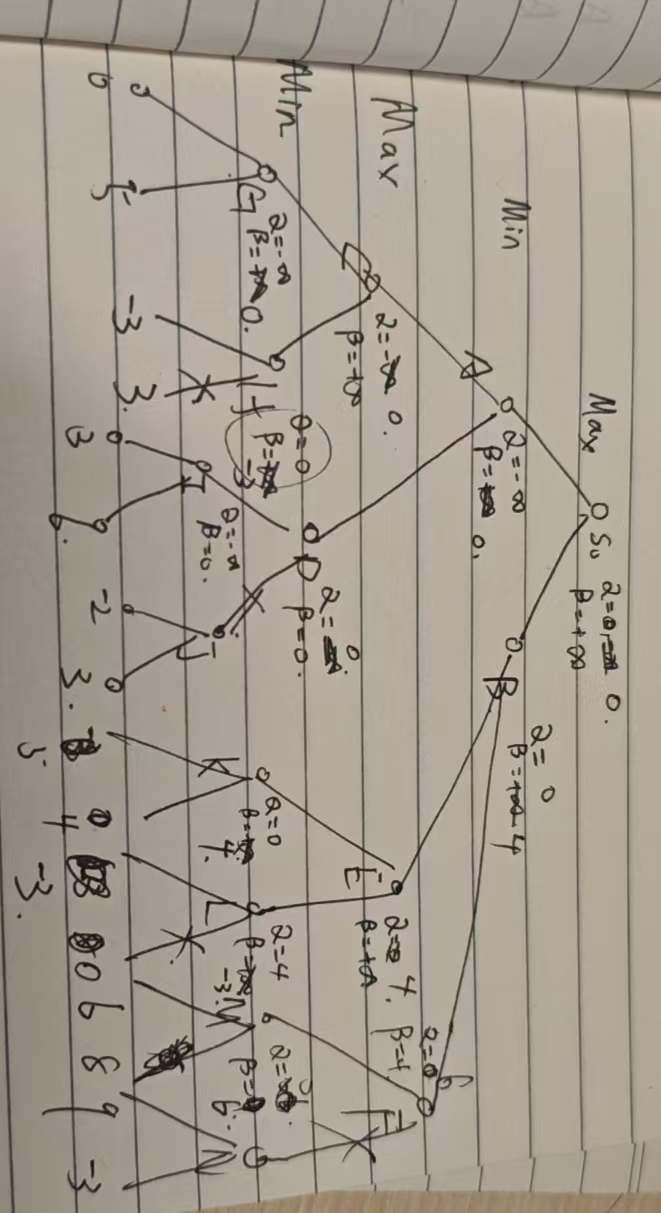
设置a b值在max层中更新a=max(a,left.value,right.value)

在min层中更新b=min(b,left.value,right.value)

利用深度优先遍历把所有节点遍历后得到减枝结果



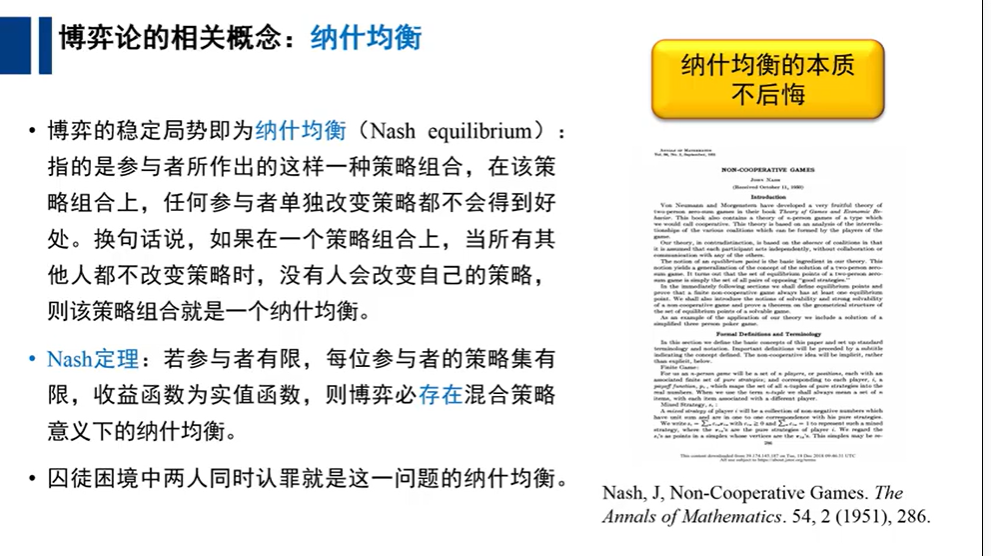


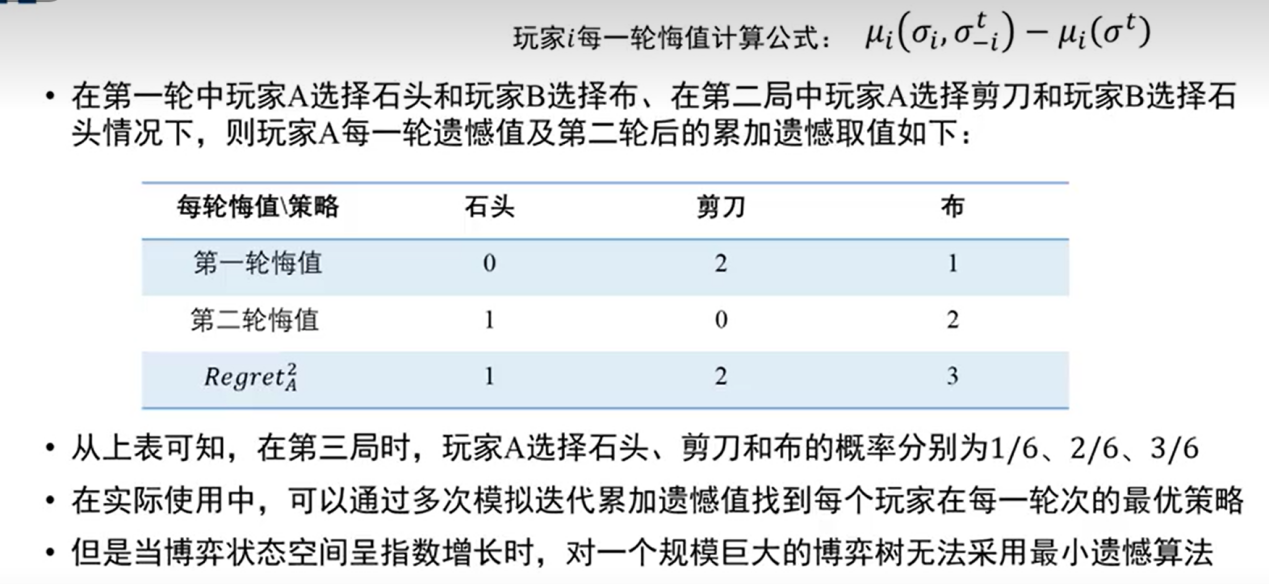


## 2.纳什均衡 必考

纳什均衡的计算(2 个参与者博弈):纯策略纳什均衡的求解(补充知识点:划线法)，混合策略纳什均衡计算，混合策略达到纳什均衡时参与者收益的期望。

[11.1 博弈论的相关概念\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1su411d7uK/?spm_id_from=333.788.top_right_bar_window_history.content.click&vd_source=7adf5059039c9fd15984302e08b131c6)





