Tutorial 4: Aprendizaje por refuerzo

Grado en Ingeniería Informática

Aprendizaje Automático

Aitor Alonso Núñez NIA 100346169 Gr. 83
100346169@alumnos.uc3m.es
Daniel Gak Anagrov NIA 100318133 Gr. 83
100318133@alumnos.uc3m.es

Aitor Alonso Núñez NIA 100346169
Daniel Gak Anagrov NIA 100318133

uc3m | Universidad Carlos III de Madrid

Índice

Aitor Alonso Núñez NIA 100346169
Daniel Gak Anagrov NIA 100318133

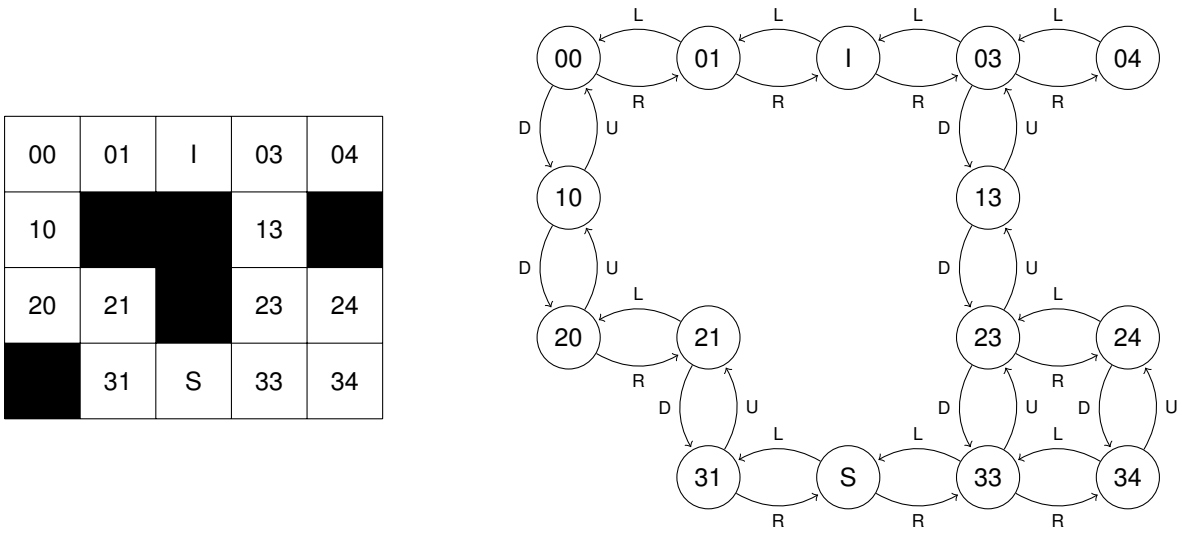

Figura 1: MDP determinista

Introducción

El presente documento responde a las preguntas realizadas en el *tutorial 4: Aprendizaje por refuerzo* de la asignatura Aprendizaje Automático. Asimismo, detalla el trabajo que hemos desarrollado durante el tutorial y argumenta cada una de las decisiones y respuestas realizadas.

MDP determinista

7. Ilustración del modelo

Hemos nombrado a cada estado posible del MDP como XY, donde X es la fila del laberinto (comenzando por la fila de arriba en cero) e Y es la columna del laberinto (comenzando por la columna de la izquierda en cero). De esta forma, el agente solo puede transicionar a estados cuyo nombre difiere solo en una cifra respecto al estado actual.

Así pues, el MDP queda como se muestra en la figura superior. Las acciones posibles son Izquierda (L), Derecha (R), Arriba (U) y Abajo (D).

10. ¿Cuántos ciclos crees que serán necesarios para aprender la política optima?¿Por qué?

Se necesitarán al menos 5 ciclos, ya que es la distancia mínima (5 acciones o transiciones) desde el origen a la meta, todas las acciones son deterministas, no hay acciones con refuerzo negativo, y las únicas acciones con refuerzo positivo son las que transitan al estado meta desde algún estado vecino.

Por ello, para que el refuerzo de la acción “L” del nodo 33 al nodo “S” (la más cercana de las dos únicas acciones con refuerzo positivo) se propague hasta las acciones posibles en el nodo “I” serán necesarios cinco ciclos.

11. Generar la tabla Q ejecutando el código anterior y responder a las siguientes preguntas:

Se ha generado la siguiente tabla. En verde se muestran el estado inicial y estado meta. En rojo se muestran las acciones elegidas por la política generada.

