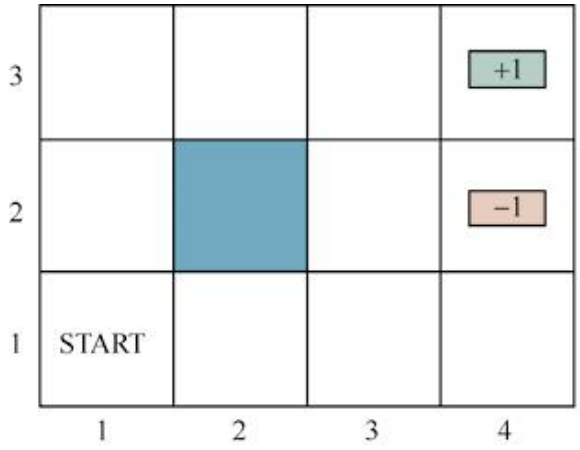
提纲

南京大学 NANJING UNIVERSITY

- **强化学习**
  - 被动强化学习
  - 主动强化学习
  - 强化学习中的泛化

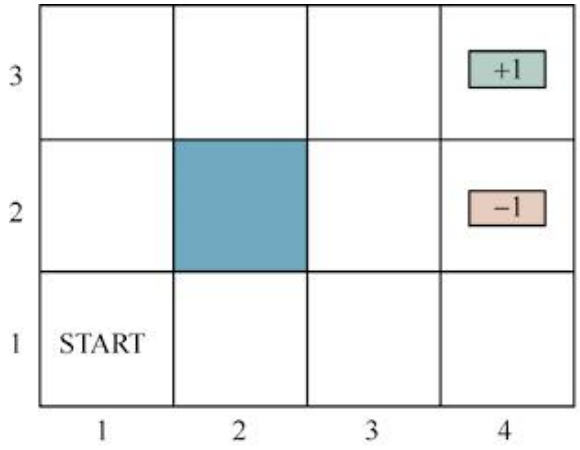
回顾：序贯决策问题

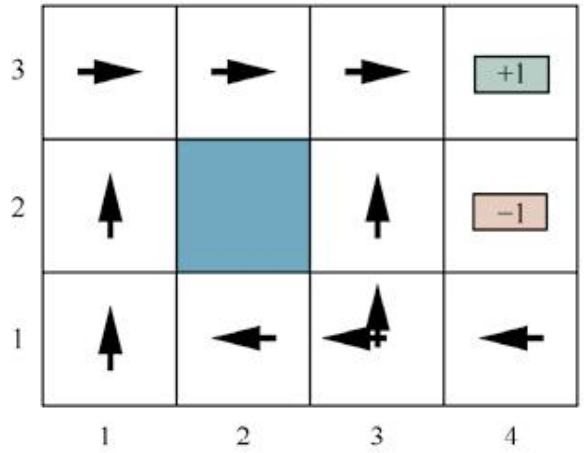


一个简单的、随机的4×3环境，它向智能体呈现一个序贯决策问题。环境的转移模型：“预期的”结果以0.8的概率出现，但在0.2的概率下，智能体以垂直于预期方向的角度运动。与墙的碰撞不会导致任何运动。转移到两种终止状态的（长时）奖励分别为+1和−1，所有其他转移的（即时）奖励为−0.04

目标：得到一个能最大化奖励的动作序列

回顾：序贯决策问题



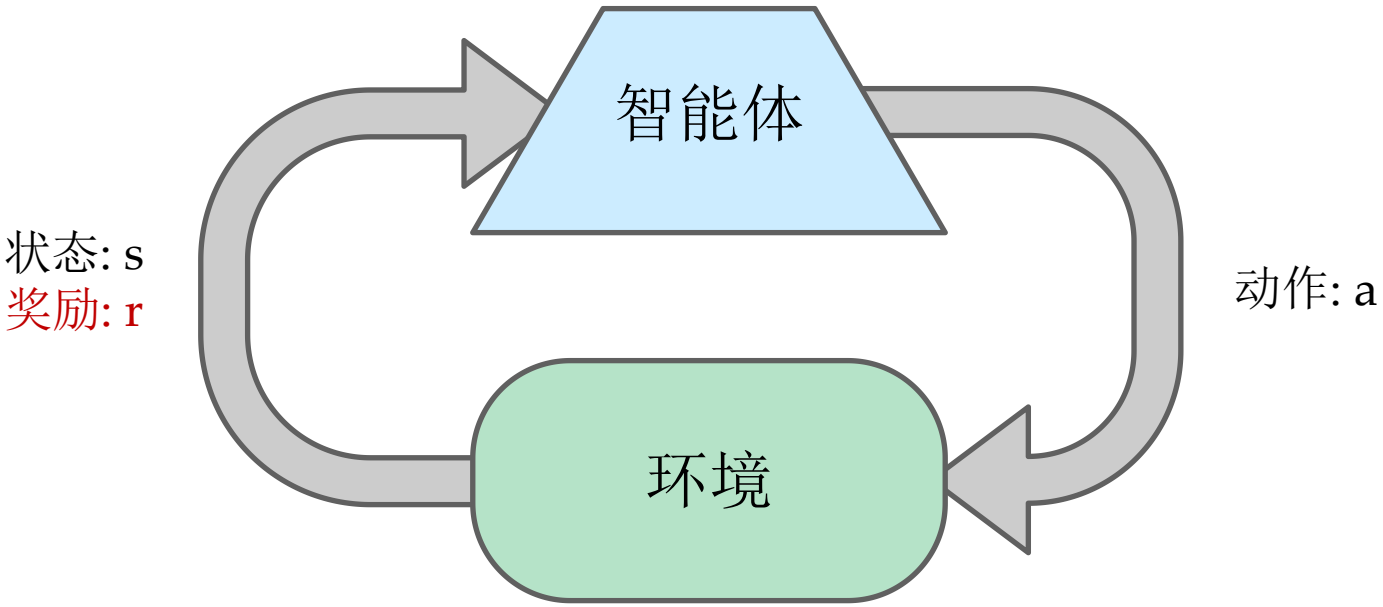


一个简单的、随机的4×3环境，它向智能体呈现一个序贯决策问题。环境的转移模型：“预期的”结果以0.8的概率出现，但在0.2的概率下，智能体以垂直于预期方向的角度运动。与墙的碰撞不会导致任何运动。转移到两种终止状态的（长时）奖励分别为+1和−1，所有其他转移的（即时）奖励为−0.04

目标：得到一个能最大化奖励的动作序列

回顾：序贯决策问题

- **基本思想:**
  - 以奖励的形式获取反馈
  - 智能体的效用定义为奖励函数
  - 需要学习动作序列去最大化期望奖励
  - 所有学习都是基于观测到的输出样本



轨迹序列： $s_1, a_1, r_1, s_2, a_2, r_2, s_3, a_3, r_3, \ldots$

回顾：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 $R(s, a, s)$;
  - 衰减因子 γ.

