南京大学
人工智能学院
SCHOOL OF ARTIFICIAL INTELLIGENCE, NANJING UNIVERSITY

第四部分：完全可观察环境中的概率规划系统

章宗长

2021年5月8日

内容安排

- 规划
- 马尔科夫决策过程
- **精确动态规划**
- 近似动态规划
- 在线规划
- 直接策略搜索

精确动态规划

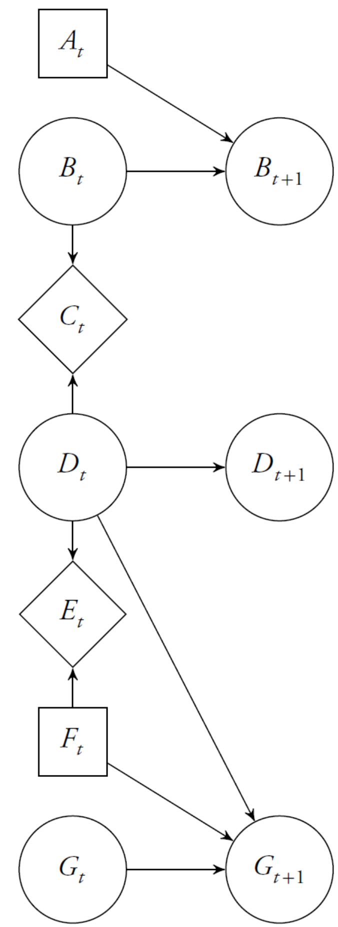
- 策略迭代
- 值迭代
- **结构化表示**
- 线性表示

结构化表示

- 维数灾难（curse of dimensionality）：如果状态空间由n个二值变量构成，离散状态的数量为2^n
- 这种指数增长限制了值迭代和策略迭代仅能用在状态变量数量不多的问题上
- 讨论利用状态变量的结构来求解更高维的问题
  - MDP问题的因子化表示
  - 结构化的动态规划

因子化的MDPs

- 因子化的MDPs：使用动态决策网络来压缩表示转移函数和奖赏函数



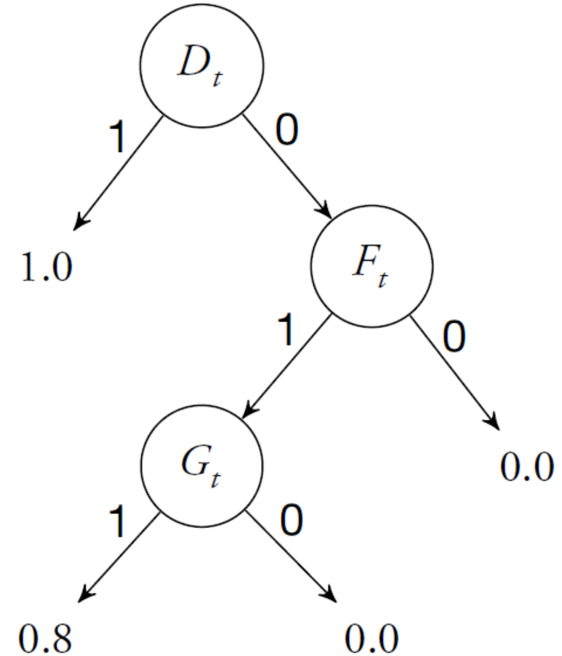
- 示例
  - 因子化状态、行动和奖赏为多个结点
  - 3个状态变量：B、D、G
  - 2个决策变量：A、F
  - 2个奖赏变量：C、E

因子化的MDPs（续）

- 用决策树来压缩表示条件概率分布和奖赏函数

D_t	F_t	G_t	$P(g_{t+1}^1 \mid D_t, F_t, G_t)$
1	1	1	1.0
1	1	0	1.0
1	0	1	1.0
1	0	0	1.0
0	1	1	0.8
0	1	0	0.0
0	0	1	0.0
0	0	0	0.0

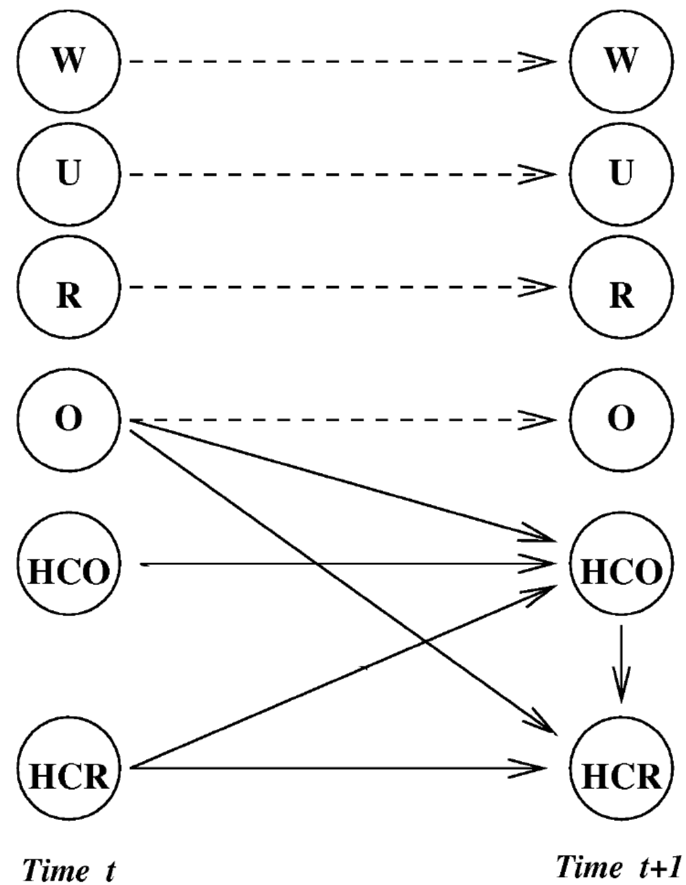
(a) Tabular form



(b) Decision tree form

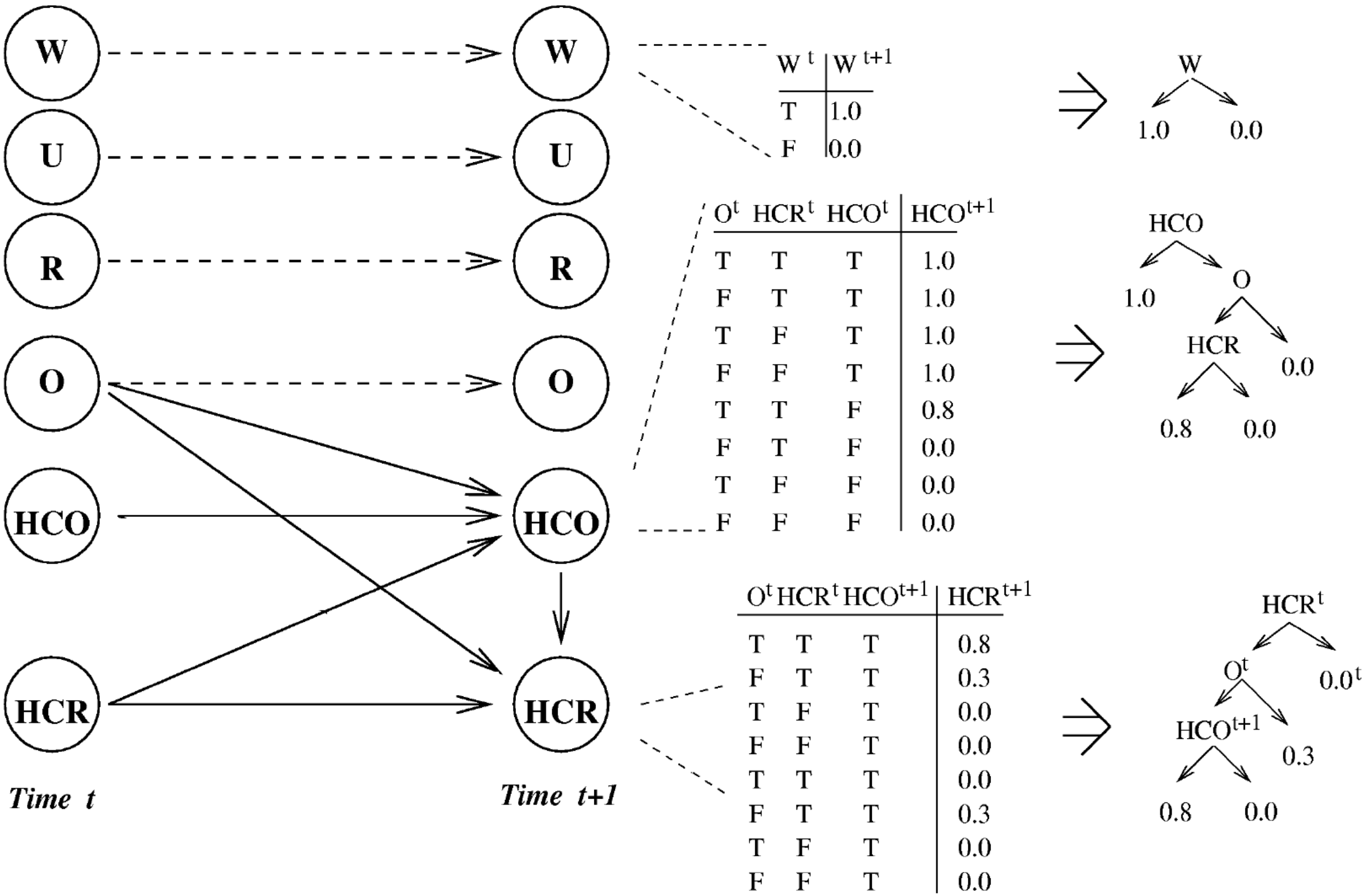
例子：咖啡问题

- 机器人买咖啡并送到主人手中
- 状态变量
  - W：机器人是湿的
  - U：机器人有伞
  - R：正在下雨
  - O：机器人在办公室
  - HCO：主人有咖啡
  - HCR：机器人有咖啡
- 行动
  - Go：移动机器人到不同位置
  - BuyC：买咖啡
  - DelC：机器人把咖啡给主人
  - GetU：机器人拿伞