S/A	L	R	U	D
00	0.0	0.0	0.0	0.41
01	0.0	0.0	0.0	0.0
I	0.0	0.67	0.0	0.0
03	0.54	0.0	0.0	1.05
04	0.67	0.0	0.0	0.0
10	0.0	0.0	0.33	0.84
13	0.0	0.0	0.84	1.48
20	0.0	1.31	0.67	0.0
21	1.05	0.0	0.0	1.85
23	0.0	1.18	1.18	1.95
24	1.48	0.0	0.0	1.59
31	0.0	2.48	1.48	0.0
S	1.98	1.98	0.0	0.0
33	2.59	1.59	1.59	0.0
34	2.07	0.0	1.27	0.0

Tabla 1: Tabla Q MDP determinista para 5 ciclos, α = 1.0, γ = 0.8

a) ¿Qué alfa y gamma has utilizado? ¿Por qué?

Hemos utilizado un valor de alfa de 1,0 y un valor de gamma de 0,8.

Hemos situado alfa a su máximo valor dado que todas las acciones posibles son deterministas, por lo que el ratio de aprendizaje del agente será constante. Por este motivo y en este caso, un valor máximo de alfa ayuda a ajustar antes los valores de refuerzo de la política con mayor precisión.

Hemos elegido un valor elevado para gamma buscando favorecer y potenciar el refuerzo futuro, acelerando la propagación de este. En realidad y para este caso, cualquier valor elevado funcionaría bastante bien (mejor que un valor gamma reducido), ya que todas las acciones son deterministas, no hay acciones con refuerzo negativo, y las únicas acciones con refuerzo positivo son las que transitan al estado meta desde algún estado vecino.

b) ¿Se genera la política óptima?

Sí, el agente realiza el camino más corto que consiste en el siguiente:

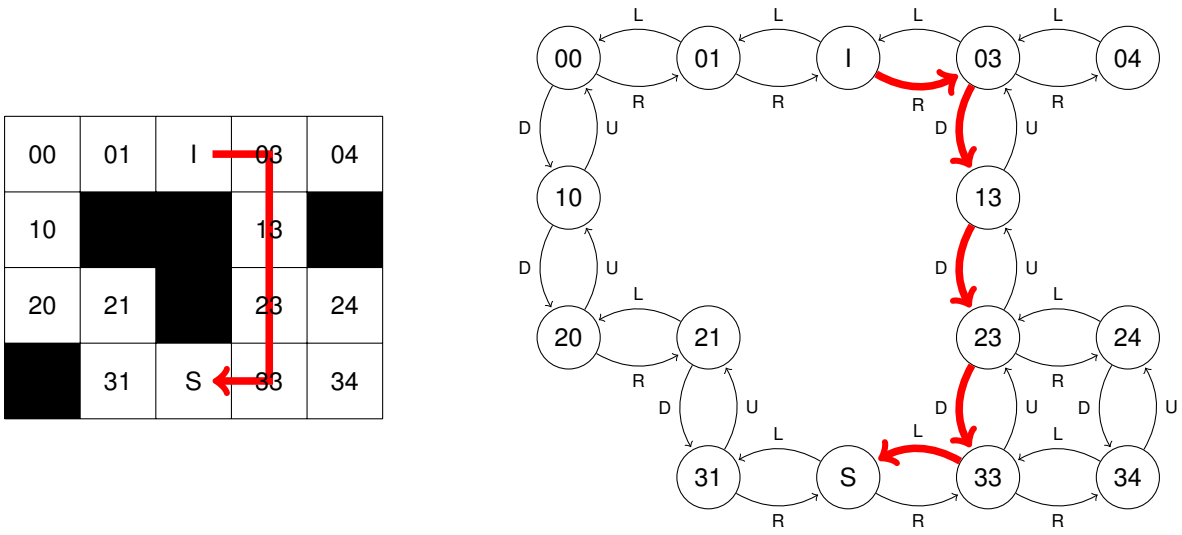

Figura 2: Política MDP determinista

c) ¿Cuántos ciclos se necesitan para obtener la política óptima y cuál es esta política?

Como se ha explicado en diversos apartados anteriores. El agente es capaz de calcular la política óptima en 5 ciclos como mínimo, y dicha política es la representada en la imagen superior, así como en la *tabla Q anterior*.

MDP estocástico

12. Ilustración del modelo

El MDP sigue siendo prácticamente el mismo, pues la mayoría de acciones transitan al estado correspondiente con probabilidad 1. Solo la acción Izquierda (L) del estado 33 es no determinista. Se refleja este cambio en rojo en la siguiente imagen.

