



DeepL

订阅DeepL Pro以翻译大型文件。

欲了解更多信息, 请访问[www.DeepL.com/pro](http://www.DeepL.com/pro)。

# 强化学习 (RL) (RL)

第三章。

马尔科夫决策过程(MDPs)

动态编程（政策迭代、价值迭代和修正的政策迭代）。

赛义德-赛义德万德，博士。

## 在本章中。Chapter:

- ✓ 环境动态
- ✓ 具有马尔科夫假设的随机过程
- ✓ 具有静止假设的随机过程
- ✓ 政策迭代
- ✓ 价值迭代
- ✓ 修改后的政策迭代

## 本章的目的。

- ✓ 理解有限马尔科夫决策过程的形式问题的概念。讨论在不同情况下选择不同行动的关联性，理解动态编程与策略迭代和价值迭代算法，并举例说明。

## 什么是马尔可夫过程?

- ✓ 马尔科夫链或马尔科夫过程是一个随机模型，描述了一连串的可能事件。
- ✓ 每个事件的概率只取决于前一个事件的状态

## 马尔科夫过程的定义

- ✓ 正如我们所知，在控制循环中，我们有一系列的状态、行动和奖励，在一定的时间步骤（状态）。

环境可以有两种情况。

- ✓ 环境是决定性的（对未来的规划更简单）。
- ✓ 我们知道，由于不确定性，我们也有随机的环境（对于我们想要建模的奖励和下一个状态，过程是动态的）。

$$(S_!, A_!, R_!), (S^*, A^*, R^*), (S_{\#}, A_{\#}, R_{\#}), \dots$$

## 马尔科夫过程的重要属性 properties

- ✓ 每个动态系统的基本过程都有一个一般结构
- ✓ 有了足够的历史（通常很短），我们可以在动态系统中进行预测

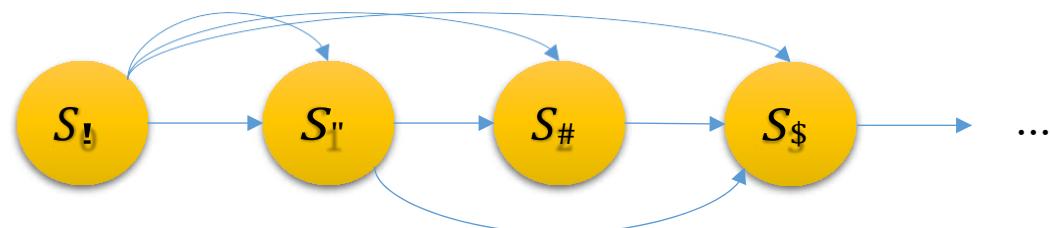
例如温度预测。

- 虽然有动态的变化，但有一些基础性的规则
- 我们可以用简短的历史来预测未来
- 环境可以是决定性的，也可以是概率性的

## 随机过程 Random Process

- ✓ 状态空间中的状态S
- ✓ 环境的随机动态可以表示如下。

$$p(S_1 | S_0, \dots, S_{t-1})$$



注意: 这个条件分布可能非常大 (由于关系) large (due to relations)

## 解决方案

- ✓ 对于大型状态空间的建模，我们需要考虑两个假设。

马尔科夫假设



状态  $S'$  只取决于之前的  $k$  个状态  $S$  或有限的状态历史

静态过程假设



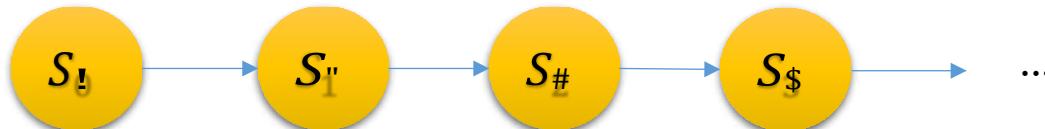
环境的动态性在时间上是静态的

# 马尔可夫过程

## 马尔科夫假设

- ✓ 状态 $S'$ 只取决于之前的 $k$ 个状态 $S$ 或有限的状态历史

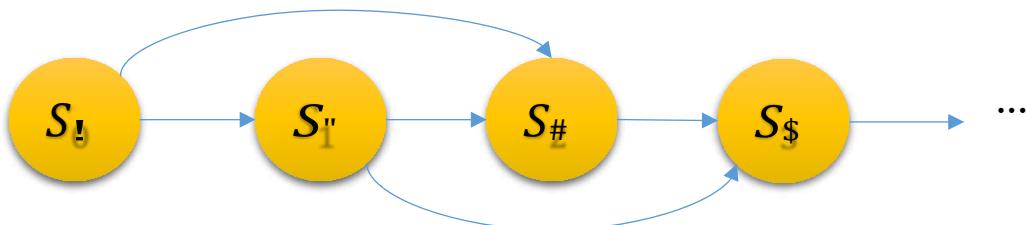
$$P(S_{\%} | S_{\%&#8226;})$$



一阶  
马尔可夫过程

$$p(S_{\%} | S_{\%}&#8226;, S_{\%&#8226;})$$

优势  
Advantage



K阶马尔科夫  
过程( $k=2$ )

- ✓ 降低计算复杂性（注意我们可能需要增加更多的变量才能实现）。

## 静态过程假设

How to achieve stationary process?

- ✓ 向系统添加新的变量，直到环境的动态变得静止。

如果我们这样做，我们可以说明下一个状态可以通过条件分布来实现。  
 $P(S'|S)$

## 静态过程假设 Static Process Assumption

例子:

考慮到只有右臂的末端执行器。

$$\langle !R, R^", R\#, R$, R\%, \rangle$$

$R^"$

是否有其他动态? Dynamics?

不断增加动态性  
进入模型

视应用而定 这里的速度

$$\langle R!, R^", R\#, R\$ \& R\%_0^{*+}, R^", -, \rangle$$
$$\langle R!, R^", R\#, R\$ \& ", R\%_0^{*+}, R^", -, R., !, R.^", R\#, R.-\$& ", R.-\%_0^{*+}, R.-, - \rangle$$



## 挑战 Challenge

- ✓ 增加更多的变量和动态，就会增加计算的复杂性！

## 解决方案 Solutions

- ✓ 在添加更多的变量与保持静止和马尔科夫假设之间做出权衡

◦

## 理念

- ✓ 假设我们有马尔可夫假设和静止性，那么这个想法是：
  - ▶ 预测行动是如何影响未来状态的

$$P(S_{\text{next}} | S_{\text{current}})$$

- ▶ 它被称为计算状态转换形式S到S'。

$$P = \begin{bmatrix} P(s_1|s_1) & P(s_2|s_1) & \cdots & P(s_N|s_1) \\ P(s_1|s_2) & P(s_2|s_2) & \cdots & P(s_N|s_2) \\ \vdots & \vdots & \ddots & \vdots \\ P(s_1|s_N) & P(s_2|s_N) & \cdots & P(s_N|s_N) \end{bmatrix}$$

因此，利用这个状态转换，我们必须做出决定

因此，我们进入马尔可夫决策过程

# 马尔科夫决策过程(MDPs) (MDPs)

## 什么是MDPs?

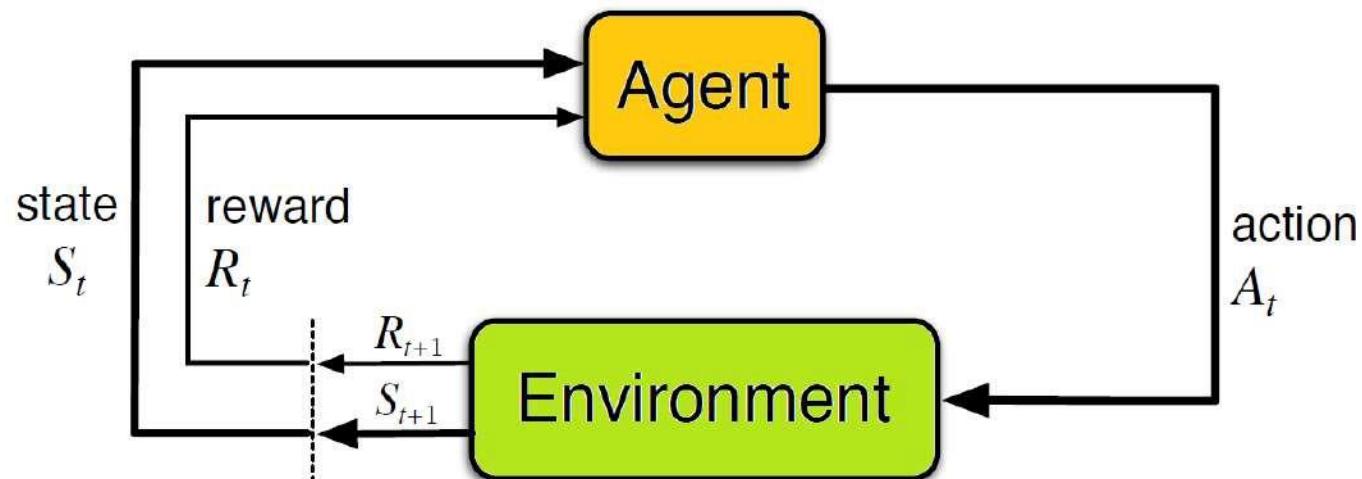
- ✓ MDP是顺序决策的经典形式化，在这种情况下，行动不仅影响眼前的回报，而且还影响后续的情况或状态以及未来的回报。
- ✓ MDPs涉及延迟奖励，以及需要权衡即时和延迟奖励。
- ✓ 在MDP中，我们估计每个状态s中每个行动a的价值 $q_*(s, a)$ ，或者我们估计每个状态的价值 $v_*(s)$ ，给出最佳行动选择。

