Inteligencia Artificial (IS) 2019/20
Propuesta de trabajo

Aprendizaje por Refuerzo

Antonia M. Chávez González

1. Introducción y objetivos

El aprendizaje por refuerzo es una técnica de aprendizaje automático que se asemeja al proceso natural de aprendizaje por ensayo/error.
En el *aprendizaje supervisado tradicional*, facilitamos al agente un conjunto de datos de atributos junto con etiquetas (clases) y solicitamos la predicción (tras valoraciones estadísticas o mediante entrenamiento) de la clasificación de unos nuevos datos. En el *aprendizaje no supervisado*, facilitamos al sistema un conjunto de datos y se solicita que se descubran patrones o estructuras ocultas en los mismos.
Para el aprendizaje por refuerzo no facilitamos datos ni etiquetas, no hay conjuntos de entrenamiento; el proceso viene marcado por las recompensas que el entorno concede al agente según las acciones que decida realizar. Éste aprende a interactuar para maximizar la recompensa final. Este enfoque del aprendizaje es el utilizado en los videojuegos.

En [1] puede leerse una sencilla introducción que ayuda a conocer:

- los elementos (estados/acciones/probabilidades/recompensas/política...) de un pequeño problema didáctico que se pone como ejemplo
- el funcionamiento de un algoritmo de aprendizaje aplicado a ese problema
- el papel de algunos de los parámetros que intervienen en dicho algoritmo

El **objetivo 1** de este trabajo consiste en estudiar el documento del enlace para dominar su funcionamiento y el papel de cada uno de los parámetros que intervienen en él.

En base a ese estudio, como **objetivo 2** del trabajo, proponemos la implementación en Python de varios procedimientos para resolver el ejercicio que se plantea en cada fase. El resultado se entregará en un notebook de Jupyter debidamente comentado y documentado.

2. El Ejercicio

2.1. Fase 1

Se pretende recorrer el tablero siguiente desde la casilla 0 a la 6.

8 | 7 | 6

3 | 2 | 5

0 | 1 | 4

Se elegirá una representación del tablero, como grafo, matriz, etc.

Se definirá la matriz R de recompensas con valores -1 para las transiciones no válidas (por ejemplo de 0 a 4), valores 0 cuando la transición de una casilla a otra sea válida (por ejemplo, de 0 a 1,2 o 3) y la recompensa será 100 cuando alcancemos el objetivo: 6

A continuación se inicializará a cero la matriz Q de valores.

Se establecerá un número de episodios de entrenamiento.

Se implementará el algoritmo de entrenamiento de la matriz de valores Q, tal como se lee en el pseudo-código del artículo de referencia:

The Q-Learning algorithm goes as follows:

For each episode:

Select a random initial state.

Do While the goal state hasn't been reached.

- Select (randomly) one among all possible actions for the current state.
- Using this possible action, consider going to the next state.
- Get maximum Q value for this next state based on all possible actions.
- Compute: Q(state, action) = R(state, action) + Gamma * Max[Q(next state, all actions)]
- Set the next state as the current state

End Do

End For

Una vez entrenada la matriz Q, comprobar que el agente encuentra el mejor camino, en este caso el más corto.

Nota: Todos los parámetros mencionados en este trabajo deberán ser configurables por el usuario, el tablero, dimensiones y celdas, objetivo e inicio, el número de episodios de entrenamiento, el factor de aprendizaje gamma, etc. Se coleccionará el rendimiento de cada episodio de entrenamiento para representar gráficas que irán acompañadas de su interpretación. Puede considerarse la medida del rendimiento como la suma de (los valores de Q divididos entre el máximo valor de Q) por 100.

Con este fin, se implementará una interfaz o formulario donde el usuario seleccionará/introducirá dichos valores y los que en las diferentes fases se indiquen.

2.2. Fase 2

En esta fase se implementará una variante del algoritmo de entrenamiento en la que aparecen dos nuevos parámetros: épsilon y alpha, ambos en [0,1].

El algoritmo introduce épsilon al elegir cuál va a ser el next_state del bucle while: se generará un número aleatorio. Si éste es menor que épsilon, la transición desde el estado action se elegirá aleatoriamente entre las válidas desde ese estado. En caso contrario, se elegirá la transición que aparezca en la tabla Q que tenga el mayor valor desde el estado action, tal como dice el algoritmo que se implementa en la Fase 1.