表示行动DelC的状态转移的贝叶斯网络

条件概率分布的决策树表示

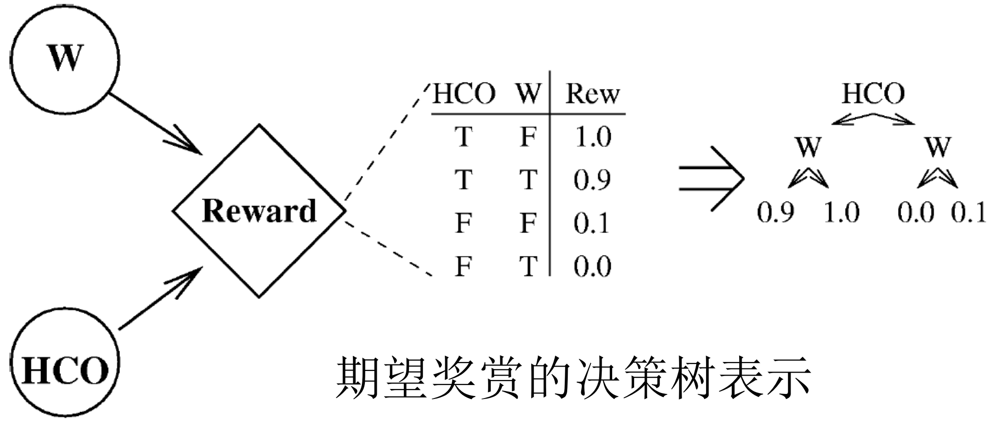


W^t	W^{t+1}
T	1.0
F	0.0

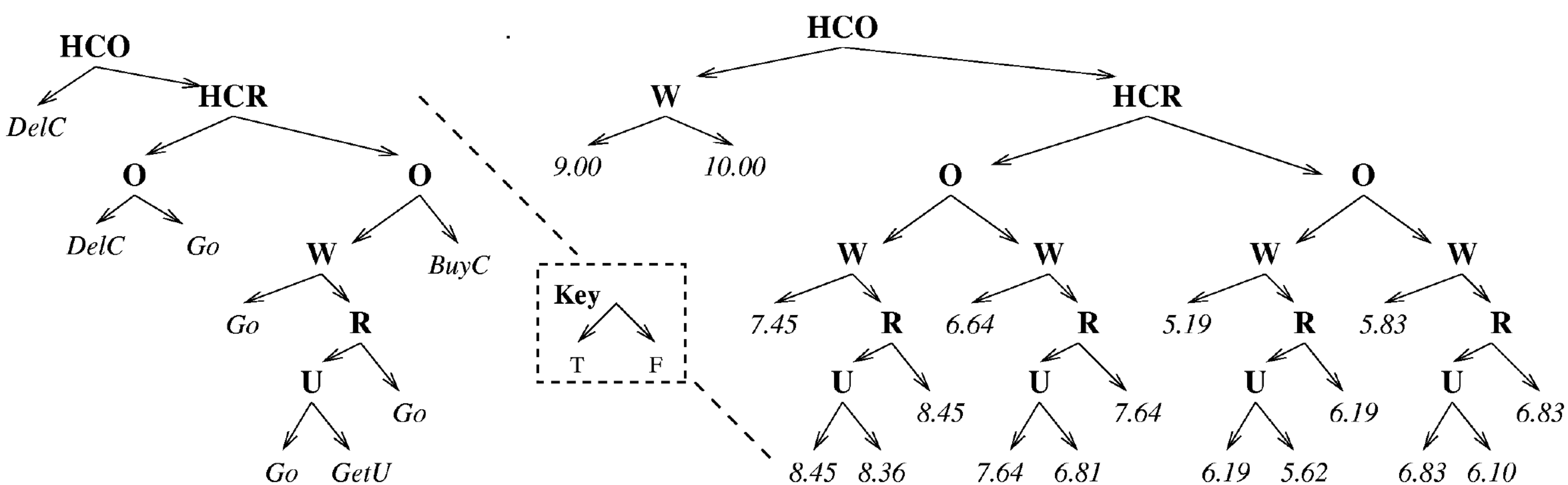
O^t	HCR^t	HCO^t	HCO^{t+1}
T	T	T	1.0
F	T	T	1.0
T	F	T	1.0
F	F	T	1.0
T	T	F	0.8
F	T	F	0.0
T	F	F	0.0
F	F	F	0.0

O^t	HCR^t	HCO^{t+1}	HCR^{t+1}
T	T	T	0.8
F	T	T	0.3
T	F	T	0.0
F	F	T	0.0
T	T	T	0.0
F	T	T	0.3
T	F	T	0.0
F	F	T	0.0

期望奖赏、策略、值函数的决策树表示



期望奖赏的决策树表示



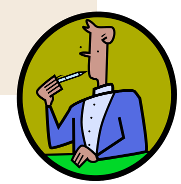
策略的决策树表示

值函数的决策树表示

结构化的动态规划

- 基于表格表示的策略迭代、值迭代
  - 用表格来存储状态转移矩阵、期望奖赏、策略和值函数
- 基于决策树表示的策略迭代、值迭代
  - 用决策树来存储状态转移矩阵、期望奖赏、策略和值函数

当使用决策树表示时，如何用Bellman方程来更新策略或值函数？



基于决策树表示的值迭代

- 值迭代的更新公式

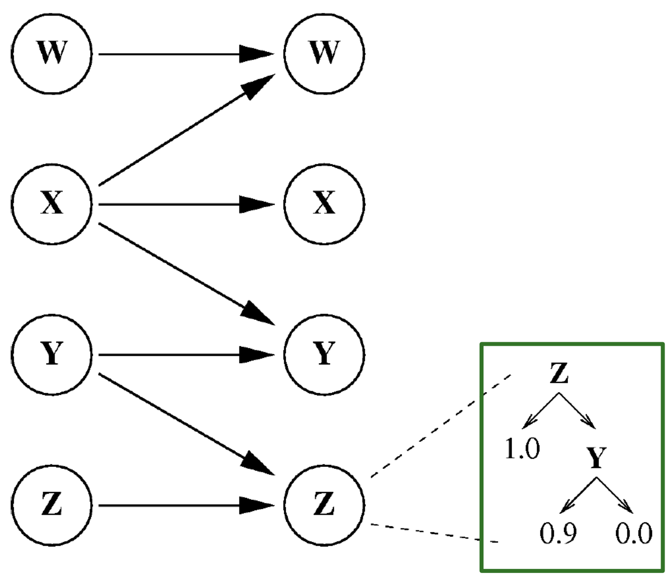
简化为：$T(z' \mid z, y)$

$$U_n(s) \leftarrow \max_a \left(R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) U_{n-1}(s') \right)$$

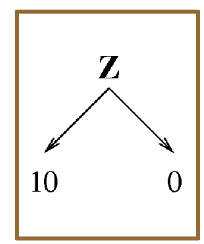
简化为：$R(z)$

令$U_1(s) = R(z)$

二值状态变量



(a) Action Network



(b) Reward Tree

基于决策树表示的值迭代（续）

$$R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) U_1(s')$$

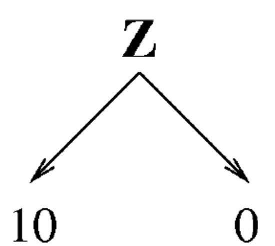
$$\Downarrow$$

$$\mathrm{Tree}(Q_a^2) = R(z) + \gamma \sum_{z'} T(z' \mid z, y)\, R(z')$$

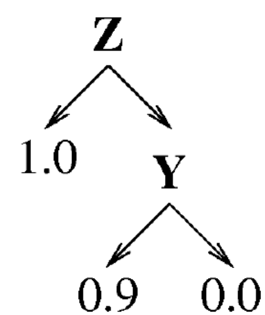
$$= R(z) + \gamma T(z' = 1 \mid z, y) R(z' = 1) + \gamma T(z' = 0 \mid z, y) R(z' = 0)$$