## 什么是MDPs?

- ✓ MDPs是一个离散时间随机控制过程
- ✓ MDPs提供了一个数学框架，在结果是的情况下对决策进行建模。
  - 部分是随机的，部分是由决策者控制的
- ✓ 一种工具，作为数学上的理想化和正式的环境呈现，以分析和开发强化学习算法。

# 马尔科夫决策过程(MDPs) (MDPs)

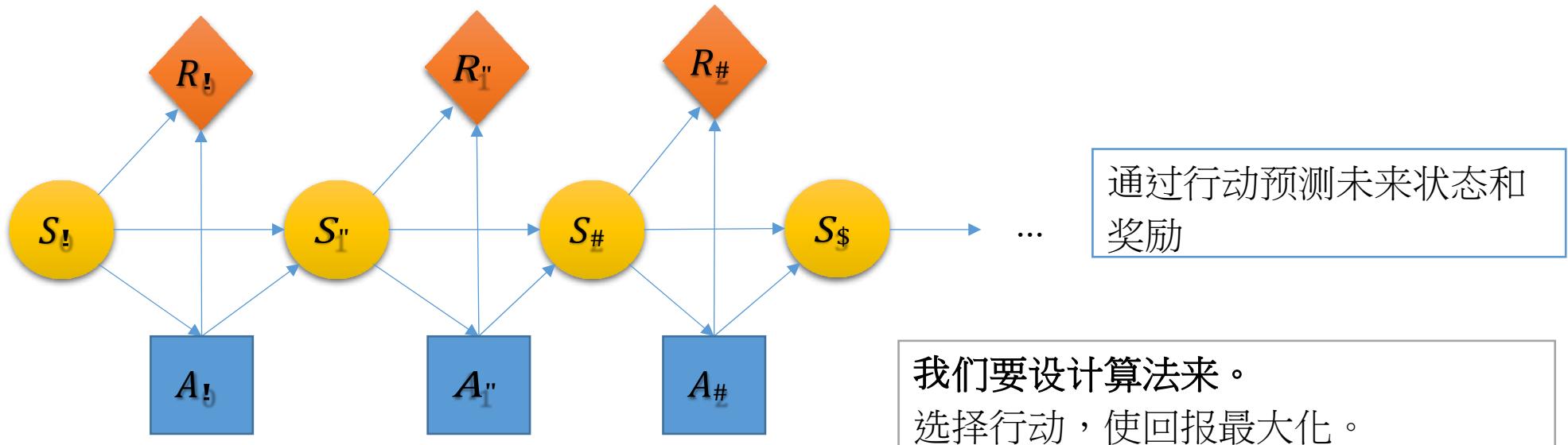
- ✓ 马尔科夫决策过程中的代理人-环境互动 (类似的概念)  
(similar concept)



# 马尔科夫决策过程(MDPs) (MDPs)

✓ 有限的MDP。

MDPs包括状态 ( $S_{\%}$ ) 、行动 ( $A_{\%}$ ) 、奖励 ( $R_{\%}$ )  
的集合，在马尔可夫假设下。



# 马尔科夫决策过程(MDPs)

奖励(记住)Reward(Remember)

- 奖励是对代理人当前情况的评价，我们为它定义了一个函数。

$$\%r = R(\%S, \quad) \\ sa\%$$

通常我们假设它是静止的，并且在时间上没有变化，具有相同的状态参数！  
注：我们通常为终端定义个人的奖励（成功/赢）

我们希望设计的算法能够选择行动以使回报最大化。

*m m m m m(m)*

## 我们目前的假设。 Our current assumptions:

- 完全可观察的隐蔽环境
- 我们有一个完整的模型和环境的过渡动态（还没有学习）。
- 随机过程(不确定性)
- 行动选择是有顺序的，可能取决于以前的行动

# 马尔科夫决策过程(MDP)的挑战

## 如果环境中的过程是无限的呢？

我们可以使用折扣系数 $\gamma$

- 折扣系数是指有折扣的奖励，如下所示。

$$\mathbf{O} \quad \gamma^{\%} R \% S, \quad ) \\ \% \qquad \qquad AS \% \quad$$

- ✓ 其中 $\gamma$ 在[0,1]之间（1表示不贴现）。
- ✓ 想法是，预期的功能奖励会更高，所以我们对其进行折算。

# 马尔科夫决策过程(MDPs) (MDPs)

## Challenge

如果环境中的过程是无限的呢？

On  $R^S, \gamma$   
% AS%

我们也可以使用奖励的平均值

$$\frac{1}{t} \rightarrow \mathbb{E}(R_t | S_t)$$

注：计算成本很高，在未来我们可以看到改进的技术

# 马尔科夫决策过程(MDPs) (MDPs)

## MDPs的定义? What are the MDPs?

✓ 有一组状态S, 行动A

✓ 奖励模式

$$R = R(S, A)$$

✓ 过渡模式。

$$P(S' | S, a)$$

✓ 折扣系数 ( $\gamma$ )

✓ 地平线 ( $h$ ) (集, 或时间步骤)。

目标是将状态映射到行动 (Finding Optimal policy)  
(寻找最佳政策)

## 贸易市场的MDP实例? Examples for Trade Market?

- ✓ 国家=共享状态
- ✓ 行动=买入、卖出、持有
- ✓ 奖励模式=利润
- ✓ 过渡模型=市场的随机变化
- ✓ 折扣系数 ( $\gamma$ ) = 0.9
- ✓ 地平线 ( $h$ ) (插曲, 或时间步骤) =无穷大

什么时候买，什么时候卖，以实现利润最大化

# MDP的目标是什么? What is the goal of the MDPs?

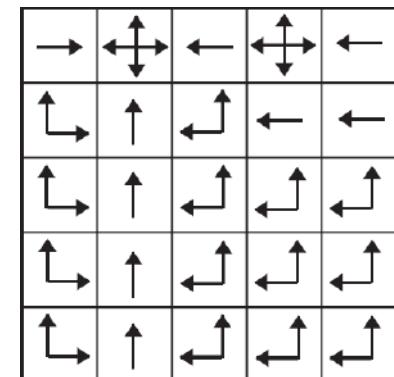
- ✓ MDP和RL的目标是**找到最佳策略 $\pi$** （优化以获得最大回报）。

请记住。

- 政策规定了一个行动a，在状态s下采取的行动。
- 更确切地说， $\pi$ 是一个概率，即在一个状态s中采取的行动a

- ✓ 在马尔科夫假设下，我们可以将政策表述为：。

$$a_{\%} = \pi(s_{\%})$$



## What is Policy Optimization

- ✓ 政策优化是为了优化将状态转化为行动的映射
- ✓ 因此，我们需要**估计价值函数**来评估环境的当前状态
- ✓ 最优政策的价值。

$$\begin{matrix} {}^*v(s) \\ - \% \end{matrix}$$

- ✓ 因此。

$$\begin{matrix} v_*(s) \geq v_!(s\%) \\ \% \end{matrix}$$

## ✓ 政策优化算法

- 政策迭代
- 价值迭代
- ...

## 政策迭代(基于MDPs) (Policy Iteration (Based on MDPs))

- ✓ 在这个算法中，我们直接优化政策
- ✓ 在政策迭代中， 我们首先选择一个随机或任意的政策。
- ✓ 然后， 我们反复评估和改进政策， 直到收敛。
- ✓ 因此， 政策迭代执行了两个步骤， 直到收敛。
  1. 政策评估
  2. 政策改进

# 动态编程(Dynamic Programming) (Policy Iteration )

## 政策迭代(基于MDPs) Policy Iteration (based on MDPs)

1

### 政策评估步骤 Policy evaluation step

- ✓ 在这一步中， 我们对状态 $s$ 下的政策 $\pi$ 进行评估， 从而用**贝尔曼方程**计算出Q值。

贝尔曼方程 Bellman equation

$$V_1(s) = \sum_r r \sum_{s'} p(s', r | s, \pi(s)) V_1(s')$$

# 动态编程(Dynamic Programming) (Policy Iteration)

## 政策迭代(基于MDPs) Policy Iteration on MDPs

2

### 政策改进步骤 Policy improvement step

✓ 在政策改进步骤中。

- 在每个步骤中寻找使Q值最大化的行动
- 通过贪婪地执行搜索来更新政策

$$\pi_{(s)} = \underset{\circ}{\operatorname{argmax}} - \underset{V_1}{\underset{)}{!} \in +} p(s', r | s, a)_{(s, )}$$

直到收敛到最佳状态  
 $(s)$

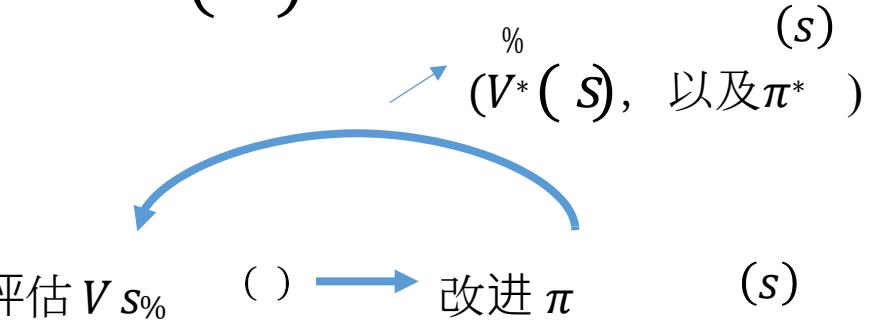
重复这两步，直到值函数收敛到最佳值函数为止。

(用最佳行动取代当前行动)

saeedvand@ntnu.edu.tw, 强化学习(RL)



