提纲

- **时间上的概率推理**
  - 时间与不确定性
  - 时序模型中的推断
  - 隐马尔可夫模型
  - 卡尔曼滤波
  - 动态贝叶斯网络

提纲

- **做简单决策**
  - 效用理论基础
  - 效用函数
  - 决策网络
  - 信息价值与未知偏好
- **做复杂决策**
  - 序贯决策问题
  - MDP的算法
  - 老虎机问题

提纲

南京大学 NANJING UNIVERSITY

- **做简单决策**
  - 效用理论基础
  - 效用函数
  - 决策网络
  - 信息价值与未知偏好
- **做复杂决策**
  - 序贯决策问题
  - MDP的算法
  - 老虎机问题

在不确定性下结合信念与愿望

南京大学 NANJING UNIVERSITY

- 智能体赋予每个可能的当前状态s一个概率$P(s)$。动作结果也可能存在不确定性；转移模型由$P(s^t \mid s, a)$给出：

$$P(\text{RESULT}(a) = s^t) = \sum_{s'} P(s)P(s^t \mid s, a) .$$

- 给定证据，一个动作的**期望效用** $EU(a)$是结果的加权平均效用值，其中权值是结果发生的概率:

$$EU(a) = \sum_{s'} P(\text{RESULT}(a) = s^t)U(s^t) .$$

- **最大期望效用**（*maximum expected utility*）原则认为理性的智能体应该选择能使其期望效用最大化的动作：

$$action = argmax_a EU(a) .$$

- 如果一个智能体采取的动作是为了使正确反映性能度量的效用函数最大化，那么智能体将获得最高的可能性能分数（在所有可能的环境中取平均值）。

效用理论基础

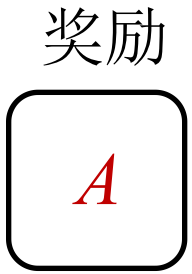
南京大学 NANJING UNIVERSITY

理性偏好的约束

- 智能体有以下偏好选择:
  - 奖励: A, B, 等.
  - 彩票: 奖励不确定的情况

    $L = [p, A;\ (1\text{-}p), B]$

    $L = [p_1, S_1; p_2, S_2; \cdots; p_n, S_n]$

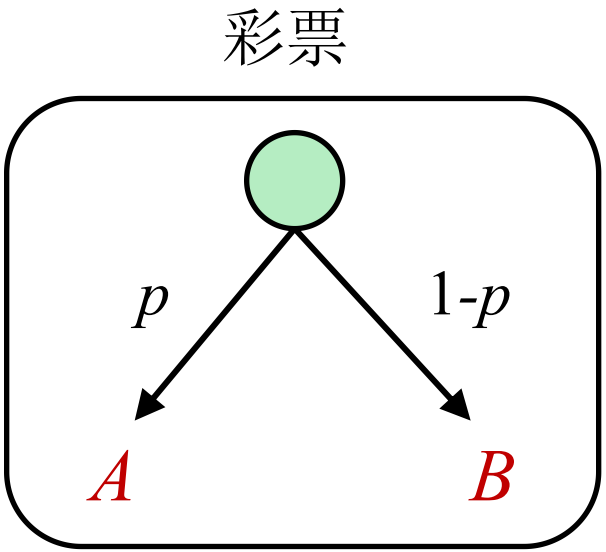


- 描述智能体偏好的符号:

$A \succ B$ 智能体偏好A甚于B

$A \sim B$ 智能体对A和B偏好相同

$A \succsim B$ 智能体偏好A甚于B或者对A和B偏好相同



效用理论基础

南京大学 NANJING UNIVERSITY

理性偏好的约束

- **有序性**（orderability）：给定任意两种彩票，理性的智能体必须偏好其中一种，或者将它们评为偏好相同的。也就是说，智能体不能避免做决策。正如12.2.3节所指出的，拒绝赌博就像拒绝时间流逝一样。

$(A \succ B)$、$(B \succ A)$ 或 $(A \sim B)$ 中有且只有一个成立

- **传递性**（transitivity）：给定任意3种彩票，如果一个智能体偏好A甚于B，偏好B甚于C，那么它一定偏好A甚于C。

$$(A \succ B) \wedge (B \succ C) \Rightarrow (A \succ C)$$

- **连续性**（continuity）：如果某彩票B在偏好上位于A和C之间，那么存在某种概率p，理性智能体将在肯定获得B和以概率p生成A并以概率$1-p$生成C的彩票之间偏好相同。

$$A \succ B \succ C \Rightarrow \exists p \ [p, A; 1-p, C] \sim B$$

效用理论基础

南京大学 NANJING UNIVERSITY

- **可替换性**（substitutability）：如果一个智能体在两种彩票A和B中偏好相同，那么这个智能体会在两种更复杂的彩票中偏好相同，这两种彩票除在其中一种彩票中用B代替了A之外，其他部分都是相同的。这无须考虑彩票的概率和其他结果即是成立的。

$$A \sim B \Rightarrow [p, A; 1-p, C] \sim [p, B; 1-p, C]$$

在这个公理中使用$\succ$替换$\sim$后仍然成立。

- **单调性**（monotonicity）：假设两种彩票有相同的两种可能的结果A和B。如果一个智能体偏好A甚于B，那么它必然偏好A的概率较高的彩票（反之亦然）。

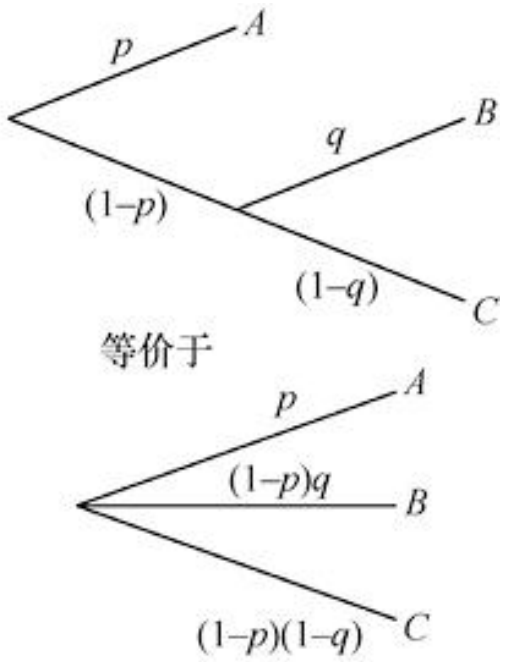
$$A \succ B \Rightarrow (p > q \Leftrightarrow [p, A; 1-p, B] \succ [q, A; 1-q, B])$$

- **可分解性**（decomposability）：利用概率法则，可以将复合彩票简化为更简单的彩票。这就是所谓的“无趣赌博”规则：如图16-1b所示，它将两个连续的彩票压缩成一个等价的彩票。[2]

$$[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$$

效用理论基础

$$[p, A; 1-p, [q, B; 1-q, C]] \sim [p, A; (1-p)q, B; (1-p)(1-q), C]$$



可分解性公理

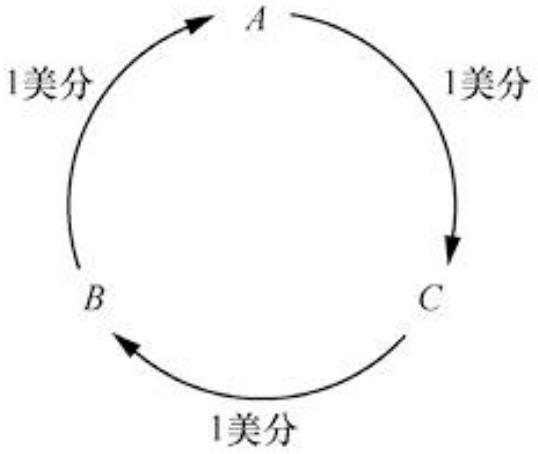
效用理论基础

南京大学 NANJING UNIVERSITY

理性偏好的约束

- 违反约束会导致明显的非理性
- 例如，我们可以通过让一个具有非传递性偏好的智能体把它所有的钱给我们来体现传递性公理的合理性。

- 如果 B > C, 那么拥有 C 的智能体将会支付 1 美分来获得 B
- 如果 A > B, 那么拥有 B 的智能体将会支付 1 美分来获得 A
- 如果 C > A, 那么拥有 A 的智能体将会支付 1 美分来获得 C



非传递性偏好$A \succ B \succ C \succ A$可能导致非理性行为：每次花费1美分的交换环。

效用理论基础

南京大学 NANJING UNIVERSITY

理性偏好导致效用

- **效用函数的存在性**（existence of utility function：如果一个智能体的偏好服从效用公理，那么存在一个函数U，使得$U(A) > U(B)$当且仅当偏好A甚于B，且$U(A) = U(B)$当且仅当智能体在A与B之间偏好相同。也就是

$$U(A) > U(B) \Leftrightarrow A \succ B \text{ 且 } U(A) = U(B) \Leftrightarrow A \sim B$$

- **彩票的期望效用**（expected utility of a lottery）：彩票的效用是每个结果的概率乘以该结果的效用之和。也就是

$$U([p_1, S_1; \cdots; p_n, S_n]) = \sum_i p_i U(S_i)$$

效用理论基础

南京大学 NANJING UNIVERSITY

理性偏好导致效用

- 效用函数U(S)在经过正仿射变换后，智能体的行为不会发生变化：

$$U'(S) = aU(S) + b$$

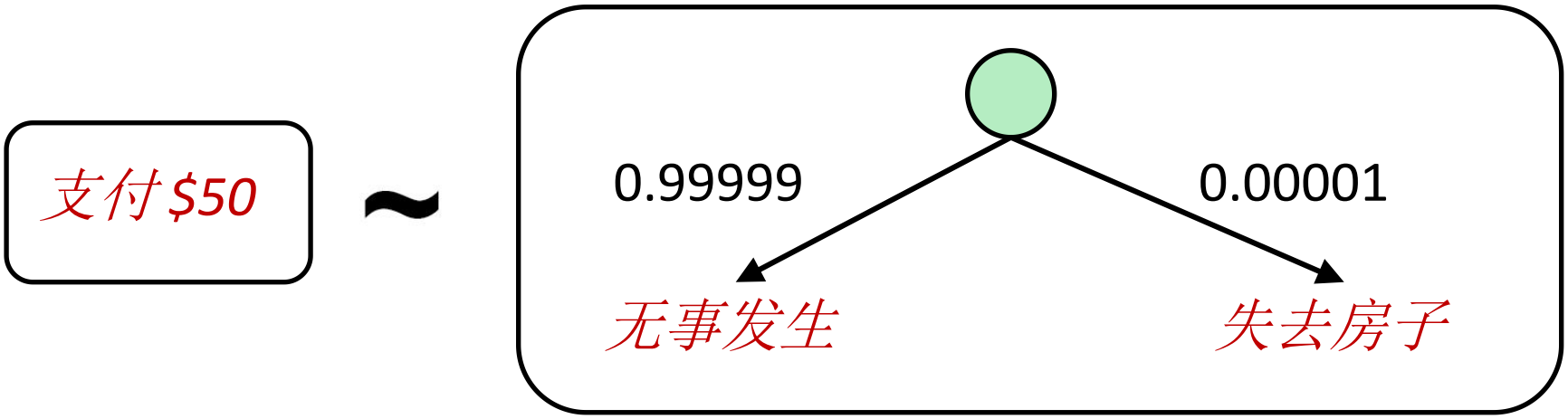
- 在确定性环境中，智能体只需要对状态进行偏好排序，其数值 并不重要。我们把这称为价值函数（value function）或序数效用函数（ordinal utility function）。

- 重要的一点是，描述智能体偏好行为的效用函数的存在并不一定意味着智能体在自己的思考中明确地最大化了效用函数。理性行为可以通 过多种方式产生。一个理性的智能体可以通过查表来实现。

效用函数

效用评估和效用尺度

- 效用函数将偏好映射为实数. 如何确定这个数?
- 人类效用评估（启发）的标准方法:
  - 比较奖励 A 与**标准彩票**L_p
    - “最佳可能奖励” $u_\top$ 拥有概率 p
    - “最坏可能灾难” $u_\bot$ 拥有概率1-p
  - 调整标准彩票概率 p 直到偏好相同: $A \sim L_p$
  - 所得到 p 就是A的效用，其值位于 [0,1]之间。



