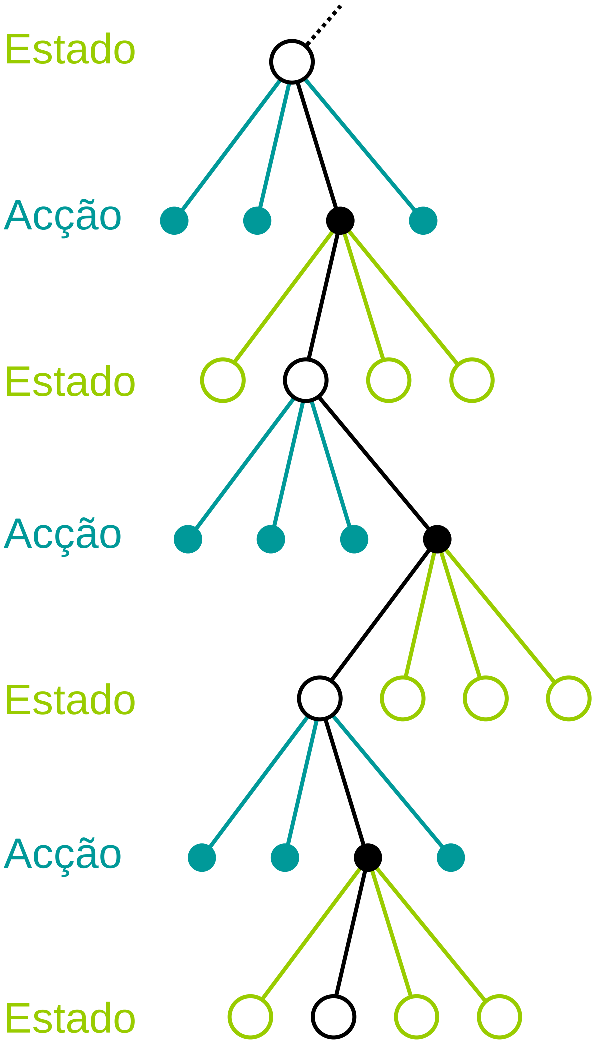
PROCESSOS DE DECISÃO SEQUENCIAL

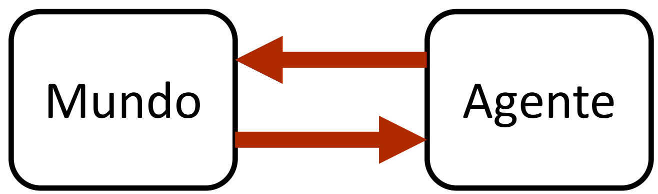
Luís Morgado

2021

PROCESSOS DE DECISÃO SEQUENCIAL

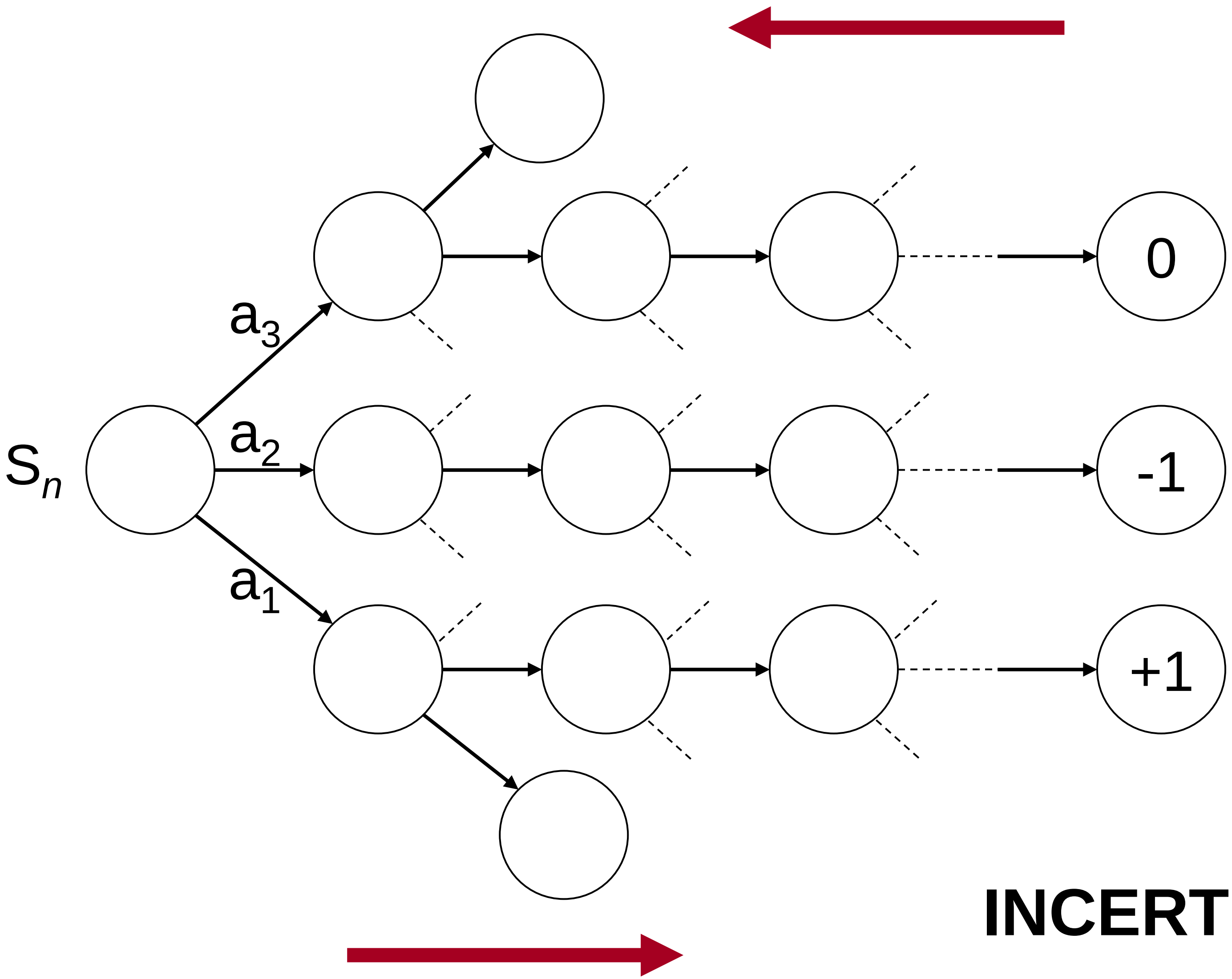


Como prever e controlar o desenrolar da interacção entre agente e ambiente ao longo do tempo para um **objectivo de longo prazo**?



[Sutton & Barto, 1998]

PROCESSOS DE DECISÃO SEQUENCIAL



INCERTEZA

Processos de Decisão Sequencial

- Problema da decisão ao longo do tempo
  - Utilidade de uma acção depende de uma sequência de decisões
  - Possibilidade de ganhos e perdas
  - Incerteza na decisão
  - Efeito cumulativo