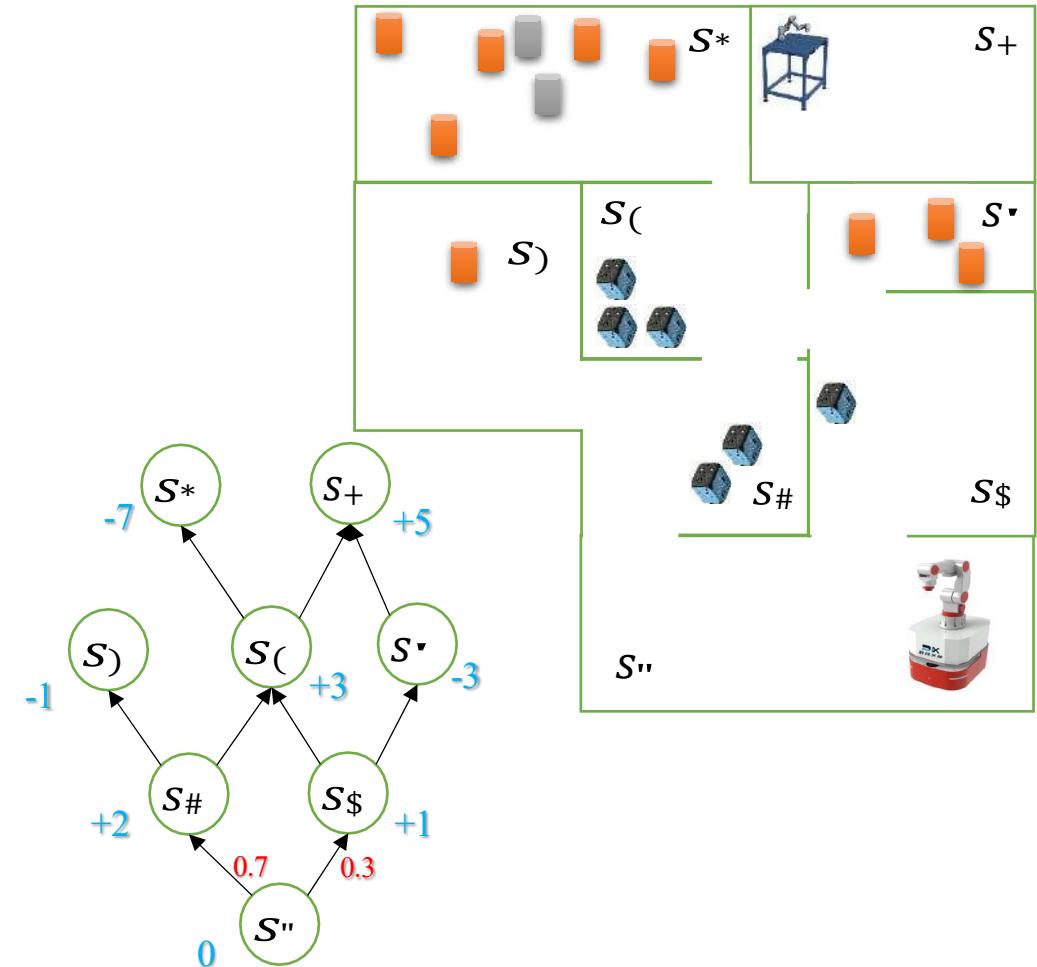
评估  $V_s$



# 动态编程 Programming (Example)

## 例子 Example

- ✓ 每个交叉点作为一个状态有两个动作，"右"和"左"，并且只能向上移动。
- ✓ 环境是随机的
- ✓ 我们需要建立一个状态转换概率矩阵（MDPs）。
- ✓ 为简单起见，让我们假设环境中左的概率总是0.7，而右的概率是0.3。



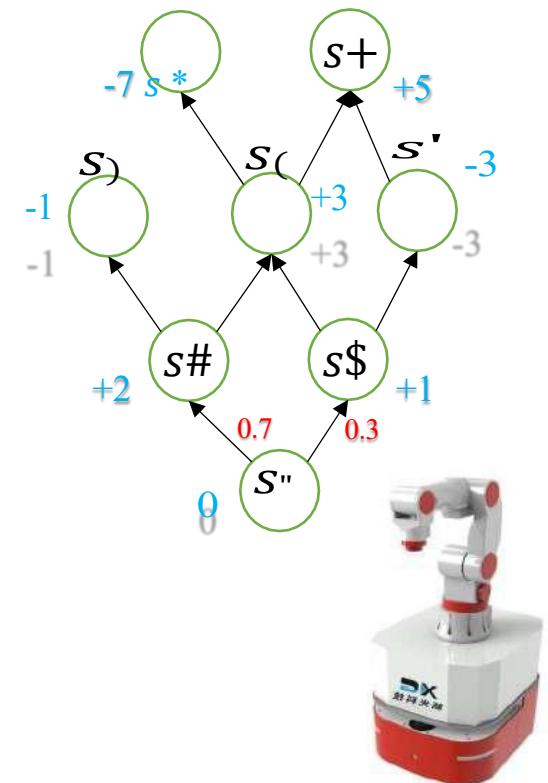
# 动态编程 (Dynamic Programming) (Example)

- ✓ 由于我们有两个行动，我们需要两个 过渡概率矩阵。对于所选择的行动左

- 

状态转换图 (左)。

$$T[a(\text{left})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

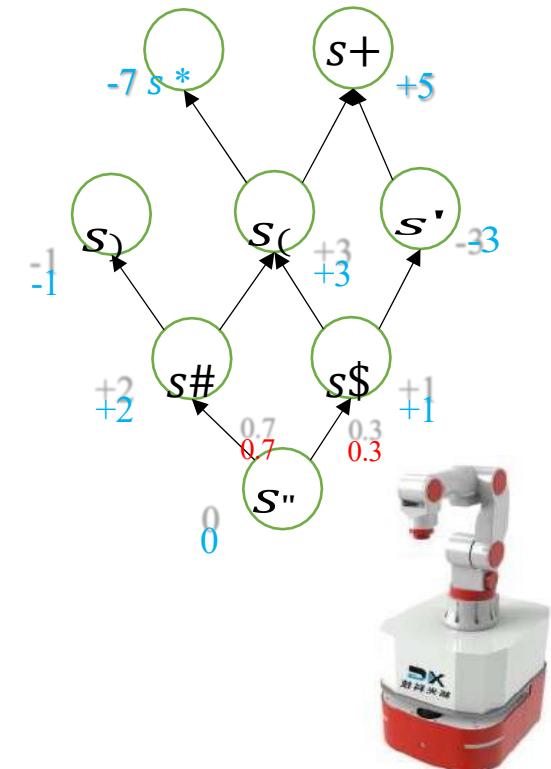


# 动态编程 (Dynamic Programming) (Example)

对于选择的行动权。

状态转换图 (右)。

$$T[a(\text{right})] = \begin{bmatrix} s_1 & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



✓ 都是基于概率和状态转换图的。

# 动态编程 Programming (Example)

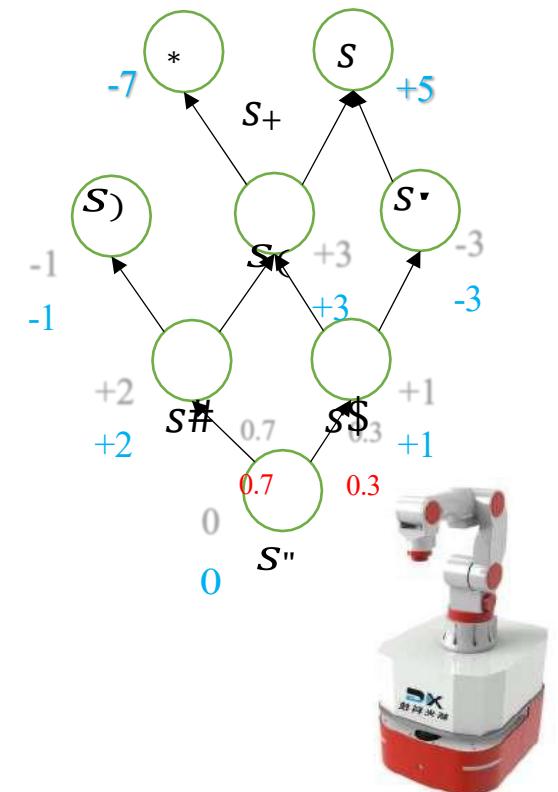
假设 Assumptions:

- 折算系数0.9
- 所有状态的初始值V(s)在大生成中为0

最初的随机政策。

政策 ( $\pi$ )