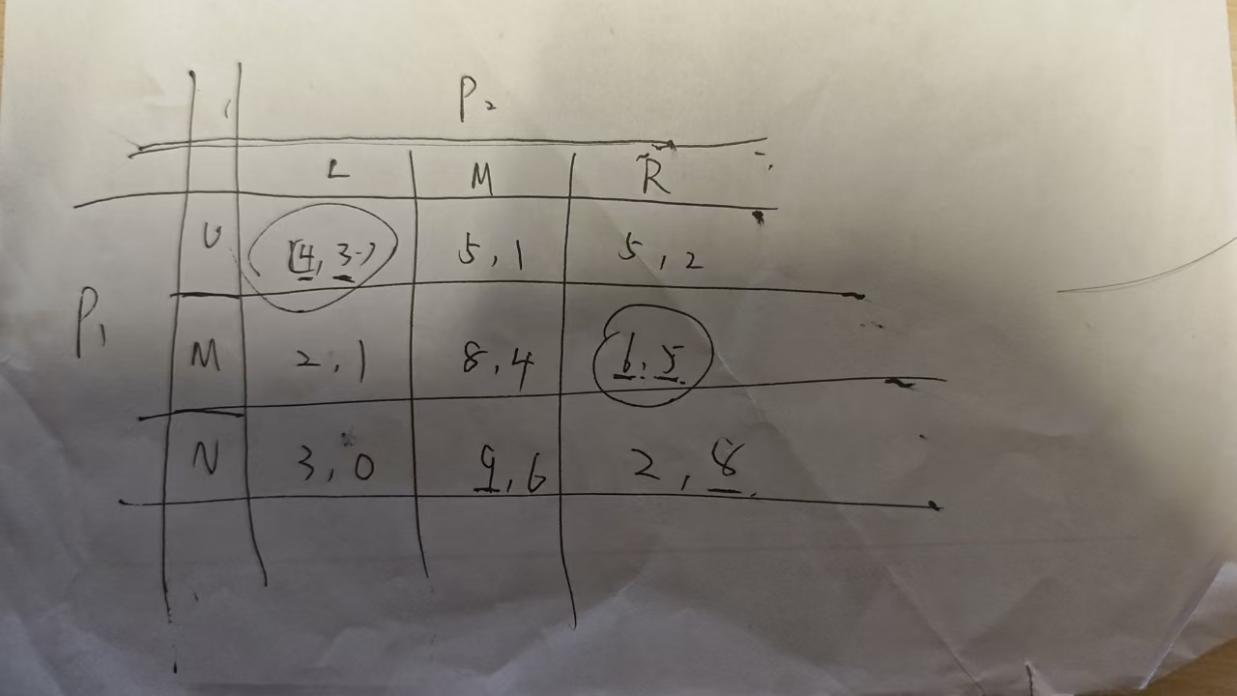
寻找严格劣策略横竖策略:划线排除

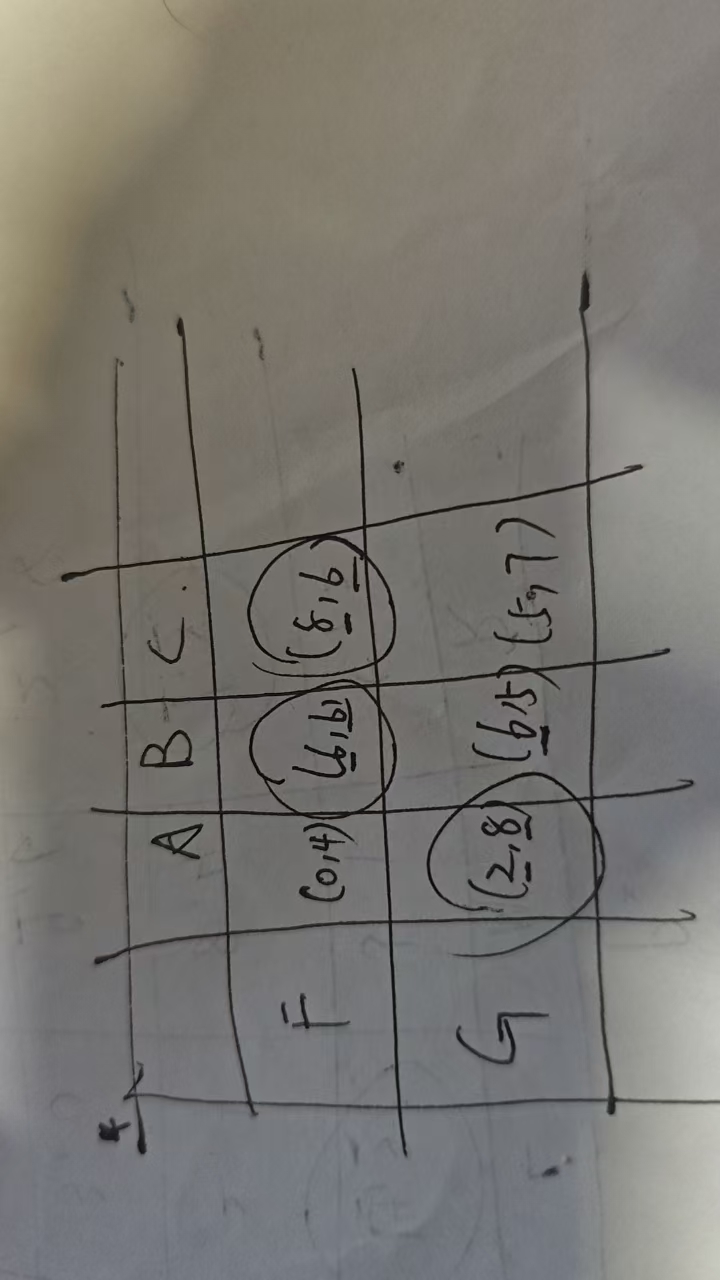
### 1.纯策略纳什均衡的求解（划线法）

纯策略纳什均衡是指参与者选择一个确定的策略（没有随机化），使得在给定对方策略的情况下，没有人能通过改变自己的策略来获得更高的收益。

1. 查看是否有支配策略： 如果一个策略在任何情况下都能给出比其他策略更高的收益，那么这个策略就是支配策略。如果某个参与者的某个策略支配了另一个策略，那么另一个策略就可以被划掉，缩小可选策略集。
2. 使用划线法（Elimination of Dominated Strategies）： 划线法的关键在于通过消除那些被支配的策略，逐步简化博弈，最终找到纯策略纳什均衡。具体做法是：

如果某个策略在所有情况下的收益都小于其他策略（即被支配策略），则将其从博弈矩阵中划去。重复这个过程，直到没有可以被支配的策略。





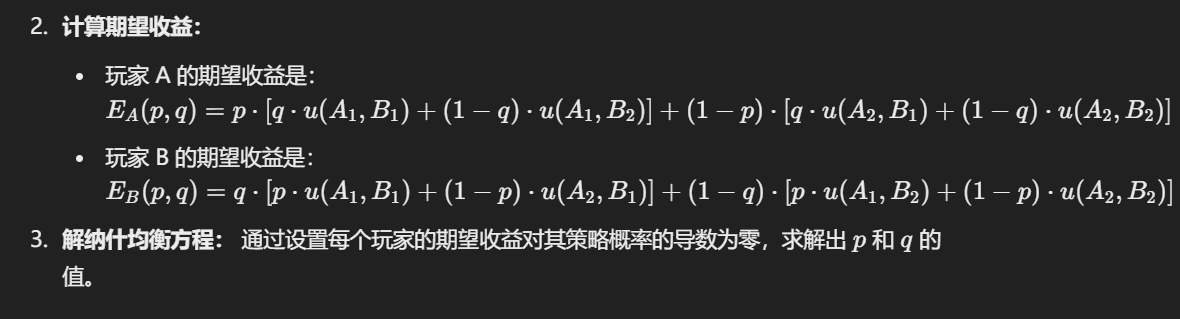
### 2. 混合策略纳什均衡的求解

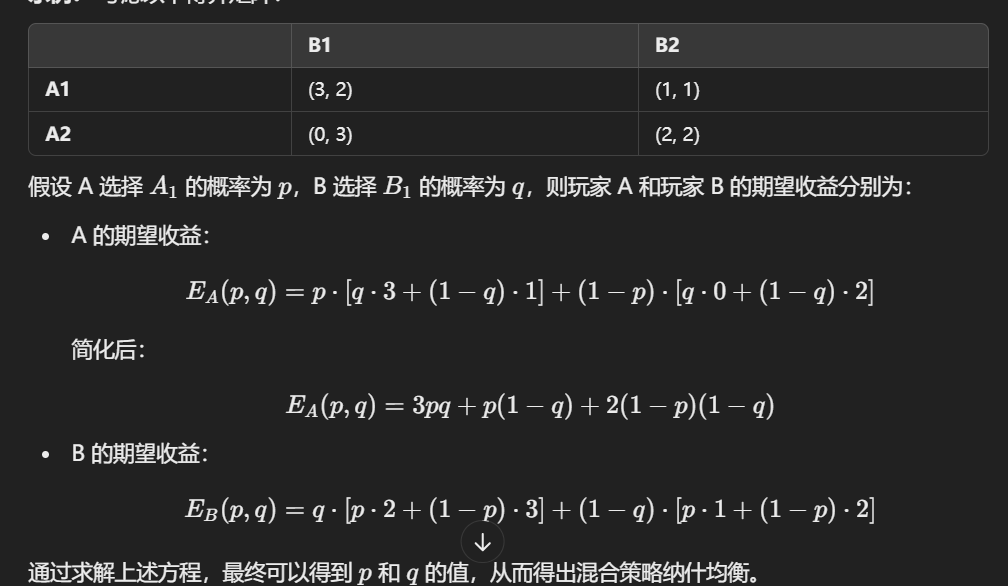
混合策略纳什均衡是指参与者不仅可以选择纯策略，还可以选择按照某种概率分布随机选择策略。在这种情况下，参与者的目标是最大化自己的期望收益，假设每个玩家都知道对方的混合策略，并且在此基础上调整自己的混合策略。