$$= R(z) + \gamma T(z' = 1 \mid z, y) R(z' = 1)$$

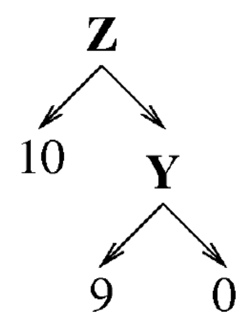
$\gamma = 0.9$



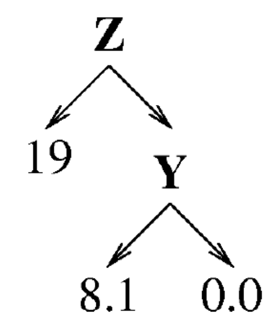
$R(z), U_1(z)$



$T(z' = 1 \mid z, y)$



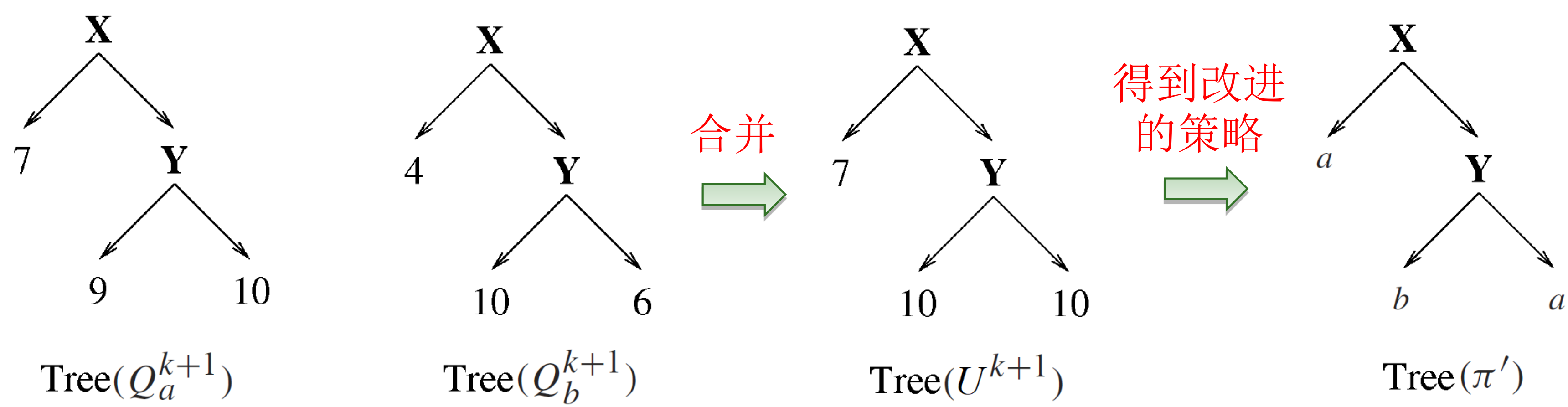
$T(z' = 1 \mid z, y) R(z' = 1)$



$\mathrm{Tree}(Q_a^2)$

基于决策树表示的值迭代（续）

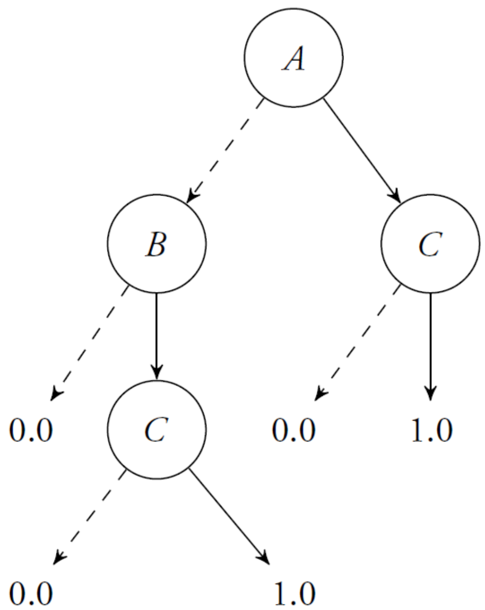
- 最大化：合并两棵Q树，得到改进的策略π'



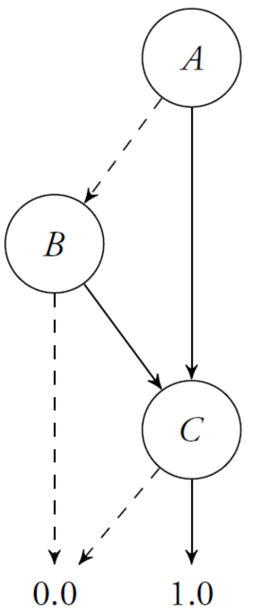
关于结构化的动态规划，参考文献：C. Boutilier, R. Dearden, M. Goldszmidt. **Stochastic Dynamic Programming with Factored Representations**. Artificial Intelligence, 121: 49-107, 2000

决策图

- 把决策树压缩表示成决策图
  - 决策树中所有结点（除根结点）有且只有一个父结点
  - 决策图中的结点可以有多个父结点



(a) Decision tree



(b) Decision diagram

一张条件概率表的决策树表示和决策图表示，其中虚线表示变量测试的结果为假

与决策树中要求有5个叶子结点不同，决策图中仅要求有2个叶子结点

精确动态规划

- 策略迭代
- 值迭代
- 结构化表示
- **线性表示**

带二次奖赏的线性系统

- 求解满足某些条件的连续状态和行动空间的MDPs的最优策略
- 状态转移函数是线性的：

$$T(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) = \mathbf{T}_s \mathbf{s} + \mathbf{T}_a \mathbf{a} + \mathbf{w}$$

矩阵$\mathbf{T}_s$和$\mathbf{T}_a$：基于$\mathbf{s}$和$\mathbf{a}$来确定下一个状态$\mathbf{s}'$的均值

$\mathbf{w}$：均值为0，方差有限的噪声

- 期望奖赏是二次的：

$$R(\mathbf{s}, \mathbf{a}) = \mathbf{s}^\top \underline{\mathbf{R}_s} \mathbf{s} + \mathbf{a}^\top \underline{\mathbf{R}_a} \mathbf{a}$$

$$\boxed{\mathbf{R}_s = \mathbf{R}_s^\top \leq 0} \qquad \boxed{\mathbf{R}_a = \mathbf{R}_a^\top < 0}$$

例子：直流电机

- 直流电机的二阶离散时间模型：

$$T(\mathbf{s}' \mid \mathbf{s}, \mathbf{a}) = \mathbf{T}_s\mathbf{s} + \mathbf{T}_a\mathbf{a}$$

$$\mathbf{s} = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} \quad \mathbf{a} = [a]$$

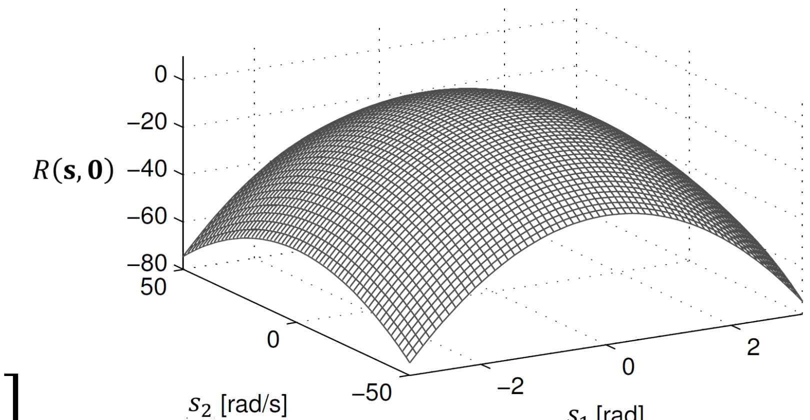
$$\mathbf{T}_s = \begin{bmatrix} 1 & 0.0049 \\ 0 & 0.9540 \end{bmatrix} \quad \mathbf{T}_a = \begin{bmatrix} 0.0021 \\ 0.8505 \end{bmatrix}$$

$s_1 \in [-\pi, \pi]$ rad
$s_2 \in [-16\pi, 16\pi]$ rad/s
$a \in [-10, 10]$ V

- 控制目标：使直流电机稳定在零平衡状态，即$\mathbf{s} = \mathbf{0}$

$$R(\mathbf{s}, \mathbf{a}) = \mathbf{s}^\top \mathbf{R}_s \mathbf{s} + \mathbf{a}^\top \mathbf{R}_a \mathbf{a}$$

$$\mathbf{R}_s = \begin{bmatrix} -5 & 0 \\ 0 & -0.01 \end{bmatrix} \quad \mathbf{R}_a = [-0.01]$$



线性系统：值迭代

- 假设一个有限步数的无折扣奖赏问题，有：

$$U_{h+1}(\mathbf{s}) = \max_{\mathbf{a}} \left(R(\mathbf{s}, \mathbf{a}) + \int_{\mathbf{s}'} \boxed{T(\mathbf{s}' \mid \mathbf{s}, \mathbf{a})} U_h(\mathbf{s}') d\mathbf{s}' \right)$$

概率密度函数

- 由T和R的假设，有：

$$U_{h+1}(\mathbf{s}) = \max_{\mathbf{a}} \left(\mathbf{s}^\top \mathbf{R}_s \mathbf{s} + \mathbf{a}^\top \mathbf{R}_a \mathbf{a} + \int_{\mathbf{w}} \boxed{p(\mathbf{w})} U_h(\mathbf{T}_s \mathbf{s} + \mathbf{T}_a \mathbf{a} + \mathbf{w}) d\mathbf{w} \right)$$

噪声的概率密度函数

- 第1步的最优状态值函数：

$$U_1(\mathbf{s}) = \max_{\mathbf{a}} \left(\mathbf{s}^\top \mathbf{R}_s \mathbf{s} + \mathbf{a}^\top \mathbf{R}_a \mathbf{a} \right) = \mathbf{s}^\top \mathbf{R}_s \mathbf{s}$$

对应的最优行动：$\mathbf{a} = \mathbf{0}$

线性系统：值迭代（续）

- 下面用数学归纳法证明：$u_n(\mathbf{s})$可以写成$\mathbf{s}^\top \mathbf{V}_n \mathbf{s} + q_n$的形式
- 当$n = 1$时，有： $U_1(\mathbf{s}) = \mathbf{s}^\top \mathbf{V}_1 \mathbf{s} + q_1$ $\boxed{\mathbf{V}_1 = \mathbf{R}_s,\ q_1 = 0}$
- 假设$U_h(\mathbf{s}) = \mathbf{s}^\top \mathbf{V}_h \mathbf{s} + q_h$，从而有：

$$U_{h+1}(\mathbf{s}) = \mathbf{s}^\top \mathbf{R}_s \mathbf{s} + \max_{\mathbf{a}} \Big(\mathbf{a}^\top \mathbf{R}_a \mathbf{a} + \qquad \boxed{U_h(\mathbf{T}_s \mathbf{s} + \mathbf{T}_a \mathbf{a} + \mathbf{w})}$$

$$\int_{\mathbf{w}} p(\mathbf{w}) \Big(\underline{(\mathbf{T}_s \mathbf{s} + \mathbf{T}_a \mathbf{a} + \mathbf{w})^\top \mathbf{V}_h (\mathbf{T}_s \mathbf{s} + \mathbf{T}_a \mathbf{a} + \mathbf{w}) + q_h} \Big) d\mathbf{w} \Big)$$

$$\boxed{\int_{\mathbf{w}} p(\mathbf{w}) d\mathbf{w} = 1}$$

$$U_{h+1}(\mathbf{s}) = \mathbf{s}^\top \mathbf{R}_s \mathbf{s} + \mathbf{s}^\top \mathbf{T}_s^\top \mathbf{V}_h \mathbf{T}_s \mathbf{s}$$

$$+ \max_{\mathbf{a}} \Big(\mathbf{a}^\top \mathbf{R}_a \mathbf{a} + 2\mathbf{s}^\top \mathbf{T}_s^\top \mathbf{V}_h \mathbf{T}_a \mathbf{a} + \mathbf{a}^\top \mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a \mathbf{a} \Big) + \int_{\mathbf{w}} p(\mathbf{w}) \Big(\mathbf{w}^\top \mathbf{V}_h \mathbf{w} \Big) d\mathbf{w}$$