S1	S2	S3	S4	S5	S6	S7	S8
R	R	R	-	R	R	-	-



# 动态编程(Policy Iteration) (Policy Iteration )

譬如说

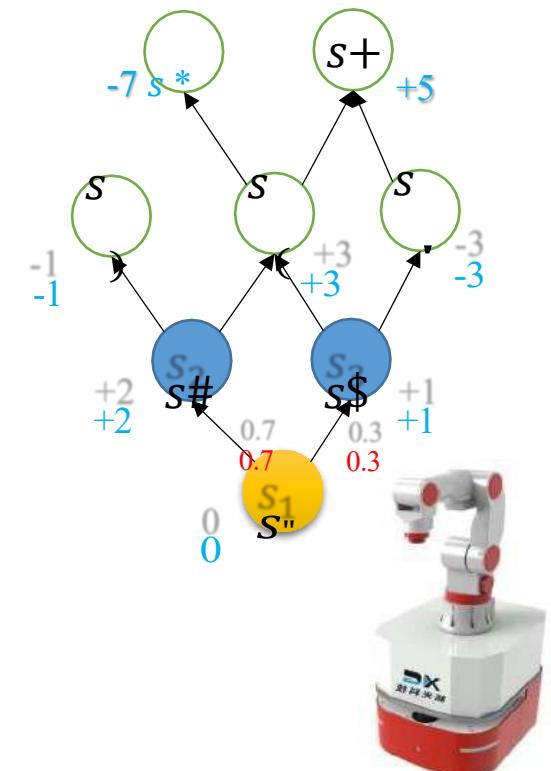
政策 ( $\pi$ )							
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
R	R	R	-	R	R	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	0	0	0	0	0	0	0

## 1 政策评估步骤 Evaluation step

在政策评估的第一次迭代中 (只有  $s_1$ )。

$$V_1(s_1) = r + 0.9 \sum_{s''} p(s', r | s_1, \pi) V_0(s')$$

$$V_0(s_1) = 0 + 0.9 * (0.7 * 0 + 0.3 * 0) = 0$$



价值:  $V_{\%}$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0							

# 动态编程(Policy Iteration) (Policy Iteration )

譬如说

政策 ( $\pi$ )							
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
R	R	R	-	R	R	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	0	0	0	0	0	0	0

价值:  $V_{\%}$

$$V_7(s) = \sum_r p(s^r, r | s, \pi(s)) V_7(s^r)$$

## 1 政策评估步骤 Policy Evaluation step

在政策评估的第一次迭代中 (只有  $s_2$  )。

$$V(s\#) = r(s\#) + 0.9 \sum_{s' \in S} p(s', r | s\#, \pi(s\#)) V(s')$$

$$V(s_9) = 2 + 0.9 * (0.7 * 0 + 0.3 * 0) = 2$$

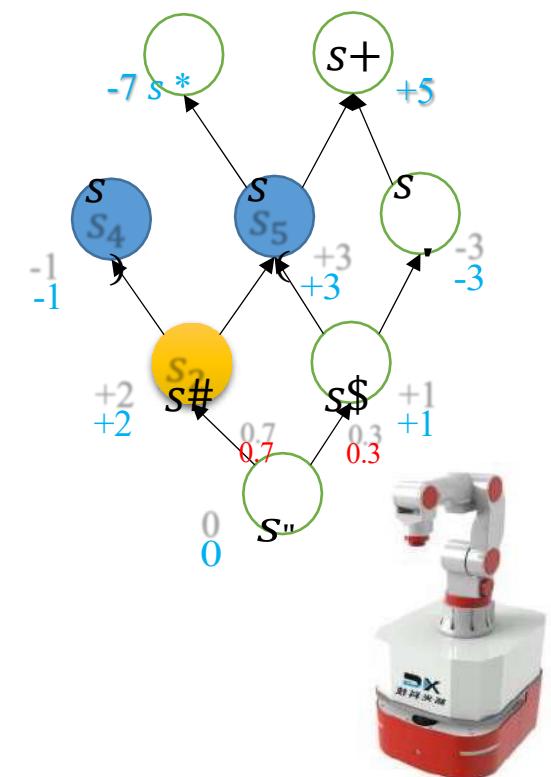
如果我们继续对所有的状态进行政策评估步骤 (

价值:  $V_{\%}$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

Reinforcement Learning  
(RL)/强化学习 (RL)。

只有奖励来了, 因为  $V_{\%}$  是零。 )



$$V_7(s) = \max_{r \in \{0, 1\}} p(s', r | s, \pi(s)) V^{\pi}(s')$$

# 动态编程(Dynamic Programming) (Policy Iteration )

例子 Example

政策 ( $\pi$ )

价值:  $V_{\pi}$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	R	-	R	R	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

$$\pi(s) = \underset{a}{\operatorname{argmax}} \gamma^{\#} p(s^{\$}, r|s, a) \quad (s\$)$$

2

政策改进步骤

✓ 政策评估的第一次迭代  $s_1$ 。

对于  $s^{\&}$  和 a(left)

$$V_7(s^{\&}) = 0.9 \& \underset{! \in \#}{p}(s^{\$}, r|s^{\&}, \pi(s^{\&})) V_7(s^{\$})$$

$$(s^{\&}) = 0.9 * (0.7 * 1.53 + 0.3 * 1) = 1.27$$

$2s, s_3$

对于  $s^{\&}$  和 a(right)

$$V_7(s^{\&}) = 0.9 \& \underset{! \in \#}{p}(s^{\$}, r|s^{\&}, \pi(s^{\&})) V_7(s^{\$})$$

$$( ) = ( ) =$$

$$T[a(\text{left})] = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



(s\$)

$$V_7 = 0.9 * (0.3 * 2 + \boxed{0.7} * 1) \\ V_7 = 1.17$$

(s&

(0.7 \* 1)

# 动态编程(Policy Iteration) (Policy Iteration )

例子 Example

政策 ( $\pi$ )

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	R	-	R	R	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

价值:  $V$   
%

$$\pi(s) = \underset{a}{\operatorname{argmax}} \gamma V^{\pi}(s)$$

2

Policy improvement steps

✓ 政策评估的第一次迭代  $s_2$ 。

对于  $s_{\#}$  和 a(left)

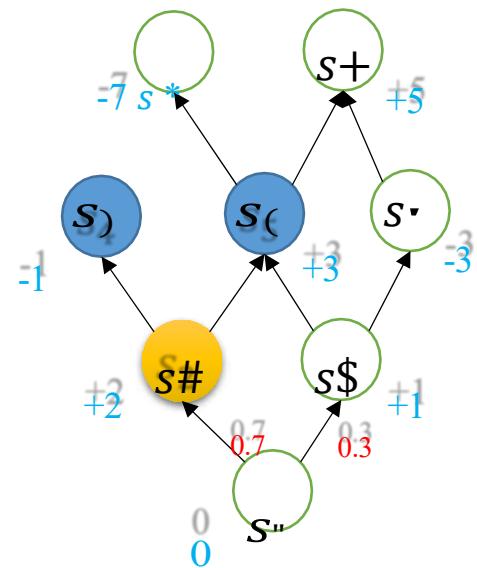
$$V_7(s_{\#}) = 0.9 \& \sum_{a \in \{L, R\}} p(s^{\#}, r | s_{\#}, a) V^{\pi(a)}(s^{\#})$$

$$V_7(s_{\#}) = 0.9 * ((s_{\#}) * (-1)) + (0.3 * 3)$$

对于  $s_{\#}$  和 a(right)

$$V_7(s_{\#}) = 0.9 \& \sum_{a \in \{L, R\}} p(s^{\#}, r | s_{\#}, a) V^{\pi(a)}(s^{\#})$$

$$V_7(s_{\#}) = 0.9 * (0.3 * (-1)) + (0.7 * 0)$$



$$V_7 = 0.18$$

$$7 \quad V =$$

(0.7 \* 3)

(*s*) 1.62

# 动态编程(Policy Iteration) (Policy Iteration )

例子 Example

政策 ( $\pi$ )

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	L	-	R	R	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

价值:  $V$   
%

$$\pi(s) = \underset{a}{\operatorname{argmax}} \gamma \sum_{t=1}^{\infty} p(s^t, r|s, a) V^t$$

2

Policy improvement steps

✓ 政策评估的第一次迭代  $s_3$ 。

对于  $s_3$  和 a(left)

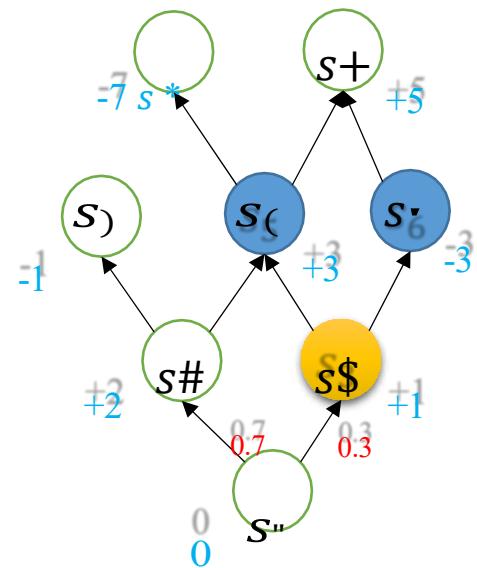
$$V_7(s_3) = 0.9 \& \sum_{a \in \{L, R\}} p(s_3^a, r|s_3, a) V_7(s^a)$$

$$V_7(s_3) = 0.9 * (0.7 * 3) + (0.3 * (-3))$$

对于  $s_3$  和 a(right)

$$V_7(s_3) = 0.9 \& \sum_{a \in \{L, R\}} p(s_3^a, r|s_3, a) V_7(s^a)$$

$$V_7(s_3) = 0.9 * ((s_3) * 3) + (0.3 * (-3))$$



7  $(s|\gamma = \boxed{1.08})$

$V_7 = -1.89$

$(0.7 * (-3))$

# 动态编程(Policy Iteration) (Policy Iteration )

例子 Example

政策 ( $\pi$ )

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	L	-	R	R	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

价值:  $V$   
%

$$\pi(s) = \underset{a}{\operatorname{argmax}} \gamma \sum_{t=1}^{\infty} p(s^t, r|s, a) V(s^t)$$