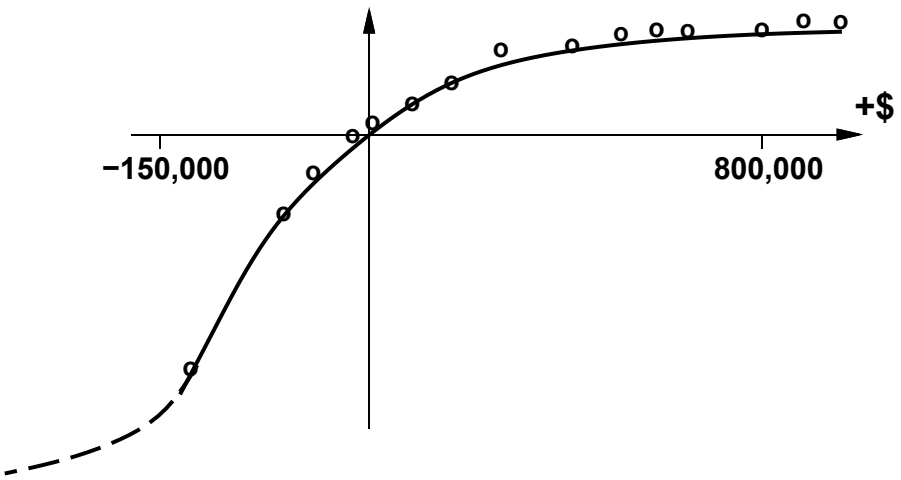
金钱的效用

- 金钱并不表现为效用函数。
- 给定彩票L = [p, $X; (1-p), $Y]，期望货币价值：EMV(L) = pX + (1-p)Y
- 假设我们用S_n表示拥有总财富n美元的状态，你的当前财富为k美元。那么接受赌局和拒绝赌局的两种行为的期望效用为：

$$EU(Accept) = \frac{1}{2}U(S_k) + \frac{1}{2}U(S_{k+2\,500\,000})$$

$$EU(Decline) = U(S_{k+1\,000\,000})$$

- 效用曲线:



通常在正财富区域 U (L) < U (EMV (L)), 即人们是风险厌恶的，但在大量负财富的“绝望”区域，人们的行为呈现出风险寻求（risk-seeking）

多属性效用函数

- 如何处理有多个变量 X1 . . . Xn 的效用函数？

- 机场选址问题中的属性可以是：
  - Throughput，每天的飞行次数；
  - Safety，负的每年期望死亡人数；
  - Quietness，负的居住在飞行路径下的人数；
  - Frugality，负的建筑成本。

- 如何根据偏好行为评估复杂的效用函数？

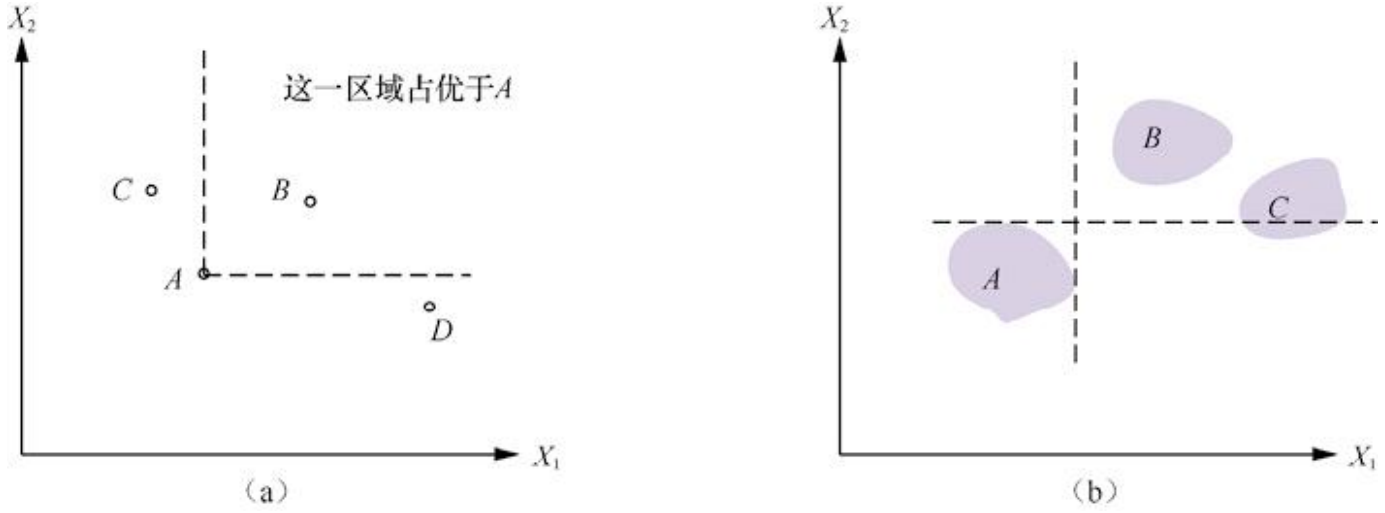
- 我们首先考察无须将属性值组合为单个效用值就可以做出决策的情况。然后我们再 探究在哪些情况下，属性组合的效用可以被非常简明地指定。

多属性效用函数

南京大学 NANJING UNIVERSITY

严格占优

- 选项B严格占优于选项A当且仅当
  $\forall\ i\ X_i(B) \geq X_i(A)$ (因此 $U(B) \geq U(A)$)



（a）确定性：对A有严格占优的是B，而不是C或D。（b）不确定性：对A有严格占优的是B，而不是C

多属性效用函数

南京大学 NANJING UNIVERSITY

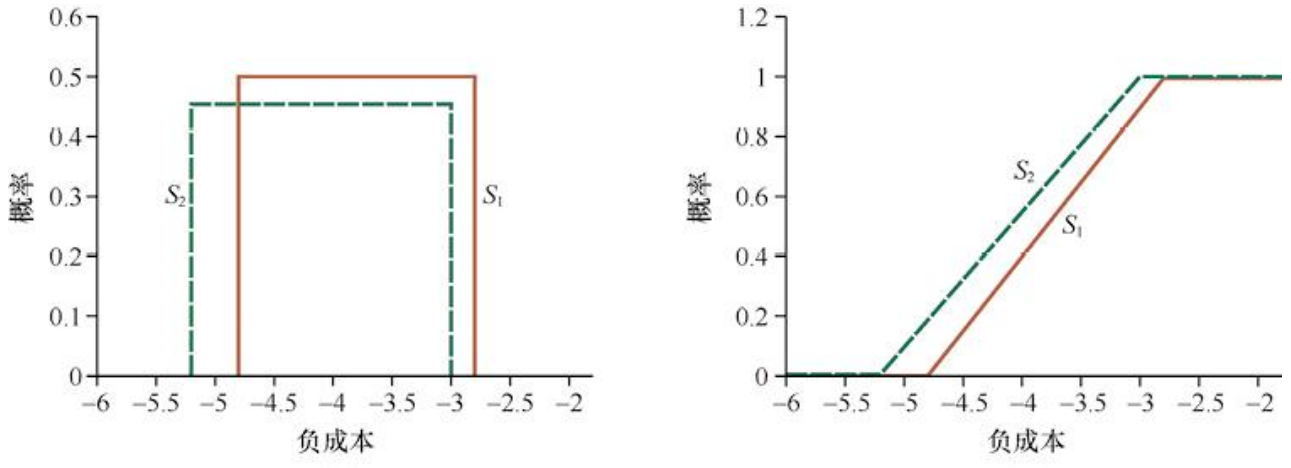
随机占优

- 分布 p_1 随机占优于分布 p_2 当且仅当

$$\forall x \int_{-\infty}^{x} p_1(x')\mathrm{d}x' \leqslant \int_{-\infty}^{x} p_2(x')\mathrm{d}x'$$

- 如果A1 随机占优于A2 ，那么对于任意单调不减的效用函数U(x)，A1 的期望效用至少和A2 的期望效用一样高。

$$\int_{-\infty}^{+\infty} p_1(x)U(x)\mathrm{d}x = \int_0^1 U(P_1^{-1}(y))\mathrm{d}y \geqslant \int_0^1 U(P_2^{-1}(y))\mathrm{d}y = \int_{-\infty}^{+\infty} p_2(x)U(x)\mathrm{d}x$$



（a）在节省成本（负成本）上，S_1随机占优于S_2。（b）S_1与S_2的节省成本的累积分布

多属性效用函数

南京大学 NANJING UNIVERSITY

随机占优

随机占优通常可以在没有精确分布的情况下通过**定性推理**确定

例如，建筑运输成本随着距城市距离的增加而增加

S_1 相比 S_2 要更靠近城市

$\Rightarrow$ S_1 在成本上随机占优于 S_2

可以用随机占优信息标注置信度网络:

$X \xrightarrow{+} Y$ (X 正面影响 Y) 意味着对 Y 的其他父节点 Z的每一个 z

$\forall\ x_1, x_2\ \ x_1 \geq x_2 \Rightarrow \mathrm{P}(Y \mid x_1, \mathrm{z})$ 随机占优于 $\mathrm{P}(Y \mid x_2, \mathrm{z})$

多属性效用函数

南京大学 NANJING UNIVERSITY

偏好结构：确定性

- **表示定理**证明具有特定种类的偏好结构的智能体具有效用函数：

$$U(x_1, \cdots, x_n) = F[f_1(x_1), \cdots, f_n(x_n)]$$

X_1 和 X_2 **偏好独立于** X_3 当且仅当 (x_1, x_2, x_3) 和 (x^t_1, x^t_2, x_3) 之间的偏好不依赖于 x_3

例如, *(Noise, Cost, Safety)*:

(20,000 suffer, $4.6 billion, 0.06 deaths/mpm) vs.

(70,000 suffer, $4.2 billion, 0.06 deaths/mpm)

相互偏好独立性 $\Rightarrow$ $\exists$ **加性价值函数**:

$$V(S) = \Sigma_i V_i(X_i(S))$$

在含有n个属性时，我们可以通过评估n个单独的一维价值函数来评估一个加性价值函数，而不需要直接评估一个n维函数。

多属性效用函数

南京大学 NANJING UNIVERSITY

偏好结构：随机性

- **效用独立性（utility independence）**的基本概念将偏好独立性推广到彩票领域：一组属性X的效用独立于另一组属性Y，如果关于属性X的彩票间的偏好独立于属性Y的特定值。

- 一组属性是**相互效用独立的（mutually utility independent， MUI）**，如果它的每个属性子集都效用独立于其他属性。

相互效用独立

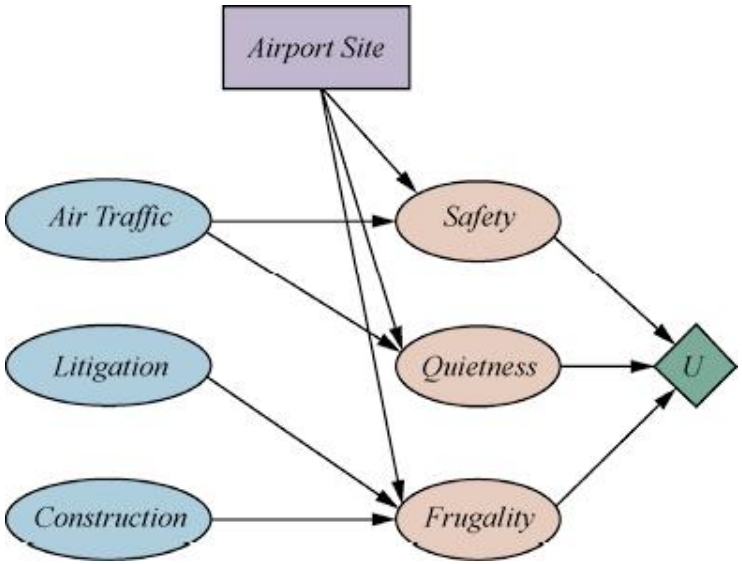
⇒ ∃**乘性效用函数：**

$$U = k_1U_1 + k_2U_2 + k_3U_3 + k_1k_2U_1U_2 + k_2k_3U_2U_3 + k_3k_1U_3U_1 + k_1k_2k_3U_1U_2U_3$$

