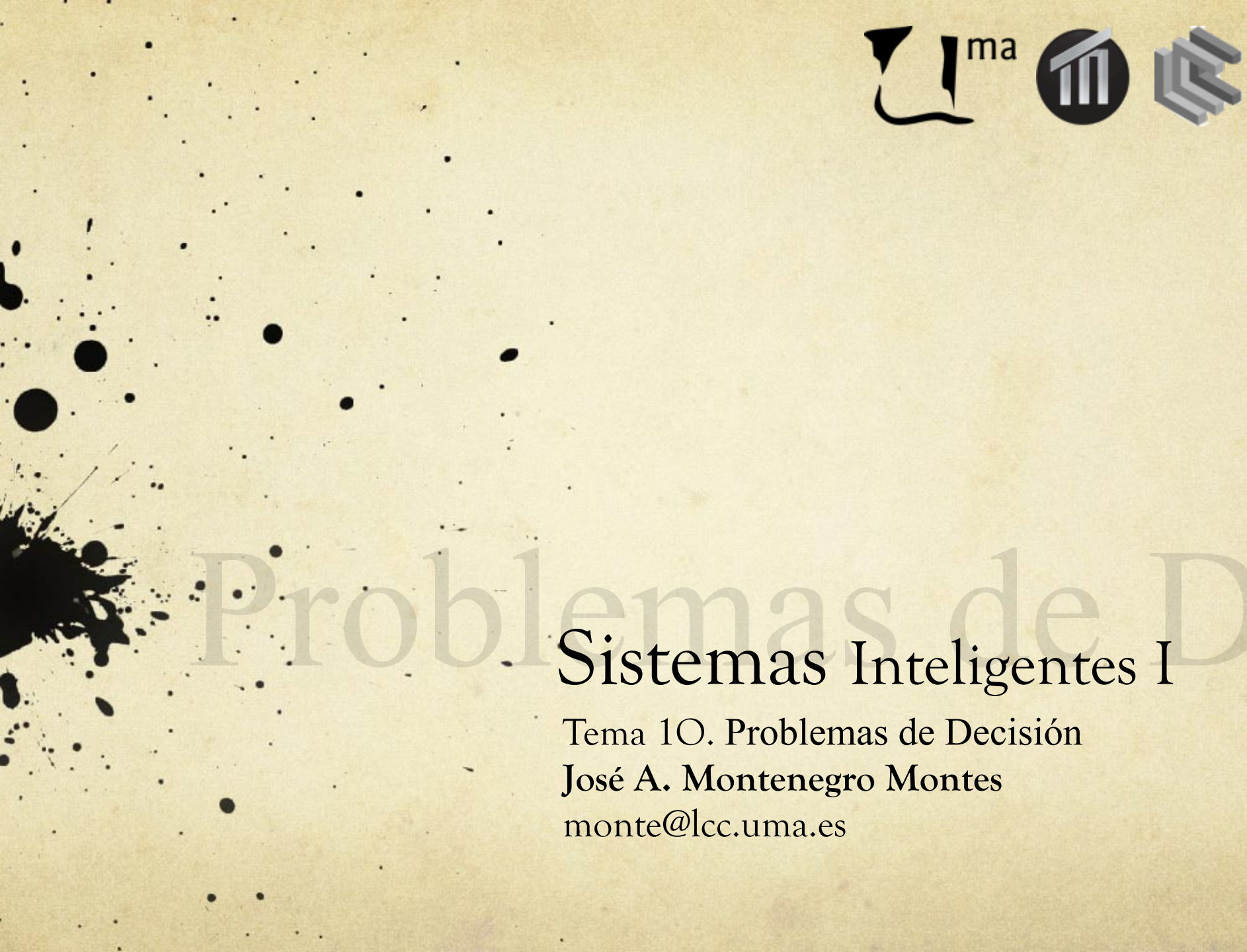


Sistemas Inteligentes I

Tema 10. Problemas de Decisión

José A. Montenegro Montes
monte@lcc.uma.es

Resumen

- Introducción
- Problemas de decisión secuenciales
- Procesos de decisión de Markov
- Ejemplo de Entorno
- Algoritmo Iteración Valores
- Ejercicios
- Conclusiones