1. 设定混合策略：假设玩家 A 和玩家 B 分别采用混合策略，

其中 p 是 A 选择策略 A1概率（1−p 为选择策略 A2的概率），

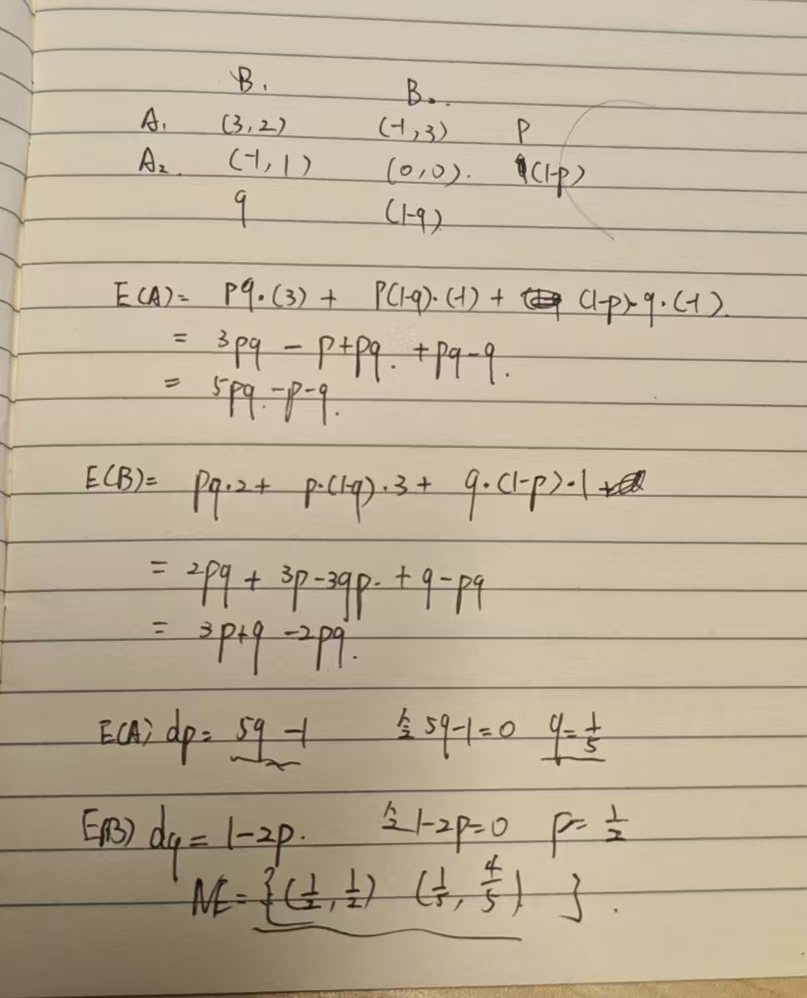
而 q 是 B 选择策略 B1 的概率（1−q1 为选择策略 B2的概率）。



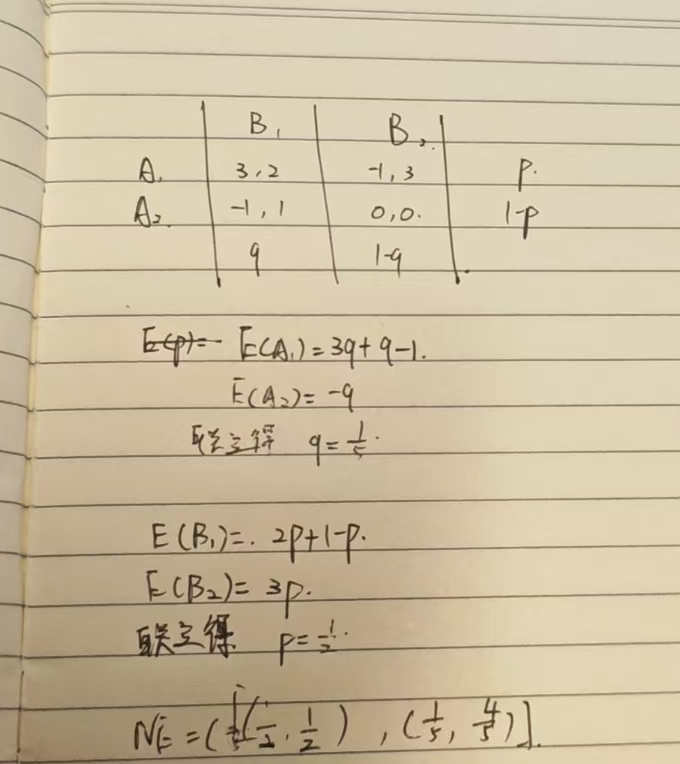


|  |  |  |
| --- | --- | --- |
|  | B1 | B2 |
| A1 | 3，2 | -1，3 |
| A2 | -1，1 | 0，0 |

法一：应讲的：求解效用函数再求解极值点联立得解

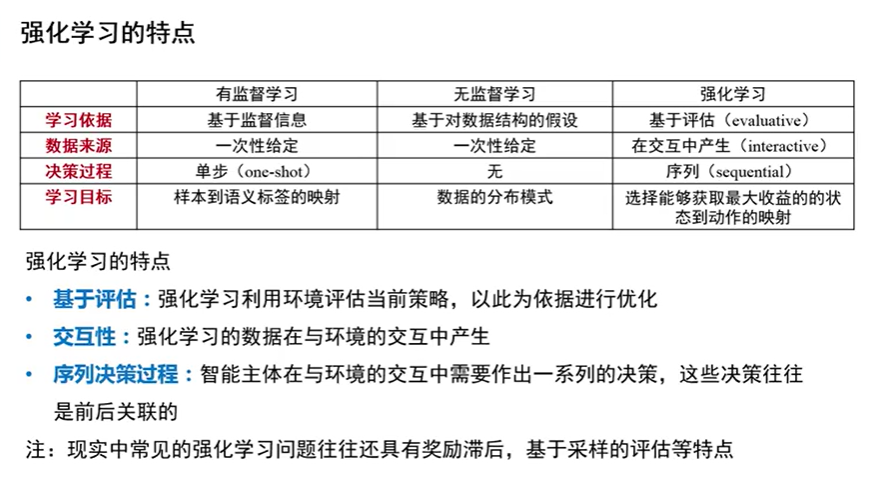


法2：网上的方法（可用于检查）



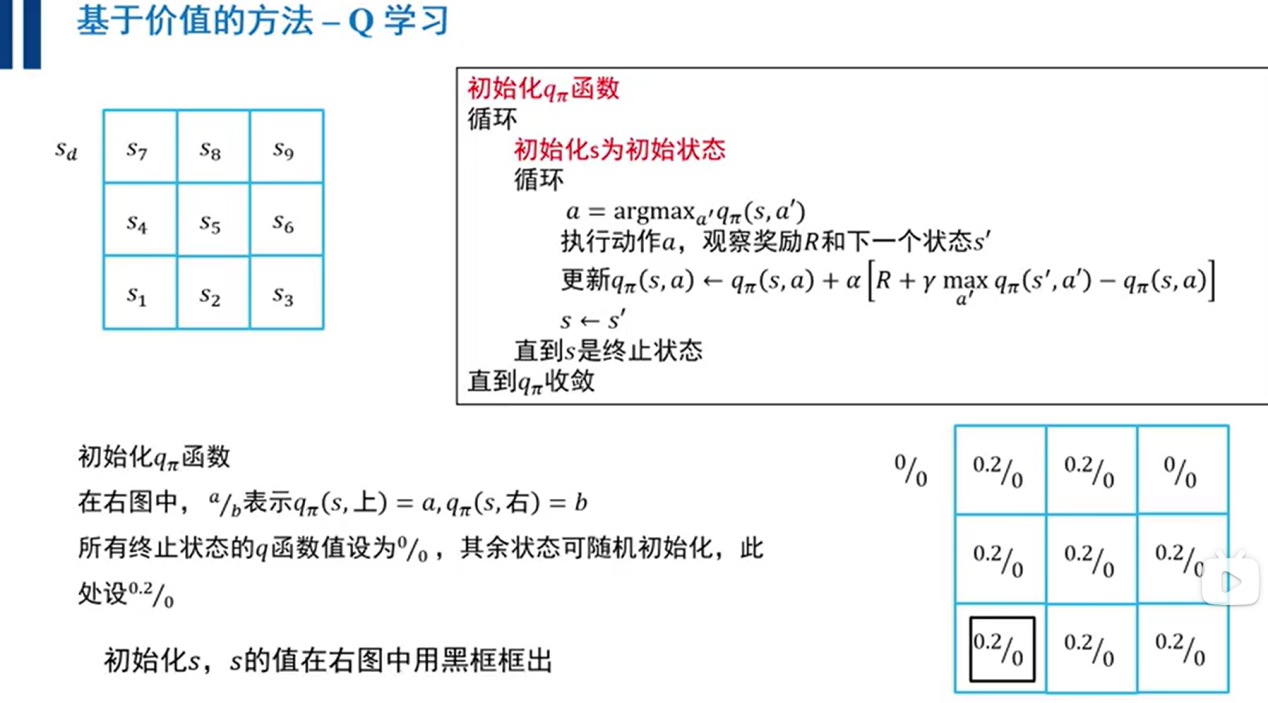
## 3.q-learnig

[10.1 