2

Policy improvement steps

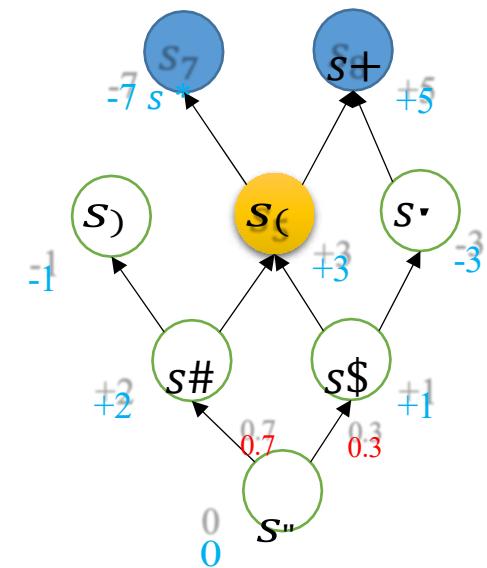
✓ 政策评估的第一次迭代  $s_5$ 。

对于  $s_5$  和 a(left)

$$V_7(s) = 0.9 \& \sum_{a \in \{L, R\}} p(s^t, r|s, a) V^{(s^t)} \\ V_7(s) = 0.9 * (0.7 * (-7) + 0.3 * 5)$$

对于  $s_5$  和 a(right)

$$V_7(s) = 0.9 \& \sum_{a \in \{L, R\}} p(s^t, r|s, a) V^{(s^t)} \\ V_7 = 0.9 * (0.3 * (-7) + 0.7 * 5)$$



$$V_7 = -3.6$$

$$7 \quad V =$$

$$(0.7 * (5)) \\ (s) ) \quad \boxed{-0.54}$$

# 动态编程(Policy Iteration) (Policy Iteration )

例子 Example

政策 ( $\pi$ )							
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	L	-	R	L	-	-
价值: $V$							
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

$$\pi(s) = \underset{a}{\operatorname{argmax}} \gamma \sum_{t=1}^T p(s^t, r|s, a) V(s^t)$$

2

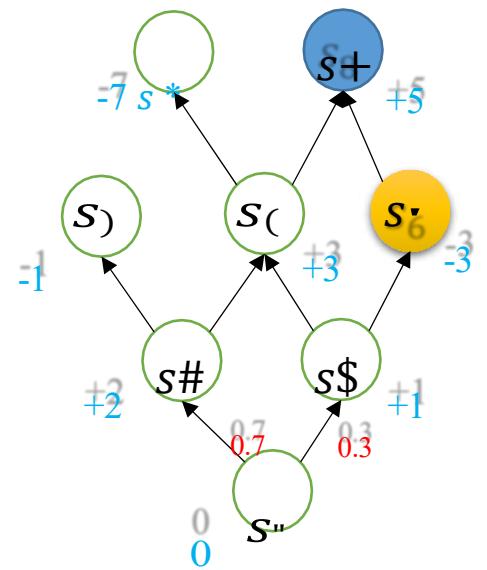
Policy improvement steps

✓ 政策评估的第一次迭代  $s_6$ 。

对于  $s^*$  和  $a(\text{left})$

$$V_7(s^*) = 0.9 \& \sum_{a \in A} p(s^*, r|s^*, \pi(s^*)) V(s^*)$$

$$V_7(s^*) = 0.9 * ((1 * 5))$$



$$7 \quad (s^*)V = 4.5$$

# 动态编程(Dynamic Programming) (Policy Iteration)

例子 Example

政策 ( $\pi$ )

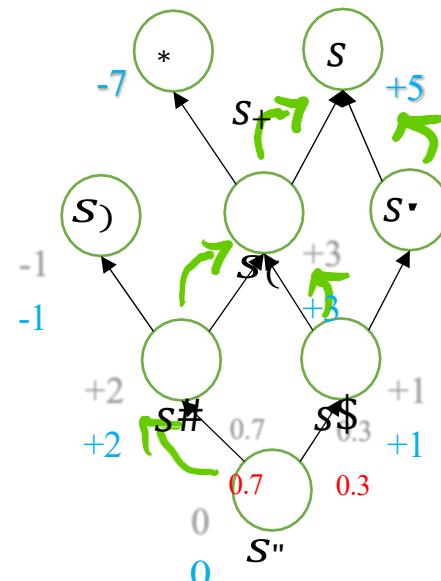
价值:  $V_{\%}$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	L	-	R	L	-	-
$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

$$T[a(\text{左})] = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

在MDP上实现可视化  
(only after one iteration)  
仅在一次迭代后)



# 动态编程(Policy Iteration) (Policy Iteration )

## 1. Initialization

$V(s) \in \mathbb{R}$  and  $\pi(s) \in \mathcal{A}(s)$  arbitrarily for all  $s \in \mathcal{S}$

## 2. Policy Evaluation

Repeat

$$\Delta \leftarrow 0$$

For each  $s \in \mathcal{S}$ :

$$v \leftarrow V(s)$$

$$V(s) \leftarrow \sum_{s',r} p(s', r | s, \pi(s)) [r + \gamma V(s')]$$

$$\Delta \leftarrow \max(\Delta, |v - V(s)|)$$

until  $\Delta < \theta$  (a small positive number)

## 3. Policy Improvement

*policy-stable*  $\leftarrow$  true

For each  $s \in \mathcal{S}$ :

$$a \leftarrow \pi(s)$$

$$\pi(s) \leftarrow \arg \max_a \sum_{s',r} p(s', r | s, a) [r + \gamma V(s')]$$

If  $a \neq \pi(s)$ , then *policy-stable*  $\leftarrow$  false

If *policy-stable*, then stop and return  $V$  and  $\pi$ ; else go to 2

# 动态编程(价值迭代)(Value Iteration )

## 价值迭代(基于MDPs)

- ✓ 另一种解决贝尔曼方程的方法被称为**价值迭代**，即直接评估价值。
- ✓ 通过迭代更新估计值 $V$ ，计算出**最佳状态值函数**- $v^{(s)}$ 。
- ✓ 价值迭代算法在一个单一的时间内更新状态**价值函数步**。
- ✓ 通过展望未来，计算出所有可能的回报，这才有可能。
- ✓ 价值迭代算法也被保证收敛到最佳的价值。

# 动态编程(价值迭代)(Value Iteration )

## 价值迭代(基于MDPs)

- ✓ 从一个随机值函数 $V$ 开始- (s) 并在每个步骤中更新它。

## 贝尔曼方程

$$V_1(s) = r(s) + \gamma \max_{a \in A} \sum_{s'} p(s', r | s, a) V_0(s')$$

这个想法是（类似于政策迭代，但增加了最大值）。

- 在价值迭代算法中，取所有可能的行动的**最大值**。

# 动态编程(价值迭代)(Value Iteration )

## 价值迭代(基于MDPs)

$$V_1(s) = r(s) + \gamma \max_{\pi} \sum_{s'} p(s', r|s, a) V_0(s')$$

如何?

例如对于任何任意问题的状态 $s_1$ 。

$$V_0(s_8) = r(s_8) + \gamma \max_{\pi} \left[ \sum_{s' \in \text{next}} p(s', r|s_8, a) V_0(s') \right]$$

如果我们有两个行动L和R。

$$V(s'') = r(s'') + \gamma \max_{\pi} \left[ \begin{array}{l} \mathbf{T}^{\pi(s'')} \left[ p(s^3, r|s'', \pi(s'') = L) V(s^3) \right] \\ \mathbf{T}^{\pi(s'')} \left[ p(s^3, r|s'', \pi(s'') = R) V(s^3) \right] \end{array} \right]$$

$0/\epsilon^2$

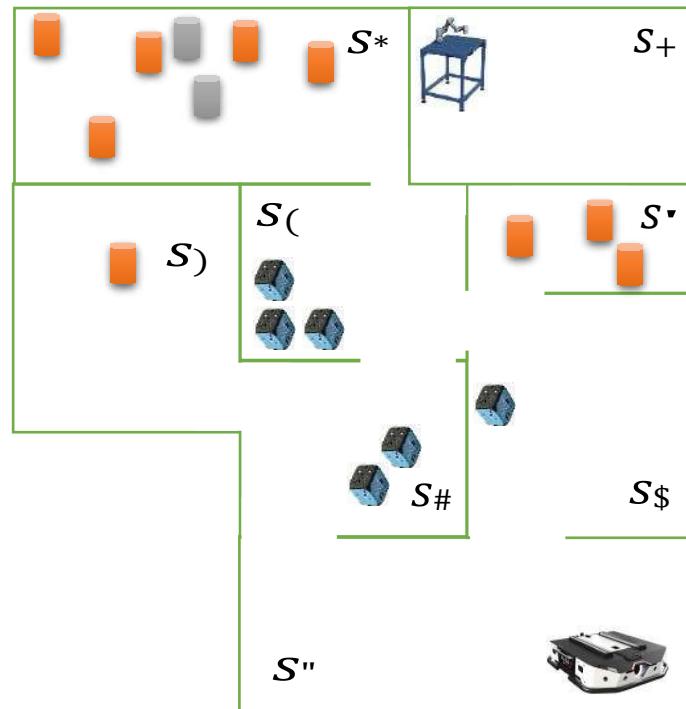
"