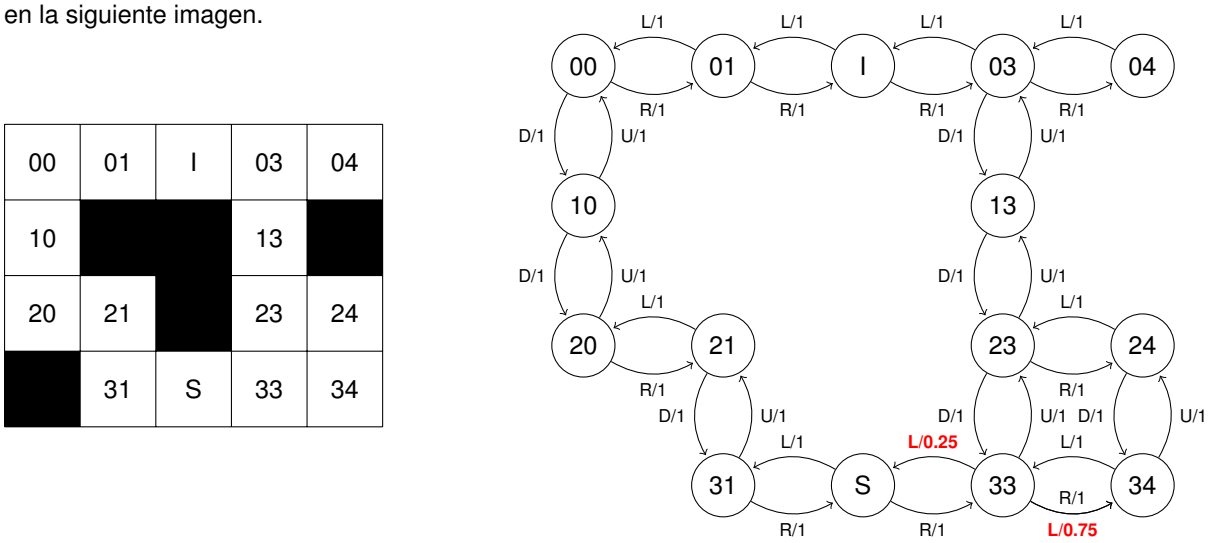

Figura 3: MDP **NO** determinista

13. Generar las tuplas de todos los episodios para el problema del apartado anterior y generar la tabla Q utilizando distintos valores para alfa y gamma. Responder a las siguientes preguntas:

Respecto al MDP anterior, se ha modificado el código de `main.java` para incluir 4 tuplas para el estado 33 y la acción "L". Tres de estas tuplas transitan al estado "34" (probabilidad 0.75) y solo la tupla restante transita al estado "S" (probabilidad 0.25).

Al igual que en la tabla anterior, se muestran en verde el estado inicial y estado meta y en rojo las acciones elegidas por la política generada.

S/A	L	R	U	D
00	0.0	0.0	0.0	0.01
01	0.0	0.0	0.0	0.0
I	0.0	0.02	0.0	0.0
03	0.01	0.0	0.0	0.17
04	0.02	0.0	0.0	0.0
10	0.0	0.0	0.01	0.1
13	0.0	0.0	0.08	0.57
20	0.0	0.37	0.05	0.0
21	0.2	0.0	0.0	0.86
23	0.0	0.31	0.31	1.19
24	0.57	0.0	0.0	0.76
31	0.0	1.55	0.53	0.0
S	1.05	1.19	0.0	0.0
33	1.94	0.76	0.76	0.0
34	1.37	0.0	0.46	0.0

Tabla 2: Tabla Q MDP **NO** determinista para 5 ciclos, α = 0.5, γ = 0.8

a) ¿Qué valores has utilizado para alfa y gamma y por qué esos valores?. Intenta explicar para qué sirven alfa y gamma.

Hemos utilizado un valor de alfa de 0,5 y un valor de gamma de 0,8.

Hemos situado alfa a un valor medio dado que **no** todas las acciones posibles son deterministas, por lo que queremos "filtrar" el ratio de aprendizaje del agente. Con este valor, que es distinto de 1, conseguimos que el agente tenga en cuenta no solo el refuerzo conseguido en cada ciclo, sino también el refuerzo conseguido en ciclos anteriores.

Hemos elegido un valor elevado para gamma buscando favorecer y potenciar el refuerzo futuro, acelerando la propagación de este. En realidad y para este caso, cualquier valor elevado funcionaría bastante bien (mejor que un valor gamma reducido), ya que **casi** todas las acciones son deterministas, no hay acciones con refuerzo negativo, y las únicas acciones con refuerzo positivo son las que transitan al estado meta desde algún estado vecino.

Así pues, podemos concluir que **alfa es el ratio de aprendizaje del agente**, determina hasta que punto la nueva información sustituye a la anterior, siendo este ritmo de sustitución mayor cuanto más elevado es el valor de alfa.

Por otra parte, **gamma es el factor de descuento del refuerzo** o recompensa, y determina la importancia de recompensas futuras.