决策网络

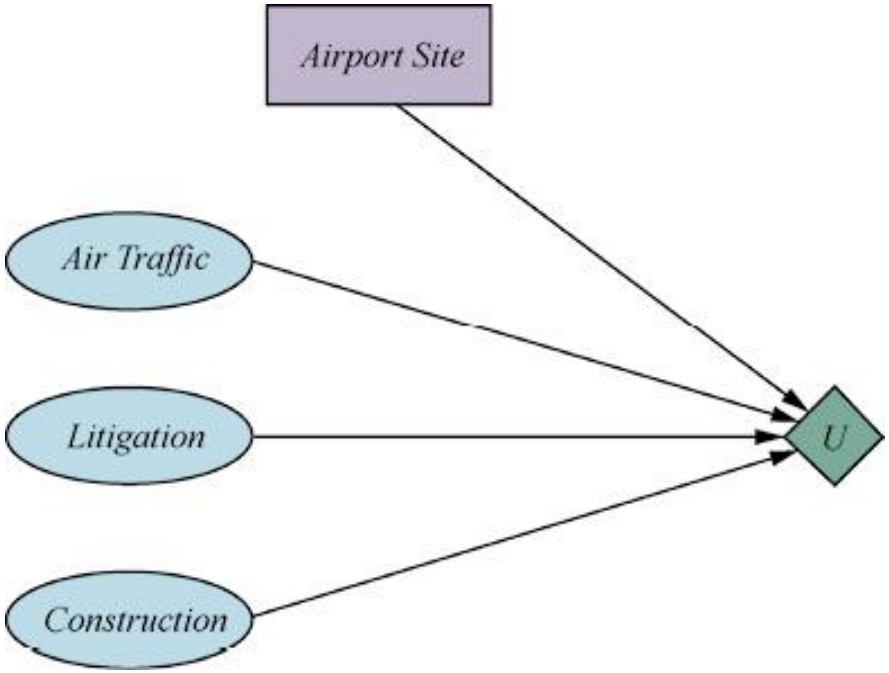
- **决策网络**表示关于智能体的当前状态信息、可能动作、由智能体的动作导致的状态以及该状态的效用。

- 三种类型节点：
  - **机会节点**（椭圆形）表示随机变量（就像在贝叶斯网络中一样）。
  - **决策节点**（矩形）代表决策制定者可以选择动作的点。
  - **效用节点**（菱形）代表智能体的效用函数。



机场选址问题的决策网络

决策网络



机场选址问题的简化表示。与结果状态对应的机会节点已被略去

决策网络

评估决策网络

1. 为当前状态设置证据变量.
2. 对决策节点的每个可能值：
   (a) 将决策节点设置为该值；
   (b) 使用标准概率推断算法计算效用节点的父节点的后验概率;
   (c)计算动作的结果效用。
3. 返回具有最高效用的动作。

信息价值

- **信息价值理论**使智能体可以选择获取什么信息。这里假设在选择由决策节点表示的实际动作之前，智能体可以获得模型中任何潜在可观测的机会变量的值。

示例: 购买石油开采权

两个区域 A 和 B, 只有一个有石油, 价值为 k

先验概率为 0.5

每个区域开采权的价格为 $k/2$

顾问提供了关于区域A是否有石油的准确调查.

应该为这些信息支付多少钱?

求解: 信息的期望值

= 给定信息最佳动作的期望值 - 没有信息最佳动作的期望值

= [0.5 × 给定A有石油买A的值

+ 0.5 ×给定A没有石油买B的值]

– $(k/2 - k/2)$

= $(0.5 \times k/2) + (0.5 \times k/2) - 0 = k/2$

信息价值

南京大学 NANJING UNIVERSITY

完美信息的一般公式

- 在智能体只有原始信息的状态下，当前最佳动作的价值为：

$$EU(\alpha) = \max_{a} \sum_{s'} P(\text{RESULT}(a) = s')U(s')$$

- 在获得新证据$E_j = e_j$后，新的最佳动作的价值为：

$$EU(\alpha_e \mid e_j) = \max_{a} \sum_{s'} P(\text{RESULT}(a) = s' \mid e_j)U(s')$$

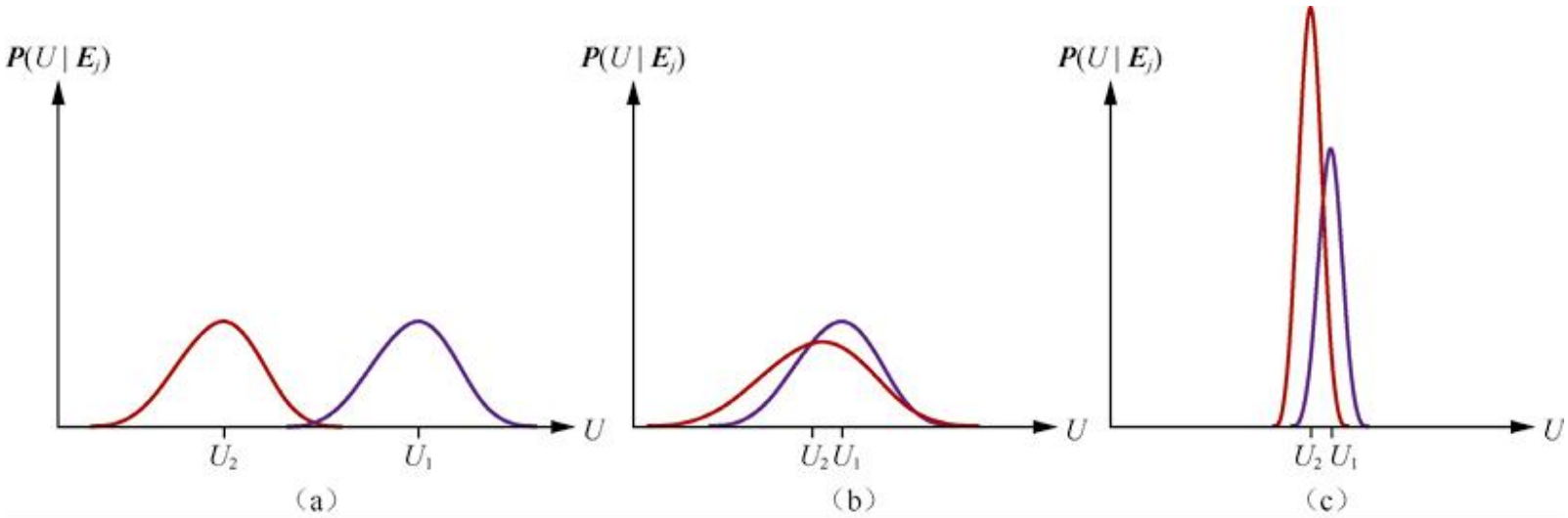
- 但是E_j是一个随机变量，它的值目前是未知的，所以要确定新信息E_j的价值，我们必须使用其值的当前概率分布，对所有可能观测的E_j值e_j取平均：

$$VPI(E_j) = \left(\sum_{e_j} P(E_j = e_j) EU(\alpha_{e_j} \mid E_j = e_j) \right) - EU(\alpha)$$

(VPI = 完美信息价值)

信息价值

完美信息的一般公式



信息价值的3种一般情况。（a）a_1几乎肯定会优于a_2，因此我们不需要这些信息。（b）选择是不明确的，信息至关重要。（c）选择是不明确的，但信息因为带来的差异很小，信息价值不高。注意：U_2在（c）中有一个峰值，说明其期望值比U_1有着更高的确定性

信息价值

南京大学 NANJING UNIVERSITY

价值信息的性质

- **非负性**：信息的期望价值是非负的，即

$$\forall j \; VPI(E_j) \geqslant 0$$

- 不可加性

$$VPI(E_j, E_k) \neq VPI(E_j) + VPI(E_k) \qquad \text{（一般情况下）}$$

- 次序独立的

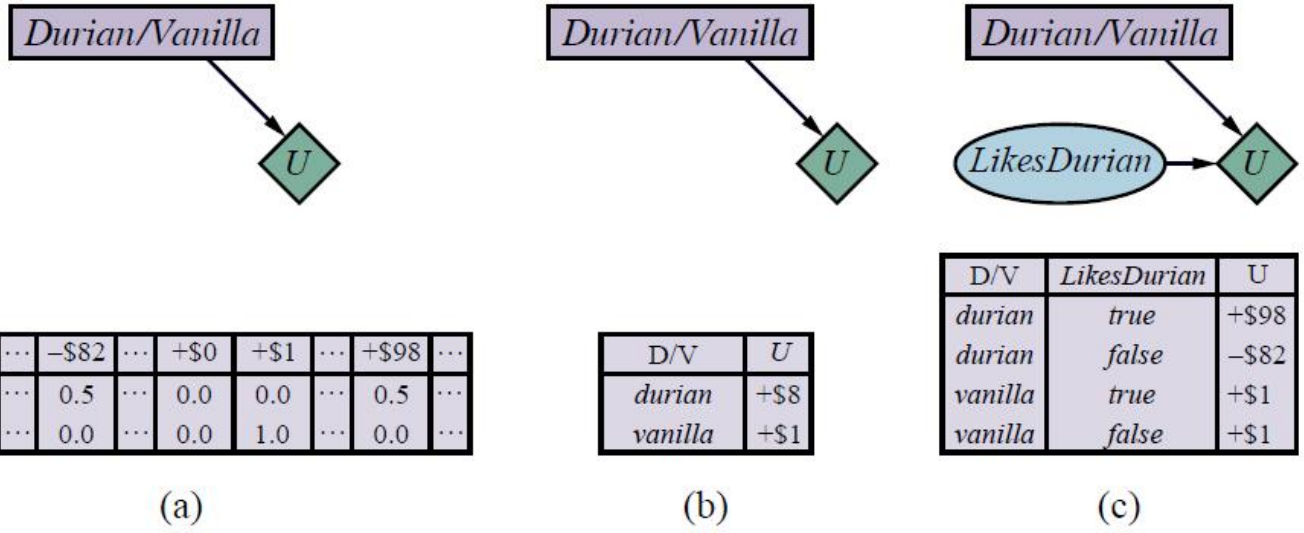
$$VPI(E_j, E_k) = VPI(E_j) + VPI(E_k \mid E_j) = VPI(E_k) + VPI(E_j \mid E_k) = VPI(E_k, E_j)$$

未知偏好

想象一下，你正位于泰国的一家冰激凌店，但店里只剩下两种口味的冰激凌：香草味（Vanilla）冰激凌和榴莲味（Durian）冰激凌。它们的成本都是2美元。你知道你比较喜欢香草口味，并愿意在这样一个大热天花3美元买一个香草味冰激凌，所以选择香草味冰激凌净收益是1美元。但是，你不知道你是否喜欢榴莲口味，你已经在维基百科上调查过，不同的人对榴莲有着不同的反应：有人发现“它的味道超过了世界上所有其 他水果”，而其他人把它比作“污水、陈腐的呕吐物、臭鼬喷雾和用过的外科拭子”。

未知偏好

具体来说，假设有50%的机会你会觉得榴莲很棒（收益为+100美元），而有50%的 概率你会讨厌它（收益为−80美元）。在这个问题中，你将赢得什么奖品没有不确定性。无论哪种方式，榴莲味冰激凌都是一样的，但你自己对奖品的偏好有不确定性。



D/V	⋯	−$82	⋯	+$0	+$1	⋯	+$98	⋯
durian	⋯	0.5	⋯	0.0	0.0	⋯	0.5	⋯
vanilla	⋯	0.0	⋯	0.0	1.0	⋯	0.0	⋯

(a)

D/V	*U*
durian	+$8
vanilla	+$1

(b)

D/V	*LikesDurian*	U
durian	*true*	+$98
durian	*false*	−$82
vanilla	*true*	+$1
vanilla	*false*	+$1

(c)