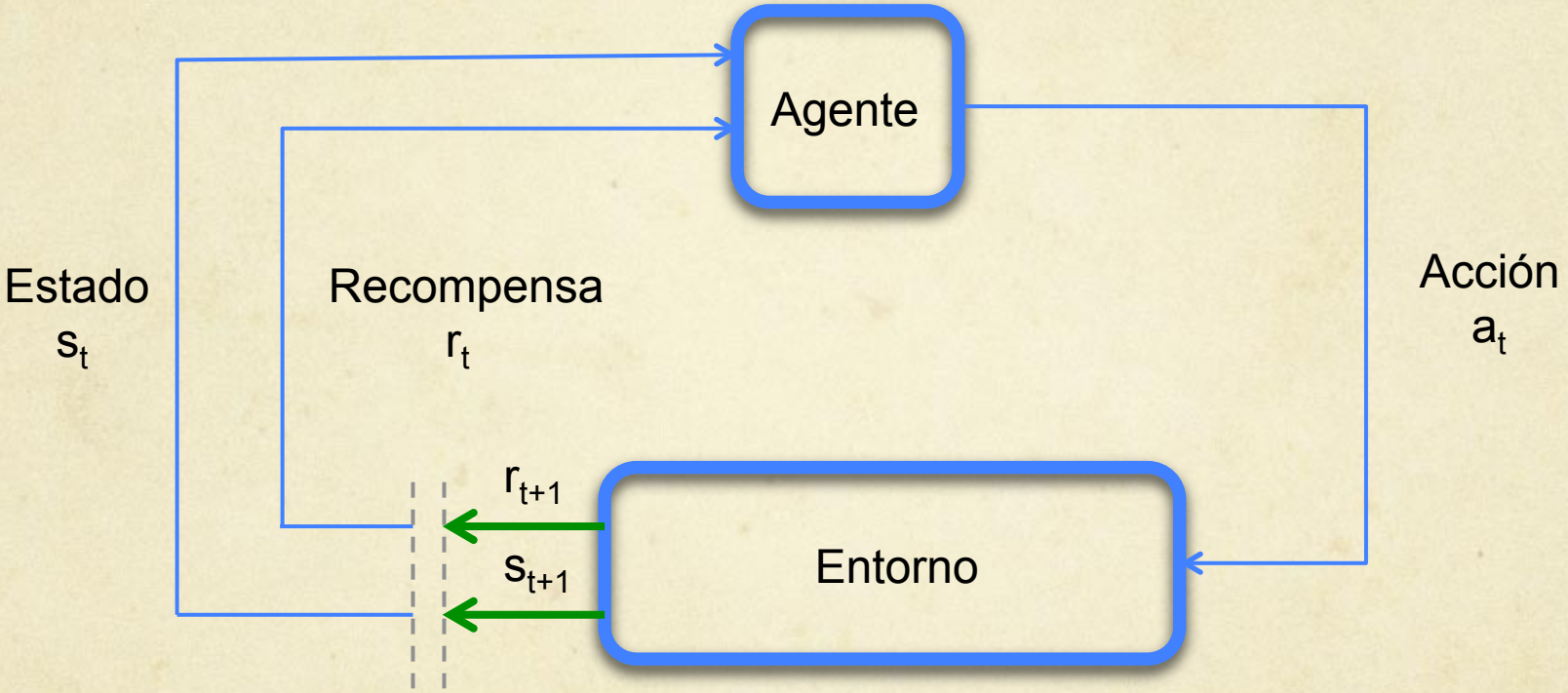
Introducción

Generalidades

- En este tema examinamos métodos para tomar decisiones en entornos estocásticos
- Estamos interesados en los problemas de decisión secuenciales, en los que la utilidad (rendimiento) depende de una secuencia de decisiones
- Calcularemos la utilidad de una política, es decir, una estrategia para tomar decisiones

