Universidad Carlos III de Madrid

Aprendizaje Automático

Grado en Ingeniería Informática

Grupo 83

Tutorial 4: Introduccíon al aprendizaje por refuerzo

Autores:
Daniel Medina García
Alejandro Rodríguez Salamanca

7 de abril de 2016

Índice

1. Ejercicio 1 **3**

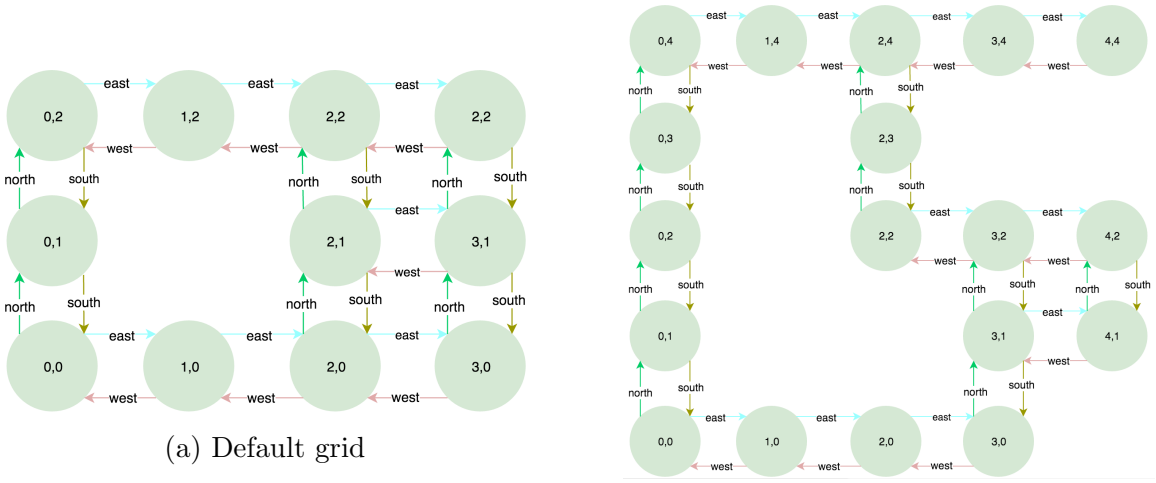
2. Ejercicio 2 **5**

3. Ejercicio 3 **5**

Introducción

1. Ejercicio 1

En este primer ejercicio tomamos contacto con el programa ejecutando tanto el agente manual como el estándar. Por cada movimiento ejecutado, en la salida por consola aparecen el estado inicial, la acción tomada, el estado al que se llega y la recompensa obtenida. El agente por defecto es *random*, que toma acciones aleatorias entre las permitidas. La Figura 1 muestra los MDP deterministas requeridos en el enunciado del tutorial:



(a) Default grid

(b) AAGrid

Figura 1: Deterministic MDPs

Estos laberintos, al igual que los demás incluidos en el código proporcionado, se almacenan en forma de arrays bidimensionales. Cada posición de la matriz representa una posición del laberinto, con los siguientes símbolos:

- **Un espacio** representa una casilla común, sin recomensa ni coste algunos, por la que el agente puede pasar.
- **Un valor numérico** representa una casilla con una recompensa o un precio. Si el agente se mueve a esta casilla cambiará su puntuación añadiéndole el valor contenido en dicha casilla.
- **Una almohadilla (#)** es considerada un muro, o casilla inalcanzable por el agente. Ninguna acción legal puede llevar al agente a estas posiciones.
- Por último, la posición de la que parte el agente se guarda como **una 'S'**.

De esta forma, podemos crear un nuevo laberinto siguiendo las instrucciones mencionadas. La siguiente figura muestra un laberinto creado por nosotros, en la que el agente deberá buscar una ruta sorteando los muros si quiere conseguir la mayor recompensa:

```
[[' ',' ',' ',' ',]
 ['8','#','3',' ',]
 ['#',' ',' ','#',]
 [' ',' ',' ',' ',]
 [' ','S','#',' ',]]
```

Figura 2: New grid

Aquí contamos cómo cojones se pueden sacar varias políticas óptimas, describiendo todas las políticas óptimas para este problema.

2. Ejercicio 2

3. Ejercicio 3