（a）效用函数不确定的冰激凌选择的决策网络。（b）带有每个动作期望效用的网络。（c）将不确定性从效用函数转移到一个新的随机变量

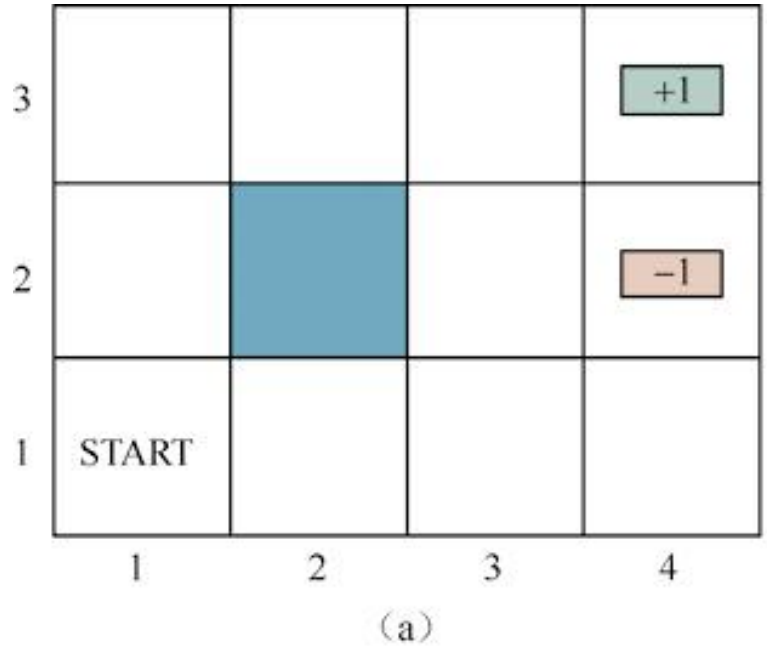
小结

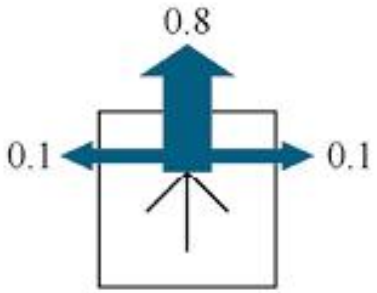
- **概率论**描述的是在证据的基础上一个智能体应该相信什么，**效用理论**描述的是一个智能体想要什么，而**决策论**将两者结合起来描述一个智能体应该做什么。
- 我们可以使用决策论来建立一个系统，通过考虑所有可能的动作，并选择其中导致最佳期望结果的一个来决策。这样的系统被称为**理性智能体**。
- 效用理论表明，对彩票的偏好与一组简单的公理一致的智能体，可以被描述为具有一个效用函数；此外，智能体选择动作就像最大化其期望效用一样。
- **多属性效用理论**研究依赖于几种不同状态属性的效用。随机占优是做出明确决策的一种特别有用的技术，即使没有精确的属性效用值。
- **决策网络**为表达和求解决策问题提供了一种简单的形式体系。它们是贝叶斯网络的自然延伸，除了机会节点，还包含决策节点和效用节点。
- 有时，求解问题需要在做出决策之前找到更多的信息。**信息价值**被定义为与在没有信息的情况下做出决策相比，期望效用的提高；这对在作出最后决策之前指导收集信息的过程特别有用。

提纲

- 做简单决策
  - 效用理论基础
  - 效用函数
  - 决策网络
  - 信息价值与未知偏好
- **做复杂决策**
  - 序贯决策问题
  - MDP的算法
  - 老虎机问题

序贯决策问题





（a）一个简单的、随机的4×3环境，它向智能体呈现一个序贯决策问题。（b）环境的转移模型的图示：“预期的”结果以0.8的概率出现，但在0.2的概率下，智能体以垂直于预期方向的角度运动。与墙的碰撞不会导致任何运动。转移到两种终止状态的奖励分别为+1和−1，所有其他转移的奖励为−0.04

序贯决策问题

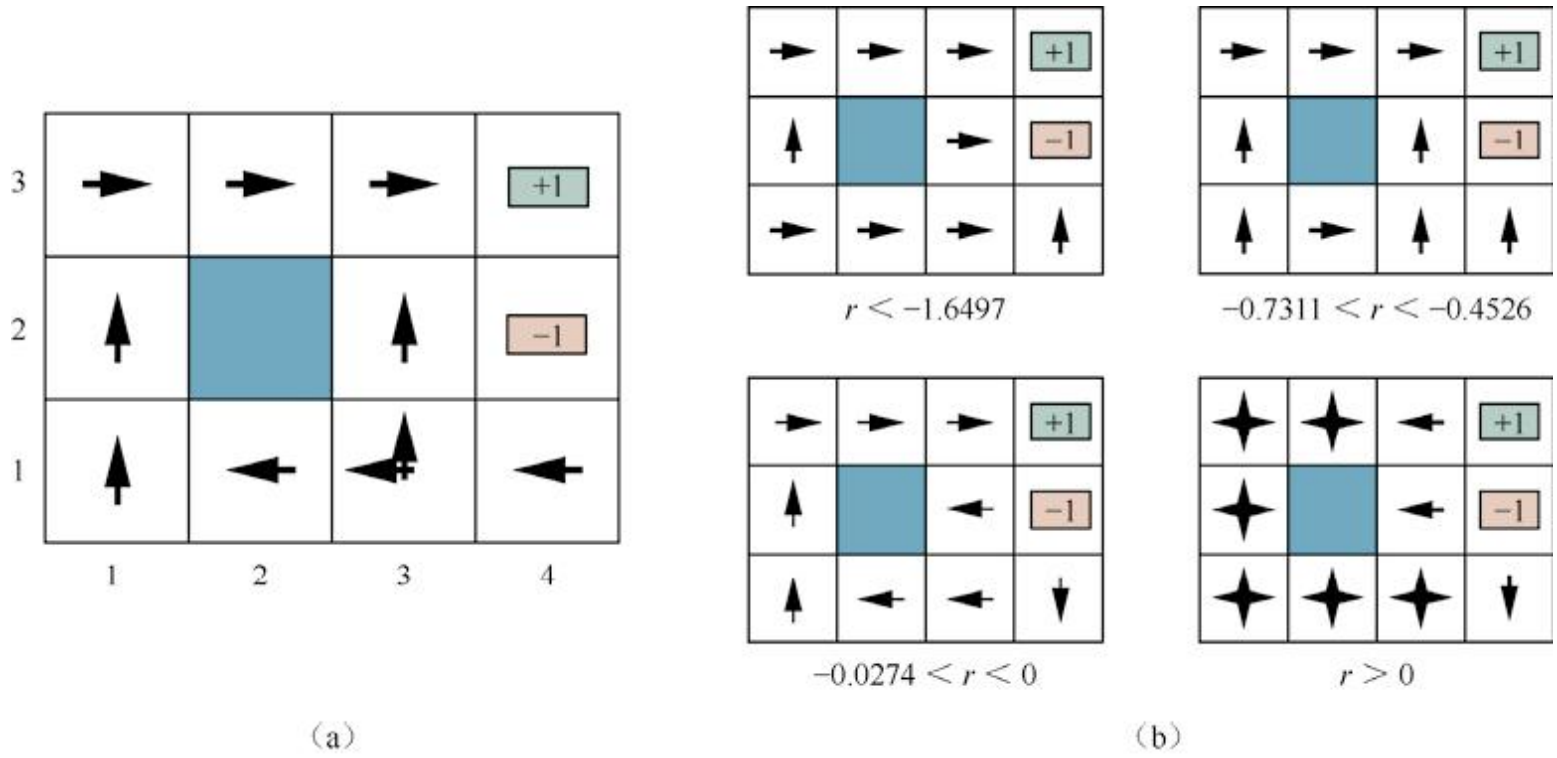
- **马尔可夫决策过程（*Markov decision process, MDP*）**：一个完全可观测的随机环境下，具有马尔可夫转移模型和加性奖励的序贯决策问题。

- *MDP* 包含:
  - 状态集合 (初始状态s_0);
  - 每个状态下的动作集合ACTIONS(s);
  - 转移模型 $P(s \mid s, a)$;
  - 奖励函数 $R(s, a, s)$.

- **动态规划**是求解*MDP*的常用方法：通过递归将问题分解成更小的部分，并考虑各部分的最优解来简化问题。

- **策略（*policy*）**：
  - 告知智能体在可能到达的任何状态下应该采取什么动作
  - 策略的质量是通过该策略所产生的可能环境历史的期望效用来衡量的
  - **最优策略**是指能够产生最大期望效用的策略

序贯决策问题



（a）在$r = -0.04$的随机环境下，非终止状态之间转移的最优策略，这里有两种策略，因为在状态(3, 1)，*Left* 和*Up*都是最优的。（b）对于r的4个不同取值范围的最优策略

序贯决策问题

南京大学 NANJING UNIVERSITY

时间上的效用 $U_h([s_0, a_0, s_1, a_1 \cdots, s_n])$

- **有限期：**存在固定时间N，对于该时间点之后的任何事件我们都不再关注
  - $Uh([s_0, a_0, s_1, a_1, \ldots, sN+ k]) = Uh([s_0, a_0, s_1, a_1, \ldots, sN])$
  - 给定状态下的最优动作可能取决于还剩多少时间
  - **非平稳的策略：**依赖于时间的策略

- **无限期:** 没有固定的时间限制
  - 智能体位于同一状态的在不同时间下的行为应该是相同的。

- 此时的最优动作只取决于当前状态，我们称此时最优策略是**平稳的**。

序贯决策问题

南京大学 NANJING UNIVERSITY

时间上的效用

- **加性折扣奖励：**
  $U_h([s_0, a_0, s_1, a_1, s_2, \ldots]) = R(s_0, a_0, s_1) + \gamma R(s_1, a_1, s_2) + \gamma^2 R(s_2, a_2, s_3) + \cdots$，
  这里**折扣因子**γ 是 0 到 1之间的数。

- **加性折扣奖励是合理的：**经验方面，经济方面，对真实奖励的不确定性，历史的偏好。

- 使用折扣奖励后，无限序列的效用会是有限的：
- 如果 $\gamma < 1$ 且奖励的界限为 $\pm R_{\max}$, 我们有

$$U_h([s_0, a_0, s_1, \ldots]) = \sum_{t=0}^{\infty} \gamma^t R(s_t, a_t, s_{t+1}) \leq \sum_{t=0}^{\infty} \gamma^t R_{\max} = \frac{R_{\max}}{1-\gamma},$$

- **适当策略**:保证到达终止状态的策略。

- 无限序列可以通过每一时间步的**平均奖励**进行比较。

序贯决策问题

南京大学 NANJING UNIVERSITY

最优策略与状态效用

- 从s开始执行π获得的期望效用：

$$U^{\pi}(s) = E\left[\sum_{t=0}^{+\infty} \gamma^t R(S_t, \pi(S_t), S_{t+1})\right]$$

- 从s开始的最优策略：

$$\pi_s^* = \underset{\pi}{\operatorname{argmax}}\, U^{\pi}(s)$$

- 效用函数U(s)允许智能体通过使用最大化期望效用的原则来选择动作，也就是说，选择使下一步奖励加上后续状态的期望折扣效用最大化的动作：

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma U(s')]$$

序贯决策问题

南京大学 NANJING UNIVERSITY

最优策略与状态效用

- **贝尔曼方程：**假设智能体选择了最优动作，状态效用是下一次转移的期望奖励加上下一个状态的折扣效用：

$$U(s) = \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma U(s')] .$$

- **动作效用函数**, 或者**Q函数**（Q-function）: $Q(s, a)$是在给定状态下采取给定动作的期望效用，Q函数显然与U(s)有关

$$U(s) = \max_{a} Q(s, a) .$$

- 最优策略可以按照如下方式从Q函数中提取出来

$$\pi^*(s) = \operatorname*{argmax}_{a} Q(s, a)$$

- Q函数在求解MDP的算法中经常出现

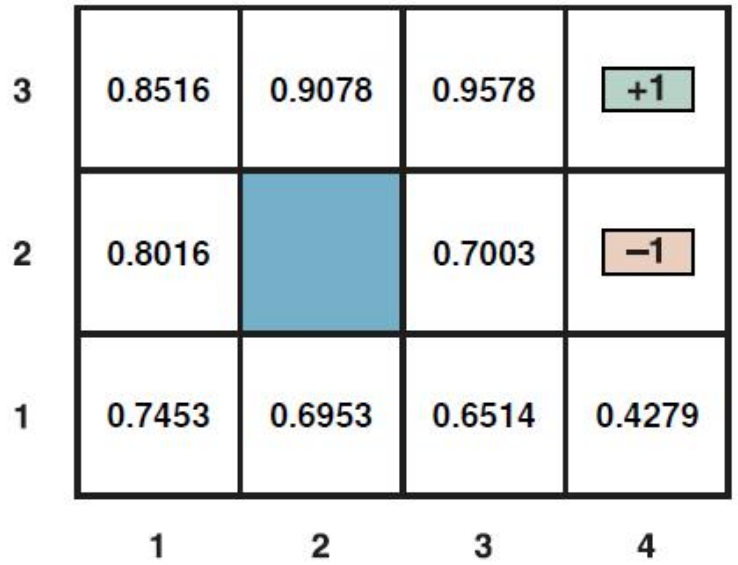
function Q-VALUE(mdp, s, a, U) **returns** 一个效用值

return $\sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma U[s']]$

序贯决策问题

南京大学 NANJING UNIVERSITY

最优策略与状态效用



$\gamma = 1$，向非终止状态转移的奖励$r = -0.04$的4×3世界的状态效用

$U(1, 1)$的表达式：

$$\begin{aligned}\max\{ &[0.8(-0.04 + \gamma U(1, 2)) + 0.1(-0.04 + \gamma U(2, 1)) + 0.1(-0.04 + \gamma U(1, 1))],\\ &[0.9(-0.04 + \gamma U(1, 1)) + 0.1(-0.04 + \gamma U(1, 2))],\\ &[0.9(-0.04 + \gamma U(1, 1)) + 0.1(-0.04 + \gamma U(2, 1))],\\ &[0.8(-0.04 + \gamma U(2, 1)) + 0.1(-0.04 + \gamma U(1, 2)) + 0.1(-0.04 + \gamma U(1, 1))]\}\end{aligned}$$