Problemas de decisión secuenciales

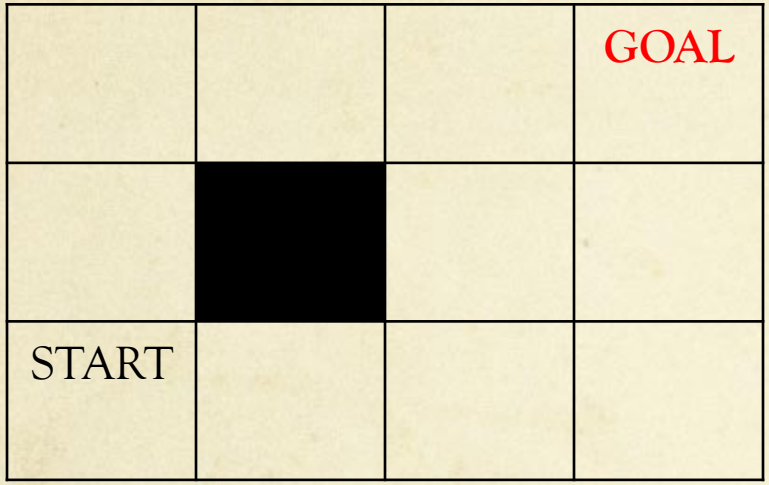
Agentes y entornos

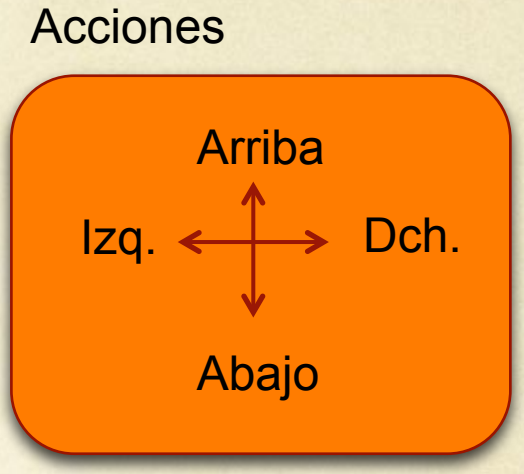


Agentes y entornos

- El agente debe elegir una acción en cada instante de tiempo
- Suponemos que el entorno es completamente observable, con lo que el agente siempre sabe donde está
- El conjunto de acciones disponibles para el agente en el estado *s* se denota Actions(*s*)

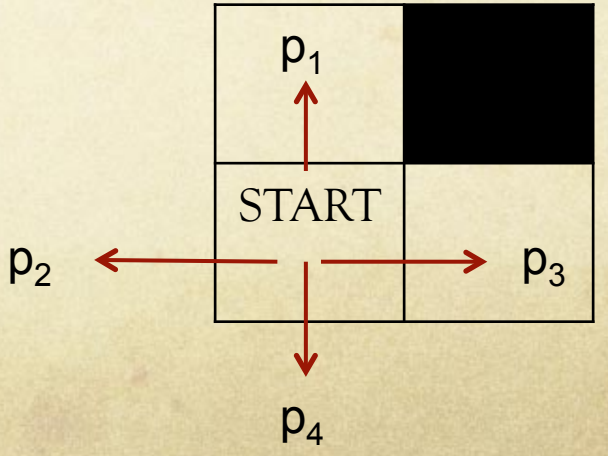
Determinista vs Estocástico





Determinista:
Posible solución:{N, N, D,D,D}

Estocástico:
Posible solución:{N ?,?,?,?,?}



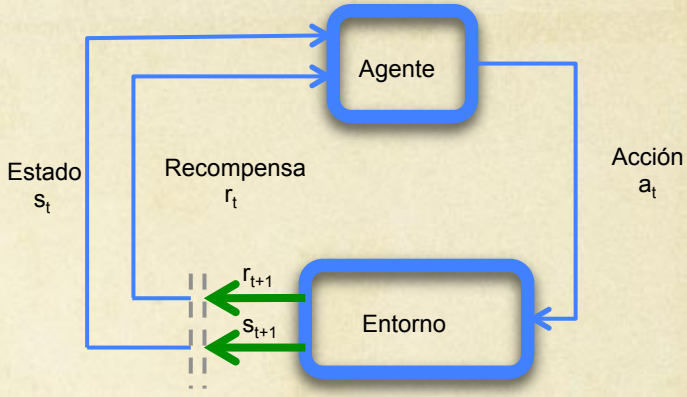
Problemas de D

Procesos de decisión de Markov

9

Procesos de decisión de Markov

- *Markov Decision Process (MDP)*
  - S: Espacio estados
  - A: Conjunto acciones
  - H: horizonte
  - T: S x A x S x {0,1, … , H} → [0,1]
    - Probabilidades (Propiedad de Markov)
  - R: S x A x S x {0,1, … , H} → *R*
    - Función de Recompensa
    - R: estado → recompensa inmediata



- **Objetivo**
  - Encontrar política. π: S x {0,1, … , H} → A

Modelos de transición y recompensa de un estado

- El modelo de transición describe el resultado de cada acción en cada estado
  - $P(s' \mid s,a)$ denota la probabilidad de alcanzar el estado s' si se realiza la acción a en el estado s
  - Suponemos que las acciones son markovianas, es decir, que la probabilidad de alcanzar s' depende solamente de s y no de la historia de los estados anteriormente visitados
- En cada estado s, el agente recibe una recompensa $R(s)$, que es un número real

