16.2

Chris considers four used cars before buying the one with maximum expected utility. Pat considers ten cars and does the same. All other things being equal, which one is more likely to have the better car? Which is more likely to be disappointed with their car’ s quality? By how much (in terms of standard deviations of expected quality)?

（1）Pat 更有可能买到更好的车。假设汽车质量服从某种分布，抽取 n 个样本并选择最大值，当 n 增加时，即可选择样本的范围增大时，最大值的期望也会增加。因此，Pat（n=10）的车的期望质量会高于 Chris（n=4）的车的期望质量。

（2）Pat 更可能对车子的质量感到失望。

决策者选择车时，是基于带有噪声的效用进行选择的，而非车的真实质量。

设真实质量 $Q\sim N(\mu, \sigma^2)$，带有噪声的效用 $S=Q+\epsilon$，其中随机误差 $\epsilon\sim N(0,\tau^2)$且独立于 Q。

决策者选择样本中 S 最大的车，预期质量 U=S，定义失望 D=S-Q=ϵ，即质量的高估程度。

当样本数量 n 增加时，选中的车更可能因正向噪声误差（$\epsilon>0$）而被高估，因此真实质量低于预期的概率更高。

（3）预期质量的标准差$\sigma_U=\sqrt{\sigma 2+\tau 2}$

平均失望 E[ϵ| 最大的 S]=(1-k)E[Sn]，其中 $k=\frac{\sigma^2}{\sigma^2+\tau^2}$为信噪比参数，$E[Sn]=cn\times\sigma_U$，c n 是标准正态分布下 n 个样本最大值的期望系数。

对于标准正态分布：c4≈1.029，c10≈1.539

不妨设$\sigma 2=\tau 2$，即 k=0.5

$E[D_{Pat}]=(1-k)\times c10\times\sigma_U=0.7695\sigma_U$

$E[D_{Chris}]=(1-k)\times c4\times\sigma_U=0.5145\sigma_U$

$E[D_{Pat}]-E[D_{Chris}]=0.255\sigma_U$

17.1

For the 4×3 world shown in Figure 17.1, calculate which squares can be reached from (1,1) by the action sequence [Right,Right,Right,Up,Up] and with what probabilities. Explain how this computation is related to the prediction task (see Section 14.2.1 for a hidden Markov model.)

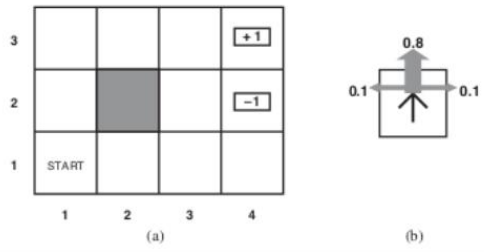


Figure 17.1 (a) A simple 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision with a wall results in no movement. The two terminal states have reward +1 and −1, respectively, and all other states have a reward of −0.04.

（1）以下坐标表示均为(行序号， 列序号)

状态转移矩阵（每一列为列状态转移到行状态的概率分布）

Right	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)
(1,1)	0.1				0.1						
(1,2)	0.8	0.2									
(1,3)		0.8	0.1			0.1					
(1,4)	0.1		0.8	0.9			0.1				
(2,1)					0.8			0.1			
(2,3)			0.1							0.1	
(2,4)				0.1		0.8	0.8				0.1
(3,1)					0.1			0.1			
(3,2)								0.8	0.2		
(3,3)						0.1			0.8	0.1	
(3,4)							0.1			0.8	0.9

Up	(1,1)	(1,2)	(1,3)	(1,4)	(2,1)	(2,3)	(2,4)	(3,1)	(3,2)	(3,3)	(3,4)
(1,1)	0.1	0.1									
(1,2)	0.1	0.8	0.1								
(1,3)		0.1		0.1							
(1,4)			0.1	0.1							
(2,1)	0.8				0.2						
(2,3)			0.8			0.1	0.1				
(2,4)				0.8		0.1	0.1				
(3,1)					0.8			0.9	0.1		
(3,2)								0.1	0.8	0,1	
(3,3)						0.8			0.1	0,8	0,1
(3,4)							0.8			0,1	0,9