序贯决策问题

南京大学 NANJING UNIVERSITY

奖励规模

- 设Φ(*s*)是关于状态s的任意函数，则根据函数设计定理，以下变换能够保持最优策略不变:

$$R'(s, a, s') = R(s, a, s') + \gamma\Phi(s') - \Phi(s)$$

- 对于MDP M的Q函数为 $Q(s,a) = \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma \max_{a'} Q(s',a')]$.
- 将 $Q'(s, a) = Q(s, a) - \Phi(s)$ 代入上述等式可得

$$\begin{aligned} Q'(s,a) &= \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma\Phi(s') - \Phi(s) + \gamma \max_{a'} Q'(s',a')] \\ &= \sum_{s'} P(s' \mid s,a)[R'(s,a,s') + \gamma \max_{a'} Q'(s',a')]. \end{aligned}$$

- 提取*M'*的最优策略

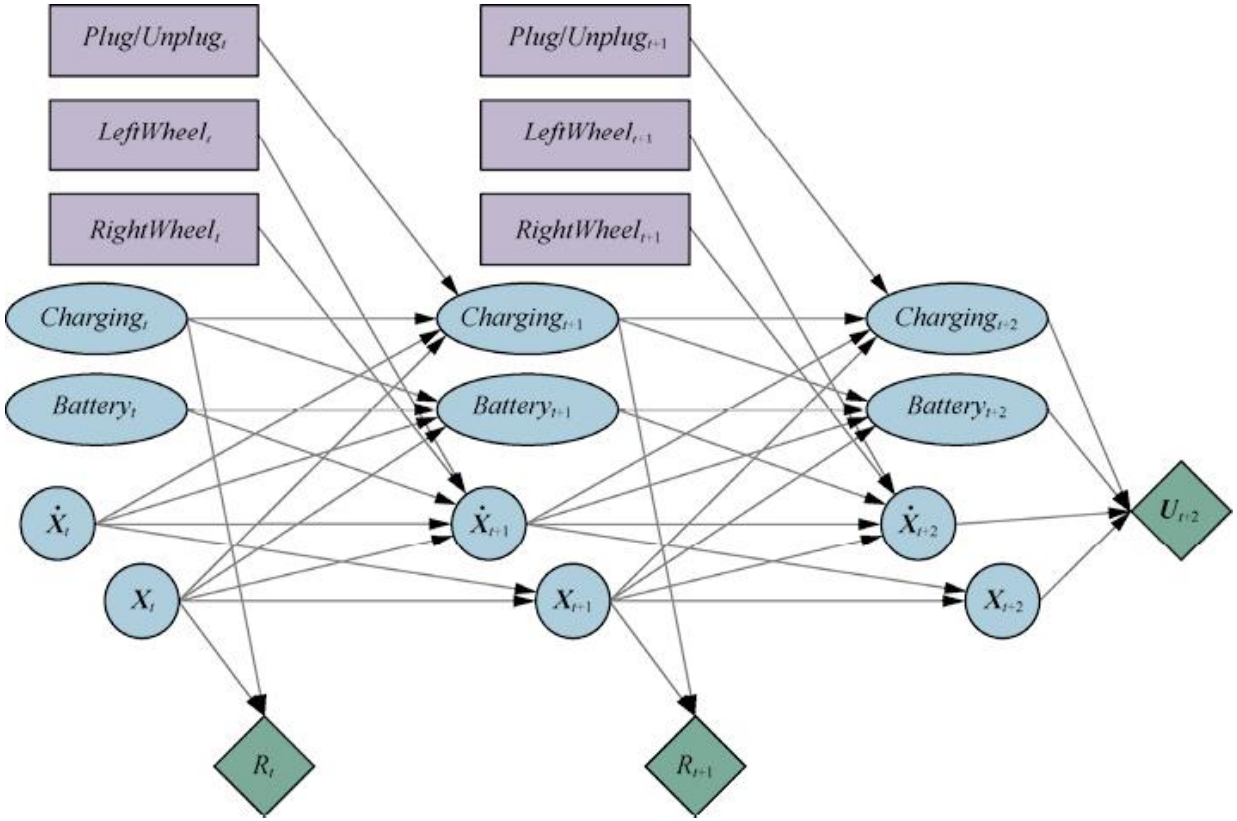
$$\pi^*_{M'}(s) = \operatorname*{argmax}_a Q'(s,a) = \operatorname*{argmax}_a Q(s,a) - \Phi(s) = \operatorname*{argmax}_a Q(s,a) = \pi^*_M(s).$$

- 函数 Φ(*s*) 通常被称为**势函数**

序贯决策问题

南京大学 NANJING UNIVERSITY

表示MDP

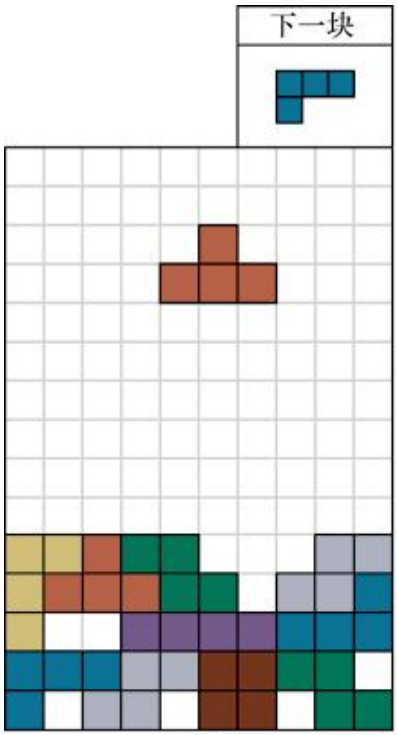


以电池电量、充电状态、位置和速度为状态变量，左右轮机动和充电为动作变量的移动机器人动态决策网络

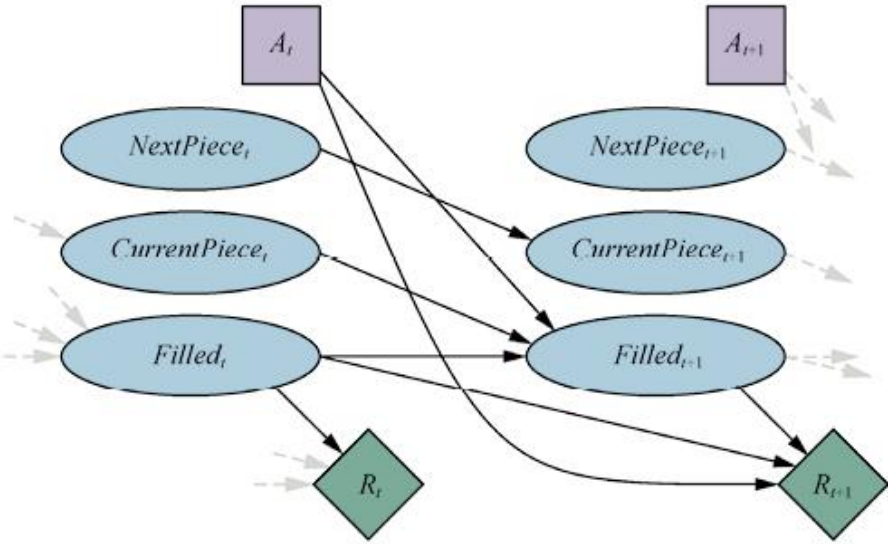
序贯决策问题

南京大学 NANJING UNIVERSITY

表示MDP



（a）



（b）

（a）俄罗斯方块游戏。位于顶部中心的T型块可以落在任何方向和任何水平位置。如果某一行被补全，则该行消失，上方的行向下移动，智能体得1分。下一块（这里是右上方的L形块）成为当前的一块，并出现一个新的下一块，从7种类型中随机选择。如果棋盘被填满，游戏结束。（b）俄罗斯方块MDP的DDN

MDP的算法

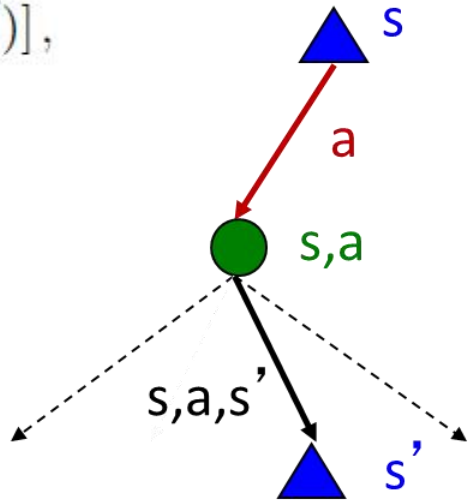
南京大学 NANJING UNIVERSITY

价值迭代

- **贝尔曼方程**是求解MDP的价值迭代算法的基础。
- 从效用的任意初始值出发计算方程的右侧，并将其值代入方程的左侧，从而根据其邻居的效用更新每个状态的效用。
- 重复这个过程，直到达到平衡。
- 这个迭代步，也称为**贝尔曼更新**：

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$

- 更新假设在每次迭代时同时应用于所有状态。



MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

function VALUE-ITERATION(mdp, ε) **returns** 一个效用函数
  inputs: mdp，具有状态S、动作$A(s)$、转移模型$P(s' \mid s, a)$、奖励 $R(s, a, s')$、折扣 γ的MDP
    ε，任意状态效用允许的最大误差
  local variables: U, U'，S中状态的效用向量，初始为0
    δ，任何状态的效用的最大相对变化

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each 状态s **in** S **do**
      $U'[s] \leftarrow \max_{a \in A(s)}$ Q-VALUE(mdp, s, a, U)
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta \leqslant \varepsilon (1 - \gamma)/\gamma$
  return U