线性系统：值迭代（续）

$$
\begin{aligned}
U_{h+1}(\mathbf{s}) =& \mathbf{s}^\top \mathbf{R}_s \mathbf{s} + \mathbf{s}^\top \mathbf{T}_s^\top \mathbf{V}_h \mathbf{T}_s \mathbf{s} \\
& + \underline{\max_{\mathbf{a}} \left(\mathbf{a}^\top \mathbf{R}_a \mathbf{a} + 2\mathbf{s}^\top \mathbf{T}_s^\top \mathbf{V}_h \mathbf{T}_a \mathbf{a} + \mathbf{a}^\top \mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a \mathbf{a} \right)} + \int_{\mathbf{w}} p(\mathbf{w}) \left(\mathbf{w}^\top \mathbf{V}_h \mathbf{w} \right) d\mathbf{w}
\end{aligned}
$$

通过计算这一项关于**a**的导数，使之为0，来求解**a**

$$
\begin{aligned}
\mathbf{0} &= 2\mathbf{R}_a \mathbf{a} + 2\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_s \mathbf{s} + \left(\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a + \left(\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a \right)^\top \right) \mathbf{a} \\
&= 2\mathbf{R}_a \mathbf{a} + 2\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_s \mathbf{s} + 2\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a \mathbf{a}
\end{aligned}
$$

$$\Downarrow$$

$$
\mathbf{a} = -\left(\mathbf{R}_a + \mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a \right)^{-1} \mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_s \mathbf{s}
$$

线性系统：值迭代（续）

- 把$\mathbf{a}$代入$\mathcal{U}_{h+1}(\mathbf{s})$后，有：

$$\mathcal{U}_{h+1}(\mathbf{s}) = \mathbf{s}^\top \mathbf{V}_{h+1}\mathbf{s} + q_{h+1}$$

其中

$$\mathbf{V}_{h+1} = \mathbf{T}_s^\top \left(\mathbf{V}_h - \mathbf{V}_h \mathbf{T}_a \left(\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a + \mathbf{R}_a\right)^{-1} \mathbf{T}_a^\top \mathbf{V}_h\right)\mathbf{T}_s + \mathbf{R}_s$$

$$q_{h+1} = \mathbb{E}_{\mathbf{w}}\left[\mathbf{w}^\top \mathbf{V}_h \mathbf{w}\right]$$

- 为了计算$\mathbf{V}_n$和q_n，先令$\mathbf{V}_0 = 0$和$q_0 = 0$，再使用上述方程迭代
- 知道了$\mathbf{V}_{n-1}$和q_{n-1}后，提取最优的n步策略：

策略矩阵

$$\pi_h(\mathbf{s}) = \boxed{-\left(\mathbf{T}_a^\top \mathbf{V}_{h-1}\mathbf{T}_a + \mathbf{R}_a\right)^{-1} \mathbf{T}_a^\top \mathbf{V}_{h-1}\mathbf{T}_s}\mathbf{s}$$

$\mathcal{U}_h$依赖于$\mathbf{w}$，但π_h不依赖于$\mathbf{w}$

例子：直流电机（续）

$$\mathbf{T}_s = \begin{bmatrix} 1 & 0.0049 \\ 0 & 0.9540 \end{bmatrix} \quad \mathbf{T}_a = \begin{bmatrix} 0.0021 \\ 0.8505 \end{bmatrix} \quad \mathbf{R}_s = \begin{bmatrix} -5 & 0 \\ 0 & -0.01 \end{bmatrix} \quad \mathbf{R}_a = [-0.01]$$

$$\mathbf{V}_{h+1} = \mathbf{T}_s^\top \left(\mathbf{V}_h - \mathbf{V}_h \mathbf{T}_a \left(\mathbf{T}_a^\top \mathbf{V}_h \mathbf{T}_a + \mathbf{R}_a \right)^{-1} \mathbf{T}_a^\top \mathbf{V}_h \right) \mathbf{T}_s + \mathbf{R}_s$$

$$\pi_h(\mathbf{s}) = -\left(\mathbf{T}_a^\top \mathbf{V}_{h-1} \mathbf{T}_a + \mathbf{R}_a \right)^{-1} \mathbf{T}_a^\top \mathbf{V}_{h-1} \mathbf{T}_s \mathbf{s}$$

$$\Downarrow$$

$$\mathbf{V}_1 = \mathbf{R}_s = \begin{bmatrix} -5 & 0 \\ 0 & -0.01 \end{bmatrix} \qquad \pi_1(\mathbf{s}) = [0, 0]\ \mathbf{s}$$

$$\mathbf{V}_2 = \begin{bmatrix} -9.9936 & -0.0195 \\ -0.0195 & -0.0154 \end{bmatrix} \qquad \pi_2(\mathbf{s}) = [-0.6085, -0.4732]\ \mathbf{s}$$

$$\mathbf{V}_3 = \begin{bmatrix} -14.9270 & -0.0451 \\ -0.0451 & -0.0168 \end{bmatrix} \qquad \pi_3(\mathbf{s}) = [-1.7716, -0.5977]\ \mathbf{s}$$

$$\mathbf{V}_4 = \begin{bmatrix} -19.7099 & -0.0724 \\ -0.0724 & -0.0172 \end{bmatrix} \qquad \pi_4(\mathbf{s}) = [-3.1139, -0.6287]\ \mathbf{s}$$

小结：精确动态规划

- 策略迭代
  - 策略评价（Bellman期望方程）、策略改进
- 值迭代
  - Bellman最优方程、 Bellman残差
  - 异步值迭代：高斯-赛德尔值迭代
- 结构化表示
  - 因子化的MDPs、动态决策网络、决策树、决策图
  - 基于决策树表示的值迭代
- 线性表示
  - 带二次奖赏的线性系统、值迭代

内容安排

- 规划
- 马尔科夫决策过程
- 精确动态规划
- **近似动态规划**
- 在线规划
- 直接策略搜索

近似动态规划

- 局部近似
- 全局近似

局部近似

- 假设：相互接近的状态有相似的值

- 如果知道一组状态$s_{1:n}$的值$\lambda_{1:n}$，则可以使用如下方程近似任意状态s的值：

$$U(s)=\sum_{i=1}^{n}\lambda_i\beta_i(s)=\lambda^\top\beta(s)$$

其中，$\beta_{1:n}$是一组权重函数（也称为核），使得$\sum_{i=1}^{n}\beta_i(s)=1$

s与s_i越接近，$\beta_i(s)$的值越大

权值函数与距离函数

- 具体形式：

$$\beta_i(s) = \frac{d(s,s_i)^{-1}}{\sum_{i=1}^{n} d(s,s_i)^{-1}}$$

- 距离函数$d(s,s')$满足：
  - 非负性：$d(s,s') \geq 0$
  - 零等价性：$d(s,s') = 0$，当且仅当$s = s'$
  - 对称性：$d(s,s') = d(s',s)$
  - 三角不等式：$d(a,c) \leq d(a,b) + d(b,c)$