强化学习的定义\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1R5411d7MF?spm_id_from=333.788.videopod.sections&vd_source=7adf5059039c9fd15984302e08b131c6)

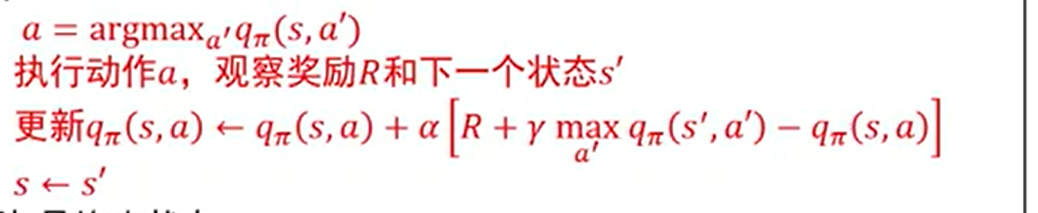


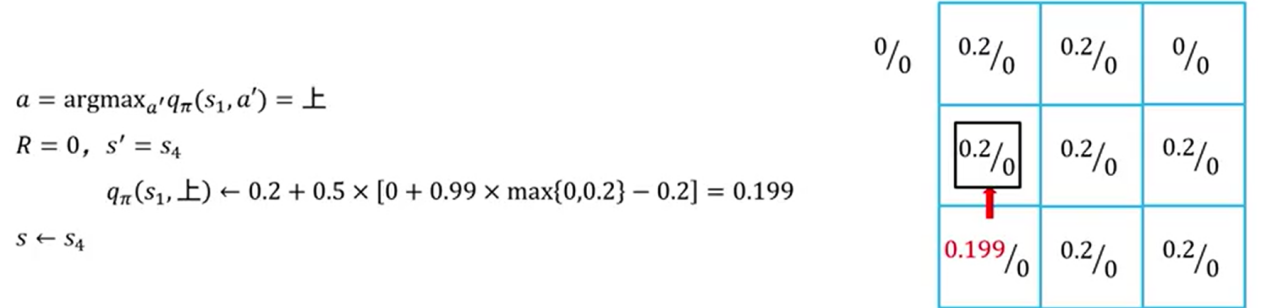


[10.3 强化学习求解\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1Ab4y177W7?spm_id_from=333.788.videopod.sections&vd_source=7adf5059039c9fd15984302e08b131c6)



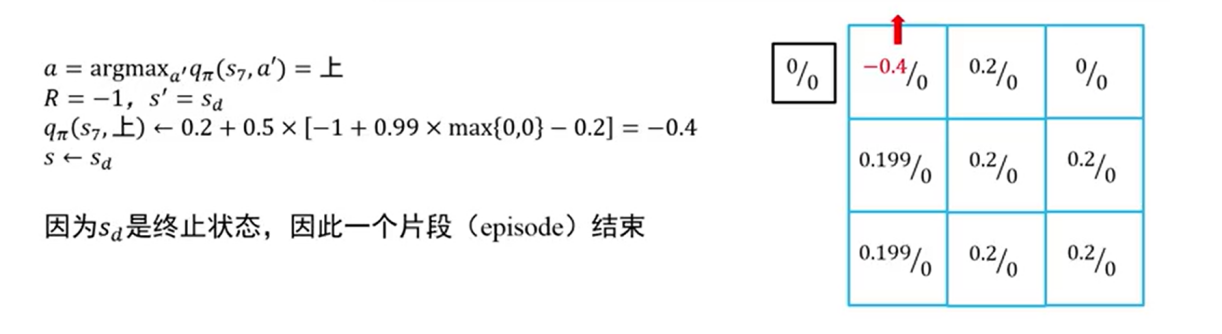
### 1.核心步骤





更新策略函数为 ：

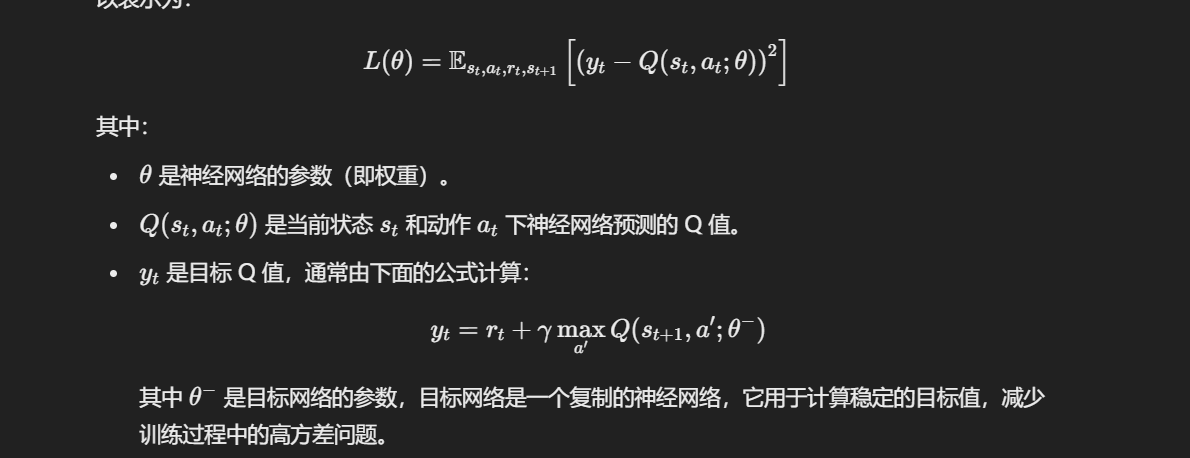
策略所选值+0.5\*（奖励值+衰减系数\*max（当前策略最大值）-上一步最大值）