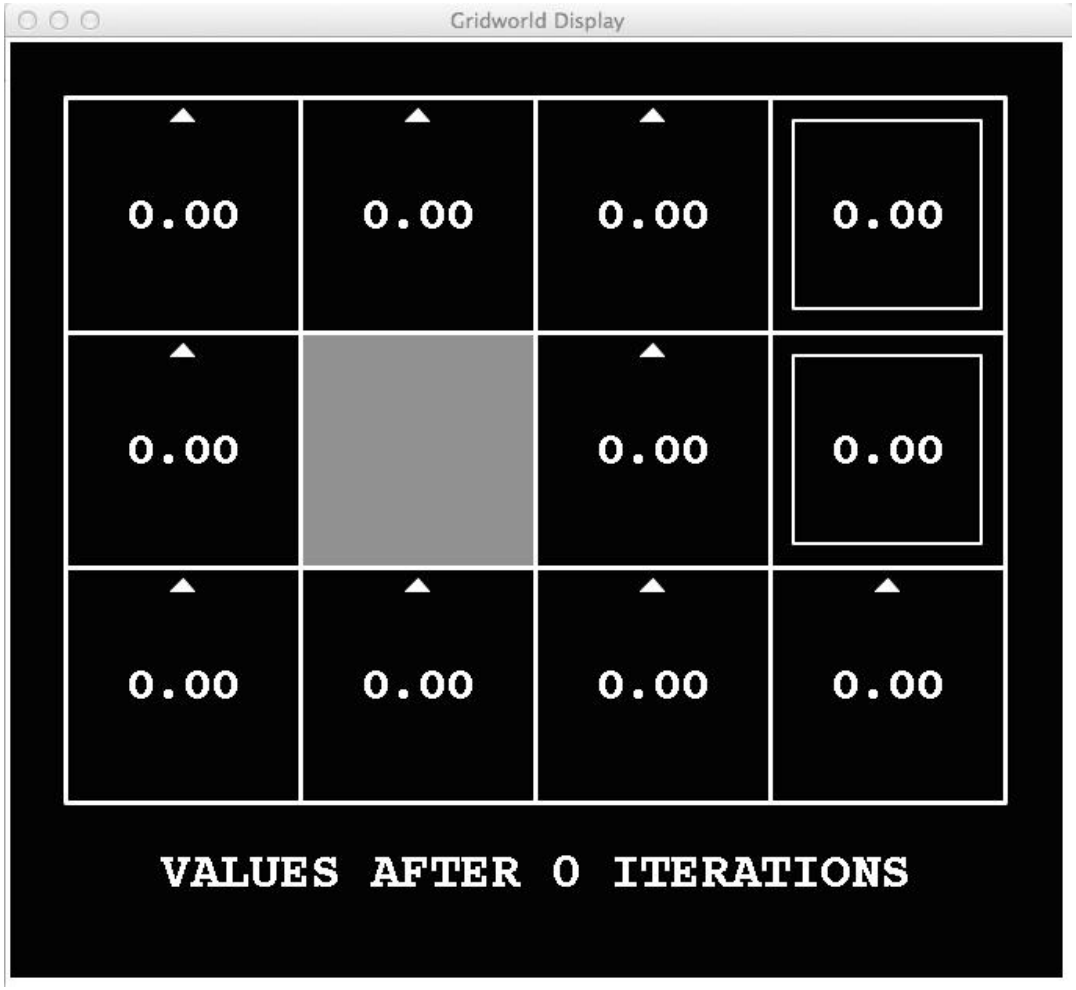
计算状态效用的价值迭代算法。

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$



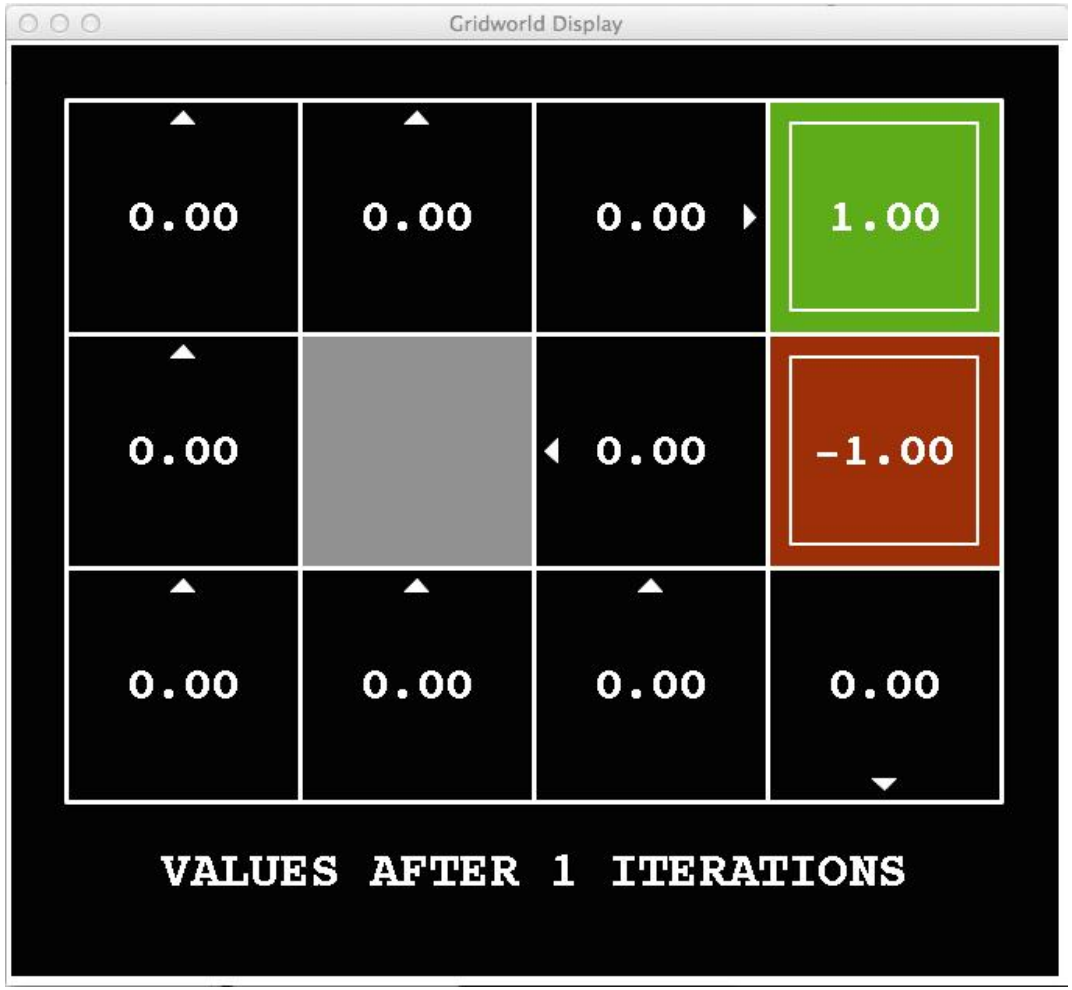
Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$



Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$

Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$

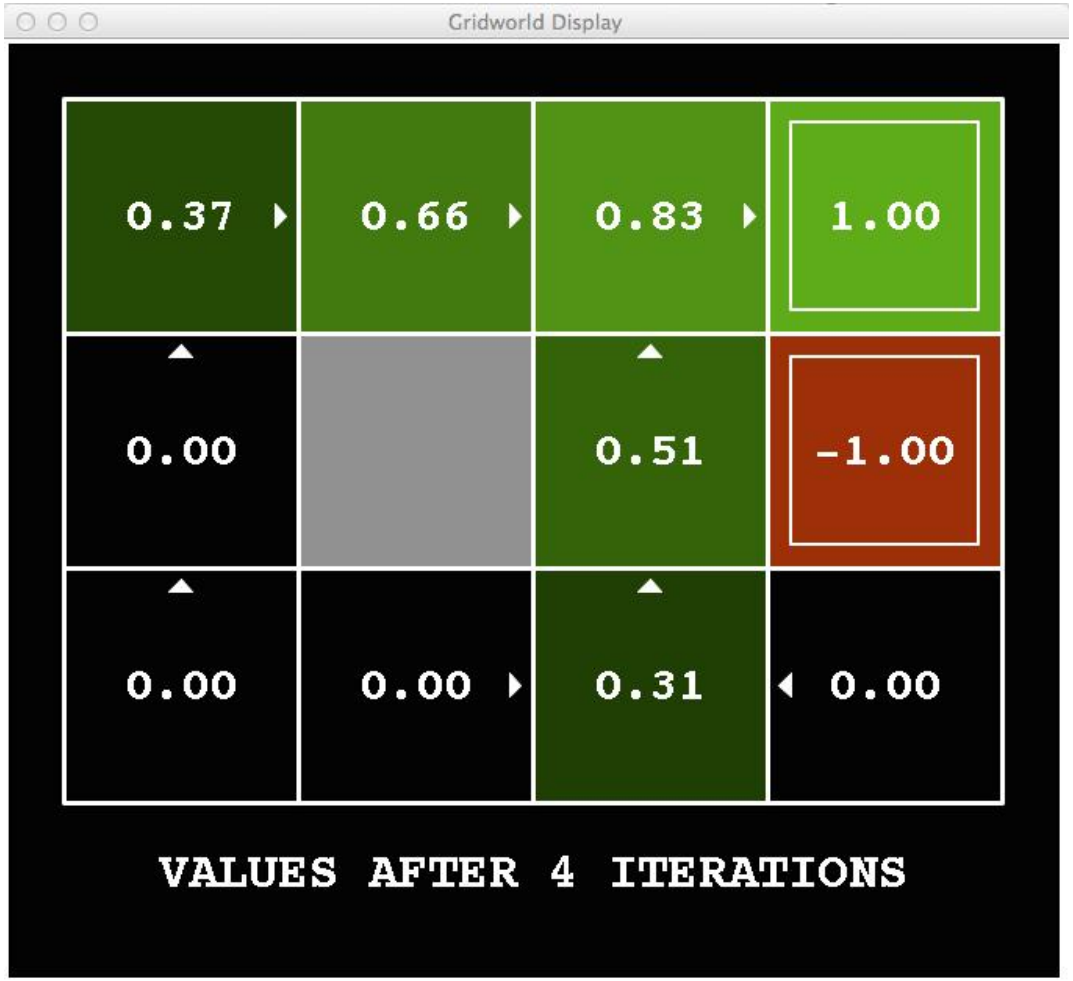
Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$



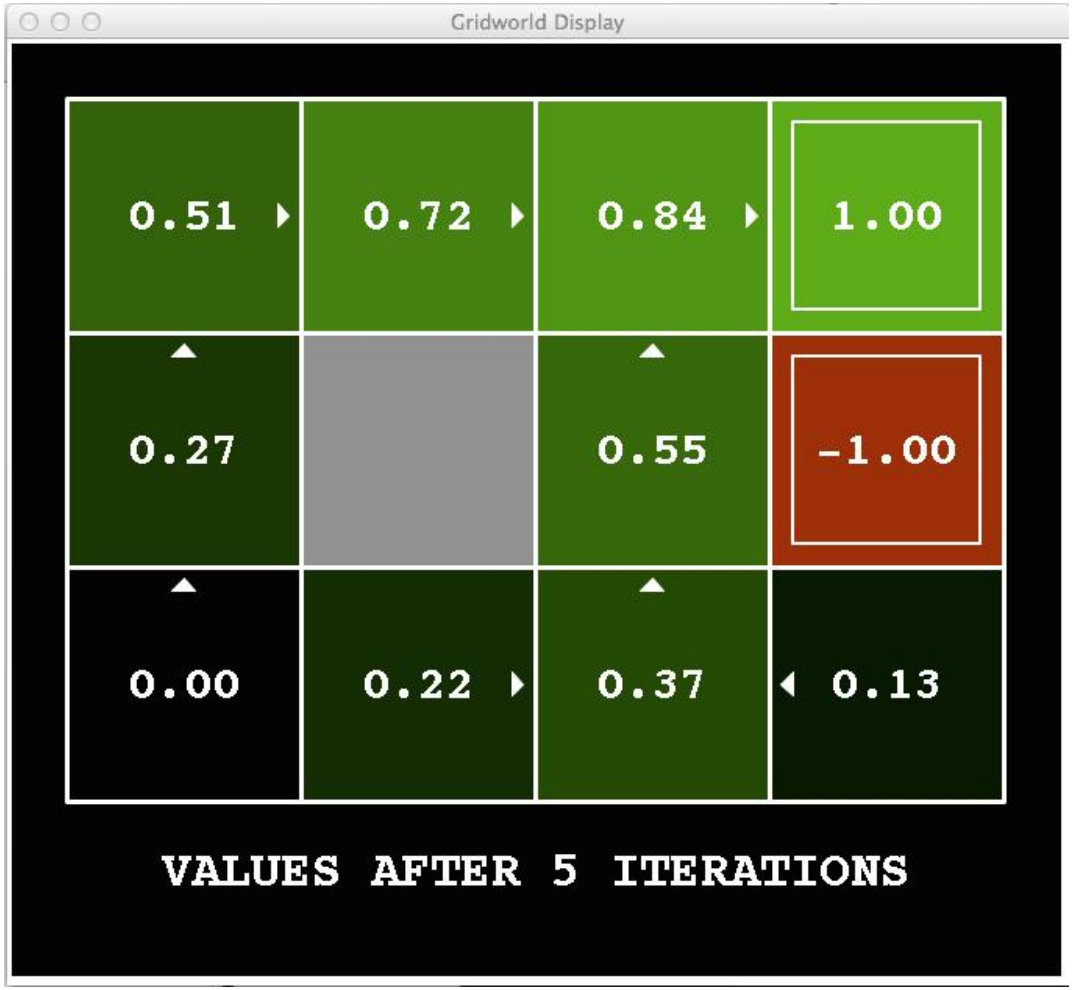
Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$