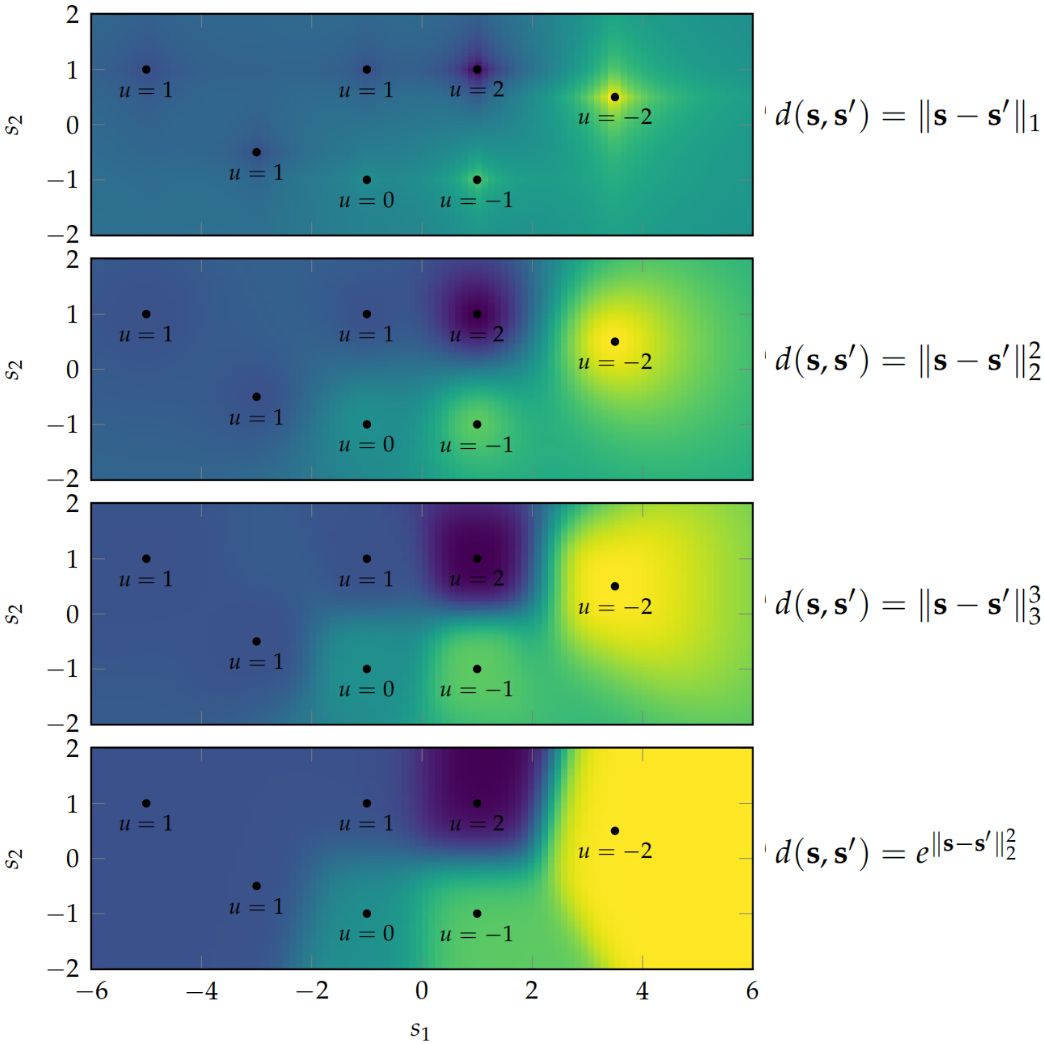
距离函数与值函数

- 给定不同的距离函数
- 值函数$U(s)$的可视化

$$U(s)=\sum_{i=1}^{n}\lambda_i\beta_i(s)=\lambda^\top\beta(s)$$

$$\beta_i(s) = \frac{d(s,s_i)^{-1}}{\sum_{i=1}^{n} d(s,s_i)^{-1}}$$

- 给定一组相同状态$s_{1:n}$的相同值$\lambda_{1:n}$
- 每个u点的位置对应的是s_i，值对应的是λ_i



局部近似（续）

- 算法4.4：通过迭代更新$\boldsymbol{\lambda}$来近似计算最优值函数

Algorithm 4.4 Local approximation value iteration

```
1: function LocalApproximationValueIteration
2:     λ ← 0
3:     loop
4:         for i ← 1 to n
5:             u_i ← max_a [R(s_i, a) + γ Σ_{s'} T(s' | s_i, a) λ^⊤ β(s')]
6:         λ ← u
7:     return λ
```

在本节中，有时把λ_i记为u_i

- 得到一个近似的最优值函数后，提取近似最优策略：

$$\pi(s) \leftarrow \arg\max_a \left(R(s,a) + \gamma \sum_{s'} T(s' \mid s,a) \boldsymbol{\lambda}^\top \beta(s') \right)$$

示例：小车上山

$\mathcal{S}$	continuous
$\mathcal{A}$	discrete
$\dim(\mathcal{S})$	2
$\lvert\mathcal{A}\rvert$	3
γ	1.0

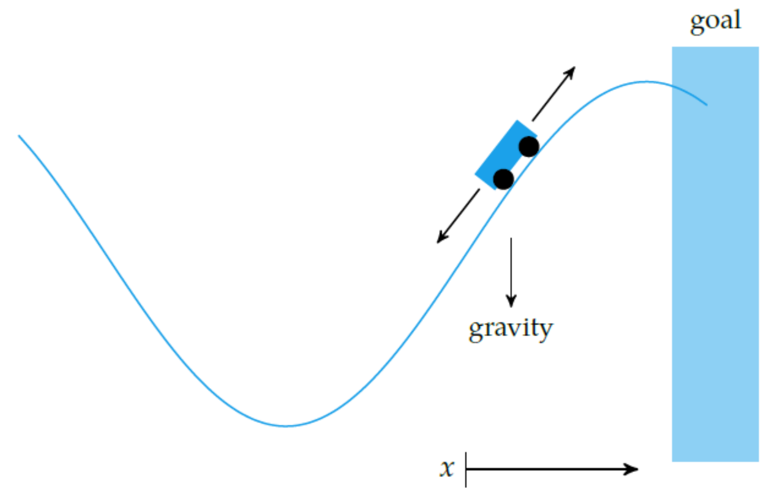
- 状态：小车的位置x和速度v
- 行动：向左加速、向右加速、不加速
- 状态转移函数：

$$v' \leftarrow v + 0.001a - 0.0025\cos(3x)$$

$$x' \leftarrow x + v'$$

其中 $x \in [-1.2, 0.6]$ $v \in [-0.07, 0.07]$

- 期望奖赏函数
  - 如果不在目标状态，则任何转移的奖赏为−1
- 当达到了目标状态，情节结束



小车上山：局部近似

- 值函数

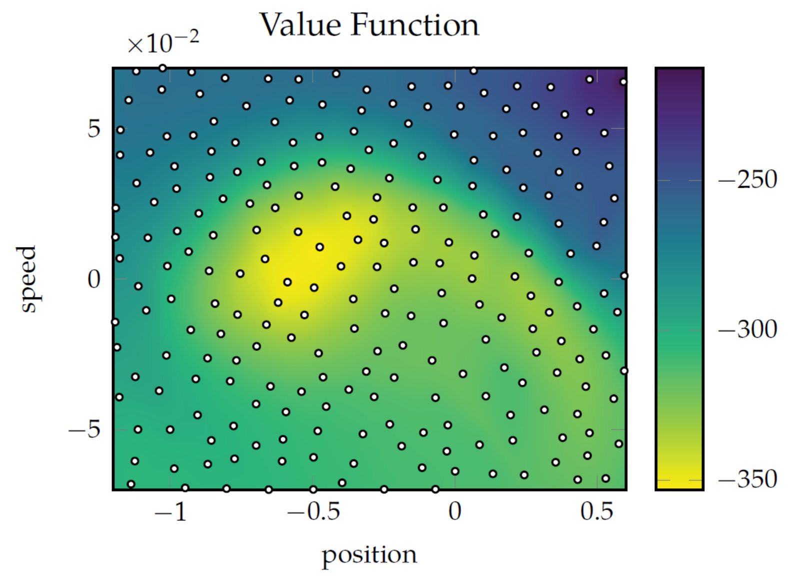
$$U(s)=\sum_{i=1}^{n}\lambda_i\beta_i(s)=\lambda^\top\beta(s)$$

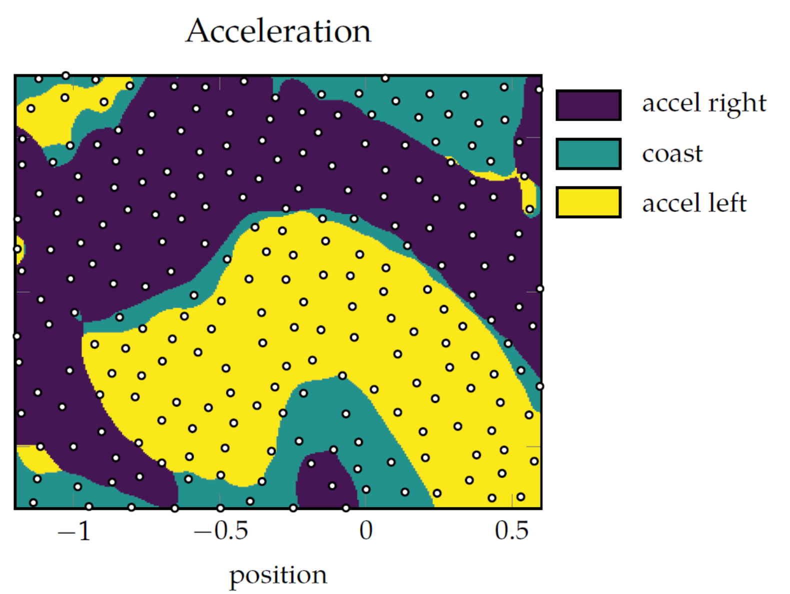
其中 $\beta_i(s)=\dfrac{d(s,s_i)^{-1}}{\sum_{i=1}^{n}d(s,s_i)^{-1}}$

$$d(\mathbf{s},\mathbf{s}')=||\mathbf{s}-\mathbf{s}'||_2+0.1$$

- 策略

$$\pi(s)\leftarrow\arg\max_a\left(R(s,a)+\gamma\sum_{s'}T(s'\mid s,a)\lambda^\top\beta(s')\right)$$