Ambiente não determinista

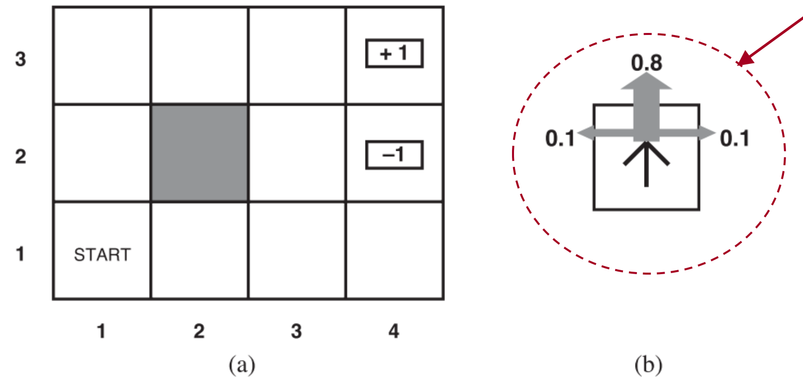
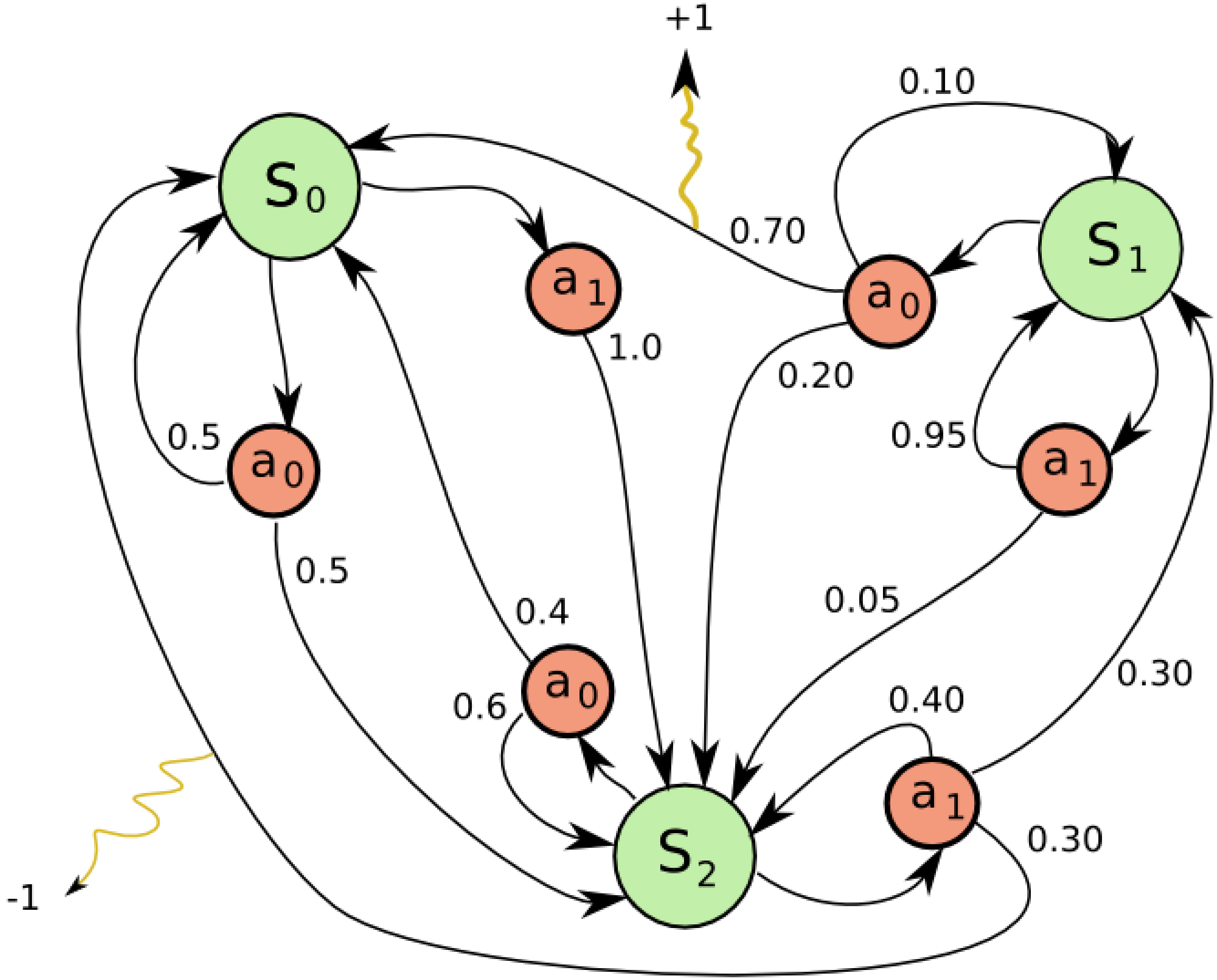


Figure 17.1 (a) A simple 4×3 environment that presents the agent with a sequential decision problem. (b) Illustration of the transition model of the environment: the "intended" outcome occurs with probability 0.8, but with probability 0.2 the agent moves at right angles to the intended direction. A collision 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.

[Russel & Norvig, 2010]

Processos de Decisão Sequencial



Propriedade de Markov

- Andrey Markov
  - Matemático Russo (1856 – 1922)
- Um processo estocástico tem a ***propriedade de Markov*** se a distribuição probabilística condicional dos **estados futuros** de um processo depender exclusivamente do **estado presente**
- **A previsão dos estados seguintes só depende do estado presente**

Processos de Decisão de Markov

- Representação do mundo sob a forma de PDM

  S – conjunto de estados do mundo

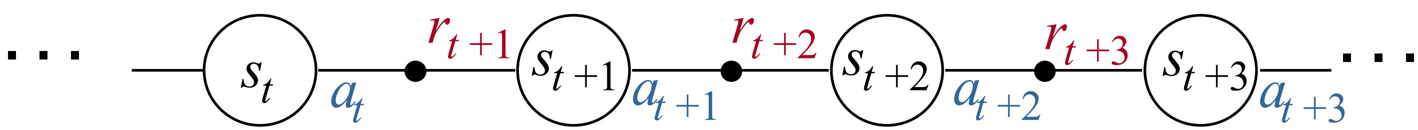
  $A(s)$ – conjunto de acções possíveis no estado $s \in S$