- **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

强化学习

- 依旧假设一个马尔可夫决策过程 (MDP):
  - 状态集合 $s \in S$
  - 每个状态的动作集合 A
  - 转移模型 T(s,a,s′)
  - 奖励函数 R(s,a,s′)

- 依旧是搜寻策略 π(s)

- 不知道 T 或者 R
  - 我们不知道哪些状态是好的，也不知道这些行为的作用是什么
  - 需要实际尝试行动和状态来进行学习

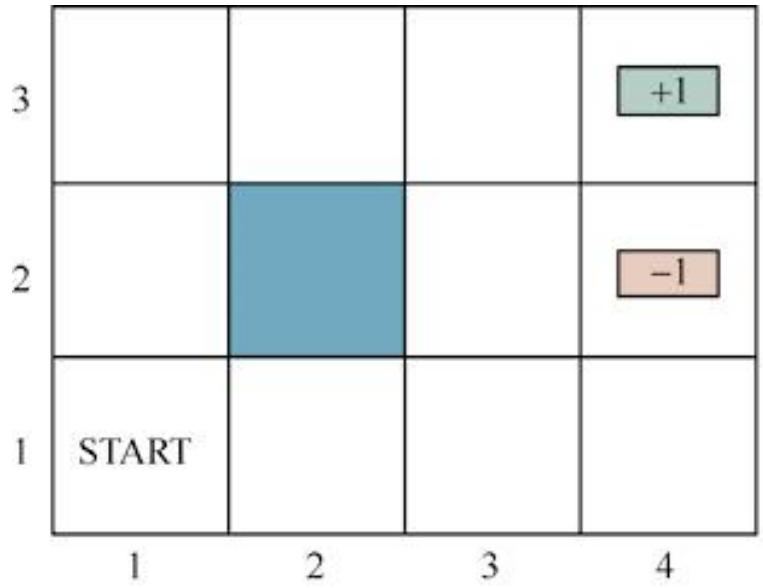
- 奖励的期望总和

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

被动强化学习

南京大学 NANJING UNIVERSITY

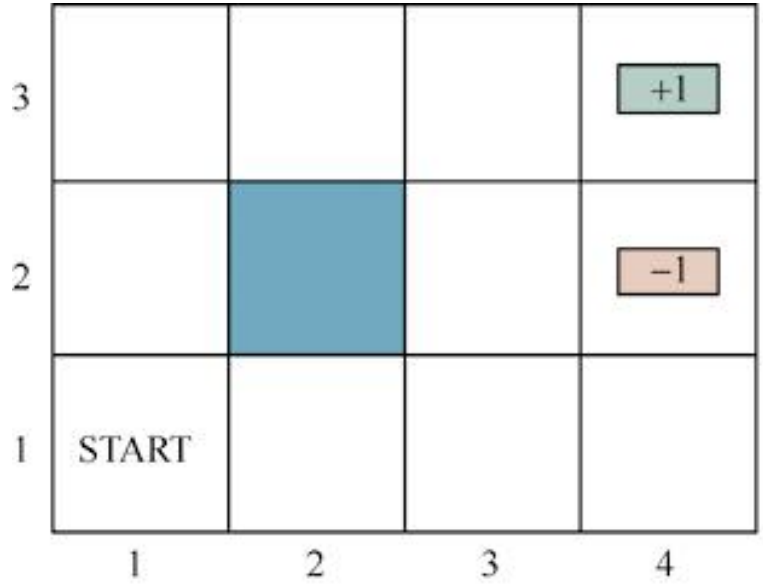
无模型学习



被动强化学习

南京大学 NANJING UNIVERSITY

无模型学习



$$(1,1)\xrightarrow[Up]{-0.04}(1,2)\xrightarrow[Up]{-0.04}(1,3)\xrightarrow[Right]{-0.04}(1,2)\xrightarrow[Up]{-0.04}(1,3)\xrightarrow[Right]{-0.04}(2,3)\xrightarrow[Right]{-0.04}(3,3)\xrightarrow[Right]{+1}(4,3)$$

$$(1,1)\xrightarrow[Up]{-0.04}(1,2)\xrightarrow[Up]{-0.04}(1,3)\xrightarrow[Right]{-0.04}(2,3)\xrightarrow[Right]{-0.04}(3,3)\xrightarrow[Right]{-0.04}(3,2)\xrightarrow[Up]{-0.04}(3,3)\xrightarrow[Right]{+1}(4,3)$$

$$(1,1)\xrightarrow[Up]{-0.04}(1,2)\xrightarrow[Up]{-0.04}(1,3)\xrightarrow[Right]{-0.04}(2,3)\xrightarrow[Right]{-0.04}(3,3)\xrightarrow[Right]{-0.04}(3,2)\xrightarrow[Up]{-1}(4,2)$$

被动强化学习

南京大学 NANJING UNIVERSITY

计算班上学生的平均年龄

已知模型 P(A)

$$E[A] = \sum_a P(a) \cdot a \quad = 0.35 \times 20 + \ldots$$

模型 P(A)未知，但是有年龄样本[a_1, a_2, … a_N]

