PROCESSOS DE DECISÃO SEQUENCIAL

Luís Morgado

ISEL-DEETC

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

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

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

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

Equações de Bellman

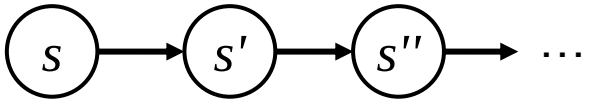
Valor esperado

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

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

Cadeia de Markov

Política: π

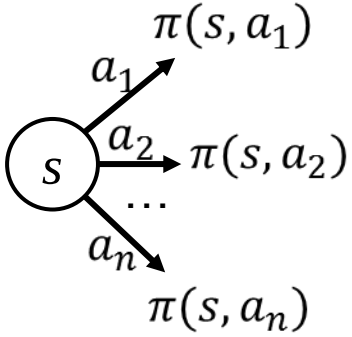


Episódio				
1	r_1^1	r_2^1	r_3^1	...
2	r_1^2	r_2^2	r_3^2	...
...				

Utilidade

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

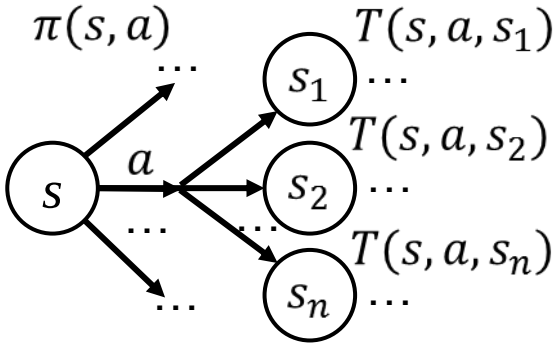
Política



$\pi(s,a)$

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

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*

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

PROGRAMAÇÃO DINÂMICA

Resolução de um problema por composição iterativa de soluções parciais

Iterar sucessivamente sobre todos os estados, actualizando cada estado com uma aproximação do valor óptimo

Iteração do Valor de Estado:

Iniciar $U(s)$:

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

Iterar $U(s)$:

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

Política de selecção de acção *greedy*

No limite:

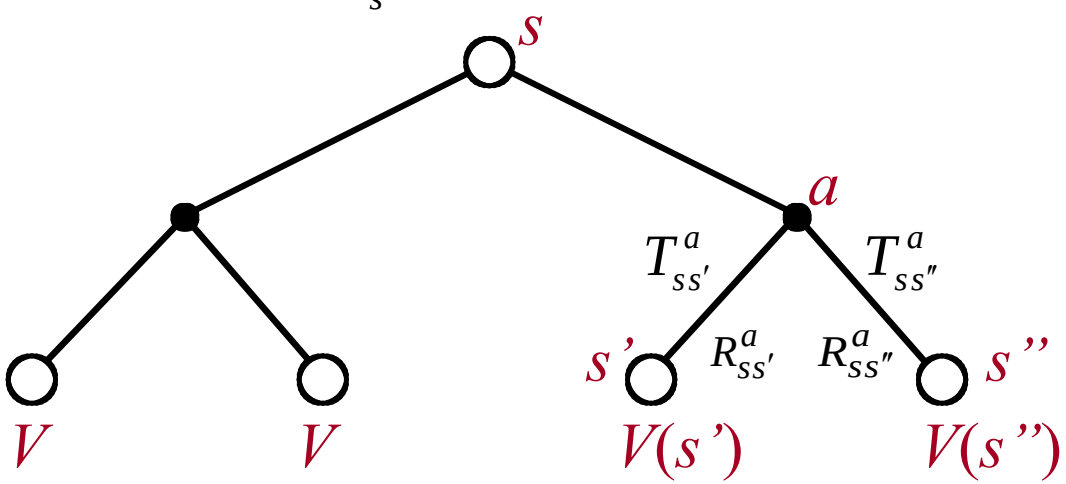
$$U \rightarrow U^{\pi *}$$

[Sutton & Barto, 1998]

