Sistemas Inteligentes

Práctica 2: Aprendizaje por refuerzo

Pablo Gomariz Martínez

Índice

[1.Introducción 3](#_Toc25951418)

[1.1 Detalles del problema 3](#_Toc25951419)

[2.Implementación 4](#_Toc25951420)

[2.1 4](#_Toc25951421)

[2.2 Mejoras para la búsqueda de la política óptima 4](#_Toc25951422)

[3.Ejecución de la práctica 4](#_Toc25951423)

[4.Estudio de los resultados 4](#_Toc25951424)

[5.Conclusiones 4](#_Toc25951425)

**No se encuentran elementos de tabla de ilustraciones.**

**No se encuentran elementos de tabla de ilustraciones.**

# 1.Introducción

En esta práctica vamos a implementar los algoritmos de aprendizaje por refuerzo ***Iteración de Valores***y ***QLearning***, aplicándolos a un problema similar al de la práctica anterior. En este caso, el problema se dersarrolla en un entorno no determinista, y tenemos que encontrar una política óptima para que un coche en un laberinto llegue a la meta (en este caso, trabajaremos solo con un coche, en lugar de los múltiples coches que existían en la práctica anterior). Una política consiste en un conjunto de acciones asociadas a cada estado, que nos permitirán llegar a un estado final obteniendo una utilidad (en el caso de la política óptima, obtendremos la mayor utilidad).

Estudiaremos el tiempo de convergencia de los algoritmos, y la solución que obtenemos tras aplicar cada uno, evaluando la utilidad de la política final obtenida según número de iteraciones que ejecutaremos en los algoritmos (en el caso de QLearning), o según los parámetros:

* **Tamaño de problema.**
* **Alpha α**: Usado en QLearning, representa la **preferencia de los valores Q nuevos frente al valor antiguo**. Con α = 1, actualizaremos siempre al nuevo valor, y nunca guardaremos información de la utilidad anterior, mientras que con α = 0, no actualizaremos a las utilidades nuevas, siempre nos quedaremos con los valores antiguos.
* **Delta δ**: Representa **la condición de parada para el algoritmo de iteración de valores**. Si tomamos un δ = 0.001, **pararemos de buscar soluciones cuando la diferencia entre la utilidad de la *penúltima* política obtenida y la última sea menor o igual a δ**, lo que quiere decir, que no habrá cambiado la utilidad de las dos últimas políticas, y por tanto no hemos encontrado una mejora que sea mayor que δ, por lo que podemos dejar de seguir buscando. **Cuanto menor sea nuestro δ, encontraremos una mejor solución a costa de un mayor tiempo de exploración, aunque es probable que no mejoremos nuestra política y hayamos malgastado el tiempo de ejecución**.
* **pCorrect**: Es **la probabilidad de elegir una acción, y que la acción aplicada resultante sea la que hemos escogido**. Al estar en un entorno no determinista, , es la probabilidad de que la acción aplicada sea distinta a la que hemos escogido, y caigamos en un estado distinto al que pretendemos.
* **Gamma γ**: Por defecto, tomaremos este valor como 1. **Representa un factor de descuento por cada acción aplicada**, por lo que cuantas mas acciones realicemos, y mas tiempo gastemos en la búsqueda de la política, menor recompensa obtendremos por cada movimiento. Suponiendo y el tiempo t:

Usando no aplicaremos ningún descuento a las recompensas obtenidas.

## 1.1 Detalles del problema

En este caso no necesitamos una representación concreta de un estado, ya que trabajaremos con todos los estados posibles del problema. En *Iteración de Valores* trataremos con una matriz de utilidades y otra de políticas, donde guardaremos la utilidad y la política asociada a cada estado del problema (cada posición del maze). En QLearning trabajaremos con la QTabla, en la que situaremos los estados que vayamos visitando, con la utilidad asociada cada acción posible en cada estado.

Las acciones posibles son las siguientes, y son escogidas en este orden:

* **UP**: El coche se desplaza a la fila de arriba.
* **DOWN**: El coche se desplaza a la fila de abajo.
* **RIGHT**: El coche se desplaza a la columna de la derecha.
* **LEFT**: El coche se desplaza a la columna de la izquierda.

**Al ser un entorno no determinista**, durante la ejecución de la política elegida **se ejecutarán las acciones de la política con una determinada probabilidad** (explicada en el apartado 2, de la Implementación). Para calcular la política en QLearning, también se aplicará esta norma, pues **el agente no tiene por qué caer en la mejor acción que elijamos todas las veces**.

Un estado final se corresponde con cualquier casilla de la última fila del *maze*, como en la práctica anterior.

# 2.Implementación

## 2.1

## 2.2 Mejoras para la búsqueda de la política óptima

Optimización del algoritmo de iteración de valores reduciendo cálculos y bucles

Alfa con decaimiento y probabilidad de exploración en QLearning

# 3.Ejecución de la práctica

# 4.Estudio de los resultados

# 5.Conclusiones