未知 P(A): “基于模型”

$$\hat{P}(a) = \frac{\text{num}(a)}{N}$$

$$E[A] \approx \sum_a \hat{P}(a) \cdot a$$

被动强化学习

南京大学 NANJING UNIVERSITY

计算班上学生的平均年龄

已知模型 P(A)

$$E[A] = \sum_a P(a) \cdot a \quad = 0.35 \times 20 + \ldots$$

模型 P(A)未知，但是有年龄样本[a_1, a_2, … a_N]

未知 P(A): “基于模型”

$$\hat{P}(a) = \frac{\text{num}(a)}{N}$$

$$E[A] \approx \sum_a \hat{P}(a) \cdot a$$

未知 P(A): “无模型”

$$E[A] \approx \frac{1}{N} \sum_i a_i$$

被动强化学习：如何从经验中学习

南京大学 NANJING UNIVERSITY

基于模型的策略评估

- 基于模型的思想:
  - 通过经验估计MDP的近似模型
  - 假设学习到的模型是正确的，并用其来计算效用值

- 步骤1: 学习经验性的MDP模型
  - 记录每一组(s, a)的输出
  - 通过归一化获得转移概率的经验估计 $\hat{T}(s,a,s')$
  - 当经历 (s, a, s')时，记录对应的奖励值 $\hat{R}(s,a,s')$

- 步骤2: 求解学习到的MDP
  - 例如, 值迭代

被动强化学习

南京大学 NANJING UNIVERSITY

基于模型的策略评估

function PASSIVE-ADP-LEARNER(*percept*) **returns** 一个动作

inputs: *percept*，指示当前状态s'与奖励信号r的某个感知

persistent: π，一个确定性策略

mdp，一个转移模型为P、奖励为R、动作为A、折扣因子为γ的MDP

U，关于状态效用的表，初始化为空

$N_{s'|s,a}$，一个表，用于某对状态与动作对应的结果状态出现次数的向量，初始为零

s, a，之前的状态与动作，初始为空

if s'是一个新的状态 **then** $U[s'] \leftarrow 0$

if s非空 **then**

增加$N_{s'|s,a}[s, a][s']$

$R[s, a, s'] \leftarrow r$

将a加入$A[s]$

$\boldsymbol{P}(\cdot \mid s, a) \leftarrow$ NORMALIZE($N_{s'|s,a}[s, a]$)

$U \leftarrow$ POLICYEVALUATION(π, U, mdp)

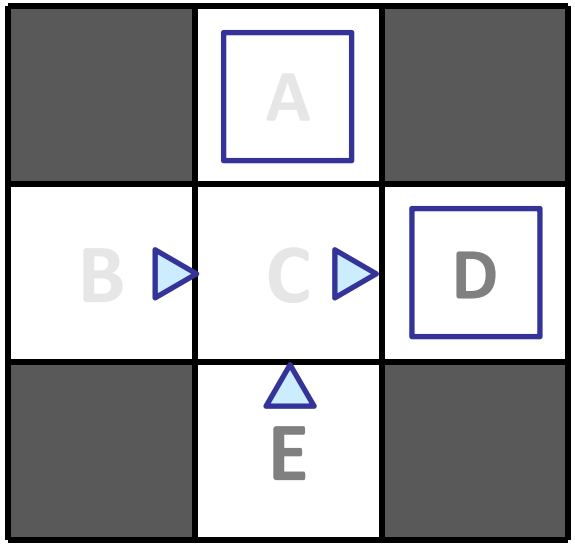
$s, a \leftarrow s', \pi[s']$

return a

基于自适应动态规划的被动强化学习智能体。智能体将选定一个值，然后逐步计算MDP中的P与R值。其中Policy-Evaluation函数将求解固定策略的贝尔曼方程，正如17.2节中所述

基于模型的策略评估

输入策略π



假设: $\gamma = 1$

观测到的样本/轨迹 (训练)

样本 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

样本2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

样本3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

样本4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

被动强化学习

南京大学 NANJING UNIVERSITY

基于模型的策略评估

观测到的样本/轨迹 (训练)

样本 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

样本2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

样本3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

样本4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

	A	B	C	D	E	x	总数
(A, exit)						1.0	1
(B, east)			1.0				2
(C, east)	.25			.75			4
(D, exit)						1.0	3
(E, north)			1.0				2

转移	奖励	总奖励	效用
(B, east, C)	-1, -1	2	-1
(C, east, D)	-1, -1, -1	3	-1
(D, exit, x)	10, 10, 10	3	10
(E, north, C)	-1, -1	2	-1
(C, east, A)	-1	1	-1
(A, exit, x)	-10	1	-10

学习到的模型

$\hat{T}(s,a,s')$

T(B, east, C) = 1.00
T(C, east, D) = 0.75
T(C, east, A) = 0.25
...

$\hat{R}(s,a,s')$

R(B, east, C) = -1
R(C, east, D) = -1
R(D, exit, x) = +10
...

被动强化学习

南京大学 NANJING UNIVERSITY

直接评估

- **目标:** 计算在π下每一个状态的值

- **思路:** 对观察到的样本值求平均
  - 根据 π 来做出动作
  - 每次访问一个状态, 记录获得的折扣奖励的总和
  - 对这些样本进行平均

被动强化学习

南京大学 NANJING UNIVERSITY

直接评估

初始化 $N(s) = 0, \quad G(s) = 0, \quad \forall s \in S$

循环

- *采样轨迹* $i = s_{i,1}, a_{i,1}, r_{i,1}, s_{i,2}, a_{i,2}, r_{i,2}, \cdots, s_{i,T_i}$
- *定义时间点t开始的第i条轨迹的回报为*
  - $G_{i,t} = r_{i,t} + \gamma r_{i,t+1} + \gamma^2 r_{i,t+2} + \cdots + \gamma^{T_i-1} r_{t,T_i}$
- *对轨迹i中的每个状态* s *，当轨迹i中t时间首次访问到状态*
  - *首次访问计数加 1:* $N(s) = N(s) + 1$
  - *累加回报* $G(s) = G(s) + G_{i,t}$
  - *更新估计* $V^{\pi}(s) = G(s)/N(s)$