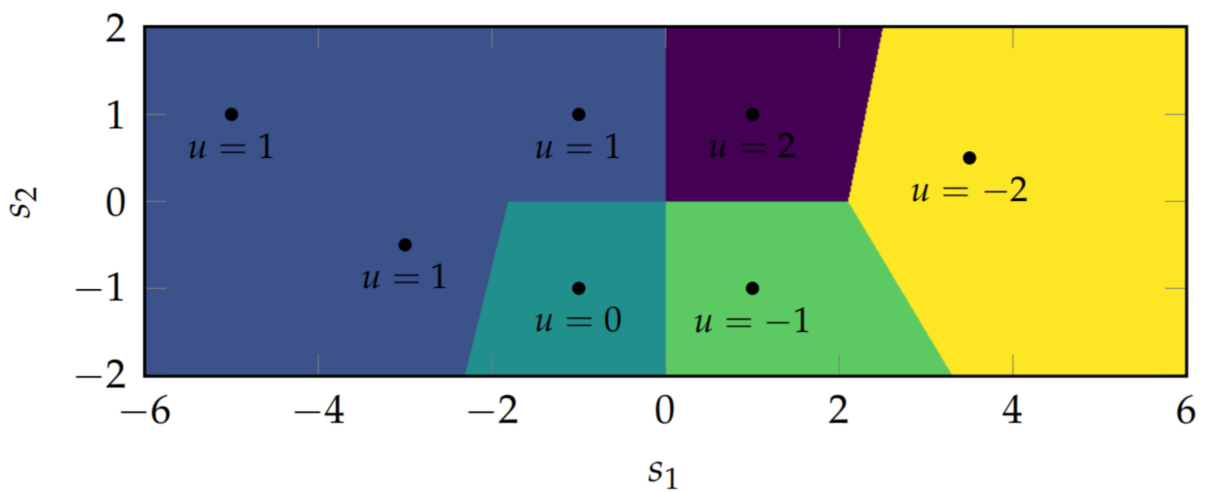


最近邻近似

- 把所有权重赋给与s最接近的状态，得到分段常值函数

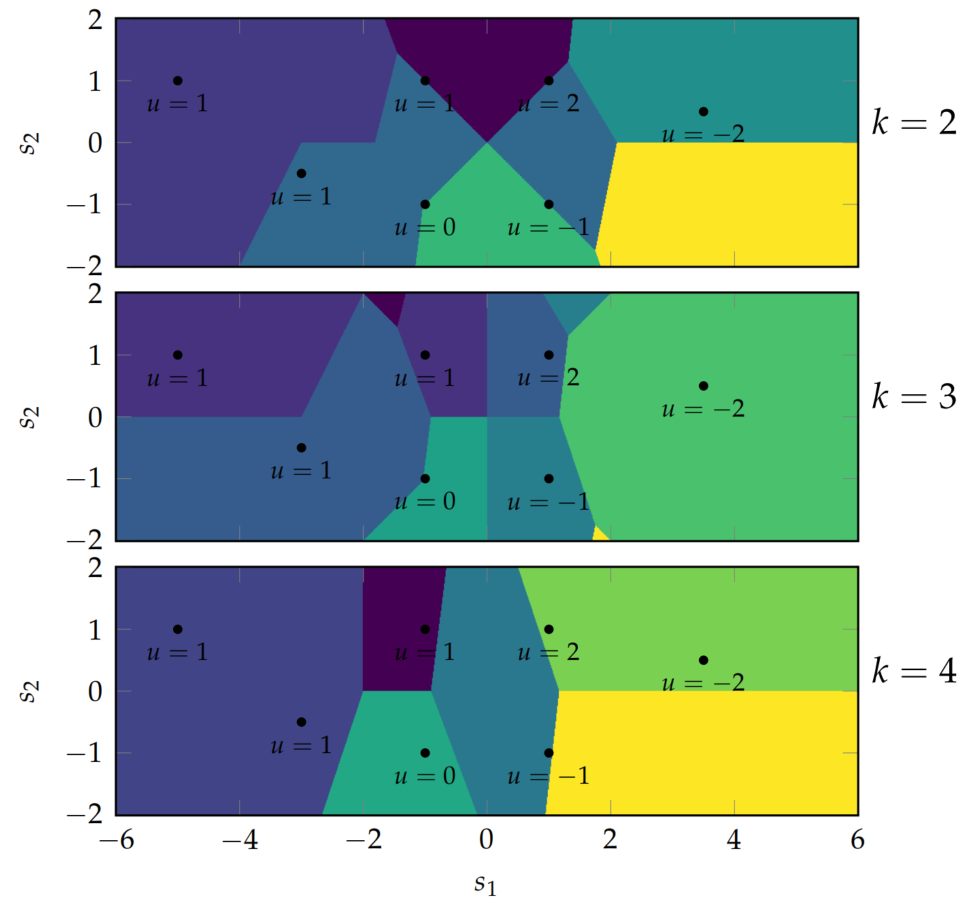
$$U(s) = u_{\arg\min_{i\in 1:n} d(s_i, s)}$$

- 使用欧式距离，用最近邻近似方法得到的值函数



k-最近邻近似

- 对与s最接近的k个状态，每个赋予$\frac{1}{k}$的权重
- 使用欧式距离，用k-最近邻近似方法得到的分段常值函数

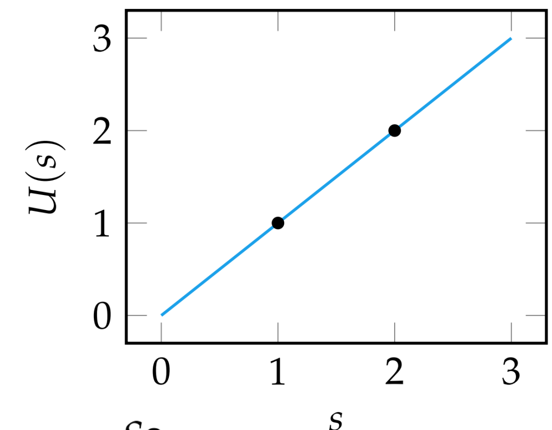


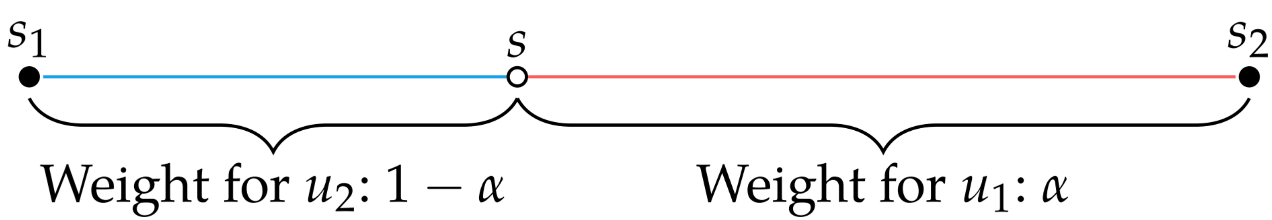
线性插值

- 邻域函数$N(s)$：从$s_{1:n}$中返回一个状态子集
- 如果状态空间是1维的，$N(s) = \{s_1, s_2\}$，则可以使用线性插值（linear interpolation）法计算s处的值：

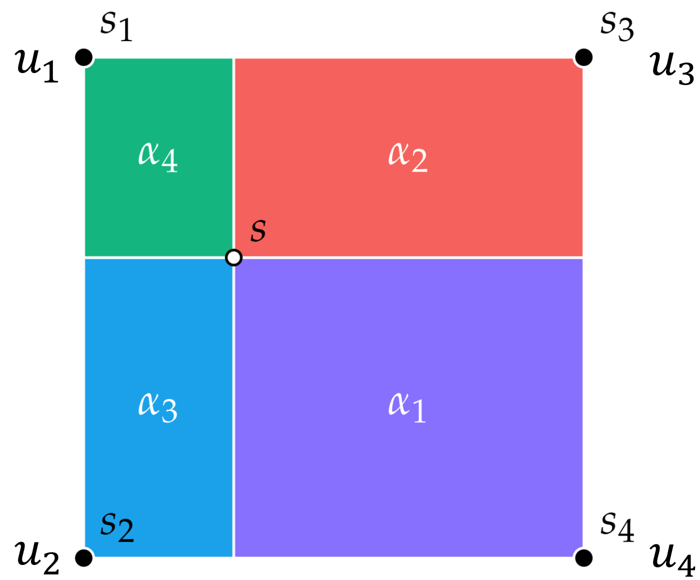
$$U(s) = \alpha u_1 + (1 - \alpha) u_2$$

$$\alpha = (s_2 - s)/(s_2 - s_1)$$





双线性插值



- 如果状态空间是2维的，则可以使用双线性插值（bilinear interpolation）

$$U(s) = \alpha_1 u_1 + \alpha_2 u_2 + \alpha_3 u_3 + \alpha_4 u_4$$

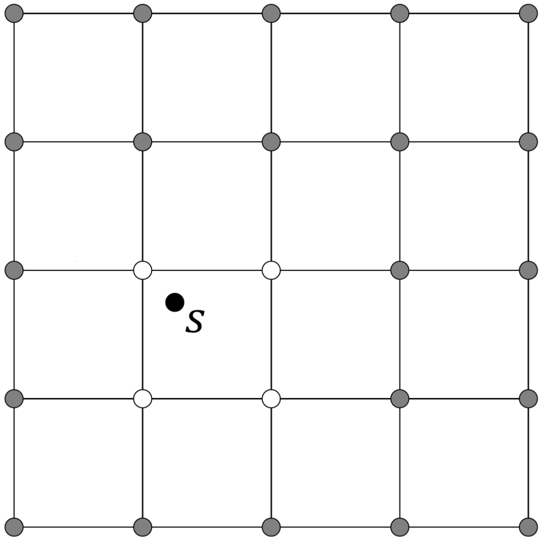
- 令

$$\mathbf{s} = \begin{bmatrix} x \\ y \end{bmatrix} \qquad \mathbf{s}_1 = \begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \quad \mathbf{s}_2 = \begin{bmatrix} x_1 \\ y_2 \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} x_2 \\ y_1 \end{bmatrix}, \quad \mathbf{s}_4 = \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}$$

$$U(s) = \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} u_1 + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} u_2 + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} u_3 + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} u_4$$

多线性插值

- 如果状态空间是高维的，则可以使用多线性插值（multilinear interpolation）
- 使用多维网格来离散化状态空间
  - 网格的顶点对应离散状态
  - $N(s)$为围绕s的矩形格子顶点的集合
- 在d维网格中，有多达2^d个邻居