  $T(s,a,s')$ – probabilidade de transição de s para s' através de a

  $R(s,a,s')$ – recompensa esperada na transição de s para s' através de a

  γ – taxa de desconto para recompensas diferidas no tempo

  $t = 0, 1, 2, \ldots$ – tempo discreto



Cadeia de Markov

Utilidade

Efeito cumulativo da evolução da situação

- História de evolução h
  - Sequência de estados (com ganhos/perdas)
- Recompensa
  - Ganho ou perda num determinado estado
  - Valor finito positivo ou negativo
  - $R(s)$
- $U_h([s_0, s_1, \ldots, s_n])$

Utilidade

- Recompensas aditivas
  - $U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + R(s_1) + R(s_2) + \ldots$
- Recompensas descontadas (no tempo)
  - $U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$
  - Factor de desconto

    $\gamma \in [0,1]$
  - Recompensas não estão limitadas a uma gama finita de valores

Utilidade (valor) de estado

Exemplo:

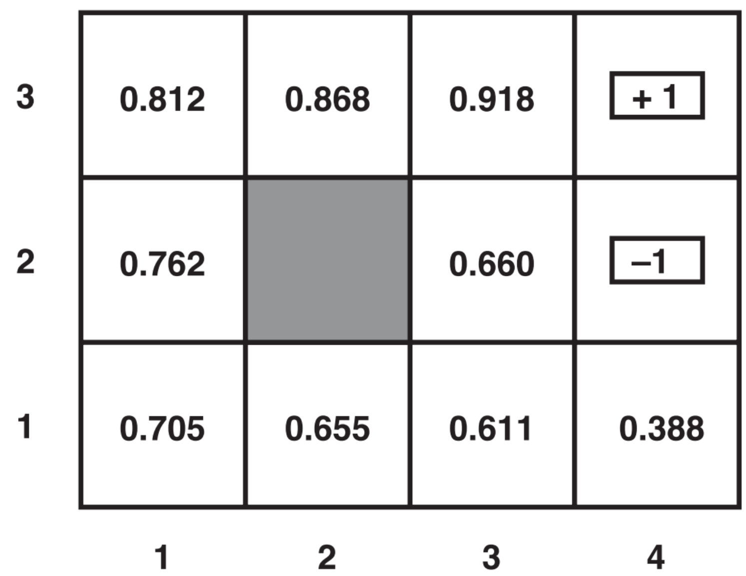


Figure 17.3 The utilities of the states in the 4×3 world, calculated with $\gamma = 1$ and $R(s) = -0.04$ for nonterminal states.

[Russel & Norvig, 2010]

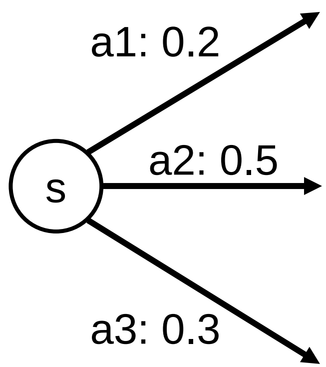
Política Comportamental

- Forma de representação do comportamento do agente
- Define qual a acção que deve ser realizada em cada estado (estratégia de acção)
- Política **determinista**

$$\pi : S \to A(s) \; ; \; s \in S$$

- Política **não determinista**

$$\pi : S \times A(s) \to [0,1] \; ; \; s \in S$$



Política Comportamental

Exemplo:

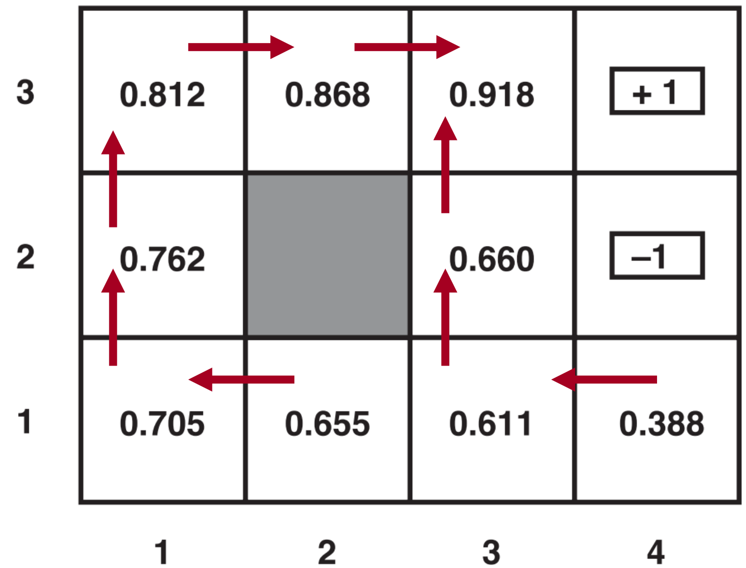


Figure 17.3 The utilities of the states in the 4×3 world, calculated with $\gamma = 1$ and $R(s) = -0.04$ for nonterminal states.