南京大学 NANJING UNIVERSITY

直接评估

观测到的样本/轨迹 (训练)

样本 1

B, east, C, -1
C, east, D, -1
D, exit, x, +10

样本2

B, east, C, -1
C, east, D, -1
D, exit, x, +10

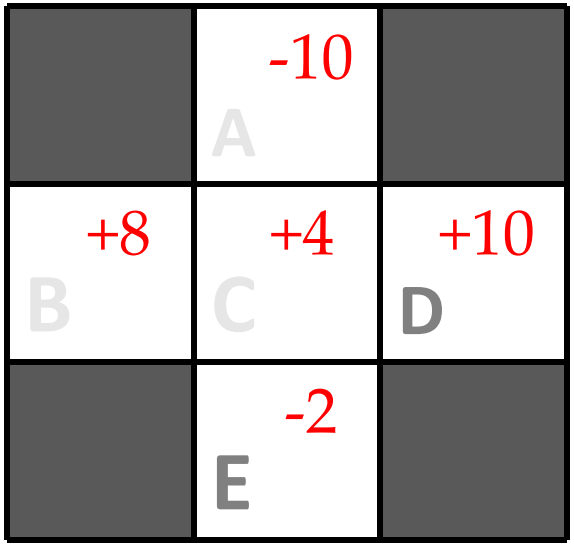
样本3

E, north, C, -1
C, east, D, -1
D, exit, x, +10

样本4

E, north, C, -1
C, east, A, -1
A, exit, x, -10

输出值



T1:

$$v^{\pi}(B) = G_{1,1} = (-1) + \gamma(-1) + \gamma * \gamma * (+10) = 8$$
$$v^{\pi}(C) = G_{1,2} = (-1) + \gamma * (+10) = 9$$
$$v^{\pi}(D) = G_{1,3} = +10 = 10$$

T2:

$$v^{\pi}(B) = G_{2,1} = 8; v^{\pi}(C) = G_{2,2} = 9; v^{\pi}(D) = G_{2,3} = 10$$

T3:

$$v^{\pi}(E) = 8; v^{\pi}(C) = 9; v^{\pi}(D) = 10$$

T4:

$$v^{\pi}(E) = -12; v^{\pi}(C) = -11; v^{\pi}(A) = -10$$

直接评估

- 直接评估方法的优点
  - 便于理解
  - 不需要关于T, R的任何知识
  - 样本数足够时可以保证收敛性
- 直接评估方法的缺点
  - 没有利用到状态之间的联系
  - 每一个状态必须单独学习
  - 因此算法收敛的非常缓慢

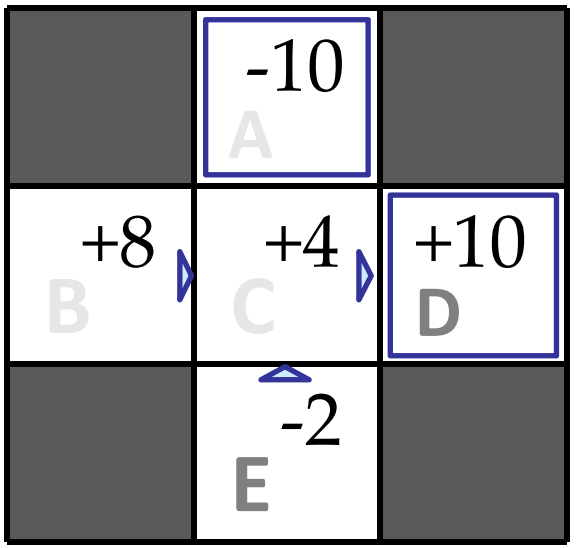
被动强化学习

南京大学 NANJING UNIVERSITY

直接评估

- 直接评估方法的优点
  - 便于理解
  - 不需要关于T, R的任何知识
  - 样本数足够时可以保证收敛性
- 直接评估方法的缺点
  - 没有利用到状态之间的联系
  - 每一个状态必须单独学习
  - 因此算法收敛的非常缓慢

输出值



B和E在这种策略下都是走向C, 但为什么他呢的估值却又很大不同？

无模型的策略评估

- 通过策略评估提升我们对 V 的估计:

$$V_{k+1}^{\pi}(s) \leftarrow \sum_{s'} T(s, \pi(s), s')[R(s, \pi(s), s') + \gamma V_k^{\pi}(s')]$$

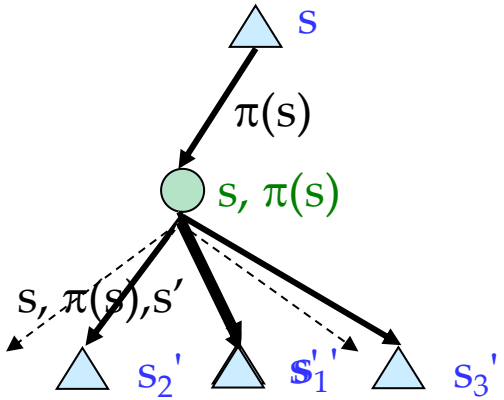
- 思路: 采样所有到达 s' 的转移并取平均

$$sample_1 = R(s, \pi(s), s_1') + \gamma V_k^{\pi}(s_1')$$

$$sample_2 = R(s, \pi(s), s_2') + \gamma V_k^{\pi}(s_2')$$

$$\cdots$$

$$sample_n = R(s, \pi(s), s_n') + \gamma V_k^{\pi}(s_n')$$



$$V_{k+1}^{\pi}(s) \leftarrow \frac{1}{n} \sum_i sample_i$$

时序差分策略评估

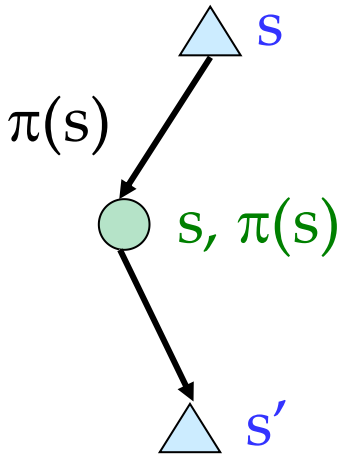
- 从每一次的经验中学习
  - 每当我们经历了一个从状态s出发的转移 (s, a, s', r)，更新 V(s)
  - 转移概率更大的状态 s' 的值对s值更新的影响更大