PROGRAMAÇÃO DINÂMICA

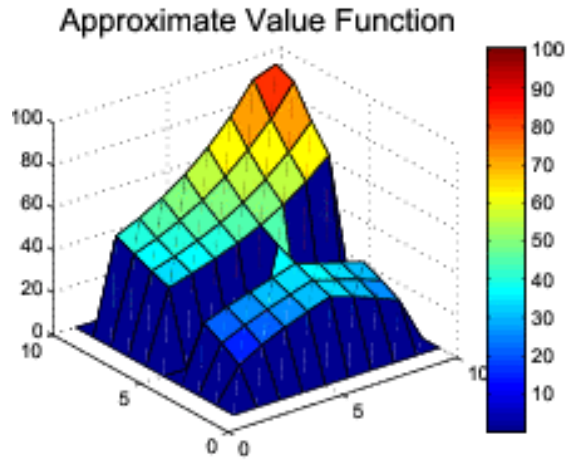
- Iteração
  - ***Backup***
    - Utilizar retornos futuros para estimar o retorno do estado actual

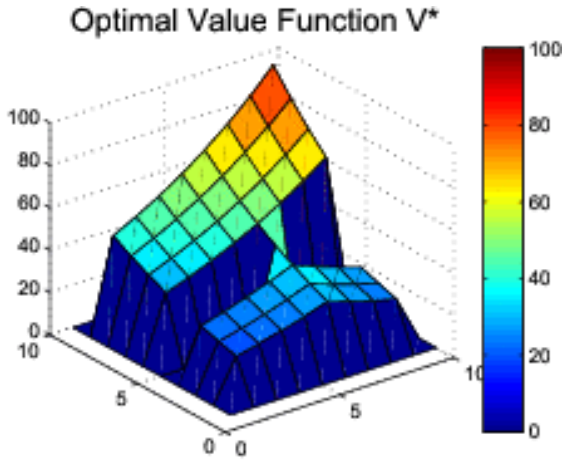
$$V(s) := \max_a \sum_{s'} T^a_{ss'} \left[R^a_{ss'} + \gamma V(s') \right] \forall s \in S$$

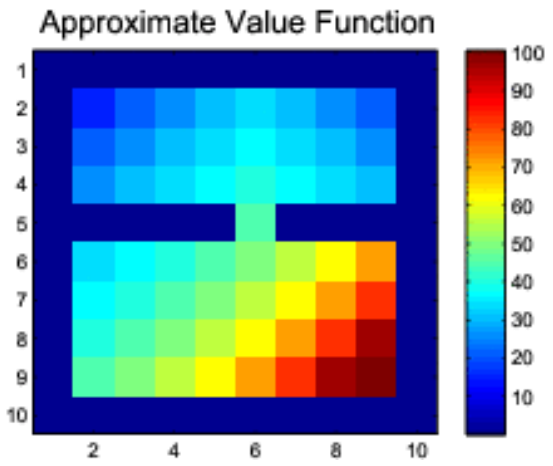


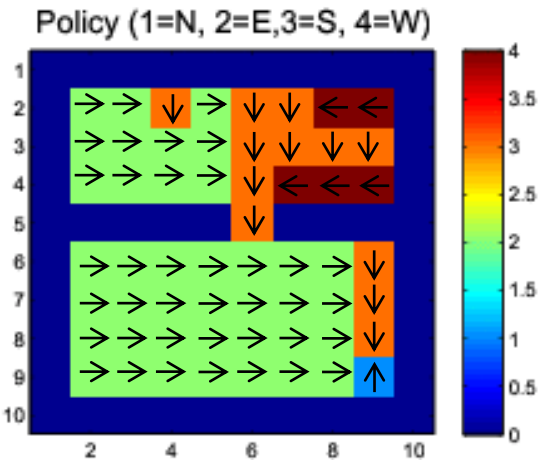
[Sutton & Barto, 1998]