[Russel & Norvig, 2010]

O Princípio da Solução Óptima

- Programação Dinâmica
  - Requer a decomposição em sub-problemas
- Num PDM isso deriva da assunção da independência dos caminhos
- As utilidades dos estados podem ser determinados em função das utilidades dos estados sucessores

Equações de Bellman

$$
\begin{aligned}
U^{\pi}(s) &= \mathrm{E}\left\langle r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \right\rangle \\
&= \mathrm{E}\left\langle r_1 + \gamma U^{\pi}(s') \right\rangle \\
&= \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma U^{\pi}(s') \right]
\end{aligned}
$$

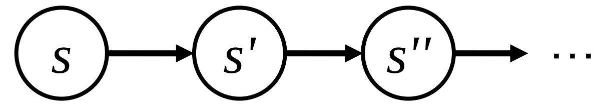
Valor esperado

$$E\langle X\rangle = \sum_{i=0}^{\infty} x_i p(x_i)$$

$$E\langle X+Y\rangle = E\langle X\rangle + E\langle Y\rangle$$

Cadeia de Markov

Política: π

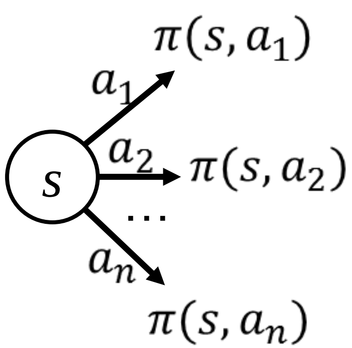


Episódio				
1	r_1^1	r_2^1	r_3^1	…
2	r_1^2	r_2^2	r_3^2	…
…				

Utilidade $U^{\pi}(s) = \mathrm{E}\left\langle r_1 + \gamma r_2 + \gamma^2 r_3 + \ldots \right\rangle$

$$= \mathrm{E}\left\langle r_1 + \gamma U\ \ (s')\right\rangle$$

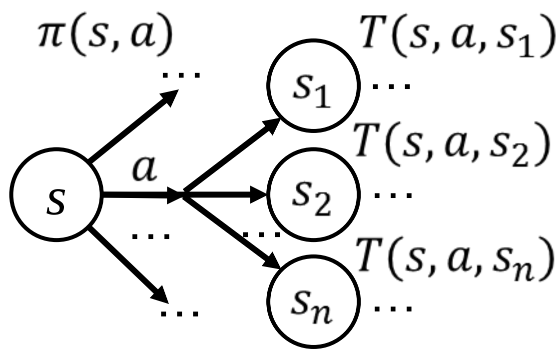
Política



$\pi(s, a)$

$a \in A(s) = \{a_1, a_2, \ldots, a_n\}$

Transição de estado com base num modelo



$T(s, a, s')$

$s' \in suc(s) \subseteq S$

Utilidade com base num modelo $U^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi}(s')]$

Processos de Decisão de Markov

Utilidade de estado para uma política π

$$U^{\pi}(s) = \sum_{a} \pi(s,a) \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi}(s')]$$

Política óptima π*

$$\pi^*(s) = \arg\max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U(s')]$$

Utilidade de estado para a política óptima π*

$$U^{\pi^*}(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U^{\pi}(s')]$$

Processos de Decisão de Markov

Iteração da utilidade de estado

Iniciar $U(s)$:

$$U(s) \leftarrow 0, \ \ \forall s \in S$$

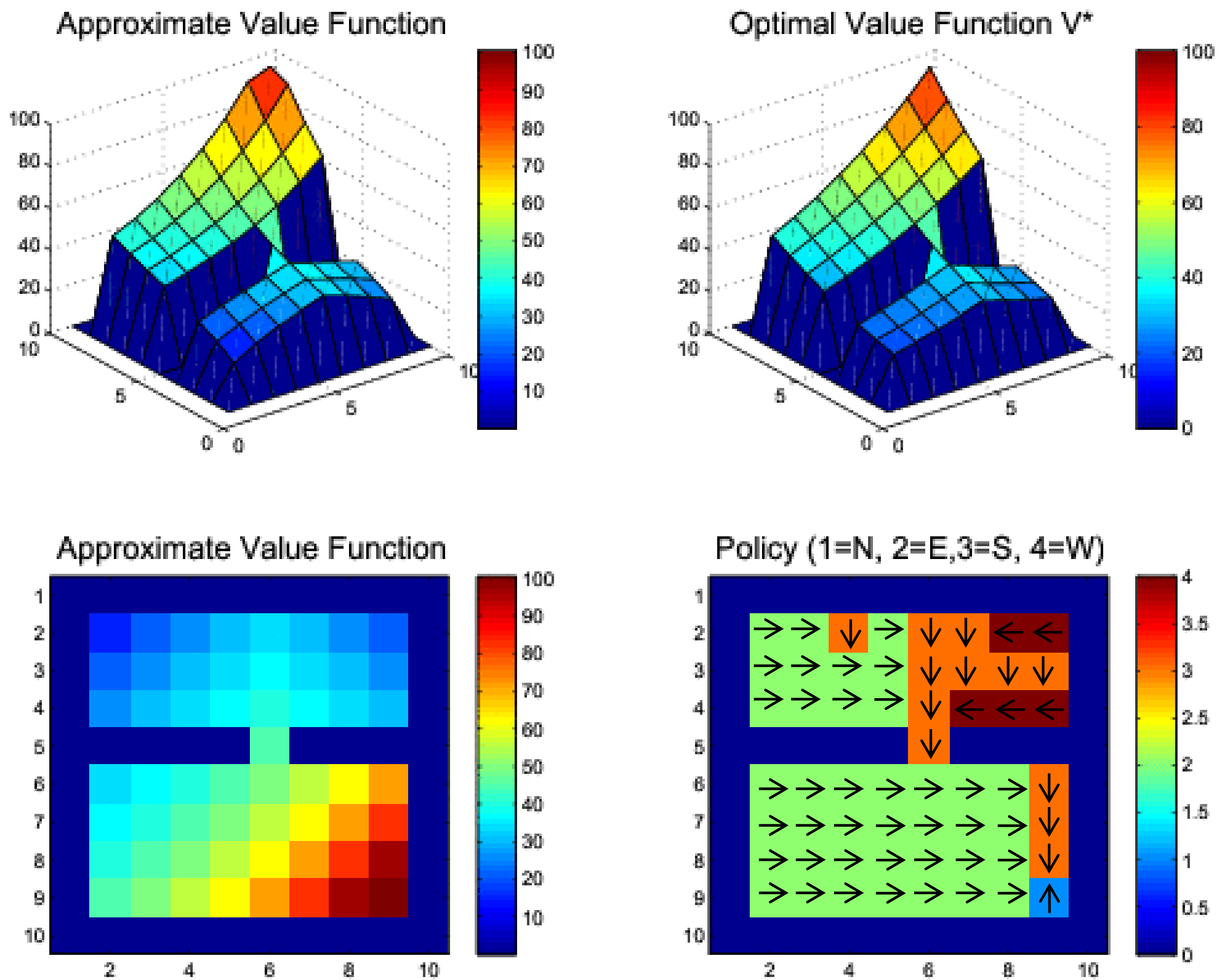
Iterar $U(s)$:

$$U(s) \leftarrow \max_{a} \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma U(s')\right], \ \forall s \in S$$

No limite:

$$U \rightarrow U^{\pi^*}$$

Processos de Decisão de Markov



[Mahadevan, 2009]