- 状态值的时序差分学习
  - 将状态值朝着后继出现的状态值靠近: 滑动平均

采样V(s): $sample = R(s, \pi(s), s') + \gamma V^{\pi}(s')$

更新V(s): $V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + (\alpha)sample$

重写成滑动平均的形式: $V^{\pi}(s) \leftarrow V^{\pi}(s) + \alpha(sample - V^{\pi}(s))$



被动强化学习

南京大学 NANJING UNIVERSITY

时序差分策略评估

- *目标：给定根据策略π生成的轨迹估计* $V^{\pi}(s)$
  - $s_1, a_1, r_1, s_2, a_2, r_2, \cdots$ ，*动作根据π采样得到*
- *朴素的TD学习：根据样本值（目标值）来更新值*

$$V^{\pi}(s_t) = V^{\pi}(s_t) + \alpha([r_t + \gamma V^{\pi}(s_{t+1})] - V^{\pi}(s_t))$$

- *TD目标：* $[r_t + \gamma V^{\pi}(s_{t+1})]$
- *TD误差：* $\delta_t = r_t + \gamma V^{\pi}(s_{t+1}) - V^{\pi}(s_t)$

- *在获得* $(s,\ a,\ r, s')$ *元组之后可立即更新*

被动强化学习

南京大学 NANJING UNIVERSITY

时序差分策略评估

输入:α

初始化$V^{\pi}(s) = 0, \quad \forall s \in S$

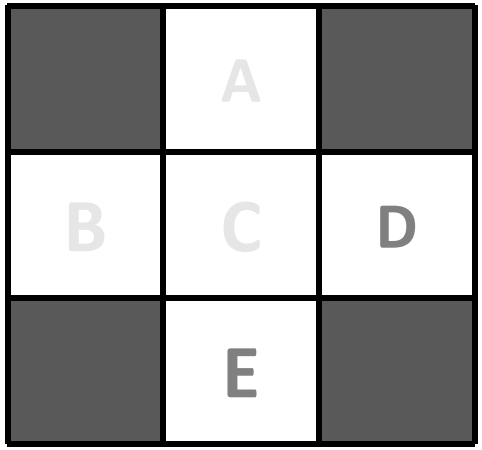
循环

- *采样元组*$(s_t, \quad a_t, \quad r_t, s_{t+1})$
- $V^{\pi}(s_t) = V^{\pi}(s_t) + \alpha([r_t + \gamma V^{\pi}(s_{t+1})] - V^{\pi}(s_t))$