-

# 动态编程(价值迭代)(Value Iteration )

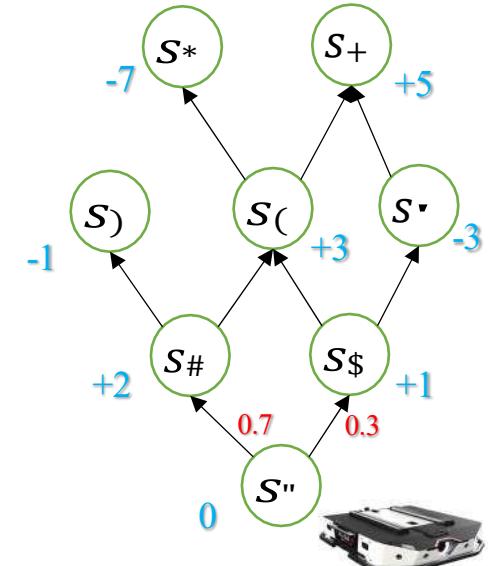
## 价值迭代(基于MDPs)

例子



$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



# 价值迭代,示例

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{s' \in \#} p(s', r | s, a) V_7(s')$$

第一次迭代和  $s_1$ 。

$$V_7(s_1) = r(s_1) + \gamma \max_{\pi} \left[ \begin{array}{l} \&_{s' \in \#} p(s', r | s_1, \pi(s_1), (s_1 = L)) V_7(s') \\ + \\ \&_{s' \in \#} p(s', r | s_1, \pi(s_1), (s_1 = R)) V_7(s') \end{array} \right]$$

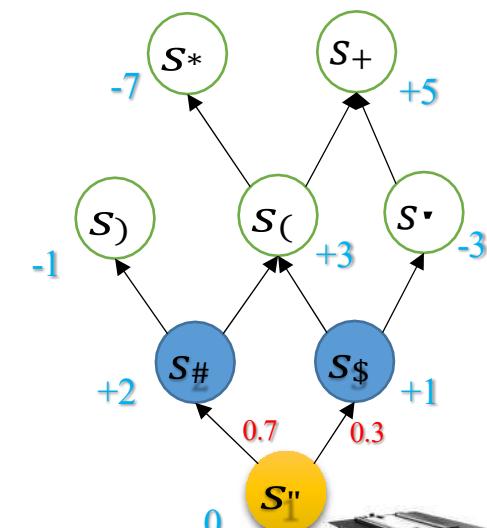
$$V_7(s_1) = r^{(0)} + 0.9 * \max_{\pi} \begin{bmatrix} 0.7 * 0 + 0.3 * 0 \\ 0.3 * 0 + 0.7 * 0 \end{bmatrix}$$

价值:  $V_0$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	0	0	0	0	0	0	0

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



0

Saeedvand@ntnu.edu.tw,  
强化学习(RL)

# 价值迭代，示例 Example

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{a \in A} p(s', r | s, a) V_7(s')$$

第一次迭代和  $s_2$ 。

$$V_7(s) = r(s) + \gamma \max_{\pi} \left[ \begin{array}{l} \&_{a \in A} p(s', r | s, a) = L \\ \&_{a \in A} p(s', r | s, a) = R \end{array} \right] V_7(s')$$

$$V_7(s) = 2 + 0.9 * \max_{\pi} \left[ \begin{array}{l} 0.7 * 0 + 0.3 * 0 \\ 0.3 * 0 + 0.7 * 0 \end{array} \right]$$

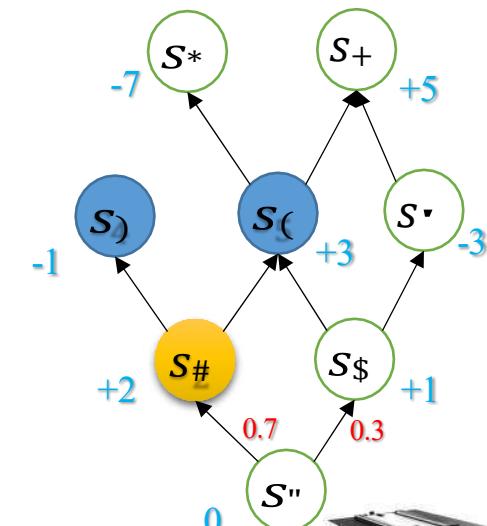
$$V_7(s) = 2$$

价值:  $V_7$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	0	0	0	0	0	0

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



0

Saeedvand@ntnu.edu.tw,  
强化学习(RL)

# 价值迭代,示例 Example

更新值 value

$$V_{-(s)} = \frac{r_{(s)} + \gamma \max_{a \in A} \sum_{s' \in S} p(s', r | s, a) V_{-(s')}}{s_1 \ s_2 \ s_3 \ s_4 \ s_5 \ s_6 \ s_7 \ s_8}$$

第一次迭代和 allstates。

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
0	2	1	-1	3	-3	-7	5

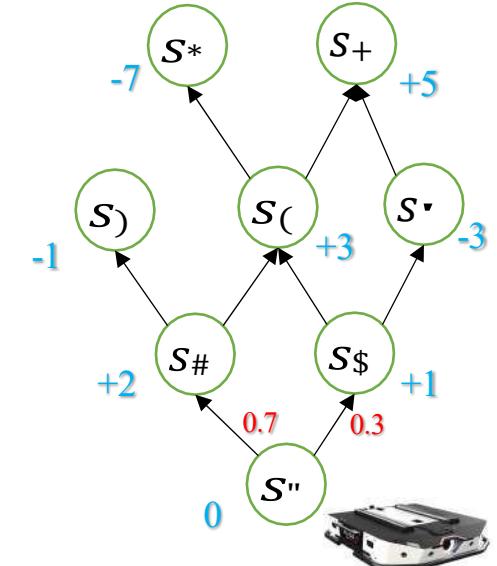
价值:  $V_{\%}$

继续对所有州进行相同的  
处理

Saeedvand@ntnu.edu.tw, 强化学习(RL)

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



# 价值迭代，示例 Example

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{a \in \#} p(s^a, r|s, a) V_7(s^a)$$

第二次迭代和  $s_1$  和

$$V_7(s\#) = \max_{\pi} \left[ \begin{array}{l} \&_{a \in \#} p(s^a, r|s\&, (s\#) = L) \\ \&_{a \in \#} p(s^a, r|s\&, (s\#) = R) \end{array} \right] V_7(s\$)$$

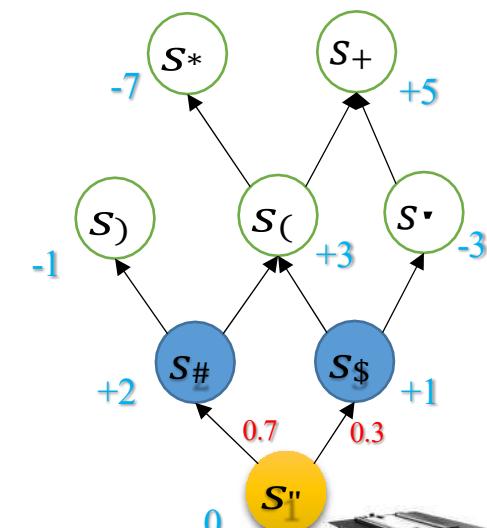
$$V_7(s\#) = r^{(0)} + 0.9 * \max_{\pi} \left[ \begin{array}{l} 0.7 * 2 + 0.3 * 1 \\ 0.3 * 2 + 0.7 * 1 \end{array} \right]$$

$$V_7(s\#) = 0 + 0.9 * \max_{\pi} \left[ \begin{array}{l} 1.7 \\ 1.3 \end{array} \right] = 1.53$$

价值:  $V_7$

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
1.53	2	1	-1	3	-3	-7	5

Saeedvand@ntnu.edu.tw, 强化学习(RL)

## 价值迭代，示例

更新值 update value

$$V_7(s) = r(s) + \gamma \max_{\substack{! \in \#}} p(s^*, r | s, a) V_7(s^*)$$

第二次迭代和 $s_2$ 。

$$V_7(s) = \max_{r(s), \pi} \left[ \sum_{s' \in \{L, R\}} p(s', r | s, \pi) V_7(s') \right]$$

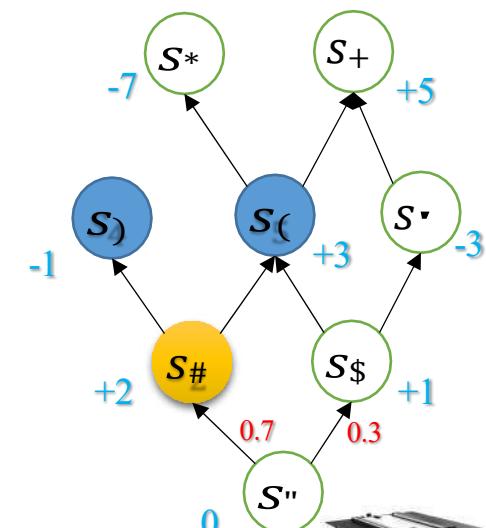
$$V_7(\underline{s}) = r^{(2)} + 0.9 * \max\% \left[ \begin{array}{l} 0.7 * -1 + 0.3 * 3 \\ 0.3 * -1 + 0.7 * 3 \end{array} \right]$$

$$V_7(\text{•}) = 2 + 0.9 * \begin{bmatrix} 0.2 \\ 1.8 \end{bmatrix} = 3.62$$

价值:  $V_{\%}$

$$T[a(\text{左})] = \begin{bmatrix} & s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
1.53	3.62	1	-1	3	-3	-7	5

Saeedvand@ntnu.edu.tw, 强化学习(RL)