Secuencias de estados

- Desde el estado inicial s_0, el agente seguirá una secuencia de estados $[s_0, s_1, s_2, \ldots]$
- La utilidad de una secuencia obedece una ley de recompensas con descuento
  - Depende de un factor de descuento $\gamma \in [0,1]$
  - Si $\gamma = 1$ tenemos recompensas aditivas

$$U_h([s_0, s_1, s_2, \ldots]) = R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \ldots$$

Procesos de decisión de Markov

- Proceso de decisión de Markov o MDP (*Markov decision process*):
  - Un problema de decisión secuencial para un entorno estocástico y completamente observable, con un modelo de transición markoviano y recompensas aditivas
- Consta de:
  - Un conjunto de estados con un estado inicial s_0
  - Un conjunto Actions(s) de acciones en cada estado s
  - Un modelo de transición $P(s' \mid s,a)$
  - Una función de recompensa $R(s)$

Políticas

- Una solución al problema de decisión debe especificar que debe hacer el agente en cualquier estado que no sea objetivo
  - Una solución de este tipo se denomina **política**, y se denota π
  - $\pi(s)$ es la acción recomendada por π en el estado s
- La utilidad esperada que se obtiene al ejecutar π empezando en s viene dada por la siguiente ecuación, donde S_t es el estado al que llega el agente en el instante t:

$$U^{\pi}(s) = E\left[\sum_{t=0}^{\infty} \gamma^t R(S_t)\right]$$

Políticas óptimas y utilidad de un estado

- De entre todas las políticas que el agente podría elegir para ejecutar empezando en *s*, aquellas que tienen una mayor utilidad esperada que todas las demás se llaman óptimas
  - Bajo nuestras suposiciones, la política óptima es independiente del estado inicial, así que la notaremos π^*

$$\pi^*(s) = \operatorname*{argmax}_a \sum_{s'} P(s' \mid s, a)\, U(s')$$

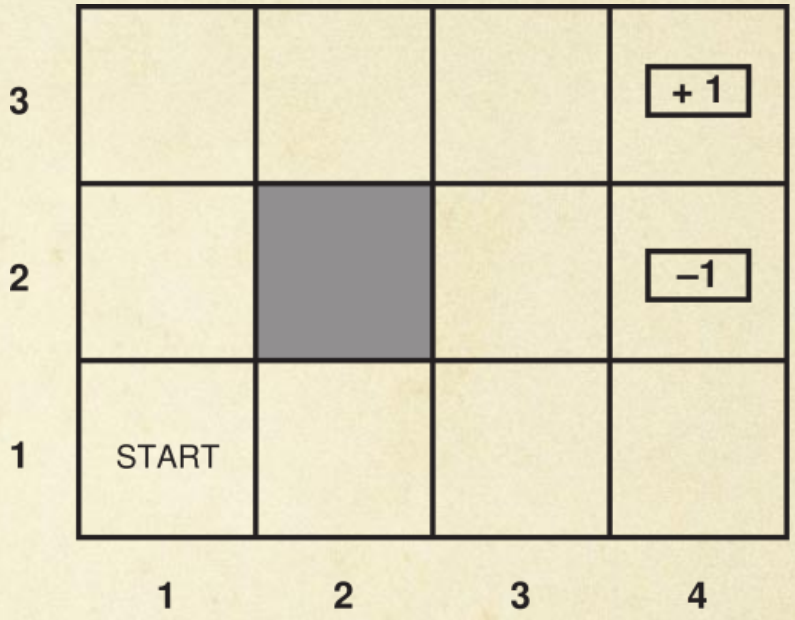
- La utilidad de un estado $U(s)$ es la utilidad esperada que se obtiene al ejecutar π^* empezando en *s*:

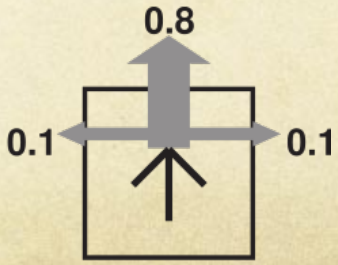
$$U(s) = U^{\pi^*}(s)$$

Ejemplo de Entorno

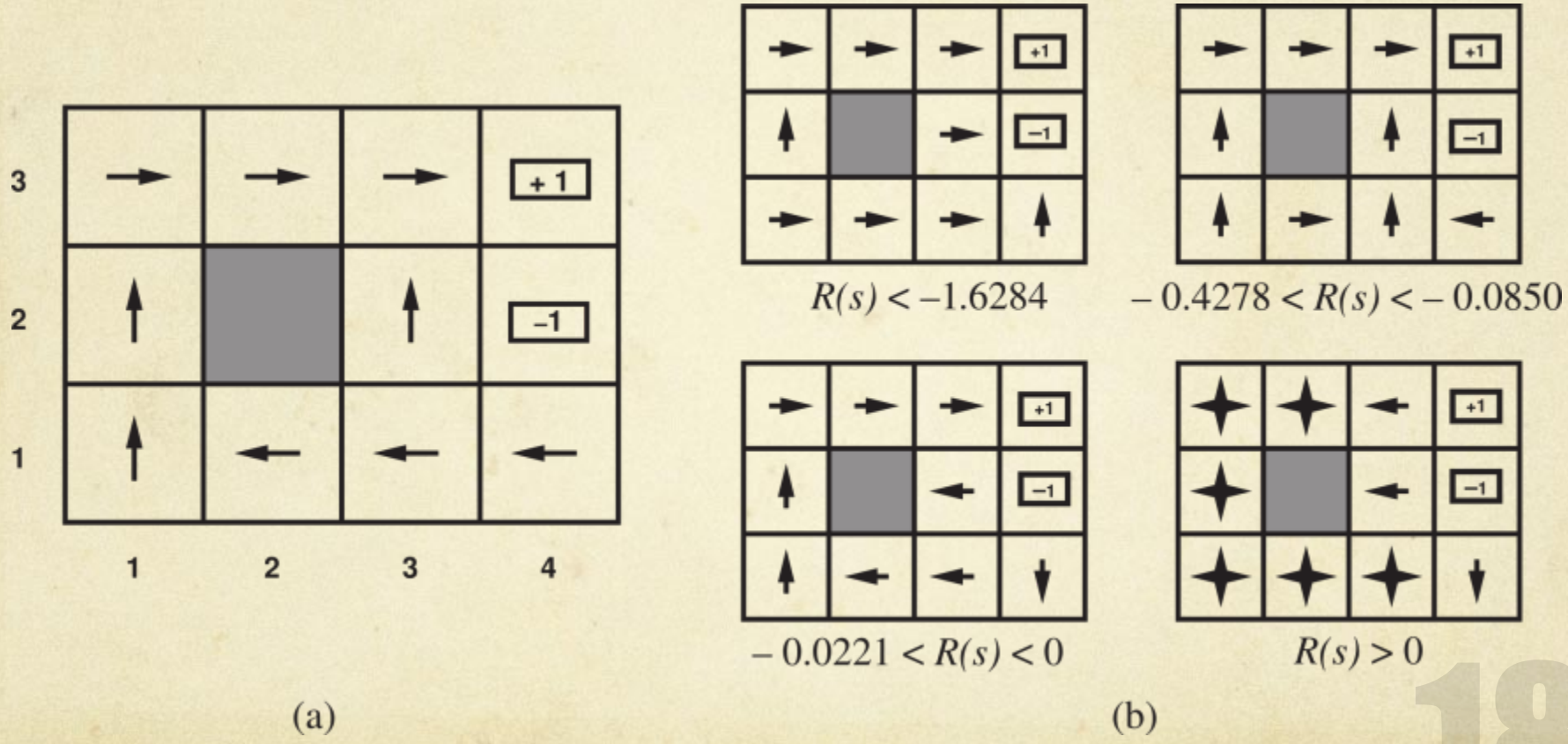
Ejemplo de entorno (I)

- El entorno es una rejilla 4×3
- La interacción finaliza cuando el agente alcanza uno de los estados objetivo, marcados –1 o +1
- Las acciones en cada estado son *Up*, *Down*, *Left* y *Right*
- Cada acción logra el efecto pretendido con probabilidad 0.8, pero el resto de las veces la acción mueve al agente en una dirección perpendicular a la pretendida
- Si el agente choca con un muro, permanece en el mismo cuadrado