一种使用时序差分方法学习效用估计的被动强化学习智能体。
我们选择适当的步长函数以确保收敛

时序差分策略评估

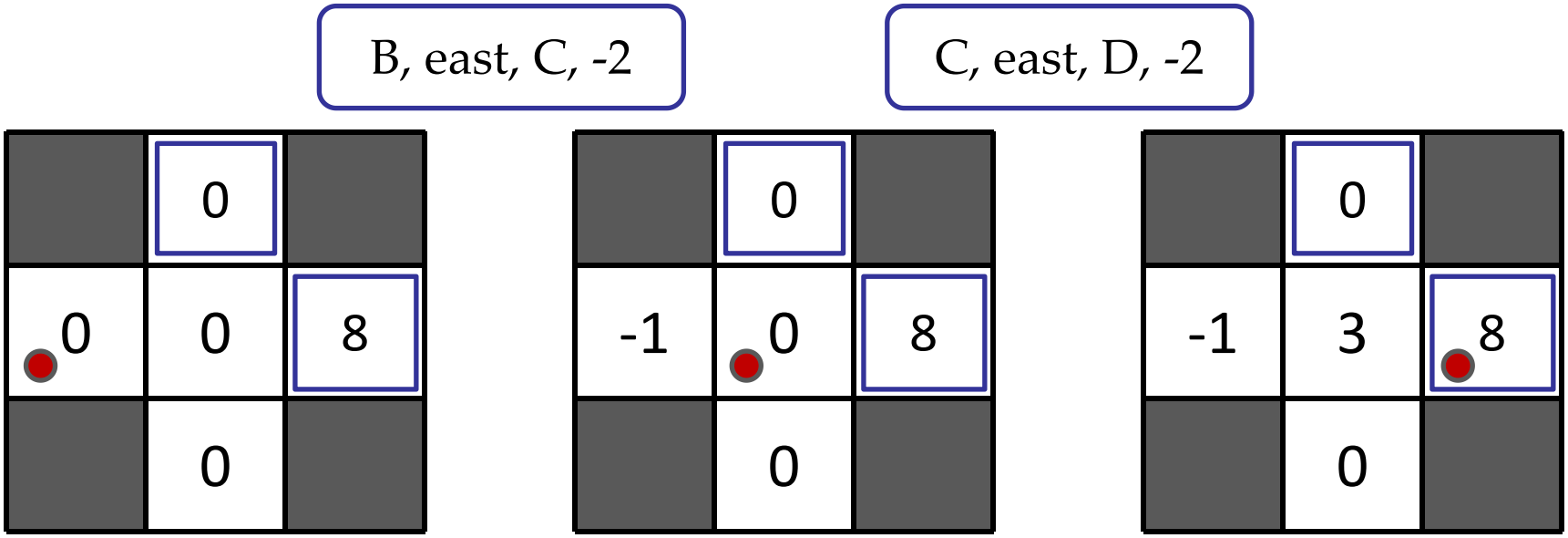
状态



假设: $\gamma = 1$, $\alpha = 1/2$

时序差分策略评估

观测到的转移



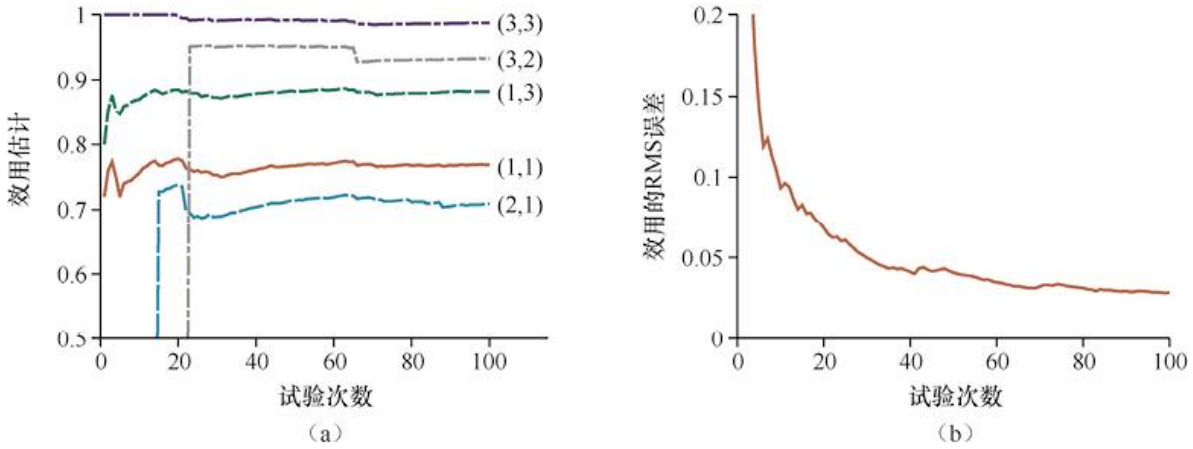
假设: $\gamma = 1$, $\alpha = 1/2$

$$V^{\pi}(s) \leftarrow (1-\alpha)V^{\pi}(s) + \alpha\left[R(s,\pi(s),s') + \gamma V^{\pi}(s')\right]$$

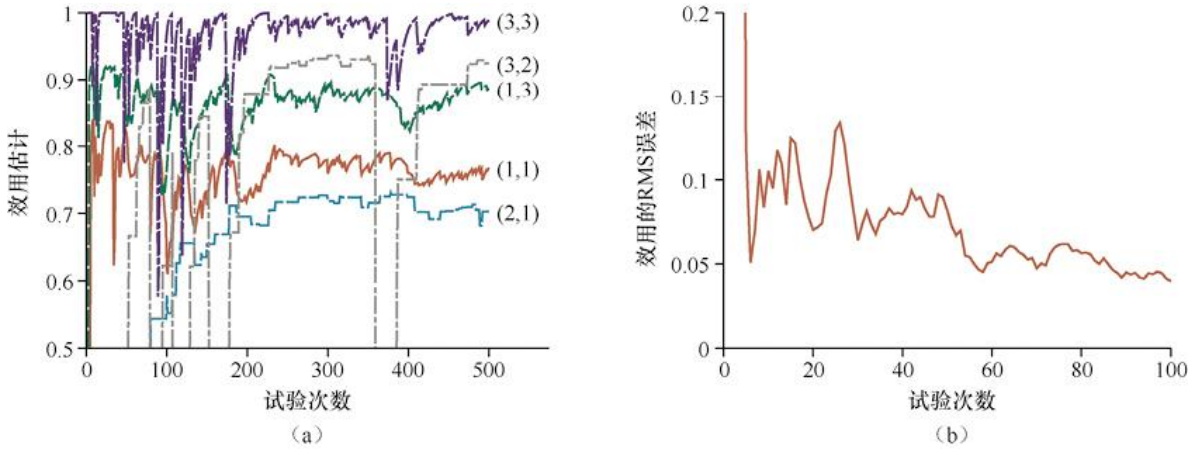
被动强化学习

南京大学 NANJING UNIVERSITY

自适应动态规划与时序差分算法比较



4×3世界问题的被动自适应动态规划的学习曲线。



4×3世界问题中，时序差分的学习曲线。

被动强化学习

南京大学 NANJING UNIVERSITY

直接评估（蒙特卡洛）与时序差分算法比较

- 对于轨迹 $s_1, a_1, r_1, s_2, a_2, r_2, \ldots, s_T$
- 都通过采样对状态值进行更新

$$V^{\pi}(s) = V^{\pi}(s) + \alpha(v^{\pi}(s) - V^{\pi}(s))$$

- 蒙特卡洛（MC）

$$v^{\pi}(s) = \sum_{k=0}^{T-1} \gamma^k r_{t+k} | S_t = s$$

- 时序差分（TD）

$$v^{\pi}(s) = r_t + \gamma V^{\pi}(S_{t+1}) | S_t = s$$

- 在TD中，使用(s，a，r，s’）来更新V(S)。
- 每次更新需要O(1)步操作,在一个长度为L的轨迹中，需要O(L)步。
- 在MC中需要等待轨迹终结,因此也是O(L)。
- TD能够利用马尔可夫结构。

回顾：MDP的算法

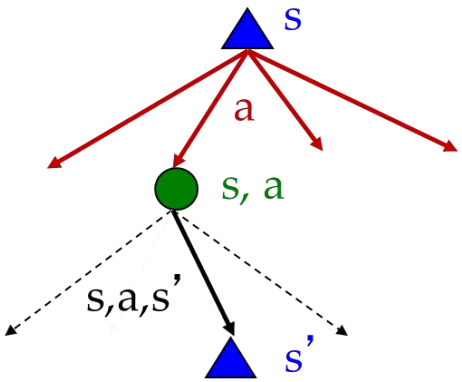
南京大学 NANJING UNIVERSITY

策略迭代

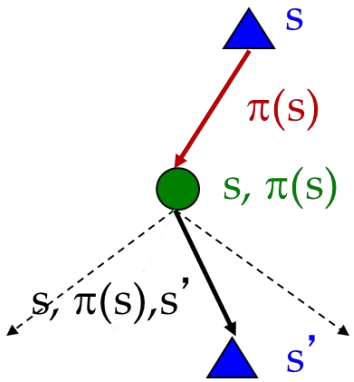
- 从某个初始策略π_0开始，交替进行以下两个步骤:
  - **策略评估:** 给定策略π_i, 计算 $U_i = U^{\pi_i}$,即执行π_i后每个状态的效用;
  - **策略改进:** 使用基于U_i的一步前瞻，计算新的MEU策略π_{i+1}