### 2.Deep Q-learning 的目标函数

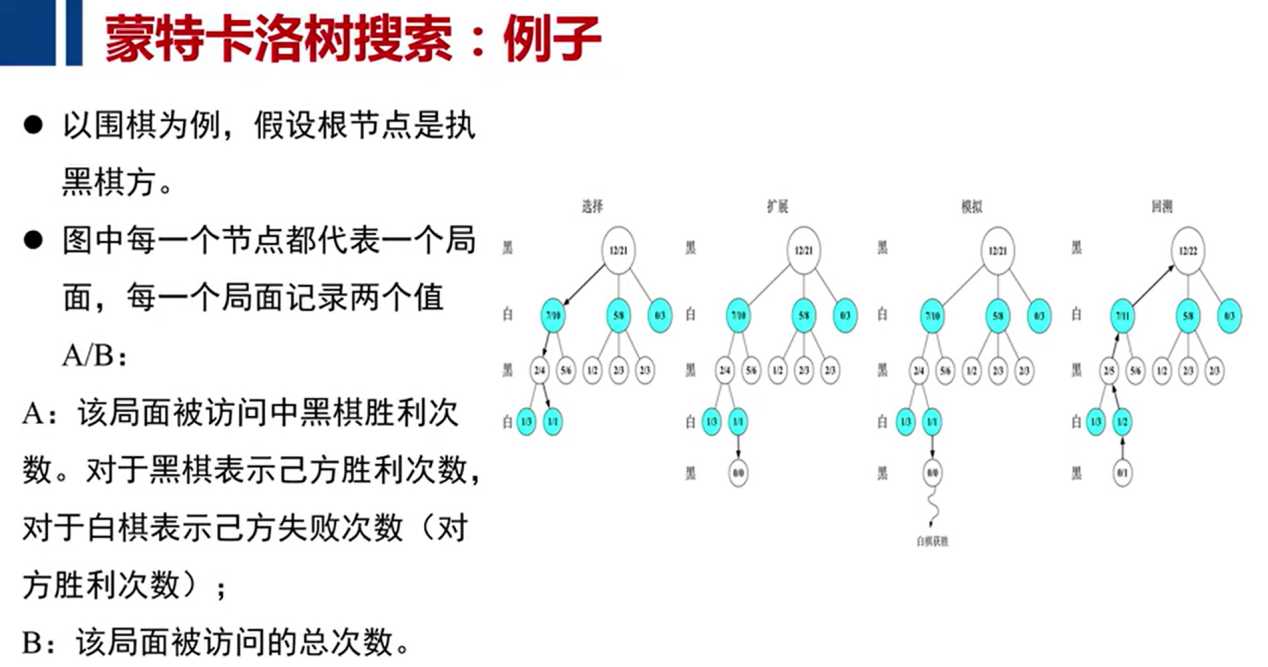
Deep Q-learning（深度 Q 学习）是 Q-learning 的扩展，它使用神经网络来逼近 Q 函数，而不是使用传统的 Q 表。这是因为对于高维状态空间（例如图像数据或大规模问题），Q 表很难存储和更新。

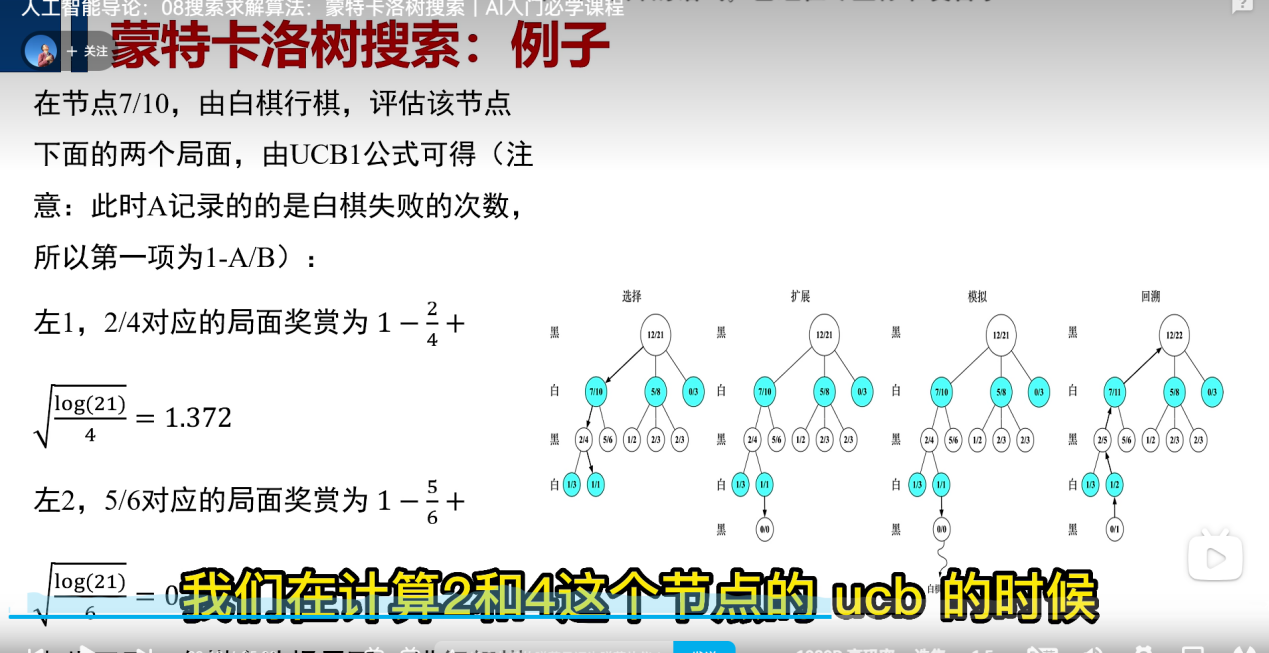
Deep Q-learning 的目标函数是通过最小化 Bellman 误差来训练神经网络。具体来说，目标函数可以表示为：



## 蒙特卡洛树搜索（考纲外）

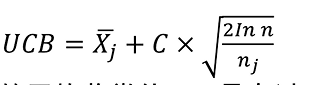
推荐视频网址 [2.3 蒙特卡洛树搜索\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1tT4y1C7jM/?spm_id_from=333.337.search-card.all.click&vd_source=7adf5059039c9fd15984302e08b131c6)