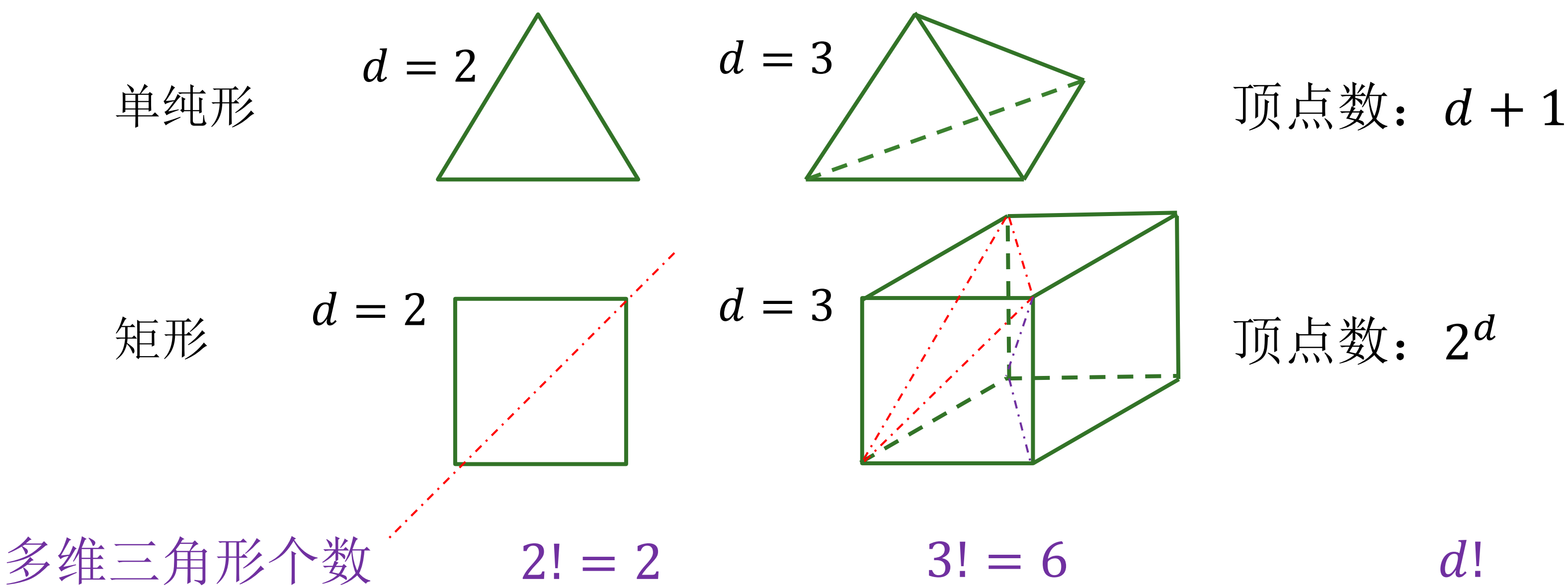


在2维状态空间中基于网格的离散化

- 在高维问题中，用2^d个邻居来计算插值是困难的

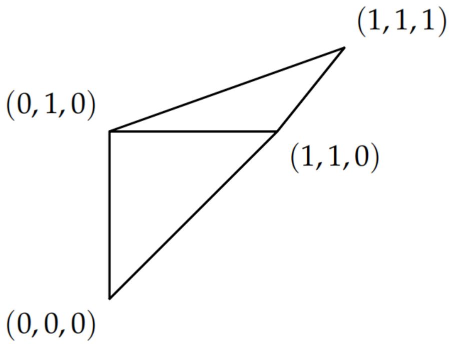
单纯形插值

- 基于单纯形（simplex）的插值
  - 把每个矩形格子分解成$d!$个多维三角形（单纯形）
  - 仅需要对由$d+1$个顶点构成的单纯形进行插值



单纯形插值：呈状态空间维数的线性增长
多线性插值：呈状态空间维数的指数次方增长

单纯形插值的例子



- 计算各个顶点的值在状态$\mathbf{s}'$的权重

$$\begin{bmatrix} s_1' \\ s_2' \\ s_3' \end{bmatrix} = w_1 \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} + w_2 \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + w_3 \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} + w_4 \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

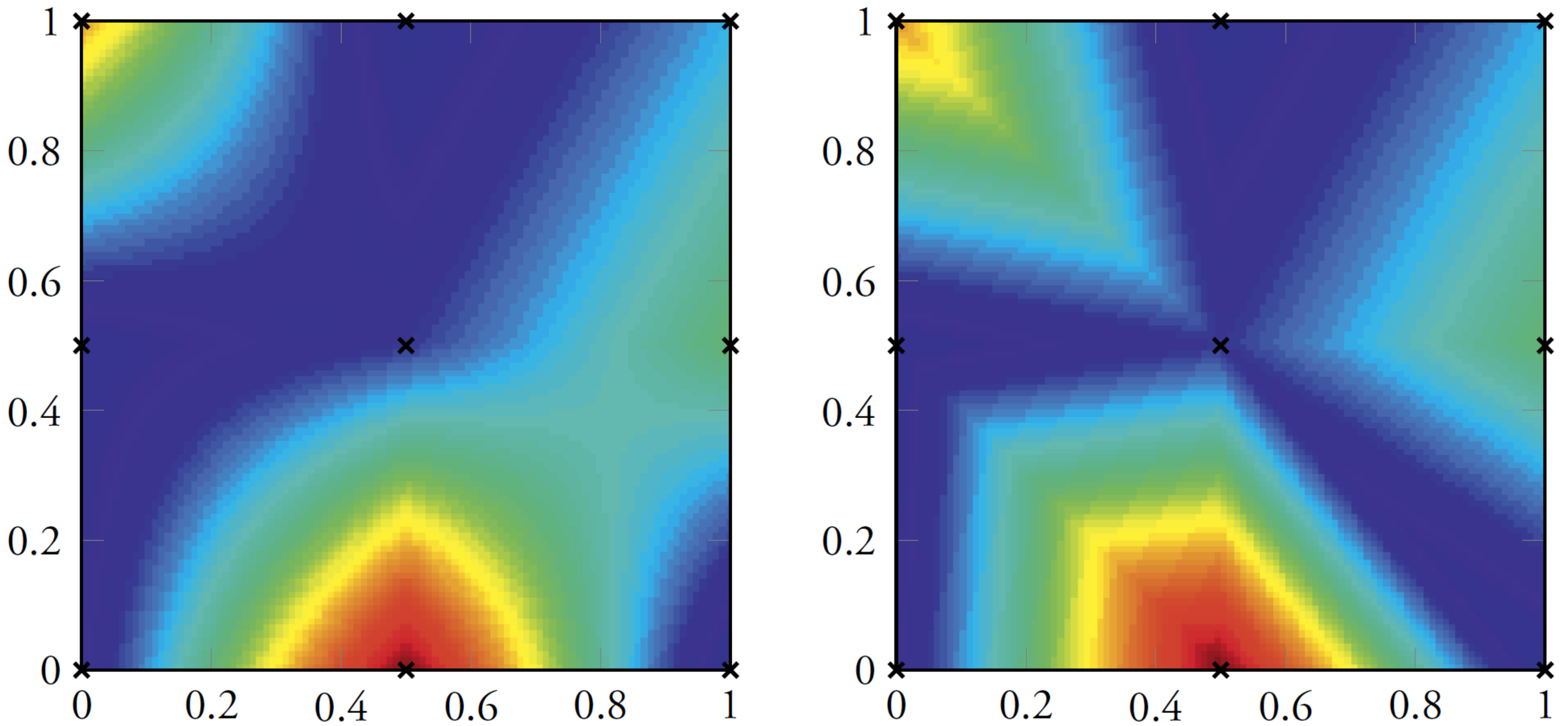
$$w_4 = s_3' \qquad w_3 = s_1' - w_4 \qquad w_2 = s_2' - w_3 - w_4$$

如果 $\mathbf{s}' = [0.3, 0.7, 0.2]^\top$

由$\sum_{i=1}^{4} w_i = 1$，可得

$$w_4 = 0.2 \qquad w_3 = 0.1 \qquad w_2 = 0.4 \qquad w_1 = 0.3$$

双线性（矩形）插值 vs. 单纯形插值



(a) Rectangular interpolation.　　(b) Simplex interpolation.

近似动态规划

- 局部近似
- 全局近似

全局近似

- 特点：用一个固定的参数集合$\lambda_{1:m}$来近似定义值函数
- **基于线性回归的全局近似**
  - 值函数：参数和基函数的线性组合

$$U(s)=\sum_{i=1}^{m}\lambda_i\beta_i(s)=\lambda^{\top}\beta(s)$$

与局部近似的$U(s)$有相同的形式，但有不同的解释

  - $\lambda_{1:m}$不与离散状态的值对应
  - 基函数$\beta_{1:m}$不必与距离度量相关，之和不必为1
- 常见的回归目标：最小化和平方误差（sum-squared error）

$$\sum_{i=1}^{n}\left(\lambda^{\top}\beta(s_i)-u_i\right)^2$$

- 线性最小二乘回归：通过简单地矩阵运算来计算使和平方误差最小的$\boldsymbol{\lambda}$

基于线性回归的值迭代

- 算法4.5：将线性回归融入到值迭代中

Algorithm 4.5 Linear regression value iteration

function LinearRegressionValueIteration
  $\lambda \leftarrow \mathbf{0}$
  loop
    for $i \leftarrow 1$ **to** n
      $u_i \leftarrow \max_a [R(s_i, a) + \gamma \sum_{s'} T(s' \mid s_i, a) \boldsymbol{\lambda}^\top \beta(s')]$
      $\boxed{\lambda_{1:m} \leftarrow \textsc{Regress}(\beta, s_{1:n}, u_{1:n})}$
  return λ

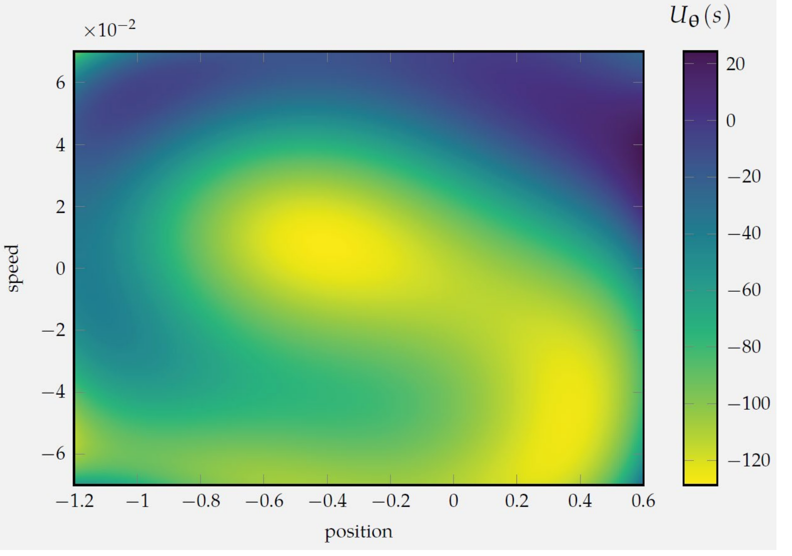
- 找到$\boldsymbol{\lambda}$，使之能在使用基函数$\boldsymbol{\beta}$时，最好地估计$s_{1:n}$上的目标值$u_{1:n}$

小车上山：使用多项式基函数的线性近似

- 一组多项式基函数

$$
\boldsymbol{\beta}(s) = \begin{matrix} [1, & & & & & & \\ x, & v, & & & & & \\ x^2, & xv, & v^2, & & & & \\ x^3, & x^2v, & xv^2, & v^3, & & & \\ x^4, & x^3v, & x^2v^2, & xv^3, & v^4, & & \\ x^5, & x^4v, & x^3v^2, & x^2v^3, & xv^4, & v^5, & \\ x^6, & x^5v, & x^4v^2, & x^3v^3, & x^2v^4, & xv^5, & v^6] \end{matrix}
$$