# 价值迭代，示例 Example

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{a \in \#} p(s^a, r|s, a) V_7(s^a)$$

第二次迭代和  $s_3$  和

$$V_7(s) = r(s) + \gamma \max_{\pi} \left[ \begin{array}{l} \&_{a \in \#} p(s^a, r|s, a) V_7(s^a) \\ \&_{a \in \#} p(s^a, r|s, a) V_7(s^a) \end{array} \right]$$

$$V_7(s) = r^{(1)} + 0.9 * \max_{\pi} \left[ \begin{array}{l} 0.7 * 3 + 0.3 * -3 \\ 0.3 * 3 + 0.7 * -3 \end{array} \right]$$

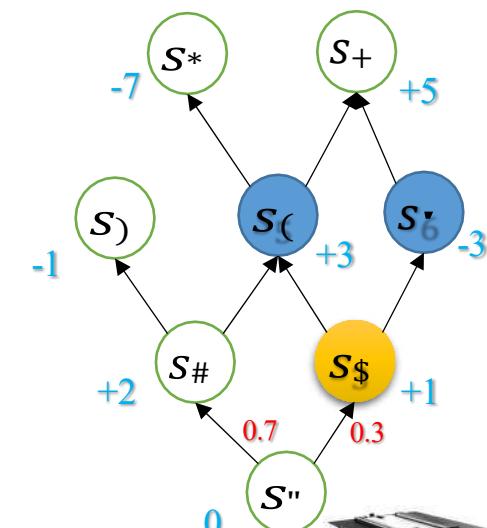
$$V_7(s) = 1 + 0.9 * \max_{\pi} \left[ \begin{array}{l} 1.2 \\ -1.2 \end{array} \right] = 2.08$$

价值:  $V_7$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
1.53	3.62	2.08	-1	3	-3	-7	5

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$





# 价值迭代，示例 Example

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{a \in A} p(s', r | s, a) V_7(s')$$

第二次迭代和  $s_4$  and

$$V_7(s) = r(s) + \gamma \max_{\pi} \left[ \begin{array}{ll} \& \sum_{a \in A} p(s', r | s, a), \pi(s') = L \\ & V_7(s') \\ \& \sum_{a \in A} p(s', r | s, a), \pi(s') = R \\ & V_7(s') \end{array} \right]$$

$$V_7(s) = -1 + 0.9 * \max_{\pi} \left[ \begin{array}{l} 0 \\ 0 \end{array} \right]$$

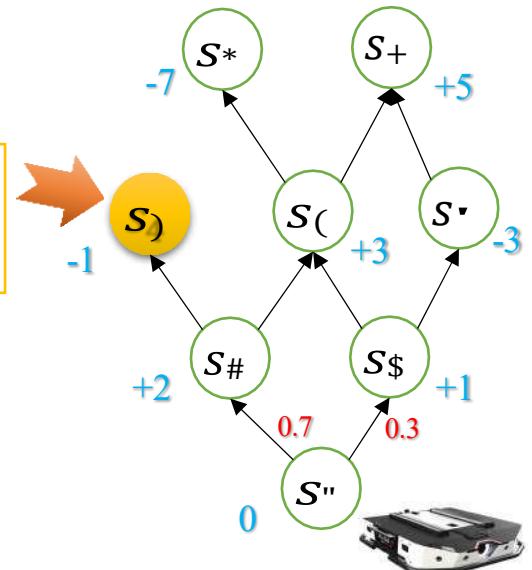
$$V_7(s) = -1$$

我们不需要运行终端状态

价值:  $V_7$

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



<b><i>s</i><sub>1</sub></b>	<b><i>s</i><sub>2</sub></b>	<b><i>s</i><sub>3</sub></b>	<b><i>s</i><sub>4</sub></b>	<b><i>s</i><sub>5</sub></b>	<b><i>s</i><sub>6</sub></b>	<b><i>s</i><sub>7</sub></b>	<b><i>s</i><sub>8</sub></b>
1.53	3.62	2.08	-1	3	-3	-7	5

Saeedvand@ntnu.edu.tw,  
强化学习(RL)

# 价值迭代，示例 Example

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{a \in A} p(s^a, r|s, a) V_7(s^a)$$

第二次迭代和  $s_5$  and

$$V_7(s) = r(s) \text{ 值 } \gamma \text{ 最大} \left[ \begin{array}{l} \&_{a \in A} p(s^a, r|s, a), \pi(s) = L \\ \&_{a \in A} p(s^a, r|s, a), \pi(s) = R \end{array} \right] V_7(s)$$

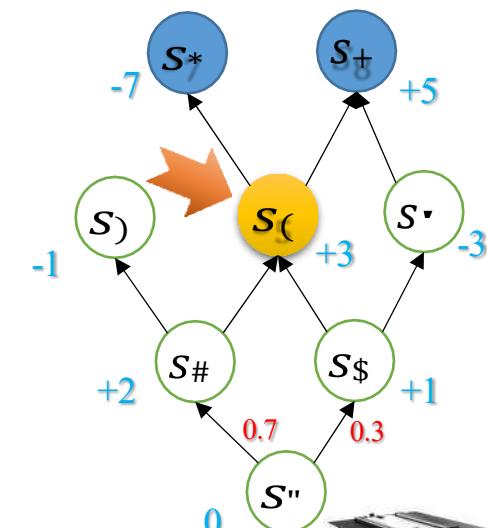
$$V_7(s) = +3 + 0.9 * \text{最大百分比} \quad [0.7 * -7 + 0.3 * 5] \\ [0.3 * -7 + 0.7 * 5]$$

$$V_7(s) = +3 + 0.9 * \text{最大百分比} \quad \begin{bmatrix} -3.4 \\ 1.4 \end{bmatrix} = 4.26$$

价值:  $V_7$

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	
$s_1$	0	0.7	0.3	0	0	0	0	0	0
$s_2$	0	0	0	0.7	0.3	0	0	0	0
$s_3$	0	0	0	0	0.7	0.3	0	0	0
$s_4$	0	0	0	0	0	0	0	0	0.7 0.3
$s_5$	0	0	0	0	0	0	0	0	0
$s_6$	0	0	0	0	0	0	0	0	0 1
$s_7$	0	0	0	0	0	0	0	0	0
$s_8$	0	0	0	0	0	0	0	0	0 0

	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$	
$s_1$	0	0.3	0.7	0	0	0	0	0	0
$s_2$	0	0	0	0.3	0.7	0	0	0	0
$s_3$	0	0	0	0	0.3	0.7	0	0	0
$s_4$	0	0	0	0	0	0	0	0	0 0
$s_5$	0	0	0	0	0	0	0	0	0.3 0.7
$s_6$	0	0	0	0	0	0	0	0	0 1
$s_7$	0	0	0	0	0	0	0	0	0 0
$s_8$	0	0	0	0	0	0	0	0	0 0



$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
1.53	3.62	2.08	-1	4.26	-3	-7	5

Saeedvand@ntnu.edu.tw, 强化学习(RL)

# 价值迭代，示例 Example

更新值 value

$$V_7(s) = r(s) + \gamma \max_{\pi} \sum_{a \in \#} p(s^+, r | s, a) V_7(s^+)$$

第二次迭代和  $s_6$  和

$$V_7(s^*) = r(s^*) + \gamma \max_{\pi} \sum_{a \in \#} p(s^+, r | s^*, a) V_7(s^+)$$

$$V_7(s^*) = -3 + 0.9 * \max_{\pi} [1, 5]$$

$$V_7(s^*) = 1.5$$

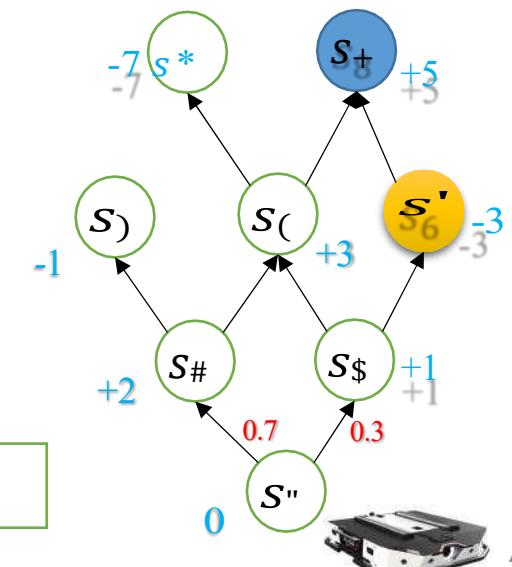
价值:  $V_7$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
1.53	3.62	2.08	-1	4.26	1.5	-7	5

更新不会改变它们，因为它们是终端状态。

$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



# 价值迭代，示例 Example

价值: $V_{\%}$	$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
	1.53	3.62	2.08	-1	4.26	1.5	-7	5

✓ 融合 Convergence

✓ 继续下一次迭代，直到收敛为止。

## 贝尔曼系数

✓ 在实践中，一旦价值函数只发生少量的变化，我们就会停止，例如贝尔曼系数，例如  $\delta=0.01$

在我们的例子中，收敛值之后。



价值:  $V^*_{\%}$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
3.88	4.41	4.09	-1	4.26	1.5	-7	5

## 汇总说明。 Tongue-in-cheek notes:

- ✓ 状态空间和行动空间应该是有限的
- ✓ 奖励值应该有一个上限和下限
- ✓ 环境应该是偶发的（如果是连续的，折扣系数应该小于1）
  -

## 价值迭代-第二步 Second Step

- ✓ 第二步是获得最佳行动，构建**最佳政策**。
- ✓ 只有一次**迭代**才能返回最大的行动序列。

$$\pi^* = \underset{\pi}{\operatorname{argmax}} \underset{M}{r}(s) + \gamma \underset{\pi \in \Pi}{\mathbf{O}} p(s' | s, a) V_{\pi}(s')$$