Cálculo da Utilidade de Estado

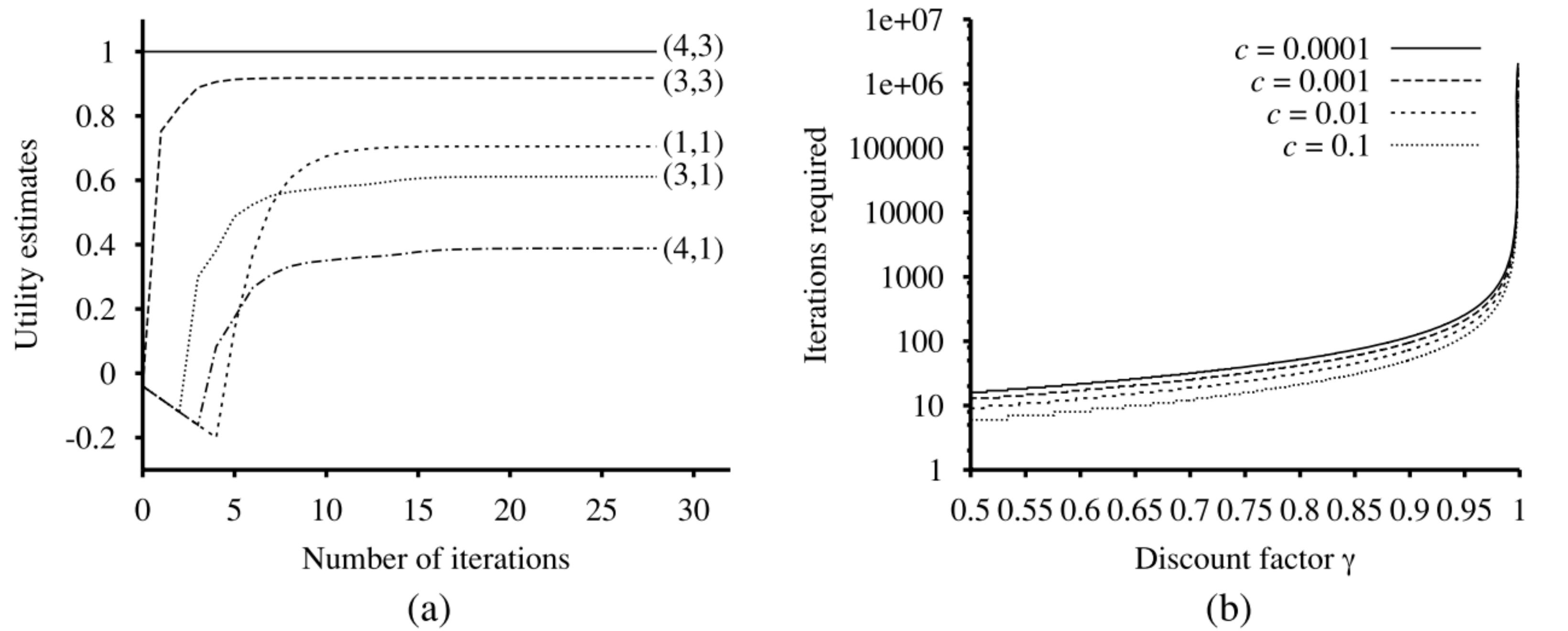


Figure 17.5 (a) Graph showing the evolution of the utilities of selected states using value iteration. (b) The number of value iterations k required to guarantee an error of at most $\epsilon = c \cdot R_{\max}$, for different values of c, as a function of the discount factor γ.

[Russel & Norvig, 2010]

Cálculo da Utilidade de Estado

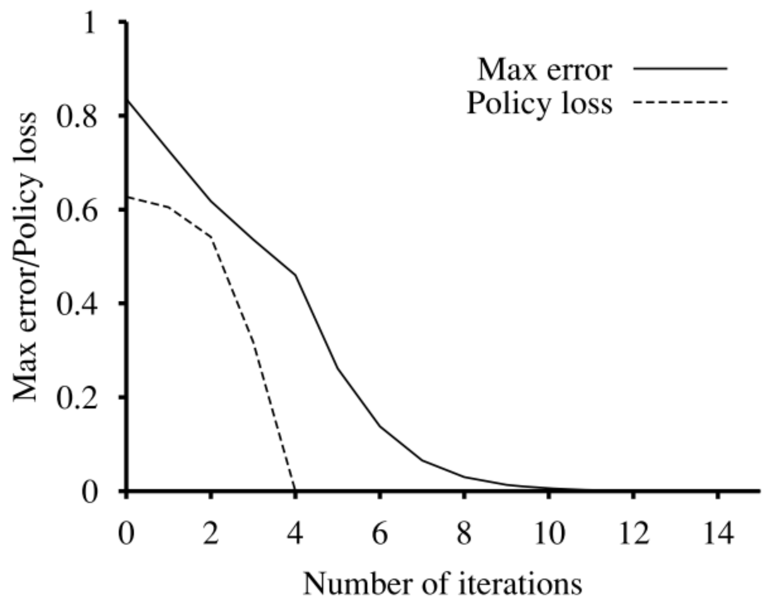


Figure 17.6 The maximum error $||U_i - U||$ of the utility estimates and the policy loss $||U^{\pi_i} - U||$, as a function of the number of iterations of value iteration.

$$\text{if} \quad ||U_i - U|| < \epsilon \quad \text{then} \quad ||U^{\pi_i} - U|| < 2\epsilon\gamma/(1-\gamma)$$

[Russel & Norvig, 2010]

Processos de Decisão de Markov

Iteração da utilidade de estado

Iniciar $U(s)$:

$$U(s) \leftarrow 0, \;\; \forall s \in S$$

Iterar $U(s)$:

$$U(s) \leftarrow \max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma U(s')\right], \; \forall s \in S$$

No limite:

$$U \rightarrow U^{\pi*}$$

Critério de paragem de iteração?