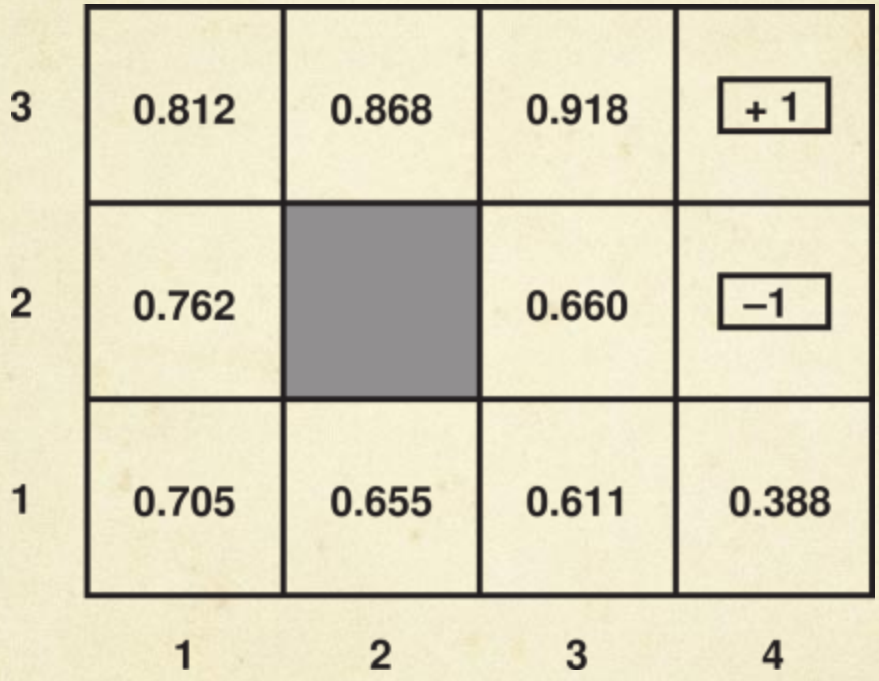


Ejemplo de políticas óptimas



(a) Política óptima para $R(s)= –0.04$. (b) Políticas óptimas para diferentes valores de $R(s)$

Utilidades de estados



Las utilidades de los estados, calculadas para γ=1 y $R(s)$= –0.04

Problemas de D

Algoritmo Iteración Valores

20

Calcular la Política Óptima

- La política óptima es definida por:

$$\pi^*(s) = \underset{a}{\operatorname{argmax}} \sum_{s'} P(s' \mid s, a)\, U(s')$$

$$U(s) = R(s) + \gamma \max_{a} \sum_{S'} P(s' \mid s, a)\, U(s')$$

- Puede ser resuelta mediante programación dinámica (Bellman)
  - Como calcular U(i) cuando la definición es recursiva

Algoritmo simplificado Iteración Valores

inicializa U'

Repetir horizonte hasta semejantes(U,U')

U← U'

Para cada estado s hacer

$$U'(s) = R(s) + \gamma \max_a \sum_{S'} P(s' \mid s, a)\, U(s')$$

finPara

devolver U

Algoritmo Iteración Valores

function VALUE-ITERATION(mdp, ϵ) **returns** a utility function
  inputs: mdp, an MDP with states S, actions $A(s)$, transition model $P(s' \mid s, a)$,
    rewards $R(s)$, discount γ
    ϵ, the maximum error allowed in the utility of any state
  local variables: U, U', vectors of utilities for states in S, initially zero
    δ, the maximum change in the utility of any state in an iteration

  repeat
    $U \leftarrow U'$; $\delta \leftarrow 0$
    for each state s **in** S **do**
      $U'[s] \leftarrow R(s) + \gamma \max_{a \in A(s)} \sum_{s'} P(s' \mid s, a)\ U[s']$
      if $|U'[s] - U[s]| > \delta$ **then** $\delta \leftarrow |U'[s] - U[s]|$
  until $\delta < \epsilon(1-\gamma)/\gamma$
  return U

Cálculo iteraciones

0		0	1
0		0	-1
0	0	0	0

-0,04	-0,04	0,76	1
-0,04		-0,04	-1
-0,04	-0,04	-0,04	-0,04

$$U'(s) = R(s) + \gamma \max_a \sum_{S'} P(s' \mid s,a)\, U(s')$$

Arriba = 0.8 x 0 + 0.1 x 0 + 0.1 x 1= 0.1
Abajo= 0.8 x 0 + 0.1 x 0 + 0.1 x 1 = 0.1
Dch= 0.8 x 1 + 0.1 x 0 + 0.1 x 0 = 0.8
Izq= 0.8 x 0 + 0.1 x 0 + 0.1 x 0 = 0

$$U'(s) = -0.04 + 1x0.8 = 0.76$$

Convergencia en las iteraciones

1

-0,04	-0,04	0,76	1
-0,04		-0,04	-1
-0,04	-0,04	-0,04	-0,04

2

-0,08	0,56	0,832	1
-0,08		0,464	-1
-0,08	-0,08	-0,08	-0,08

3

0,392	0,7376	0,8896	1
-0,12		0,572	-1
-0,12	-0,12	0,3152	-0,12

4

0,57728	0,8192	0,90616	1
0,2496		0,62888	-1
-0,16	0,18816	0,3936	0,10016

11

0,81038074	0,867748235	0,917794511	1
0,75789833		0,660235188	-1
0,68947023	0,618397936	0,582257012	0,356823359