它决定了哪种行动给我们带来最高的价值

# 价值迭代，示例 Example

政策更新 Policy Update

价值:  $V$

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
3.88	4.41	4.09	-1	4.26	1.5	-7	5

$$\pi^* = \arg\max_a V(s) + \gamma \mathbf{T}^{0/\infty} p(s'|s, a) V_{-}$$

第二次迭代和  $s_1$  and

$$\pi^{(s)} = \max r(s) + \gamma \left[ \begin{array}{c} \& \underset{! \in \#}{p(s, r | s, \pi^{(s)})} \\ \& \underset{! \in \#}{p(s, r | s, \pi^{(s)})} \end{array} \right]$$

$$\pi^{*}(s) = \arg\max r(s) + 0.9 * \left[ \begin{array}{c} 0.7 * 4.41 + 0.3 * 4.09 \\ 0.3 * 4.41 + 0.7 * 4.09 \end{array} \right]$$

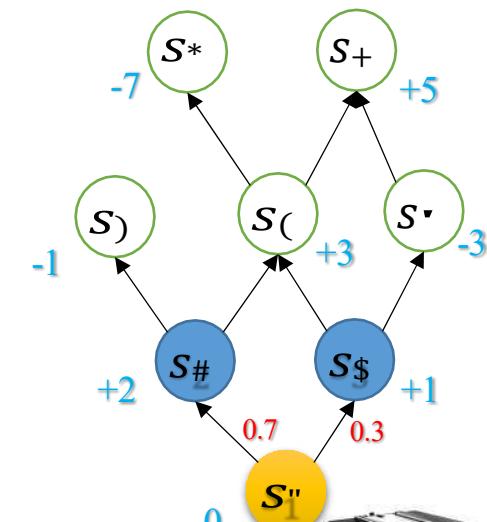
$$\pi^{*}(s) = \arg\max 0 + 0.9 \left[ \begin{array}{c} 4.314 \\ 4.186 \end{array} \right] = \arg\max (3.88, 3.76)$$

政策 ( $\pi^*$ )

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	R	-	R	R	-	-

$$T[a(\text{左})] = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{右})] = \begin{bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ s_1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s_2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s_3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 \\ s_4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s_6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s_7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



R

Saeedvand@ntnu.edu.tw,  
强化学习(RL)

# 价值迭代，示例 Example

例子 Example

政策更新的finalresult		$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
价值: $V\%$		3.88	4.41	4.09	-1	4.26	1.5	-7	5

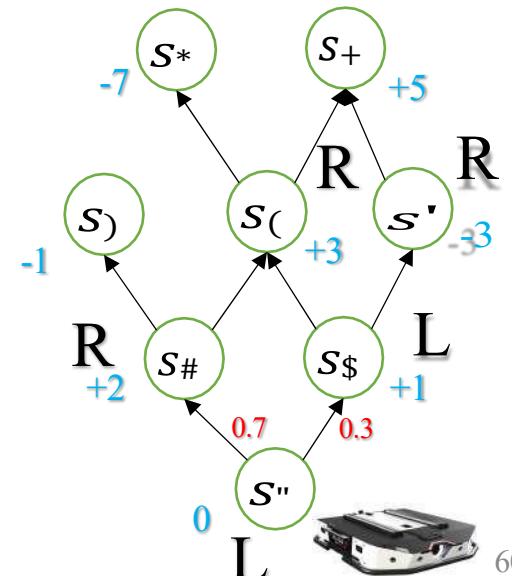
政策 ( $\pi^*$ )

$s_1$	$s_2$	$s_3$	$s_4$	$s_5$	$s_6$	$s_7$	$s_8$
L	R	L	-	R	L	-	-



$$T[a(\text{左})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.7 & 0.3 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.7 & 0.3 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0 & 0.7 & 0.3 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T[a(\text{right})] = \begin{bmatrix} s1 & s2 & s3 & s4 & s5 & s6 & s7 & s8 \\ s1 & 0 & 0.3 & 0.7 & 0 & 0 & 0 & 0 \\ s2 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s3 & 0 & 0 & 0 & 0 & 0.3 & 0.7 & 0 & 0 \\ s4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s5 & 0 & 0 & 0 & 0 & 0 & 0 & 0.3 & 0.7 \\ s6 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ s7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$



# 动态编程(价值迭代)(Value Iteration )

Algorithm parameter: a small threshold  $\theta > 0$  determining accuracy of estimation  
Initialize  $V(s)$ , for all  $s \in \mathcal{S}^+$ , arbitrarily except that  $V(\text{terminal}) = 0$

Loop:

```
|   Δ ← 0
|   Loop for each  $s \in \mathcal{S}$ :
|      $v \leftarrow V(s)$ 
|      $V(s) \leftarrow \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$ 
|     Δ ← max(Δ, |v - V(s)|)
| until Δ < θ
```

Output a deterministic policy,  $\pi \approx \pi_*$ , such that

$$\pi(s) = \arg \max_a \sum_{s',r} p(s',r|s,a)[r + \gamma V(s')]$$

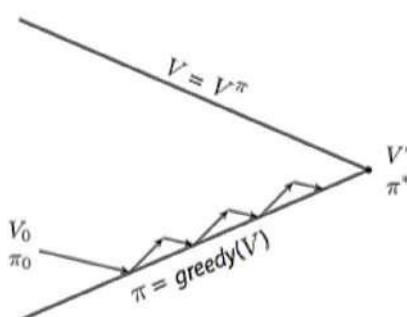
# 动态编程(Dynamic Programming)(价值迭代)(Value Iteration )

价值迭代  
Value Iteration

VS

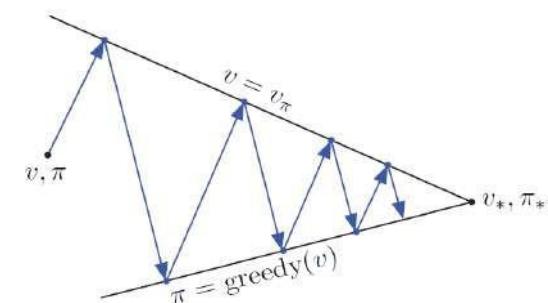
政策迭代  
Policy Iteration

- ✓ 价值迭代算法会随着更多的迭代而收敛
- ✓ 较慢
- ✓ 开始时有一个随机值函数
- ✓ 简单的算法  $O(AS^9)$



两种算法都能保证收敛到最优策略  
Both algorithms converge to an optimal policy

- ✓ 政策迭代算法的收敛速度为迭代次数较少
- ✓ 更快
- ✓ 开始时有一个随机政策
- ✓ 更复杂的  $O(AS + |S| \cdot |A| + S^2)$



# 动态编程 (修改后的政策迭代) Dynamic Programming (Modified Policy Iteration )

## 修改后的政策迭代 Modified Policy Iteration

- ✓ 为了利用价值迭代和策略迭代算法的优点，我们修改了策略迭代算法
- ✓ 我们的想法是用k次（有限的次数）更新来改进政策更新循环部分
  -

# 动态编程(Dynamic Programming) (Policy Iteration )

## 修改后的政策迭代

1

Repeat K times Policy evaluation step

$$V_0(s) = r(s) + \gamma \mathbf{B} \sum_{s' \in S} p(s', r | s, \pi(s)) V_0(s')$$

2

Policy Improvement step

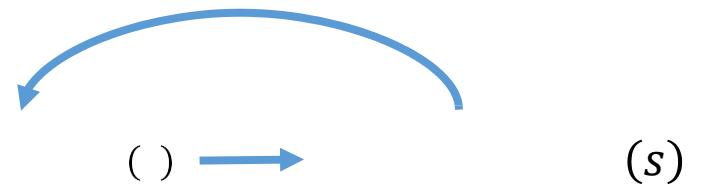
$$\pi(s) = \arg \max_a \mathbf{B} p(s', r | s, a)$$

直到收敛到最佳状态  $(V^*)^*(s)$ , 以及  $\pi^*(s)$ )

Repeating K times

评估  $V s\%$

改进  $\pi$



# 动态编程(价值迭代)(Value Iteration )

## 价值迭代

✓ 简单的算法  $O(|AS^{\#}|)$

✓ 修改后的政策迭代通常比价值迭代和政策迭代算法的表现都要快。

## 政策迭代

✓ 每迭代成本  $O(|AS^{\#}| \cdot |S|)$

## 修改后的政策迭代

✓ 每迭代成本  $O(|AS^{\#}| \cdot |A| \cdot |S^{\#}|)$

## 任务

扩展给定的示例代码（值迭代）以实现和比较算法。  
价值迭代、政策迭代和修正的政策迭代算法。

- A) 比较幻灯片中的例子问题（绘制收敛速度结果）。
- B) 对幻灯片中的例子进行扩展，增加更多的动作（机器人也可以执行向后移动的动作），并运行相同的算法来绘制收敛性。

## 有限与无限的剧情或地平线 Horizon

- ✓ 在无限地平线上，我们可以有**固定的最优政策**，因为结局总是不明确的。
- ✓ 在有限视界中，最佳政策可能**是非稳定的最佳政策**，因为当我们接近结束时，所选择的行动可能是不同的。

**挑战**对于价值迭代中的无限水平线，我们不能计算出无限的步骤