- 近似值函数：拟合来自一个专家策略的一组状态-行动对



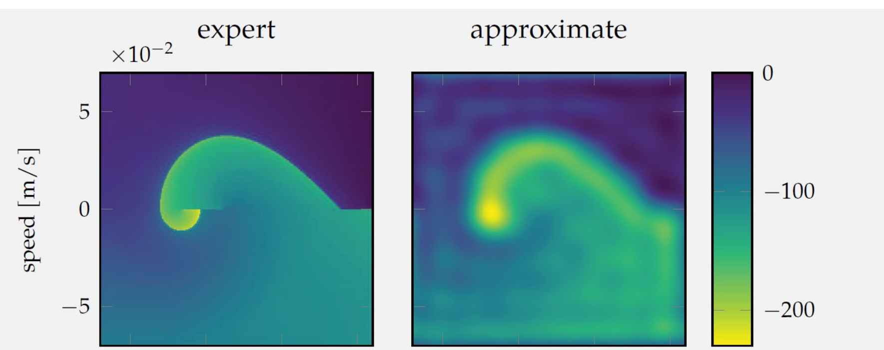
小车上山：使用傅里叶基函数的线性近似

- 一组傅里叶基函数

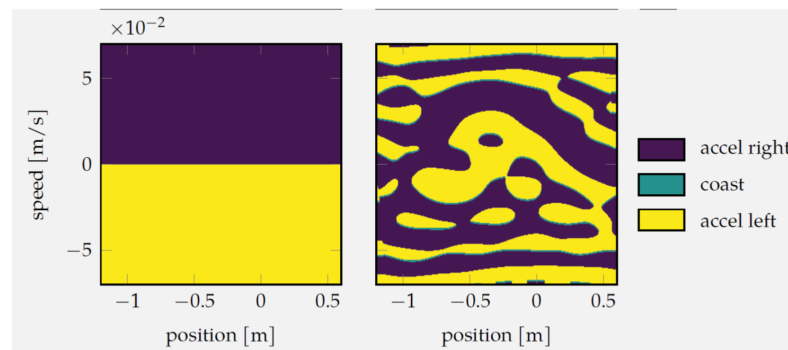
$$b_0(x) = 1/2$$
$$b_{s,i}(x) = \sin(2\pi i x / T) \quad \text{for} \quad i = 1, 2, \ldots$$
$$b_{c,i}(x) = \cos(2\pi i x / T) \quad \text{for} \quad i = 1, 2, \ldots$$

T为函数的周期

- 近似值函数



- 策略



线性插值 vs. 线性回归

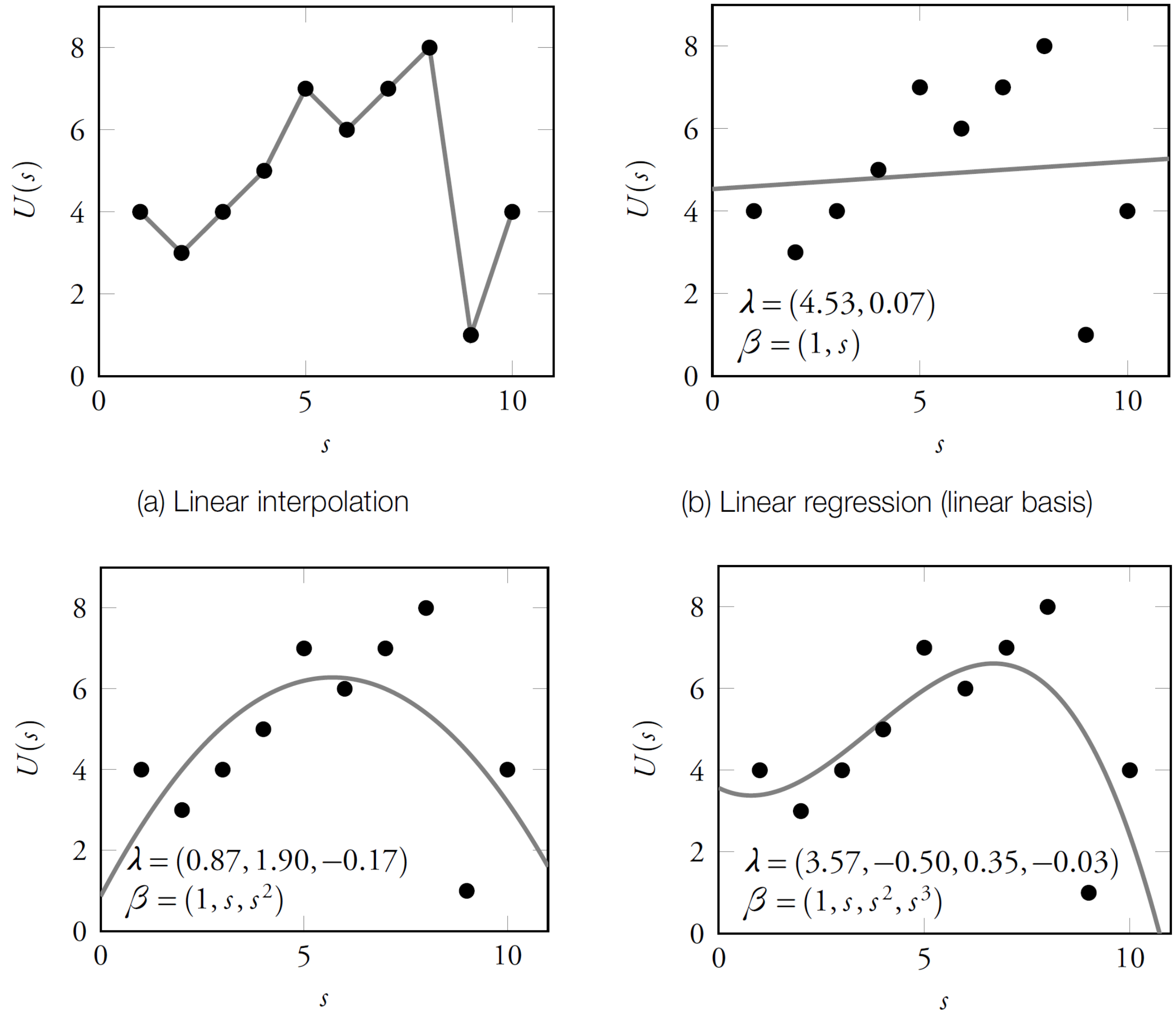
使用不同的基函数

$s_{1:10}$：在1维状态空间上均匀放置的状态

$u_{1:10}$：使用动态规划得到的目标值

- 其他基函数
  - $\sin(s)$
  - e^s

更多基函数会让函数值在已知状态上拟合得更好，但在未知状态上拟合效果可能会更差



(a) Linear interpolation (b) Linear regression (linear basis)

(c) Linear regression (quadratic basis) (d) Linear regression (cubic basis)

小结：近似动态规划

- 局部近似
  - $U(s)$的近似是值估计$\lambda_{1:n}$和权重函数$\beta_{1:n}$的线性组合：

$$U(s)=\sum_{i=1}^{n}\lambda_i\beta_i(s)=\boldsymbol{\lambda}^{\top}\boldsymbol{\beta}(s) \qquad \sum_{i=1}^{n}\beta_i(s)=1$$

  - k-最近邻近似、线性（双线性、多线性）插值、单纯形插值

- 全局近似
  - $U(s)$的近似是权重参数和基函数的线性组合：

$$U(s)=\sum_{i=1}^{m}\lambda_i\beta_i(s)=\boldsymbol{\lambda}^{\top}\boldsymbol{\beta}(s)$$

    - 权重参数$\lambda_{1:m}$不与离散状态的值对应
    - 基函数$\beta_{1:m}$不必与距离度量相关，之和不必为1
  - 基于线性回归的值迭代

课后练习4.6

提示：可以使用Matlab，套用公式算出结果即可

- 考虑一个连续的MDP。状态由位置x和速度v构成，即$\mathbf{s}=\begin{bmatrix} x \\ v \end{bmatrix}$。行动由加速度$a$构成，其每个时间步$\Delta t=1$执行一次。奖赏函数为如下二次奖赏：

$$R(\mathbf{s},a)=-x^2-v^2-0.5a^2$$

即$\mathbf{R}_s=-\mathbf{I}$，$\mathbf{R}_a=-[0.5]$。转移函数为：

$$\begin{bmatrix} x' \\ v' \end{bmatrix}=\begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ v \end{bmatrix}+\begin{bmatrix} 0.5\Delta t^2 \\ \Delta t \end{bmatrix}[a]+\mathbf{w}$$

其中，$\mathbf{w}$服从均值为$\mathbf{0}$，协方差矩阵为$0.1\mathbf{I}$的多元高斯分布。

该系统的控制目标是达到零平衡状态$\mathbf{s}=\mathbf{0}$。试求出一个从$\mathbf{s}_0=\begin{bmatrix} -10 \\ 0 \end{bmatrix}$开始的最优5步策略。

课后练习4.7

- 给定如下3个状态：

$$\mathbf{s}_1 = \begin{bmatrix} 4 \\ 5 \end{bmatrix}, \quad \mathbf{s}_2 = \begin{bmatrix} 2 \\ 6 \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} -1 \\ -1 \end{bmatrix}$$

以及它们对应的值：

$$\mathcal{U}(\mathbf{s}_1) = 2，\mathcal{U}(\mathbf{s}_2) = 10，\mathcal{U}(\mathbf{s}_3) = 30$$

分别使用L_1、L_2 、L_∞距离度量，计算状态$\mathbf{s} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$的2-最近邻局部近似值。

课后练习4.8

- 考虑如下4个状态：

$$\mathbf{s}_1 = \begin{bmatrix} 0 \\ 5 \end{bmatrix}, \quad \mathbf{s}_2 = \begin{bmatrix} 0 \\ 25 \end{bmatrix}, \quad \mathbf{s}_3 = \begin{bmatrix} 1 \\ 5 \end{bmatrix}, \quad \mathbf{s}_4 = \begin{bmatrix} 1 \\ 25 \end{bmatrix}$$

以及采样状态$\mathbf{s} = \begin{bmatrix} 0.7 \\ 10 \end{bmatrix}$，写出$\mathcal{U}(s)$的双线性插值方程。