PROCESSOS DE DECISÃO DE MARKOV



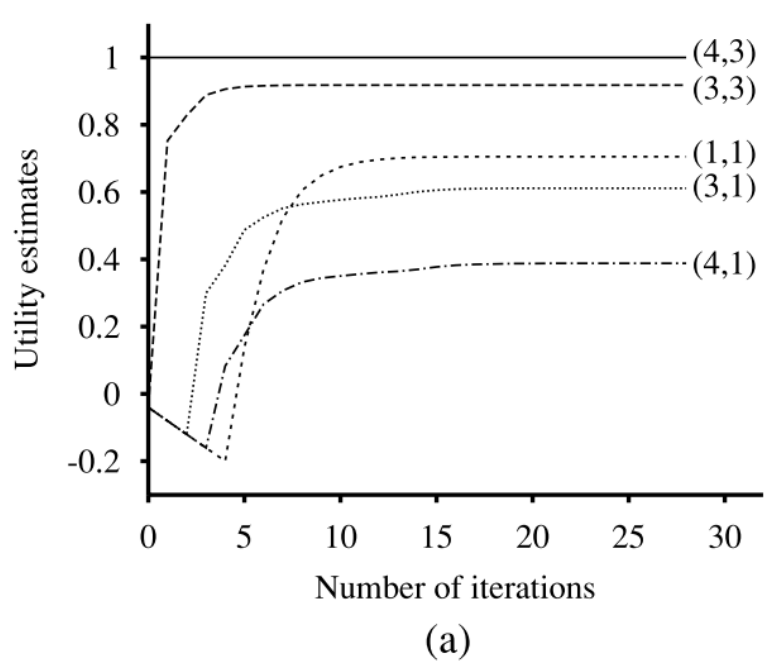






[Mahadevan, 2009]

CÁLCULO DA UTILIDADE DE ESTADO



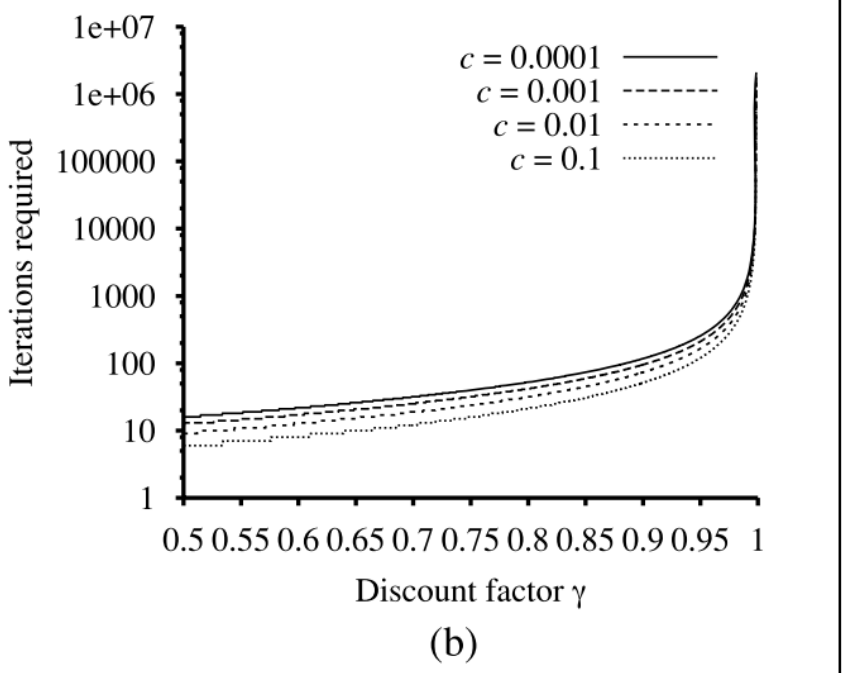


Figure 17.5 (a) Graph showing the evolution of the utilities of selected states using value iteration. (b) The number of value iterations k required to guarantee an error of at most $\epsilon = c \cdot R_{\max}$, for different values of c, as a function of the discount factor γ.

