Tarea 1 - Aprendizaje por refuerzo

Emmanuel Peto Gutiérrez

3 de marzo de 2023

1. Ejecución de los algoritmos

1.1. Generación de políticas y valores

Para generar los archivos que contienen tanto los valores (v*) como las políticas (pi*), se debe ejecutar el script de python: `entrenamiento.py`. Una vez que se ejecuta, se preguntará cuál algoritmo usar. Por cada algoritmo se generarán dos archivos:

- `politicaIP.csv`: las políticas generadas por el algoritmo de iteración de política.
- `valorIP.csv`: los valores calculados por el algoritmo de iteración de política.
- `politicaIV.csv`: las políticas generadas por el algoritmo de iteración de valor.
- `valorIV.csv`: valores calculados por el algoritmo de iteración de valor.

1.2. Juego

Una vez que se han generado las políticas, se puede poner al agente a jugar. Para esto se debe ejecutar el script `juego.py` y se elige una política: la generada por iteración de valor o la generada por iteración de política.

2. Frozen lake

2.1. Recompensa y probabilidades

En el juego de frozen lake se tiene una recompensa de -1 por cada movimiento; sin embargo, si el agente no se mueve (porque intentó avanzar hacia la pared), la recompensa será 0. Se debe calcular la recompensa promedio en cada movimiento. El agente se mueve aleatoriamente hacia 3 posibles direcciones (con la misma probabilidad a cada dirección), así que la recompensa promedio será:

- -1 si no le estorba ninguna pared.
- -2/3 si le estorba una pared.
- -1/3 si le estorban dos paredes.

Luego, la probabilidad de llegar al estado s' dado que está en s y se toma la acción a es 1/3 siempre: $p(s', s, a) = 1/3$; y en algunos casos $s' = s$, si el agente intentó moverse hacia una pared.

2.2. Estados y función hash

Como se observará, los estados están representados por solo un número, el cual se calcula (segun gymnasium) como $4f + c$, donde f es la fila donde se encuentra el agente y c es la columna. Podría pensarse que es una función hash que mapea la posición del agente a un número. Dado el estado s, se puede calcular la posición del agente de la siguiente forma: $f = s/4$ y $c = s \mod 4$. La siguiente cuadrícula muestra los posibles estados, donde 0 es el estado inicial y 15 el final.

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

3. Gato

3.1. Recompensa y probabilidades

Las recompensas en el juego del gato son las siguientes:

- 1 si gana el agente.
- -1 si pierde el agente.
- 0 si empata.
- 0 si llega a un estado no terminal.

Para cortar el número de estados, solamente se consideran aquellos donde va a tirar el agente. En mi caso, cuando va a tirar el jugador 1 (o el jugador $\times$, si se quiere ver con símbolos).

Lo siguiente es calcular la probabilidad de llegar al siguiente estado s' dado un estado s y dado una acción a, para lo cual, tira el agente en la posición a e inmediatamente tira el rival de forma aleatoria (y equiprobable) en alguna de las casillas disponibles. Por lo tanto, la probabilidad de llegar a un estado s' es 1 dividido entre la cantidad de posibles sucesores dado s y dado a: $1/|sucesores(s, a)|$.

3.2. Estados y función hash

Como se observa en los archivos, las políticas y los valores están relacionados solo con un número. Esto es porque se mapea un estado del tablero con un número, al cual llamo el "código hash" del tablero. Primero, se considera una casilla como un número del 0 al 8.

0	1	2
3	4	5
6	7	8

Luego, se le asigna un número a cada casilla dependiendo de la ficha que tenga:

- 0 si está vacía.
- 1 si tiene una $\times$.
- 2 si tiene un $\circ$.

Así, la función hash se calcula de la siguiente manera:

$$\sum_{i=0}^{8} f(i) * 3^i$$

donde i es una casilla y $f(i)$ es la ficha en esa casilla. Por ejemplo, para el siguiente estado del tablero

	$\times$	$\circ$
$\circ$	$\times$	
		$\times$

se tiene el código hash:
$1 \cdot 3^1 + 2 \cdot 3^2 + 2 \cdot 3^3 + 1 \cdot 3^4 + 1 \cdot 3^8 = 6717$
También se puede encontrar el estado del tablero dado el código hash en la misma forma en la que se calcula un número en base 3 dado un número en base 10.