步骤：选择->扩展->模拟->反向传播

UCB计算核心公式



其中Xj为当前结点过去平均reward

n为父结点经过次数

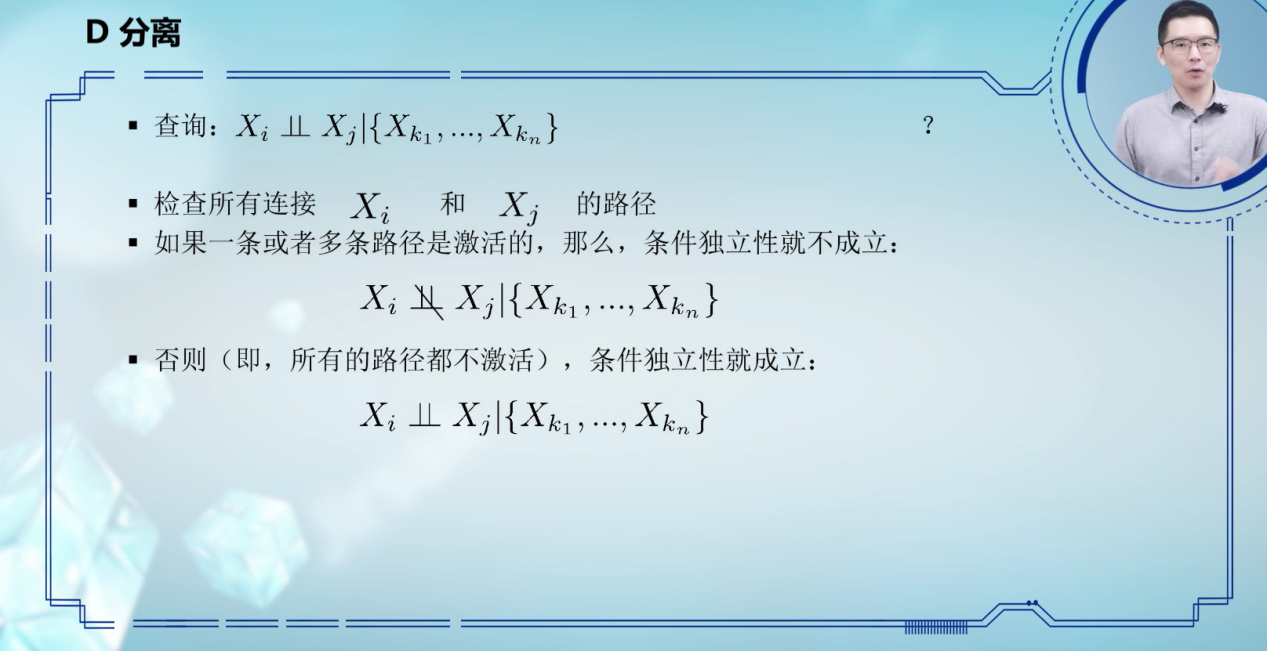
Nj为当前结点经过次数

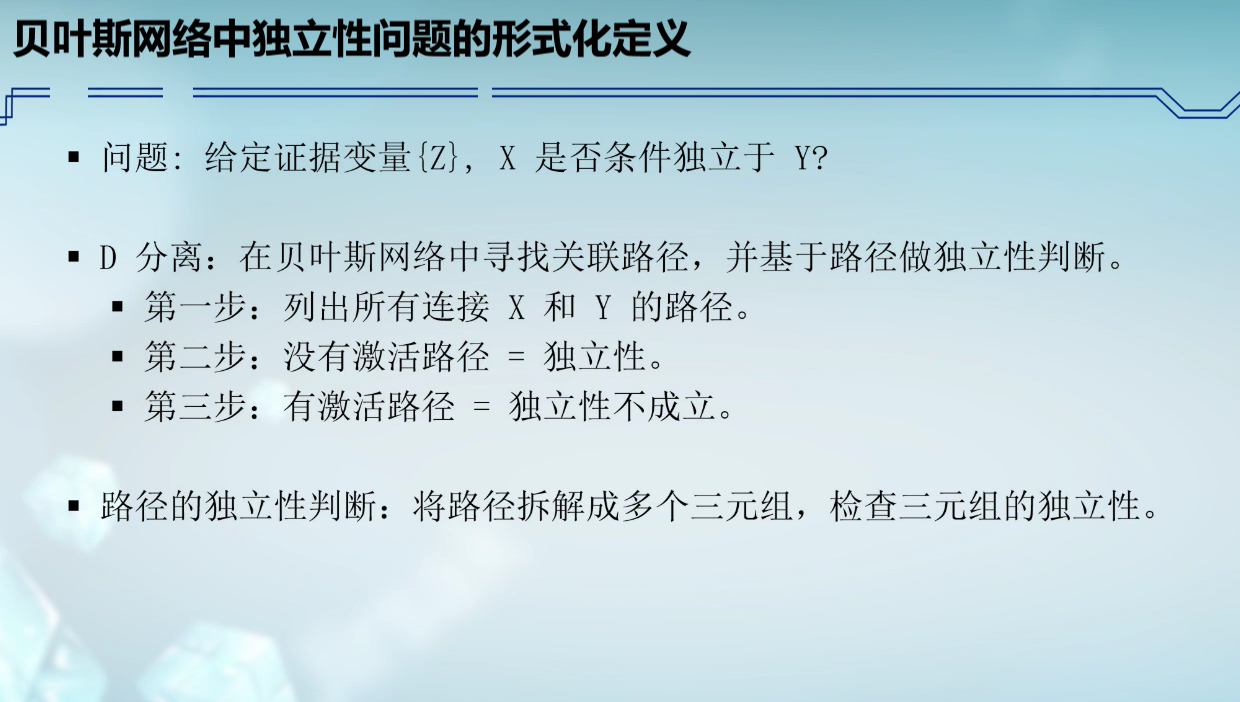
注意，在min层所计算的Xj应当取反

## d分离（考纲外）

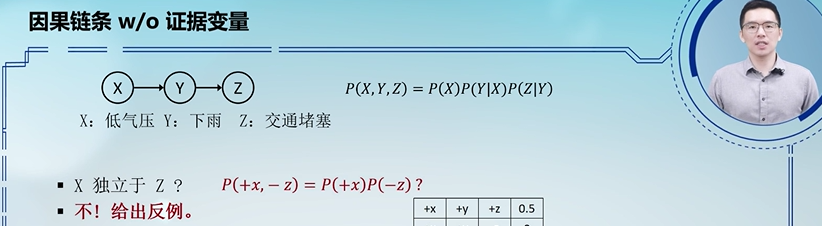
[[8.7]D分离方法\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1xF411K7aa?spm_id_from=333.788.videopod.episodes&vd_source=7adf5059039c9fd15984302e08b131c6&p=7)

[D-Separation D划分详解 大量例子 贝叶斯网络 概率图模型 因果推断\_哔哩哔哩\_bilibili](https://www.bilibili.com/video/BV1LL411J7xD?spm_id_from=333.788.recommend_more_video.5&vd_source=7adf5059039c9fd15984302e08b131c6)