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

- 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 $\pi_i(s)$



执行最优动作



执行策略π指定的动作

主动强化学习

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。

- 在模型未知的情况下，如何使用样本实现策略迭代？

主动强化学习

策略迭代

- 根据策略 π_2 采样轨迹。
  - $s_1, a_1, r_1, s_2, a_2, r_2, \cdots, s_T$
- 计算策略 π_1 下的状态值。
  - $Q^{\pi_1}(s, a)$

- 行为策略 π_2 ：训练轨迹收集。
- 目标策略 π_1 ：进行状态值估计的策略。

主动强化学习

南京大学 NANJING UNIVERSITY

策略迭代

- 行为策略 π_2 ：训练轨迹收集。
- 目标策略 π_1 ：进行状态值估计的策略。

- 同策略（On-policy）学习（ $\pi_1 = \pi_2$）
  - 使用由相同策略采集的样本对目标策略进行状态值估计。

- 异策略（Off-policy）学习（ $\pi_1 \neq \pi_2$）
  - 使用由不同策略采集的样本对目标策略进行状态值估计。
  - 当轨迹采样需要的代价很高，可采用异策略方法。
  - 希望使用旧策略采集的样本进行策略评估，可采用异策略方法。

主动强化学习

南京大学 NANJING UNIVERSITY

时序差分Q学习

- 使用TD方法进行模型无关的策略迭代
- 初始化策略π
- 循环：
  - 策略评估：基于TD策略评估方法计算 $Q^{\pi}(s,a)$，使用 ε- 贪婪策略。
  - 策略提升：将π设置成ε-贪婪策略，使用和MC方法相同的算法更新 π。

主动强化学习

南京大学 NANJING UNIVERSITY

时序差分Q学习

1: Set initial ϵ-greedy policy π, $t = 0$, initial state $s_t = s_0$
2: Take $a_t \sim \pi(s_t)$ // Sample action from policy
3: Observe (r_t, s_{t+1})
4: **loop**
5: Take action $a_{t+1} \sim \pi(s_{t+1})$
6: Observe (r_{t+1}, s_{t+2})
7: $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r_t + \gamma Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t))$
8: $\pi(s_t) = \arg\max_a Q(s_t, a)$ w.prob $1 - \epsilon$, else random
9: $t = t + 1$
10: **end loop**

SARSA算法，即状态（state）、动作（action）、奖励（reward）、状态（state）、动作（action）

主动强化学习

南京大学 NANJING UNIVERSITY

时序差分Q学习

- Q-Learning：使用目标转移状态对应的最佳Q状态，进行当前Q状态的值更新。$(s_1, a_1, r_1, s_2, argmax(s_2))$

$$Q[s, a] \leftarrow Q[s, a] + \alpha(N_{sa}[s, a])(r + \gamma \max_{a'} Q[s', a'] - Q[s, a])$$

- SARSA：使用当前策略进行目标转移状态的动作选取，并使用对应的Q状态，进行当前Q状态的值更新。$(s_1, a_1, r_1, s_2, a_2)$

$$Q(s,a) \leftarrow Q(s,a) + \alpha[R(s,a,s') + \gamma\, Q(s',a') - Q(s,a)]$$

主动强化学习

南京大学 NANJING UNIVERSITY

时序差分Q学习

1: Initialize $Q(s,a), \forall s \in S, a \in A$ $t=0$, initial state $s_t = s_0$
2: Set π_b to be ϵ-greedy w.r.t. Q
3: **loop**
4: Take $a_t \sim \pi_b(s_t)$ // Sample action from policy
5: Observe (r_t, s_{t+1})
6: $Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha(r_t + \gamma \arg\max_a Q(s_{t_1},a) - Q(s_t,a_t))$
7: $\pi(s_t) = \arg\max_a Q(s_t,a)$ w.prob $1-\epsilon$, else random
8: $t = t+1$
9: **end loop**