[Russel & Norvig, 2010]

CÁLCULO DA UTILIDADE DE ESTADO

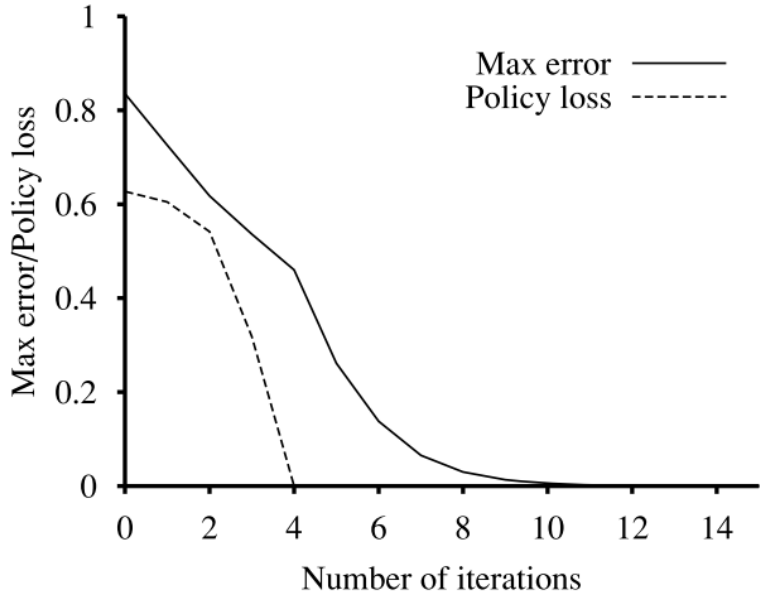


Figure 17.6 The maximum error $||U_i - U||$ of the utility estimates and the policy loss $||U^{\pi_i} - U||$, as a function of the number of iterations of value iteration.

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

[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 $U(s)$:

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

No limite:

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

Critério de paragem de iteração

$$\Delta = \left\| U_{i+1} - U_i \right\|, \quad \Delta < \Delta_{\max}$$ (limiar de convergência)

ALGORITMO DE ITERAÇÃO DE VALOR

Value Iteration, for estimating $\pi \approx \pi_*$

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

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

Output a deterministic policy, $\pi \approx \pi_*$, such that
$\pi(s) = \arg\max_a \sum_{s',r} p(s', r \mid s, a)\big[r + \gamma V(s')\big]$

[Sutton & Barto, 2020]

$T(s, a, s') = p(s' \mid s, a)$
$R(s, a, s') = p(r \mid s, a)$

UTILIDADE DE ESTADO

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

- **Iteração de valor** pode ser vista como uma *propagação de informação de valor* através do espaço de estados
  - **Síncrona**
    - As iterações actualizam sistematicamente cada estado
    - Manutenção de uma cópia da utilidade anterior
  - **Assíncrona**
    - As iterações não actualizam sistematicamente cada estado
    - Necessário continuar a iterar até que todos os estados tenham sido actualizados

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
    - S
  - Conjunto de acções possíveis num estado
    - $A(s)$
  - Modelo de transição
    - $T(s,a,s')$ – também designado $P(s,a,s')$
  - Modelo de recompensa
    - $R(s,a,s')$ – 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

- **Problemas**
  - Dimensão dos espaços de estados
    - Complexidade computacional exponencial
  - Dificuldade de definição das dinâmicas (por exemplo a partir de dados experimentais)
  - Dinâmicas desconhecidas

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

? ?

REFERÊNCIAS

[Russel & Norvig, 2010]
S. Russell, P. Norvig, “Artificial Intelligence: A Modern Approach”, 3dd Ed., Prentice Hall, 2010

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

[Sutton & Barto, 2012]
R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, 2nd Edition - Preview, MIT Press, 2012

[Sutton & Barto, 2020]
R. Sutton, A. Barto, “Reinforcement Learning: An Introduction”, 2nd Edition, MIT Press, 2020

[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