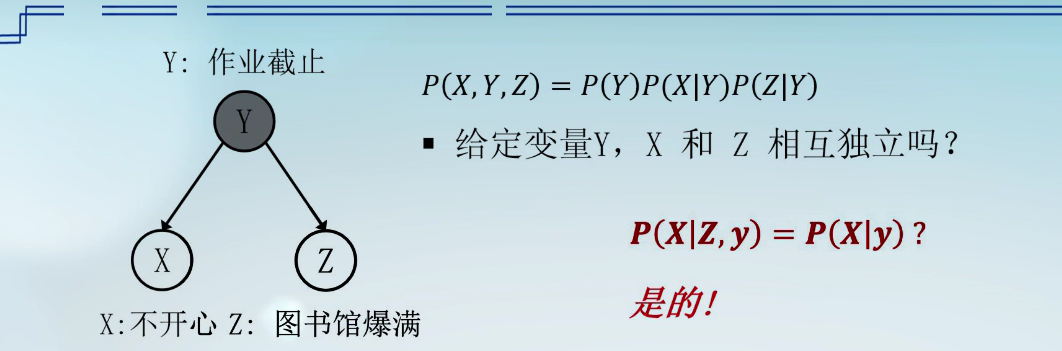


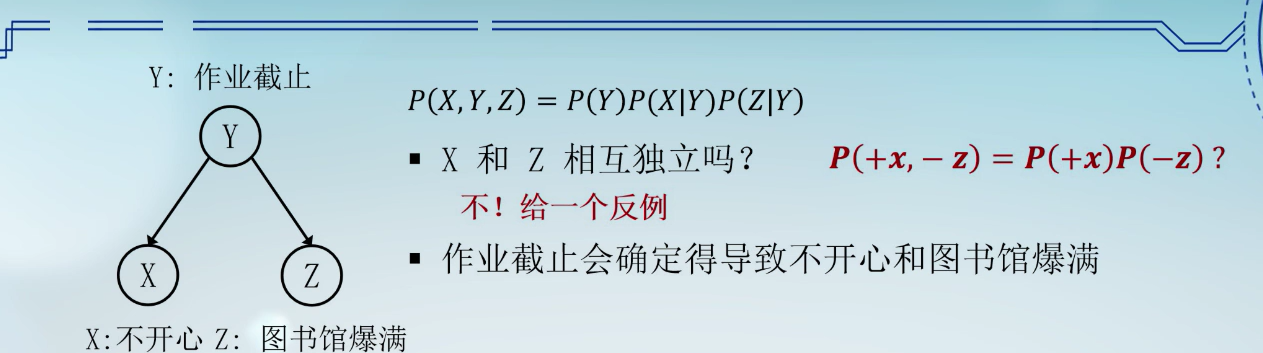
因果链



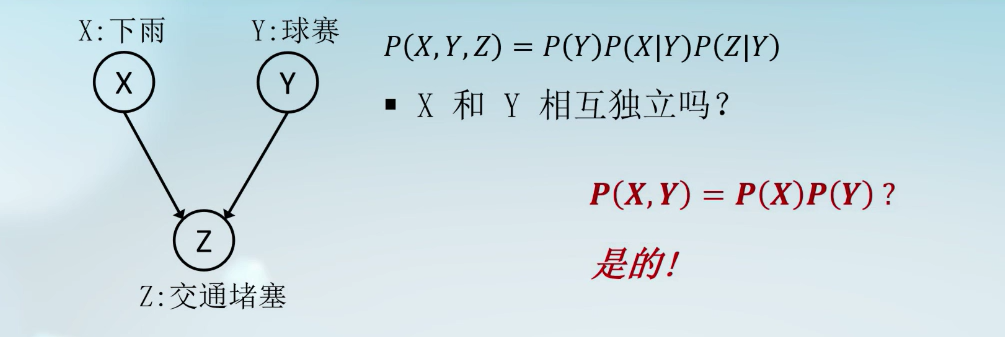


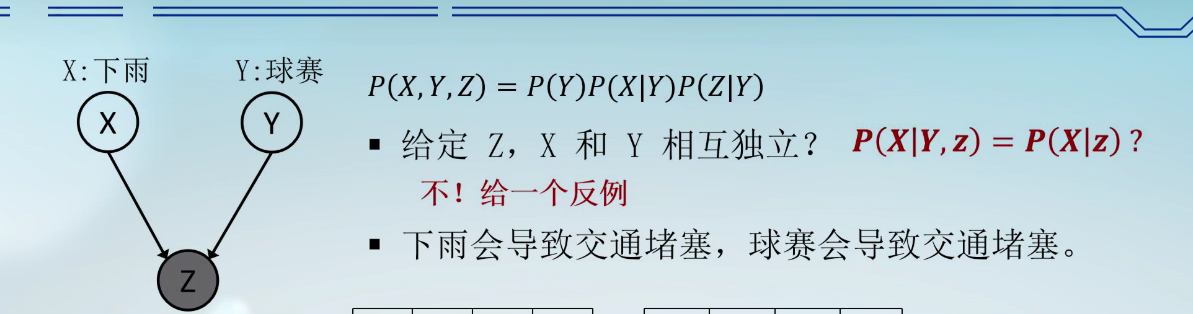
共同原因

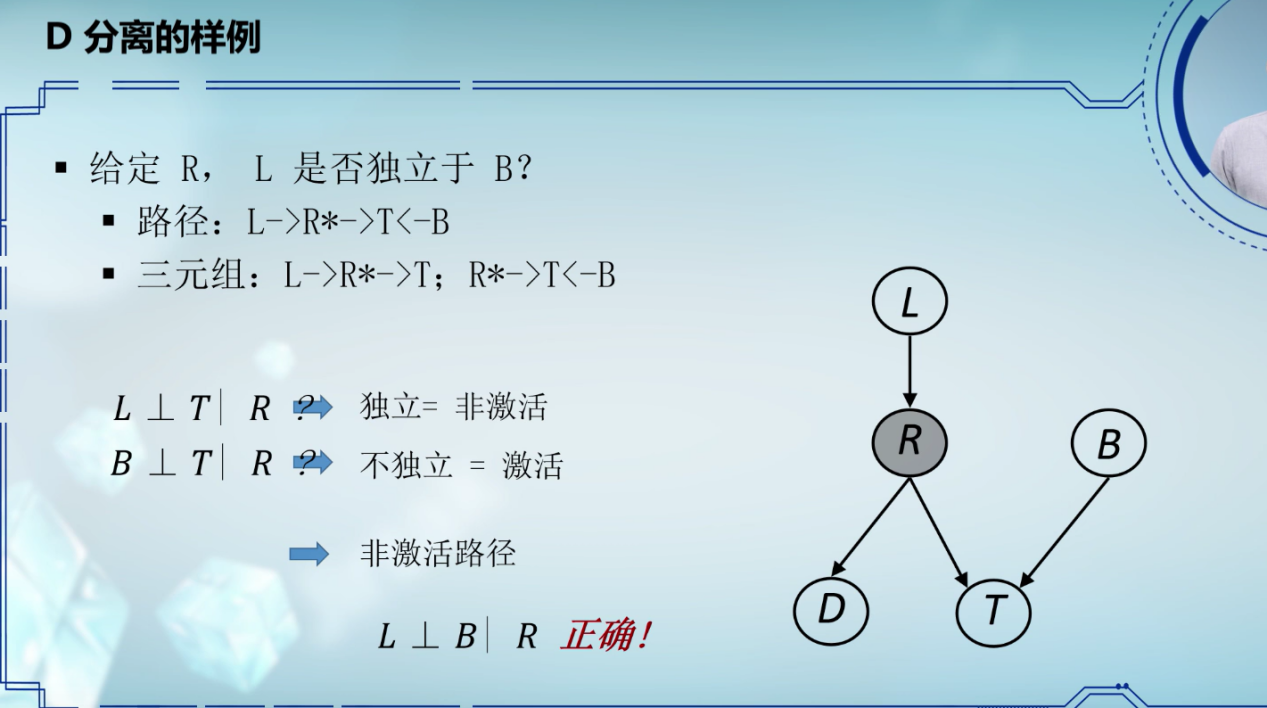


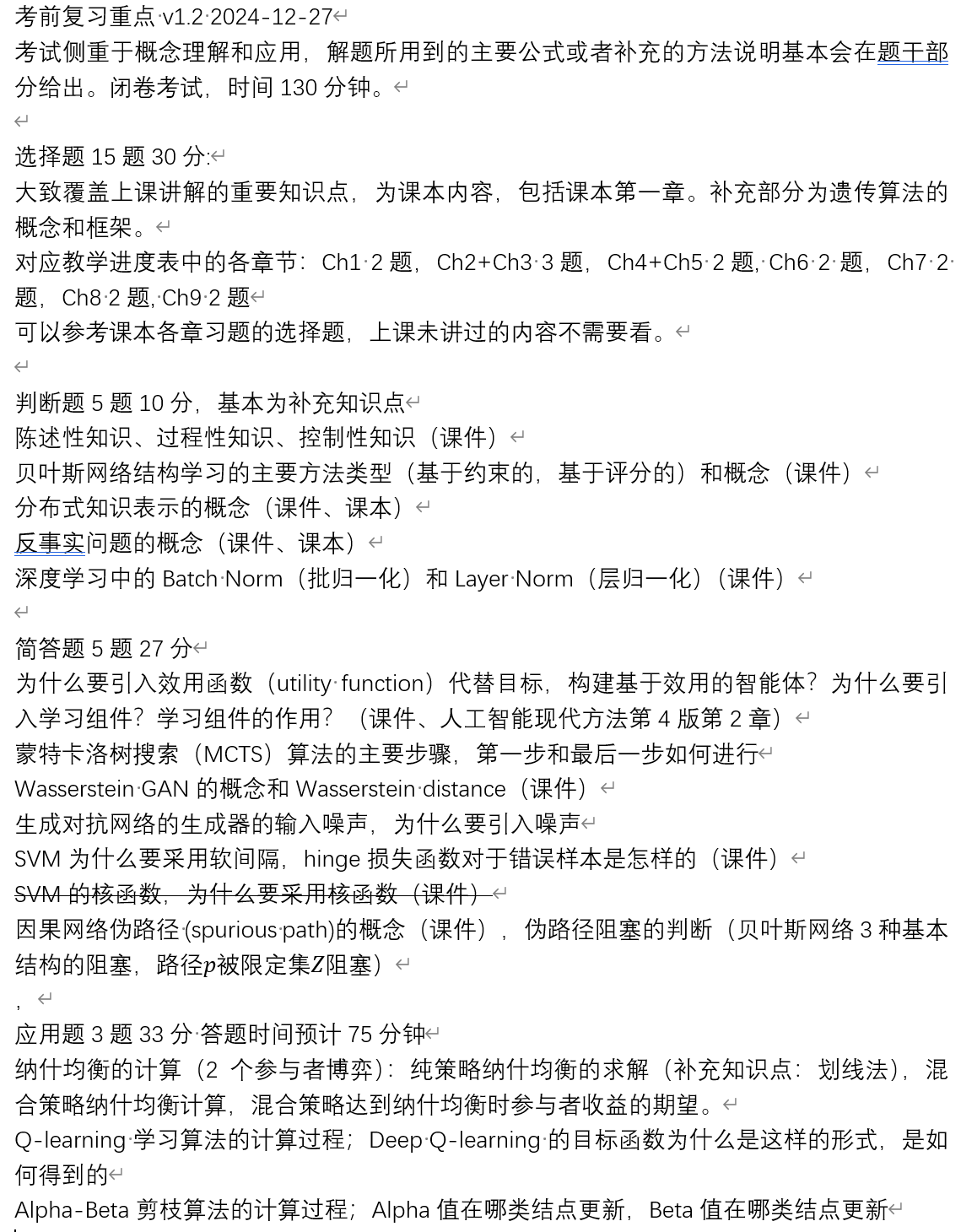


共同结果









# 简答题

## 为什么要引入效用函数(utility·function)代替目标构建基于效用的智能体?为什么要引入学习组件?学习组件的作用?

仅靠目标在很多环境中不足以生成高品质的行为。更通用的性能度量应该允许

不同的世界状态之间进行效用比较。性能度量给环境状态的任何给定序列赋了

一个值。Agent 的效用函数是性能度量的内在化。

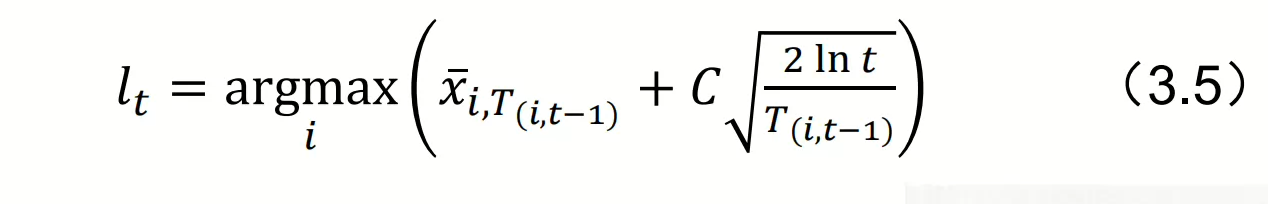
拥有显式效用函数的Agent 因此可以做出理性决策，它可以通过通用算法做到，且此算法并不依赖于要最大化的特定效用函数。

性能元件负责选择外部行动,学习元件负责改进提高

学习元件利用来自评判元件的反馈评价Agent做得如何，并确定应该如何修改执行元件以便将来做得更好。

## 蒙特卡洛树搜索(MCTS)算法的主要步骤，第一步和最后一步如何进行“

使用UCB算法



对每个节点计算UCB值,从根节点开始进行递归选择节点

进行选择->扩展->扩展->反向传播

1. 选择:向下递归选择子节点，直至到达叶子结点或者到达具有还未被扩展过的子节点.记录每个结点被选择次数和每个结点得到的奖励均值。

(4)用模拟所得结果（终止节点的代价或游戏终局

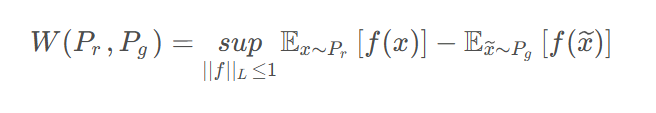
分数）回溯更新模拟路径中M以上（含M）节点的奖励均值和被访问次数。

## 3.Wasserstein·GAN的概念和Wasserstein:distance

csdn:https://blog.csdn.net/Allen\_Smath/article/details/143949616

Wasserstein Generative Adversarial Networks (WGANs) 是一种改进的生成对抗网络（GAN）模型，旨在解决原始GAN模型中的训练不稳定问题。WGAN的核心思想是使用Wasserstein距离来衡量生成数据分布与真实数据分布之间的差异，而不是传统GAN中的Jensen-Shannon散度。

Wasserstein距离的定义如下：



## 4.生成式对抗网络的噪声，为什么要引入噪声

1.控制生成样本的多样性和数量。不同的噪声向量会导致生成器生成不同的样本，从而增加了生成样本的多样性。

2.模式坍塌（Mode Collapse）是指生成器生成的样本多样性降低，开始生成非常相似或相同的样本。

3.生成器和判别器之间存在对抗性。生成器试图生成逼真的数据以迷惑判别器，而判别器则试图区分真实数据和生成数据。噪声的引入使得生成器和判别器在训练过程中进行有效的对抗，从而提升模型的性

## SVM为什么要采用软间隔，hinge损失函数对错误样本是怎样的。

SVM（支持向量机）最初是为线性可分的数据设计的，这种SVM被称为硬间隔SVM。硬间隔SVM要求所有的数据点都严格位于决策边界的两侧，即所有的数据点都必须被正确分类，不允许任何误差。然而，在现实世界中，很多数据集并不是完全线性可分的，可能存在噪声或者重叠，这时硬间隔SVM就无法处理这些数据集。