状态向量表（表格中为各步后处于各状态的概率分布）

	Start	Right	Right	Right	Up	Up
(1,1)	1	0.1	0.01	0.001	0.0057	0.00708
(1,2)		0.8	0.24	0.056	0.0651	0.05933
(1,3)			0.65	0.202	0.0668	0.01465
(1,4)		0.1	0.1	0.612	0.0814	0.01482
(2,1)					0.0008	0.00472
(2,3)				0.065	0.1699	0.12022
(2,4)			0.01	0.018	0.4979	0.1319
(3,1)						0.00064
(3,2)						0.00521
(3,3)					0.0521	0.17913
(3,4)				0.001	0.0153	0.4173

（2）在隐马尔可夫模型（HMM）中，预测过程即将状态概率向量乘以转移矩阵。这里特殊的地方在于，Up 和 Right 两种动作对应着不同的转移矩阵。

17.9

This exercise considers two-player MDPs that correspond to zero-sum, turn-taking games like those in Chapter Adversarial Search. Let the players be A and B, and let $R(s)$ be the reward for player A in state s. (The reward for B is always equal and opposite.)

1. Let $U_A(s)$ be the utility of state s when it is A's turn to move in s, and let $U_B(s)$ be the utility of state s when it is B's turn to move in s. All rewards and utilities are calculated from A's point of view (just as in a minimax game tree). Write down Bellman equations defining $U_A(s)$ and $U_B(s)$.
2. Explain how to do two-player value iteration with these equations, and define a suitable termination criterion.
3. Consider the game described in Figure 5.17 on page . Draw the state space (rather than the game tree), showing the moves by A as solid lines and moves by B as dashed lines. Mark each state with $R(s)$. You will find it helpful to arrange the states (s_A, s_B) on a two-dimensional grid, using s_A and s_B as "coordinates."
4. Now apply two-player value iteration to solve this game, and derive the optimal policy.

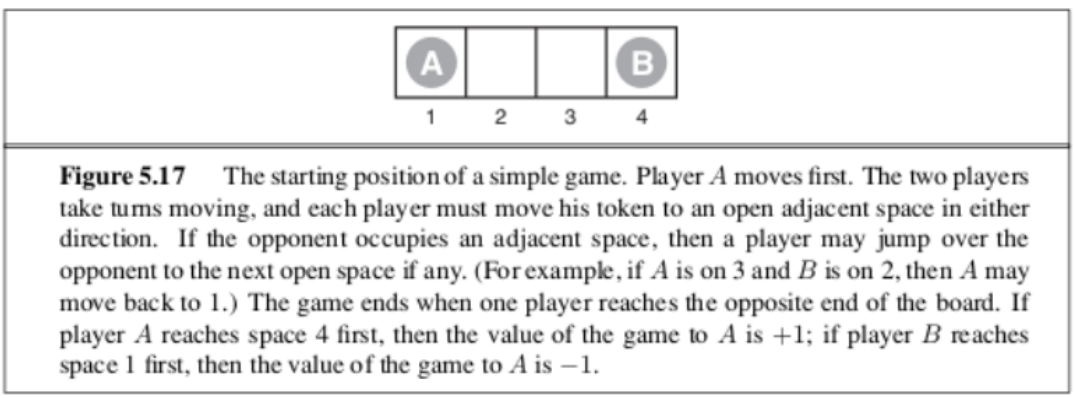


Figure 5.17 The starting position of a simple game. Player A moves first. The two players take turns moving, and each player must move his token to an open adjacent space in either direction. If the opponent occupies an adjacent space, then a player may jump over the opponent to the next open space if any. (For example, if A is on 3 and B is on 2, then A may move back to 1.) The game ends when one player reaches the opposite end of the board. If player A reaches space 4 first, then the value of the game to A is $+1$; if player B reaches space 1 first, then the value of the game to A is -1.

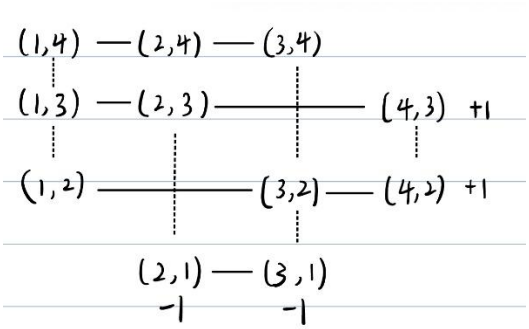
1. U_A 的贝尔曼方程为 $U_A(s)=R(s)+\max_a \sum_{s'} P(s'|a,s)U_B(s')$