Los valores límite de estos parámetros afectan al aprendizaje de la siguiente forma:

- α **= 1:** la información nueva sustituye por completo a la información conocida, común en los MDP deterministas.
- α **= 0:** no se sustituye nunca la información conocida inicialmente por el agente, por lo que este no aprende nada.
- γ **= 1:** fomenta el aprendizaje de comportamientos que reportan recompensa a largo plazo.
- γ **= 0:** fomenta el aprendizaje de comportamientos que reportan recompensa inmediata.

b) ¿Qué diferencias hay entre las distintas tablas Q y las políticas obtenidas?

La política obtenida es exactamente la misma que para el MDP determinista, al menos con los valores de α = 0.5 y γ = 0.8 utilizados para la generación de la *tabla anterior*, independientemente del número de ciclos (con ciclos $\geq$ 5). Esta política se puede visualizar sobre el tablero y el grafo del MDP determinista en la *imagen de política* de dicho MDP, pues reincidimos en que esta política es la misma.

Sin embargo, si comprobamos ambas tablas, vemos que en general el valor-acción es menor en todos los casos en la *tabla del MDP **no** determinista* frente a la *tabla del MDP determinista*. Esto se debe principalmente a que estamos filtrando el aprendizaje mediante un valor de alfa menor que 1, y a que la baja probabilidad con la que se transita del estado “33” al estado “S” con la acción “L” (que forma parte de la política) provoca una notoria reducción del refuerzo conseguido en dicha situación, lo que afecta a su propagación hacia el estado “I”.

Material entregado

Junto a esta memoria se entrega el código utilizado para la realización del tutorial 4. Este código es el mismo que ha sido descargado de Aula Global, y solo se ha modificado el fichero `main.java`.

Concretamente, el contenido de los arrays `acciones` (línea 21), `estados` (22), `estado` (24), `estadoFinal` (25), los parámetros pasados al constructor de `QLearning` en la instanciación de `ql` (31) y las tuplas de la línea 36 a la 70.

Se entrega el código en el mismo estado en el que se encontraba tras terminar este tutorial. Esto es, tras realizar el algoritmo de Q-learning sobre el MDP no de terminista o estocástico.

Problemas encontrados a la hora de realizar este tutorial

El único problema encontrado durante la realización de este tutorial ha sido a la hora de generar las transiciones del MDP estocástico en el código java proporcionado. No hemos encontrado en el código ni los comentarios del fichero `main.java` ni del resto de ficheros una opción para indicar la probabilidad de las transiciones, que nos

ha sido necesario para la acción “L” en el estado “33”. Tampoco hemos encontrado documentación o información acerca de este paso en el enunciado, o si estaba, esta no era clara.

Finalmente se ha optado por incluir 4 tuplas para la acción “L” en el estado “33”. Tres tuplas que transitan hacia el estado “34” (probabilidad de 0.75) y una sola que transita al estado “S” (probabilidad de 0.25). De esta forma se ha intentado emular el no determinismo de la acción.

Conclusiones

En general, este ha sido un tutorial fácil y que nos va a ayudar de cara a la realización de la próxima práctica, en la que tendremos que programar un agente inteligente de Mario que aprenda por refuerzo. El completar este tutorial nos ha permitido comprender mejor el funcionamiento del algoritmo de aprendizaje por refuerzo *Q-Learning*, así como la importancia y el papel que juegan en el mismo los parámetros alfa y gama.

Comentarios personales

Hemos tardado entre 3 y 4 horas en la realización de este tutorial, mucho más allá de la duración de una hora que se indicaba en la cabecera del enunciado. Asimismo, hemos preguntado a algunos de nuestros compañeros y coincidían con nosotros en que el tiempo empelado en la realización del tutorial ha sido mayor que el indicado.

Creemos que esto se debe a varios factores, como son:

- Tiempo empleado en el siempre necesario repaso de la teoría de la asignatura.
- Tiempo empleado en entender el código proporcionado que ha sido programado por un tercero.
- Tiempo empleado en la experimentación (que a veces va acompañado de una investigación, ya sea un repaso de la teoría más lento y detallado o búsquedas en Internet).
- Tiempo empleado en la documentación de la práctica, teniendo en cuenta revisiones ortográficas y de formato, y atendiendo a la corrección de las respuestas.
- Asimismo, el tiempo empleado en entender algunos comentarios del enunciado, cuyas explicaciones en ocasiones resultan ambiguas o poco clara, lo cual entendemos que es difícil de detectar para la persona que lo escribe y dada en ocasiones la diferencia de entrenamiento o conocimiento en el tipo de trabajo a realizar.

Tenemos la sensación de que no se se tiene en cuenta, probablemente por desconocimiento, el verdadero trabajo que hay detrás de una tarea aparentemente (y que de hecho es) fácil, como es este tutorial. Esto induce a un mal cálculo o aproximación, que sumado a la carga de trabajo del resto de asignaturas, en ocasiones nos impide dedicarle el tiempo y esfuerzo que nos gustaría a estas prácticas.