为了解决这个问题，软间隔SVM被引入。软间隔SVM通过引入松弛变量（slack variables）允许一些数据点违反间隔规则，即它们可以位于间隔边界的内部或者甚至在边界的另一侧。这样做的目的是为了在最大化间隔的同时，最小化分类错误的数量。通过引入正则化参数C来控制间隔的宽度和误分类之间的权衡。C的值越大，模型对误分类的惩罚就越大，越倾向于硬间隔；C的值越小，模型对误分类的容忍度就越高，越倾向于软间隔。

Hinge损失函数通常用于SVM中，特别是在二分类问题中。

Hinge损失函数的含义是，如果预测结果与真实标签之间的乘积大于1，则损失为0；否则，损失为1减去预测结果与真实标签之间的乘积。

对于错误分类的样本，即那些预测结果与真实标签之间的乘积小于等于1的样本，Hinge损失函数会产生非零的损失值，从而对这些样本进行惩罚。这意味着，如果一个样本被错误分类，其损失将等于1减去正确类别分数与错误类别分数之差；如果一个样本被正确分类但置信度不高（即分数接近于边界），也会有一个较小的损失值，这鼓励模型不仅要正确分类样本，还要以较大的间隔进行分类。

## 因果网络伪路径(spurious·path)的概念(课件)，伪路径阻塞的判断(贝叶斯网络3种基本结构的阻塞，路径p被限定集Z阻塞)

1.A spurious path in a causal graph introduces a misleading association between variables due to factors like common causes or incorrect conditioning on colliders, making variables appear related when they are not causally connected.

伪路径是指在因果网络中，尽管存在路径连接了不同的节点，但它并不代表有效的因果关系。这通常发生在网络的条件独立性被破坏的情况下。换句话说，伪路径是那些可以被阻塞的路径，这些路径在给定某些条件下不应该影响目标变量。

2.

（1） V-结构（或者说 "隐性因果结构"） V-结构是指一个节点作为共同的父节点连接两个子节点（如 A → C ← B）。在这种结构中，A 和 B 是条件独立的，只有在给定 C 的情况下，它们才会有依赖关系。如果没有给定 C，路径 A → C ← B 就会被阻塞。

（2）链式结构（Chain Structure） 链式结构是指一个节点通过直接的链路连接到另一个节点（如 A → B → C）。在这种结构中，A 和 C 是条件独立的，只要我们给定了 B，那么 A 和 C 就会不再影响对方，路径 A → B → C 会被阻塞。

（3）分支结构（Fork Structure） 分支结构是指一个节点通过分支连接到两个子节点（如 A ← B → C）。在这种结构中，B 是 A 和 C 的共同父节点。如果我们给定了 B，那么 A 和 C 就不再影响对方，路径 A ← B → C 会被阻塞。

第一章

理性

可计算性

第二章

启发式

可容性 一致性

第三章

启发式函数

遗传算法 （？)

局部最小化 (？)

补充：Astar最优性

第四章

知识图谱+知识图谱推理

第五章

贝叶斯网络

1. 机器学习（无计算题）

三要素 策略 模型 算法

归纳偏好 没有免费午餐定理

经验风险 结构风险最小化

训练 验证 测试 集

过拟合与千拟合

计算空间

假设空间

支持向量机

监督学习

无监督学习

强化学习

1. means

PCA不要考

第七章 深度学习 不考计算题

反响传播

cnn rnn 概念

卷及概念

rnn fnn cnn三者关系

rnn解决的问题

drpout 等

多层神经网络优化的困难

梯度消失等

交叉商损失函数

生成式网络（文答题）

1. 强化学习（选择题为主）无问答

马尔科夫

贝尔曼

策略改进定理

q-learing \*\*（必考计算题）看书本计算题+概念

策略梯度算法

考到8.2