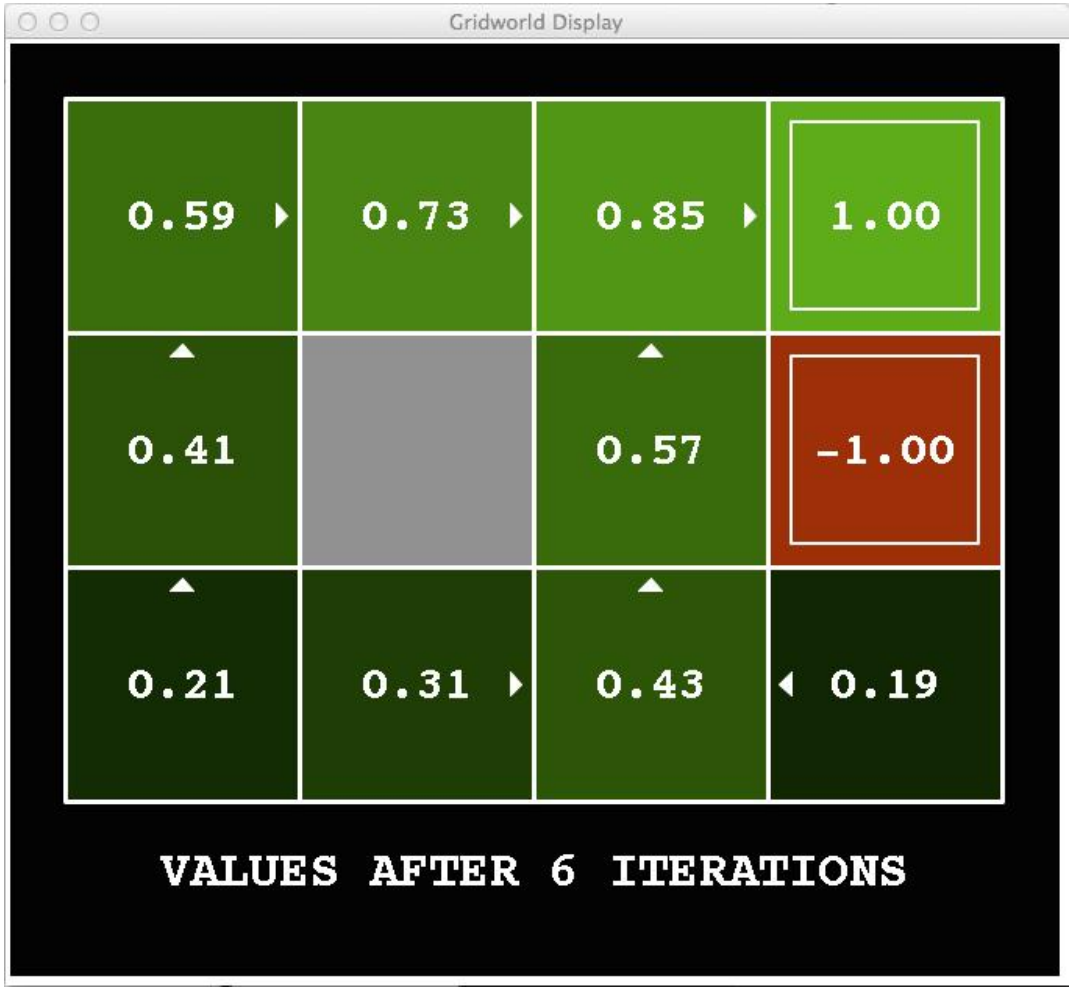
Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$



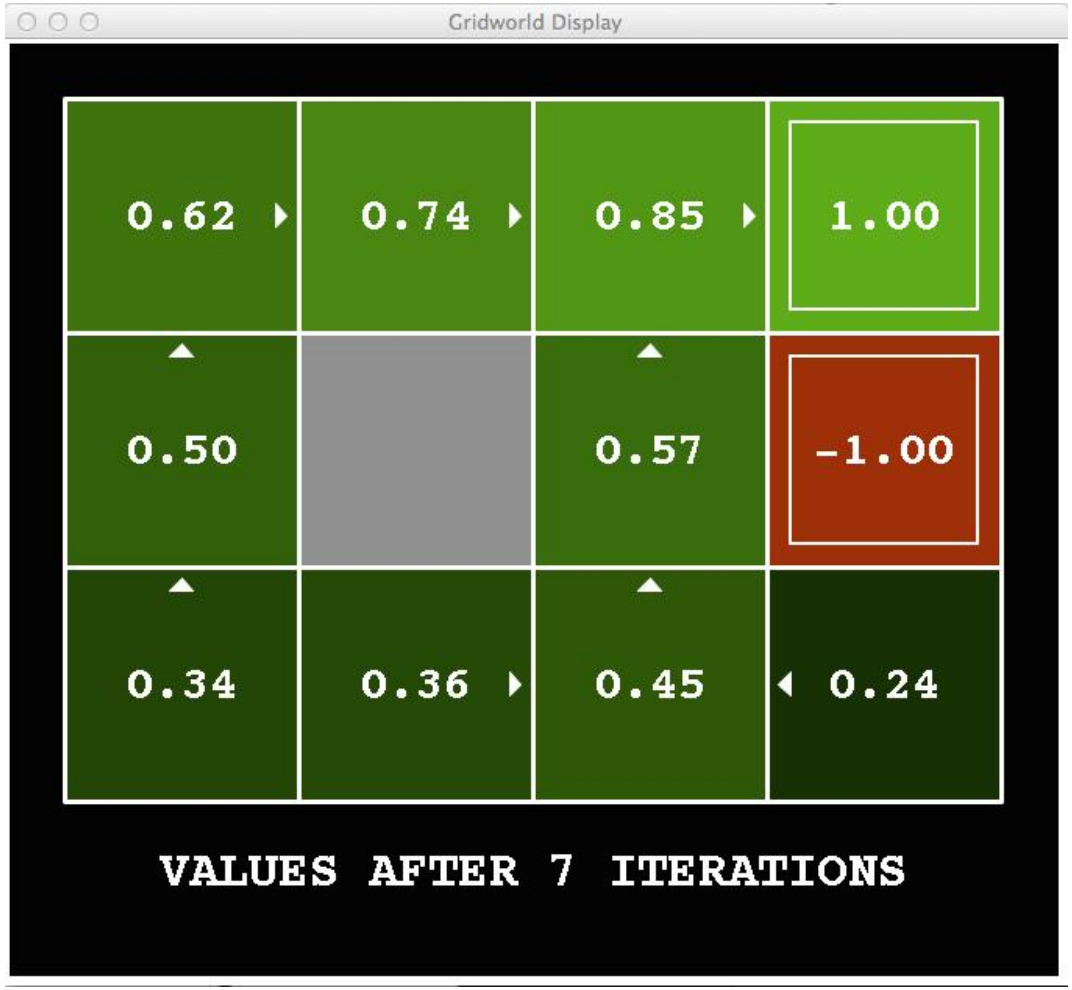
Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$



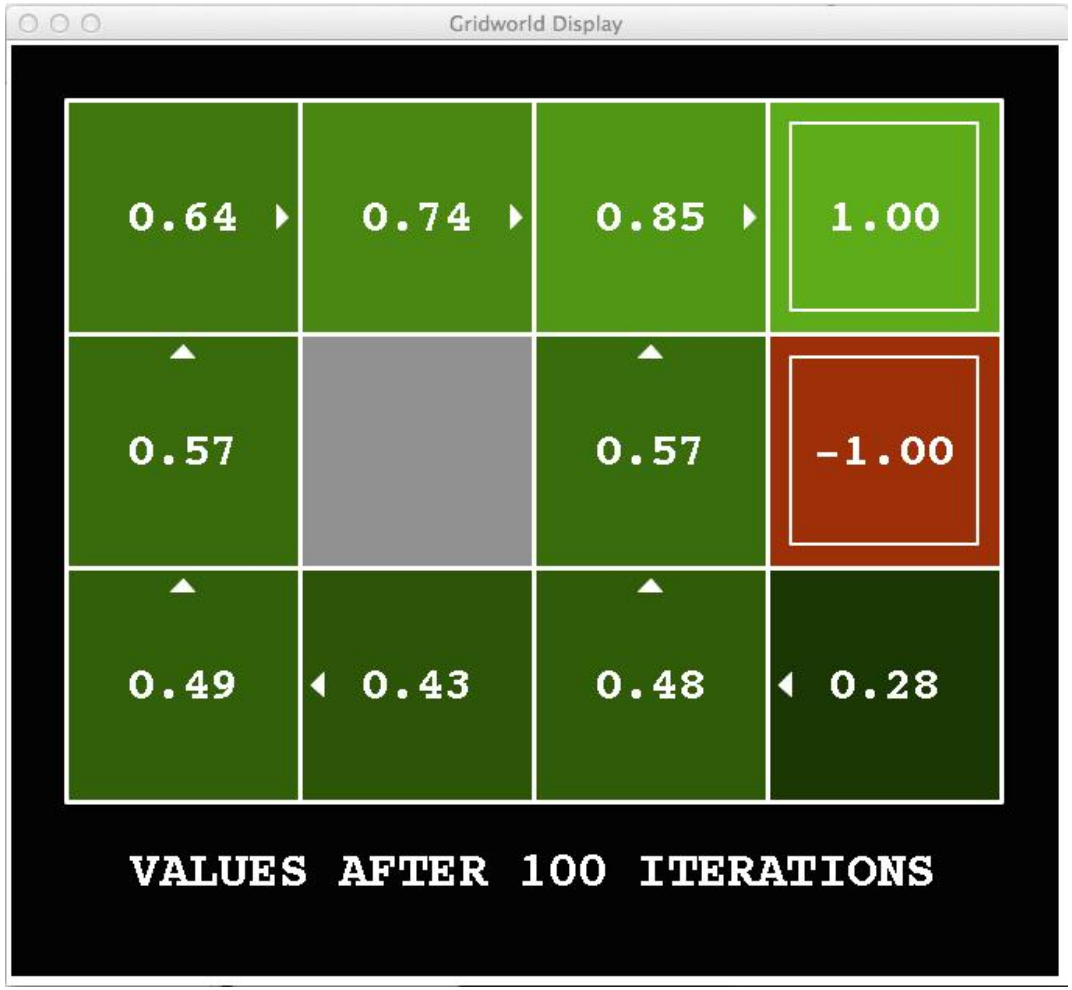
Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代

$$U_{i+1}(s) \leftarrow \max_{a \in A(s)} \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U_i(s')],$$

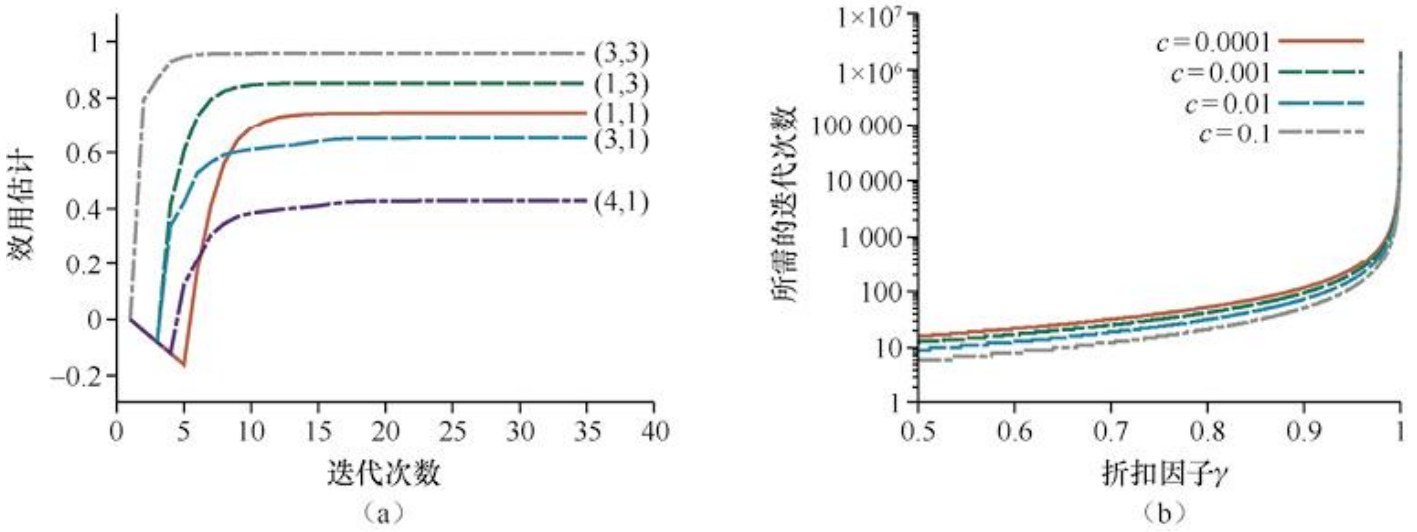


Noise = 0.2
Discount = 0.9
Living reward = 0

MDP的算法

南京大学 NANJING UNIVERSITY

价值迭代



(a) 显示被选择状态使用价值迭代的效用演变的图。 (b) 对于c不同的值，为保证误差最多为$\varepsilon = c \cdot R_{max}$所需的价值迭代次数，作为折扣因子$\gamma$的函数

MDP的算法

南京大学 NANJING UNIVERSITY

策略迭代

- 从某个初始策略π_0开始，交替进行以下两个步骤:
  - **策略评估:** 给定策略π_i, 计算 $U_i = U^{\pi_i}$,即执行π_i后每个状态的效用;
  - **策略改进:** 使用基于U_i的一步前瞻，计算新的MEU策略π_{i+1}

