



Universidad
Rey Juan Carlos

Práctica 2: Q-Learning

ÍNDICE

- 1.- Antes de arrancar.**
- 2.- Etapa de Aprendizaje.**
- 3.- Etapa de Explotación.**
- 4.- Después de la utilización.**
- 5.- Discusión sobre los resultados obtenidos.**

1.- Antes de arrancar.

En el inspector de Unity, antes de arrancar al usuario se le dan varias opciones. En primer lugar, se puede elegir si quiere empezar aprendiendo: puesto que el algoritmo almacena los aprendizajes previamente utilizados, esto permite que se utilice el aprendizaje previo en esa semilla sin tener que rehacer el proceso de aprendizaje o si, por otro lado, reaprenda sobre la información previa y pueda afinar dicha información con la posibilidad de que encuentre un camino óptimo. En segundo lugar, también se le da la opción de ver el aprendizaje, donde el usuario puede ver por qué casillas va pasando el algoritmo.

También se permite modificar el número de episodios e iteraciones por las que el algoritmo pasa para aprender, de forma que cuanto más elevados sean estos números mayor será la precisión de su aprendizaje.

2.- Etapa de Aprendizaje.

En la etapa de aprendizaje, los parámetros del algoritmo Q-Learning implementados fueron: la constante *alpha* (el valor es 0.3), que modula cuánto peso le da al nuevo aprendizaje; la constante *gamma* (el valor es 0.92), que modula la influencia de la mejor acción del estado siguiente; el número de iteraciones de aprendizaje por cada episodio (por defecto el valor es 600), el número de episodios que lleva el proceso de aprendizaje (por defecto el valor es 15) y la función de refuerzo (la meta cuenta con un valor de refuerzo de 100 y los muros cuentan con un valor de refuerzo de -10).

Estos parámetros (a excepción del número de episodios y el número de iteraciones) no se pueden modificar, puesto que son los que modulan la regla de aprendizaje empleada.

En cuanto a los ficheros necesarios, el programa no necesita ningún archivo previo para funcionar, aunque si la semilla ha sido utilizada previamente permite extraer información de aprendizajes previos y, según si está desactivada o no la opción de aprender, realizar el movimiento aprendido anteriormente (en el caso de no reaprender) o de refinar el aprendizaje de dicha semilla. Si se quiere resetear el proceso de aprendizaje, basta con eliminar el archivo que contiene la información.

Dicho aprendizaje (expresado en una tabla-Q) se almacena una vez terminado el proceso en un fichero csv, que permite ser abierto por programas similares a hojas de cálculo. La nomenclatura utilizada sobre el fichero es: *nº_semilla-nº_columnas-nº_filas.csv*.

3.- Etapa de Explotación.

En esta etapa el programa guarda el fichero csv justo después de terminar el aprendizaje sobre la semilla correspondiente, y encuentra el mejor camino utilizando los valores más altos que se encuentra en la tabla-Q. Además, cuando se encuentra en la etapa de explotación, el programa finaliza en cuanto encuentra la meta.

4.- Después de la utilización.

Desde la propia carpeta donde se encuentra la memoria, también hay una carpeta llamada *LearnedData* donde se almacenan todos los ficheros de las semillas aprendidas, los cuales contienen las respectivas tablas-Q para cada semilla.

Cabe destacar que, aunque desde el inspector no se puede cambiar el tamaño del tablero, sí se puede modificar el código del programa para cambiar el tamaño del tablero, por lo cual también se almacena un archivo distinto según el número de columnas y filas por seguridad, de modo que dos tableros distintos con la misma semilla tienen distintos ficheros.

5.- Discusión sobre los resultados obtenidos.

Tras la realización de varias pruebas sobre distintas semillas, se ha observado que no siempre se encuentra la solución más óptima en el primer aprendizaje, sin embargo, a medida que se va aprendiendo sobre la misma semilla, el resultado se va afinando hasta obtener la solución óptima.

También cabe destacar que, si el número de episodios e iteraciones es muy bajo, el proceso de aprendizaje se vuelve muy limitado y puede ocurrir que no encuentre un camino hacia la meta. Esto se puede solventar aumentando el número de episodios e iteraciones. Es recomendable subir el número de iteraciones por encima del número de episodios, ya que la complejidad del algoritmo no sube en gran medida y el aprendizaje se vuelve más profundo.