   U_B 的贝尔曼方程为 $U_B(s)=R(s)+\max_a \sum_{s'} P(s'|a,s)U_A(s')$

2. 价值迭代过程：进行值迭代时，需将每个方程转化为贝尔曼更新公式，并交替应用这些更新公式，对所有状态进行更新。

终止标准：迭代过程在某个智能体的效用向量与其前一次的效用向量完全相同时终止。

3. 状态空间如图：



4. 价值迭代过程如下表：

	(1,4)	(2,4)	(3,4)	(1,3)	(2,3)	(4,3)	(1,2)	(3,2)	(4,2)
U_A	0	0	0	0	0	+1	0	0	+1
U_B	0	0	0	0	-1	+1	0	-1	+1
U_A	0	0	0	-1	+1	+1	-1	+1	+1
U_B	-1	+1	+1	-1	-1	+1	-1	-1	+1
U_A	+1	+1	+1	-1	+1	+1	-1	+1	+1
U_B	-1	+1	+1	-1	-1	+1	-1	-1	+1

玩家 A、B 的最佳策略如下：

	(1,4)	(2,4)	(3,4)	(1,3)	(2,3)	(4,3)	(1,2)	(3,2)	(4,2)	(2,1)	(3,1)
π^*_A	(2,4)	(3,4)	(2,4)	(2,3)	(4,3)		(3,2)	(4,2)			
π^*_B	(1,3)	(2,3)	(3,2)	(1,2)	(2,1)		(1,3)	(3,1)			

17.19

In the children's game of rock–paper–scissors each player reveals at the same time a choice of rock, paper, or scissors. Paper wraps rock, rock blunts scissors, and

scissors cut paper. In the extended version rock–paper–scissors–fire–water, fire beats rock, paper, and scissors; rock, paper, and scissors beat water; and water beats fire. Write out the payoff matrix and find a mixed-strategy solution to this game.

收益矩阵：

A vs B	R	P	S	F	W
R	0, 0	−1, 1	1, −1	−1, 1	1, −1
P	1, −1	0, 0	−1, 1	−1, 1	1, −1
S	−1, 1	1, −1	0, 0	−1, 1	1, −1
F	1, −1	1, −1	1, −1	0, 0	−1, 1
W	−1, 1	−1, 1	−1, 1	1, −1	0, 0

考虑一个混合策略使得对手无论选哪个选项，期望收益都一样。

设 $p_R=p_P=p_S=a$（由对称性），$p_F=b$，$p_W=c$，且 $3a+b+c=1$

由于 B 选哪个选项，期望收益都一样：$b-c = -3a+c = 3a-b$

解得 $a=1/9$, $b=c=1/3$

即出剪刀石头布的概率均为 1/9，出水或火的概率为 1/3，是一个混合策略 NASH 均衡解

18.29

Suppose you had a neural network with linear activation functions. That is, for each unit the output is some constant c times the weighted sum of the inputs.

1. Assume that the network has one hidden layer. For a given assignment to the weights $\mathbf{w}$, write down equations for the value of the units in the output layer as a function of $\mathbf{w}$ and the input layer $\mathbf{x}$, without any explicit mention of the output of the hidden layer. Show that there is a network with no hidden units that computes the same function.
2. Repeat the calculation in part (a), but this time do it for a network with any number of hidden layers.
3. Suppose a network with one hidden layer and linear activation functions has n input and output nodes and h hidden nodes. What effect does the transformation in part (a) to a network with no hidden layers have on the total number of weights? Discuss in particular the case $h \ll n$.

1. （1）设输入层有 n 各节点，$x\in R^n$；隐藏层有 h 个节点，权重矩阵 W_1 属于 $R^{h\times n}$；输出层有 m 个节点，权重矩阵 W_2 属于 $R^{m\times h}$

则隐藏层的输出为 $h=W_1x$，输出层输入为 $y=W_2h=W_2(W_1x)$

设 $W=W_2W_1$，那么输出层（不显式包含隐藏层的输出的）方程为 $y=Wx$

(2)证明:一个无隐藏层的网络(即单层感知机)可直接使用权重矩阵 $W\in R^{m\times n}$ 计算 $y=Wx$，与上述函数相同。因此，存在无隐藏单元的网络可计算相同函数。

2. 同样使用 1.中的简化方法可以直接将一个 n 层网络简化为一个 d 单层网络。即线性激活函数限制了神经网络只能表示线性函数。

3. 原始网络共有 2hn 个权重；而简化后的网络有 n^2 个权重。当 $h\ll n$ 时，原始网络的权重要少得多，因此能更简洁地表示输入/输出映射。因此这样的网络比简化网络学得更快。