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma U(s')]$$

- 如果策略改进步骤对效用不产生任何改变，则算法终止。

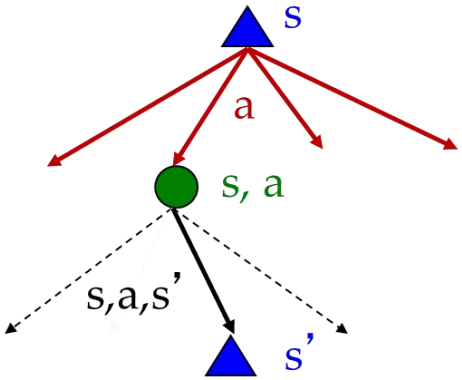
MDP的算法

策略迭代

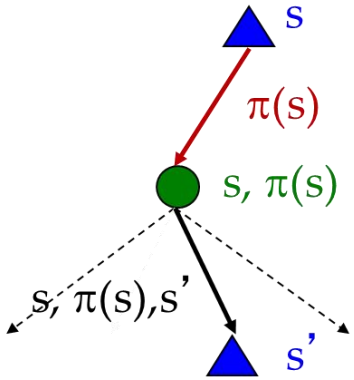
- 从某个初始策略π_0开始，交替进行以下两个步骤:
  - **策略评估:** 给定策略π_i, 计算 $U_i = U^{\pi_i}$,即执行π_i后每个状态的效用;
  - **策略改进:** 使用基于U_i的一步前瞻，计算新的MEU策略π_{i+1}

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma U(s')]$$

- 如果策略改进步骤对效用不产生任何改变，则算法终止。
- 在策略评估中，每个状态中的动作都是由策略确定的. 在第i次迭代时，策略π_i 指定了状态s下的动作 $\pi_i(s)$



执行最优动作



执行策略π指定的动作

MDP的算法

南京大学 NANJING UNIVERSITY

策略迭代

- 从某个初始策略π_0开始，交替进行以下两个步骤:
  - **策略评估:** 给定策略π_i, 计算 $U_i = U^{\pi_i}$,即执行π_i后每个状态的效用;
  - **策略改进:** 使用基于U_i的一步前瞻，计算新的MEU策略π_{i+1}

$$\pi^*(s) = \underset{a \in A(s)}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)[R(s, a, s') + \gamma U(s')]$$

- 如果策略改进步骤对效用不产生任何改变，则算法终止。
- 在策略评估中，每个状态中的动作都是由策略确定的. 在第i次迭代时, 策略π_i 指定了状态s下的动作 $\pi_i(s)$
- 简化版的贝尔曼方程将π_i下s的效用与其邻居的效用联系起来:

$$U_i(s) = \sum_{s'} P(s' \mid s, \pi_i(s))[R(s, \pi_i(s), s') + \gamma U_i(s')] .$$

- 对于大的状态空间，时间复杂度可能过高。简化贝尔曼更新修正策略迭代:

$$U_{i+1}(s) \leftarrow \sum_{s'} P(s' \mid s, \pi_i(s))[R(s, \pi_i(s), s') + \gamma U_i(s')] ,$$

MDP的算法

南京大学 NANJING UNIVERSITY

策略迭代

function POLICY-ITERATION(mdp) **returns** 一个策略
  inputs: mdp, 状态为S、动作为$A(s)$、转移模型为$P(s' \mid s, a)$的一个MDP
  local variables: U，一个S中状态的效用向量，初始为0
    π，一个由状态索引的策略向量，初始为随机

  repeat
    $U \leftarrow$ POLICY-EVALUATION(π, U, mdp)
    $unchanged? \leftarrow$ true
    for each 状态s **in** S **do**
      $a^* \leftarrow \underset{a \in A(s)}{\text{argmax}}$ Q-VALUE(mdp, s, a, U)
      if Q-VALUE(mdp, s, a^*, U) > Q-VALUE(mdp, s, $\pi[s]$, U) **then**
        $\pi[s] \leftarrow a^*$; $unchanged? \leftarrow$ **false**
  until $unchanged?$
  return π

计算最优策略的策略迭代算法

MDP的算法

南京大学 NANJING UNIVERSITY

线性规划

- 线性规划（linear programming）是一个表述约束优化问题的一般方法
- 对于每个状态s和动作a，对所有满足下式的s最小化$U(s)$

$$U(s) \geq \sum_{s'} P(s' \mid s,a)[R(s,a,s') + \gamma U(s')]$$

- 这样一来，我们就把动态规划和线性规划联系了起来.

老虎机问题

- **单臂老虎机（*one-armed bandit*）**是赌场里的一种投币机。赌徒可以投入一枚硬币，拉动杠杆，并收集奖金（如果有的话）。

- **n臂老虎机（*n-armed bandit*）**有n个杠杆。每个杠杆的背后是一个固定但未知的奖金概率分布，每次拉动杠杆都是从未知的分布中进行采样。

- 在连续的赌博中，赌徒必须在每次投币前决定拉动哪个杠杆

- 一个普遍存在于日常生活中的关于权衡的例子，即**利用**当前的最佳动作来获得奖励，还是**探索**之前未知的状态和动作来获得信息。

- n臂老虎机是在许多重要领域的真实问题的形式化模型，例如
  - 决定使用n种可能的新疗法中的哪种用来治疗一种疾病，
  - 采用n种可能的投资中的哪一种来投入你的部分储蓄，
  - 选择n个可能的研究项目中的哪一个进行资助

老虎机问题

南京大学 NANJING UNIVERSITY

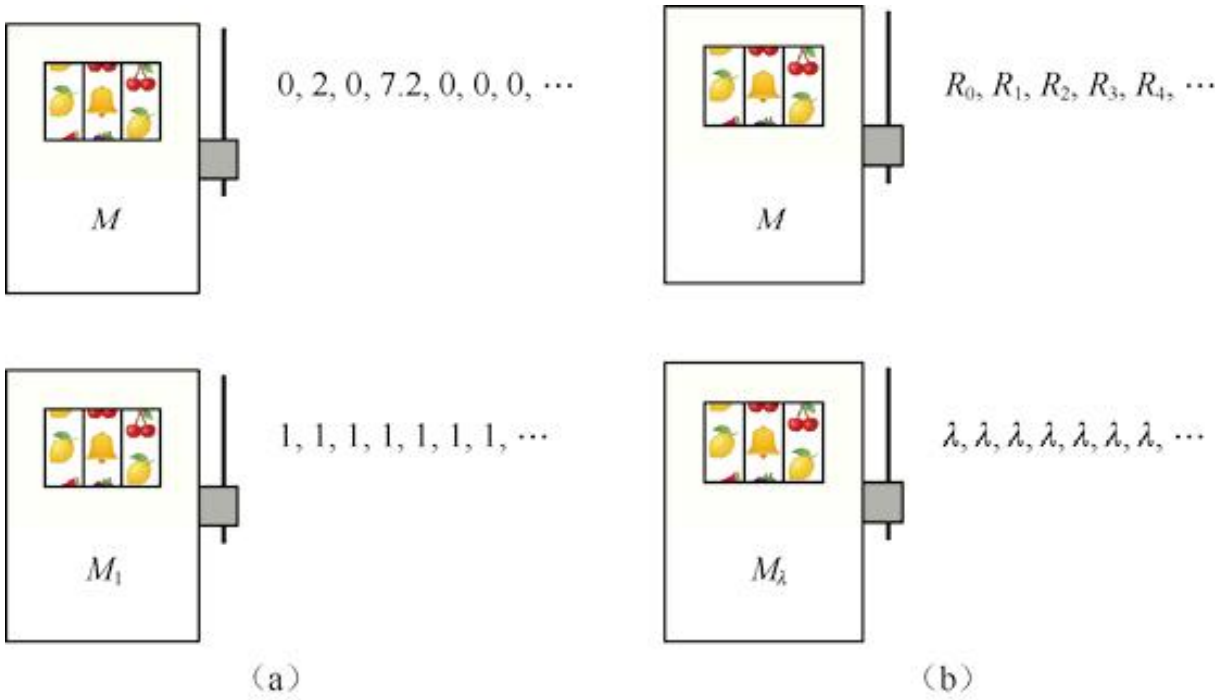
老虎机问题的定义

- 每个臂M_i是一个**马尔可夫奖励过程**（Markov reward process，MRP），也就是说，一个只有一个可能动作a_i的MDP。它拥有状态S_i，转移模型$P_i(s' \mid s, a_i)$，奖励$R_i(s, a_i, s')$。这个臂定义了奖励序列$R_{i,0}, R_{i,1}, R_{i,2}, \cdots$，的分布，其中每个$R_{i,t}$是一个随机变量。

- 整体老虎机问题是一个MDP：状态空间由笛卡儿积$\boldsymbol{S} = \boldsymbol{S}_1 \times \cdots \times \boldsymbol{S}_n$给出；动作为$a_1, \ldots, a_n$；转移模型更新被选择的臂$M_i$的状态，根据其指定的转移模型；剩下的臂保持不变；折扣因子为γ。

- 关键特性在于臂之间是独立的，并且同一时间只能有一个臂工作。

老虎机问题



（a）一个简单的有两个臂的确定性老虎机问题。臂可以以任何顺序拉动，每个臂产生所示的奖励序列。（b）关于（a）中的老虎机的一个更普遍的情况，其中第一个臂给出任意奖励序列，第二个臂给出一个固定的奖励

老虎机问题

设折扣为0.5，计算每个臂的效用（总折扣奖励）：

$$U(M) = (1.0\times 0)+(0.5\times 2)+(0.5^2\times 0)+(0.5^3\times 7.2)=1.9$$
$$U(M_1) = \sum_{t=0}^{\infty} 0.5^t = 2.0.$$

从M开始，然后在第四次奖励后切换到M_1，会得到序列$S = 0, 2, 0, 7.2, 1, 1, 1, \ldots$,则可以计算出效用为

$$U(S) = (1.0\times 0)+(0.5\times 2)+(0.5^2\times 0)+(0.5^3\times 7.2)+\sum_{t=4}^{\infty} 0.5^t = 2.025.$$

因此，在正确的时间从M转换到M_1的策略S比选择任何一种单独的臂都要好。

老虎机问题

- 最优策略将拉动臂M直到时间T，然后在剩余时间转换到M_λ
- 基廷斯指数：

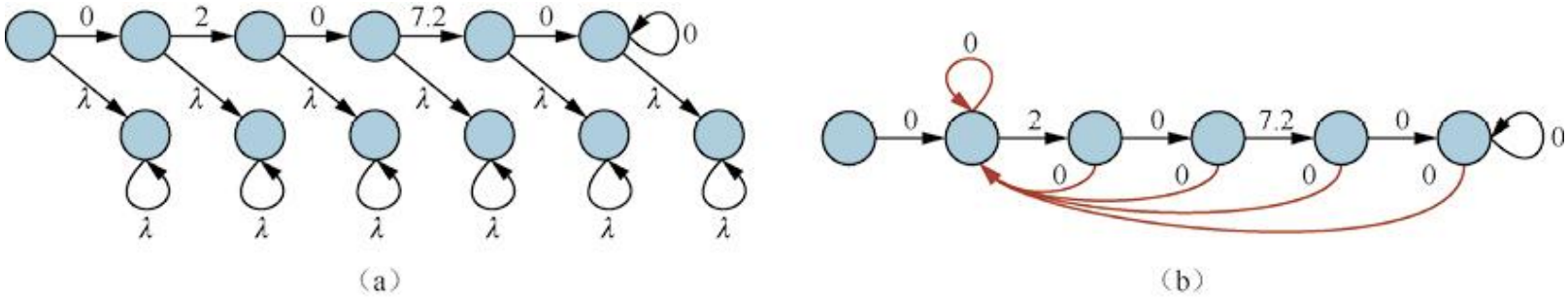
$$\max_{T>0} E\left[\left(\sum_{t=0}^{T-1}\gamma^t R_t\right)+\sum_{t=T}^{+\infty}\gamma^t\lambda\right]=\sum_{t=0}^{+\infty}\gamma^t\lambda$$

$$\lambda=\max_{T>0}\frac{E\left(\sum_{t=0}^{T-1}\gamma^t R_t\right)}{E\left(\sum_{t=0}^{T-1}\gamma^t\right)}$$

- 该值描述了每单位折扣时间可获得的最大效用

T	1	2	3	4	5	6
R_t	0	2	0	7.2	0	0
$\sum\gamma^t R_t$	0.0	1.0	1.0	1.9	1.9	1.9
$\sum\gamma^t$	1.0	1.5	1.75	1.875	1.9375	1.9687
ratio	0.0	0.6667	0.5714	1.0133	0.9806	0.9651

老虎机问题



（a）在每个点使用一个永久转向恒定臂的选择增广的奖励序列M = 0, 2, 0, 7.2, 0, 0, 0, …。（b）一个最优价值恰好等于（a）的最优价值的MDP，在每个点的最优策略在M与M_λ之间中立