强化学习中的泛化

南京大学 NANJING UNIVERSITY

状态的泛化表示

- 到目前为止，我们假设效用函数和Q函数可以用表格的形式表示，其中每个状态有一个输出值

- 在现实场景中, 几乎不可能学习到所有状态的信息
  - 在训练当中太多的状态要去遍历
  - 太多的状态要存储在内存中

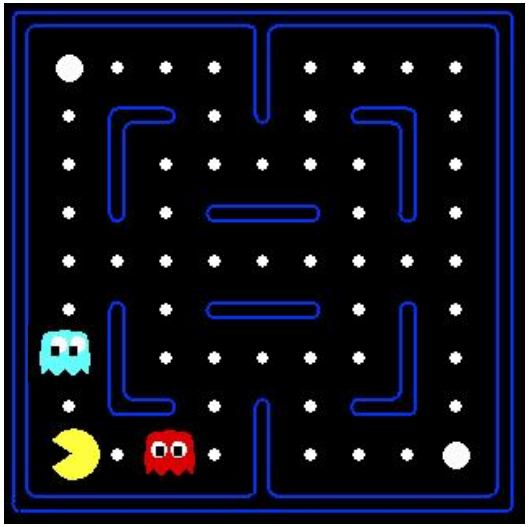
- 我们想要更紧凑的表示方式，可以概括状态或状态动作对

强化学习中的泛化

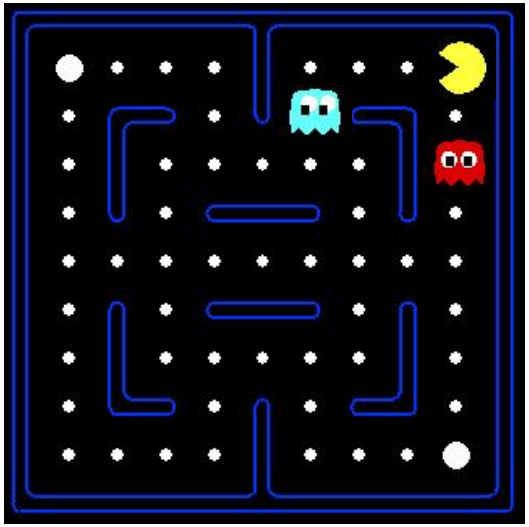
南京大学 NANJING UNIVERSITY

状态的泛化表示：吃豆人

我们从样本中发现这个状态的效用值比较低：



在朴素q学习中，我们对这个状态一无所知：



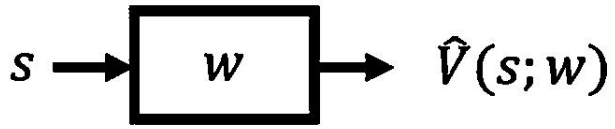
甚至对这个状态也一无所知：

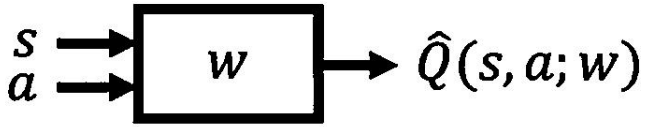
强化学习中的泛化

南京大学 NANJING UNIVERSITY

状态的泛化表示

- 使用含参数的函数来估计状态或者Q状态的值。从一个更大的空间（原始状态空间）映射到较小的空间（参数空间）。相似的状态在参数空间中的位置更靠近。





- 泛化表示的优势
  - 降低内存消耗
  - 减少对经验 / 采样的需求
  - 降低计算复杂度

强化学习中的泛化

函数近似

- 特征向量的线性组合形式:

$$V(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$$
$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

强化学习中的泛化

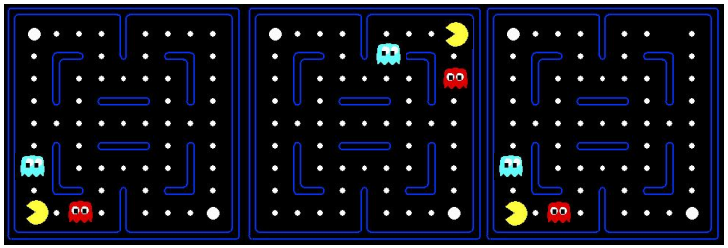
南京大学 NANJING UNIVERSITY

函数近似

- 特征向量的线性组合形式:

$$V(s) = w_1 f_1(s) + w_2 f_2(s) + \ldots + w_n f_n(s)$$
$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

- 特征是从状态到实数（通常为0/1）的映射，捕获状态的重要属性。
  - f_1：距离最近鬼怪的距离，2。
  - f_2：距离最近豆子的距离，1 。
  - f_3：鬼怪的个数，2 。
  - f_4：小黄人是否在单行通道中？ (0/1)，1 。
  - W：[1, 1, 1, 1]



$$\hat{V}(s;w) = w_1 * f_1(s) + w_2 * f_2(s) + w_3 * f_3(s) + w_4 * f_4(s)$$

强化学习中的泛化

南京大学 NANJING UNIVERSITY

近似时序差分Q学习

$$\theta_i \leftarrow \theta_i + \alpha[R(s,a,s') + \gamma \max_{a'} \hat{Q}_\theta(s',a') - \hat{Q}_\theta(s,a)] \frac{\partial \hat{Q}_\theta(s,a)}{\partial \theta_i}$$

- 使用线性Q函数的Q学习:

$$Q(s,a) = w_1 f_1(s,a) + w_2 f_2(s,a) + \ldots + w_n f_n(s,a)$$

transition $= (s, a, r, s')$

difference $= \left[r + \gamma \max_{a'} Q(s', a')\right] - Q(s, a)$

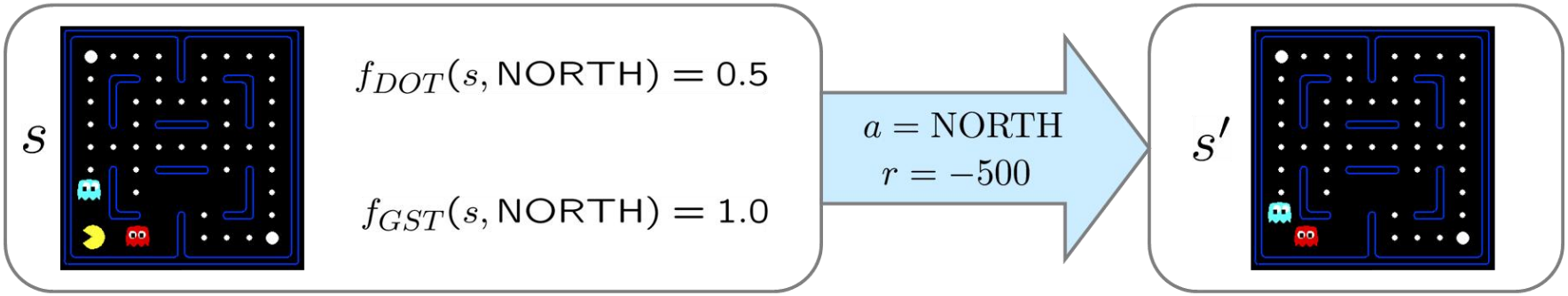
$Q(s,a) \leftarrow Q(s,a) + \alpha$ [difference] 精确Q值

$w_i \leftarrow w_i + \alpha$ [difference] $f_i(s,a)$ 近似Q值

强化学习中的泛化

近似时序差分Q学习：吃豆人

$$Q(s,a) = 4.0 f_{DOT}(s,a) - 1.0 f_{GST}(s,a)$$



$Q(s, \text{NORTH}) = +1$

$Q(s', \cdot) = 0$

$r + \gamma \max_{a'} Q(s', a') = -500 + 0$

difference $= -501$

$w_{DOT} \leftarrow 4.0 + \alpha\,[-501]\,0.5$

$w_{GST} \leftarrow -1.0 + \alpha\,[-501]\,1.0$

$$Q(s,a) = 3.0 f_{DOT}(s,a) - 3.0 f_{GST}(s,a)$$