12

0,8110265	0,867785256	0,91780297	1
0,75988426		0,660259127	-1
0,69710548	0,635255767	0,58571028	0,361487945

13

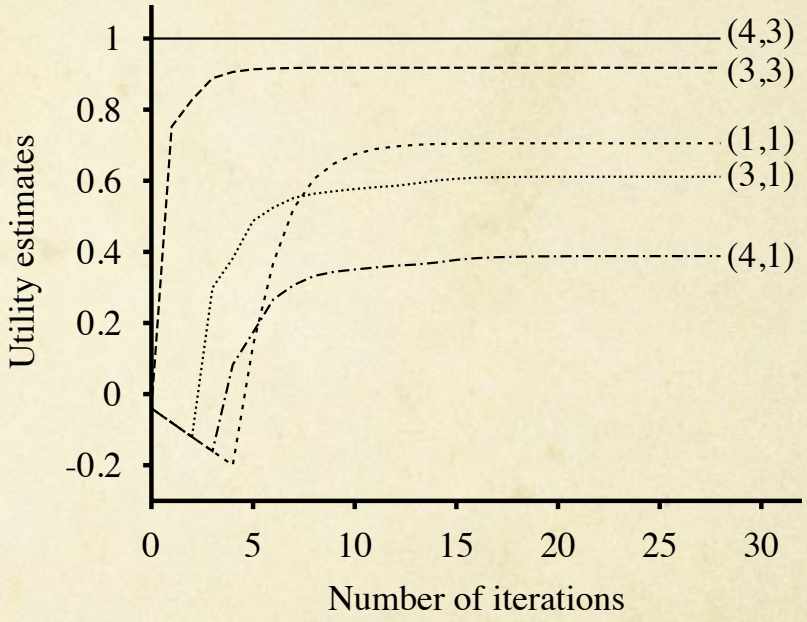
0,81131928	0,867799427	0,91780621	1
0,76079805		0,660268289	-1
0,70114353	0,644735538	0,592801555	0,364717019

14

0,81145127	0,867804853	0,91780745	1
0,76121503		0,660271797	-1
0,70322635	0,649861933	0,601095414	0,370712946

Efecto Horizonte

- h=1
  - greedy, solo considera la recompensa inmediata
- h pequeño
  - solo considera corto plazo, no considera planes a largo plazo
- h grande
  - sacrifica las decisiones a corto plazo por las recompensas a largo plazo



Problemas de P

Ejercicios

Ejercicio 1

Estado (1,1)

Arriba: 0,8 *0,762 + 0,1 * 0,705 + 0,1 * 0,655= 0,7456

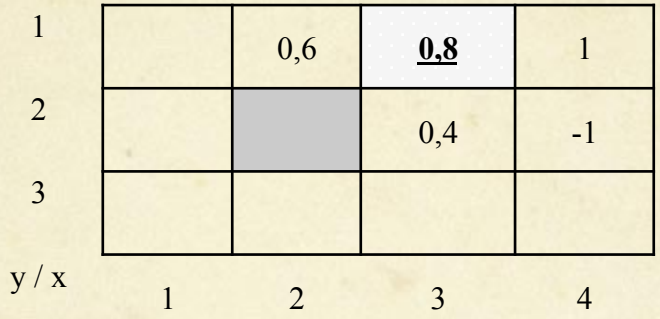
Abajo: 0,9 * 0,705 + 0,1 * 0,655= 0,7

Izquierda: 0,9 * 0,705 + 0,1 * 0,762= 0,7107

Derecha:0,8 *0,655 + 0,1* 0,705 + 0,1*0,762= 0,6707

Ejercicio 2

Halla la acción más adecuada para la celda (3,1) del mundo 4×3:

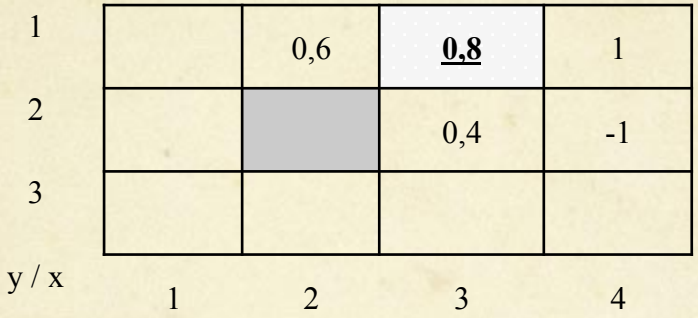


Ten en cuenta lo siguiente:

- Las acciones en cada estado son Izquierda, Derecha, Arriba y Abajo.
- Cada acción logra el efecto pretendido con probabilidad 0'8, pero el resto de las veces la acción mueve al agente en una dirección perpendicular a la pretendida, con una probabilidad 0'1 en cada dirección.
- Si el agente choca con un muro, permanece en el mismo cuadrado

Solución Ejercicio 2

Halla la acción más adecuada para la celda (3,1) del mundo 4×3:



Solución:

Izquierda: 0,6 x 0,8 + 0,8 x 0,1 + 0,4 x 0,1 = 0,6

Derecha: 1 x 0,8 + 0,8 x 0,1 + 0,4 x 0,1 = 0,92

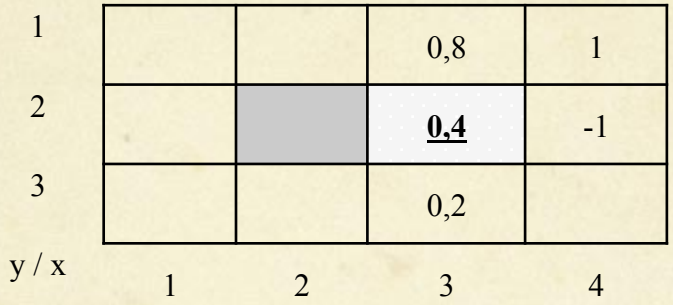
Arriba: 0,8 x 0,8 + 0,6 x 0,1 + 1 x 0,1 = 0,8

Abajo: 0,4 x 0,8 + 0,6 x 0,1 + 1 x 0,1 = 0,48

Acción Más Adecuada: **Derecha**

Ejercicio 3

Halla la acción más adecuada para la celda (3,2) del mundo 4×3:



Ten en cuenta lo siguiente:

- Las acciones en cada estado son Izquierda, Derecha, Arriba y Abajo.
- Cada acción logra el efecto pretendido con probabilidad 0'8, pero el resto de las veces la acción mueve al agente en una dirección perpendicular a la pretendida, con una probabilidad 0'1 en cada dirección.
- Si el agente choca con un muro, permanece en el mismo cuadrado

Solución Ejercicio 3

Halla la acción más adecuada para la celda (3,2) del mundo 4×3:

y / x	1	2	3	4
1			0,8	1
2			**0,4**	-1
3			0,2	

Solución:

Izquierda: 0,4x 0,8 + 0,8 x 0,1 + 0,2 x 0,1 = 0,42

Derecha: -1 x 0,8+0,2 x 0,1 + 0,8 x 0,1 = - 0,7

Arriba: 0,8 x 0,8 + -1 x 0,1 + 0,4 x 0,1 = 0,58

Abajo: 0,2 x 0,8 + -1 x 0,1 + 0,4 x 0,1 = 0,1

Marque Acción Más Adecuada: **Arriba**

Conclusiones

Sumario

- Los problemas de decisión secuenciales en entornos estocásticos, también llamados procesos de decisión de Markov, se definen mediante un modelo de transición
- La solución de un MDP es una política que asocia una decisión a cada estado que el agente podría alcanzar
- Una política óptima maximiza la utilidad de las secuencias de estados obtenidas cuando es ejecutada
- La programación dinámica adaptativa aprende un modelo y una función de recompensa, y obtiene las utilidades

Epílogo

- Los procesos de decisión de Markov se han aplicado a la optimización dinámica de aplicaciones que se ejecutan en un teléfono móvil
- Los MDP permiten a las aplicaciones incorporar las preferencias del usuario y los perfiles de usuario al proceso de decidir en tiempo real qué recursos utilizar
- Se ha observado que las políticas obtenidas dan mejores resultados que las políticas clásicas de manejo de la batería

Sistemas Inteligentes

José A. Montenegro Montes

monte@lcc.uma.es