- Diferença máxima de actualização ≤ Δ_{max} (limiar de convergência)

CÁLCULO DA UTILIDADE

```
function utilidade:
    U ← 0, ∀s ∈ S
    do:
        U_ant ← U
        δ ← 0
        for s in S:
            U[s] ← max_{a∈A(s)} U_acção(s, a, U_ant)
            δ ← max{δ, |U[s] - U_ant[s]|}
    while δ > Δ_max:
    return U
```

Iteração da utilidade de estado

Iniciar $U(s)$:

$$U(s) \leftarrow 0, \ \forall s \in S$$

Iterar $U(s)$:

$$U(s) \leftarrow \max_a \sum_{s'} T(s,a,s')\left[R(s,a,s') + \gamma U(s')\right], \ \forall s \in S$$

No limite:

$$U \rightarrow U^{\pi^*}$$

Critério de paragem de iteração?

- Diferença máxima de actualização < Δ_{max} (limiar de convergência)

```
function U_acção(s, a, U):
    return Σ_s' T(s,a,s')[R(s,a,s') + γU[s']]
```

Processos de Decisão de Markov

- **Propriedade de Markov**
  - Estados futuros dependem apenas do estado actual
    - São independentes de estados passados
- **Modelo do mundo - representação do problema**
  - Conjunto de estados
    - $\boldsymbol{S}$
  - Conjunto de acções possíveis num estado
    - $\boldsymbol{A(s)}$
  - Modelo de transição
    - $\boldsymbol{T(s,a,s')}$ – também designado $P(s,a,s')$
  - Modelo de recompensa
    - $\boldsymbol{R(s,a,s')}$ – no caso geral
    - $R(s, a)$ – se a recompensa só depende do estado e da acção
    - $R(s)$ – se a recompensa só depende do estado

Processos de Decisão de Markov

No caso geral

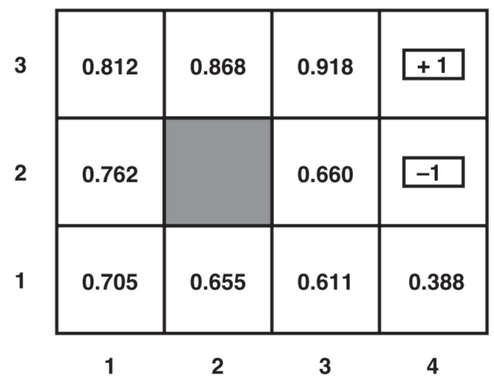
$$U(s) = \max_a \sum_{s'} T(s,a,s')[R(s,a,s') + \gamma U(s')]$$

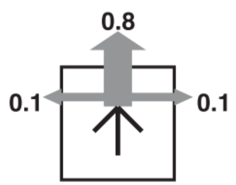
Se a recompensa só depende do estado

$$U(s) = \max_a \sum_{s'} T(s,a,s')[R(s) + \gamma U(s')]$$

$$U(s) = R(s) + \max_a \sum_{s'} T(s,a,s')[\gamma U(s')]$$

Cálculo da Utilidade de Estado





$$U_{i+1}(s) \leftarrow R(s) + \gamma \max_a \sum_{s'} T(s,a,s') U_i(s')$$

$$\begin{aligned} U(1,1) = -0.04 + \gamma \ \max\{ \ & 0.8U(1,2) + 0.1U(2,1) + 0.1U(1,1), && (Up) \\ & 0.9U(1,1) + 0.1U(1,2), && (Left) \\ & 0.9U(1,1) + 0.1U(2,1), && (Down) \\ & 0.8U(2,1) + 0.1U(1,2) + 0.1U(1,1) \ \} && (Right) \end{aligned}$$

We can think of the value iteration algorithm as *propagating information* through the state space by means of local updates.

[Russel & Norvig, 2010]

Referências

[Russel & Norvig, 2010]

S. Russell, P. Norvig, “Artificial Intelligence: A Modern Approach”, 3rd Ed., Prentice Hall, 2010

[Sutton & Barto, 1998]

R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, MIT Press, 1998

[Mahadevan, 2009]

S. Mahadevan, “Learning Representation and Control in Markov Decision Processes: New Frontiers”, Foundations and Trends in Machine Learning, 1:4, 2009

[LaValle, 2006]

S. LaValle, “Planning Algorithms”, Cambridge University Press, 2006

[Kragic & Vincze, 2009]

D. Kragic, M. Vincze, “Vision for Robotics”, Foundations and Trends in Robotics, 1:1, 2009