El parámetro alpha se multiplicará por épsilon en cada episodio de manera que épsilon irá variando.

Estos parámetros influyen en el ritmo de aprendizaje y el alumno deberá interpretar y describir cómo.

2.3. Fase 3

Finalmente, se diseñará libremente un grafo o mapa de mayores dimensiones, con diferentes recompensas intermedias y transiciones prohibidas.

Para realizar un estudio comparativo de los dos algoritmos, se aplicará el algoritmo y su variante a dicho problema una vez establecidos los parámetros necesarios. Se explicará qué tipo de recorrido es mejor, se elaborarán gráficos y tablas en las que se pueda visualizar y valorar el aprendizaje según la relación entre los parámetros épsilon, alpha, gamma, nº de episodios, y se describirán y analizarán los resultados de cada algoritmo. Se analizará la escalabilidad de los resultados.

3. Descripción de la memoria del trabajo

Este trabajo consta de dos objetivos cuya consecución se plasmará en la memoria como se indica:

- La consecución del Objetivo 1 constituirá el contenido de la Introducción que debe aparecer en la memoria.

  Respecto al Objetivo 2:
- Deberá haber un apartado donde se describan las decisiones sobre el diseño de la representación del ejercicio, las librerías elegidas y su justificación. Constarán las características del ordenador utilizado en relación con su rendimiento.
- Las implementaciones del algoritmo de las fases 1 y 2 (para el entorno inicial) y su aplicación en la fase 3 a un nuevo entorno más complejo, se entregarán usando cuadernos de Jupyter debidamente documentados y comentados. En la memoria se volcarán las tablas comparativas y las gráficas obtenidas en la experimentación que sean relevantes. Además se incluirá el análisis de la interpretación de tales gráficas y tablas.
- Deberá incluirse un apartado de Conclusiones en el que se explicarán las conclusiones extraídas del análisis anterior.
- En un apartado de Bibliografía aparecerán las referencias de cualquier documento consultado durante la realización del trabajo (incluidos los links a páginas o repositorios y fechas de último acceso a los mismos).
- No deberá incluirse código en la memoria

4. Formato de entrega

En la página web de la asignatura se pueden encontrar plantillas donde se sugiere una estructura general. Estas plantillas siguen el formato de los *IEEE conference proceedings* cuyo sitio web *guía para autores* [2] ofrece información más detallada. Se facilitan plantillas para LaTeX, LibreOffice y Word. El artículo entregado deberá tener una extensión mínima de 6 páginas.

5. Mejoras

Podrán recibir puntuación adicional mejoras o añadidos que se incorporen al trabajo como por ejemplo:

- incluir un visor que muestre gráficamente los entornos y los recorridos de entrenamiento y óptimo
- el uso de la plantilla LaTeX
- otras mejoras propuestas por los autores

6. *Presentación y* defensa

El día de la defensa se deberá realizar una pequeña presentación (PDF, PowerPoint o similar) de 10 minutos en la que participarán activamente los miembros del grupo que han desarrollado el trabajo. Esta presentación seguirá la misma estructura que el documento. Se podrá usar un portátil (personal del alumno), diapositivas y/o pizarra.

En los siguientes 10 minutos de la defensa, el profesor procederá a realizar preguntas sobre el trabajo, el documento y el código fuente.

7. Criterios de evaluación

Para que el trabajo pueda ser evaluado, deberá satisfacer la descripción del apartado 3.

La entrega consistirá de **un único fichero zip** conteniendo:

- El código fuente implementado (cuadernos de Jupyter).
- Un fichero README.txt, que resuma la estructura del código fuente, e indique cómo usar la interfaz (si se ha implementado), o al menos cómo aplicar el algoritmo al problema y cómo hacer pruebas con los parámetros del mismo, incluyendo ejemplos de uso. Es importante la coherencia de este fichero con la defensa.
- El documento – artículo en formato PDF. Deberá tener una extensión mínima de 6 páginas. Deberá incluir en el apartado correspondiente toda la bibliografía consultada (libros, artículos, technical reports, páginas web, códigos fuente, diapositivas, vídeos, etc.) y además esa bibliografía deberá estar debidamente referenciada a lo largo del documento.

Para la evaluación se tendrá en cuenta el siguiente criterio de valoración, considerando una nota máxima de 3 puntos en total para el trabajo:

- El código fuente (hasta 1,5 puntos):
  Se valorará la claridad y buen estilo de programación, corrección y eficiencia de la implementación, y calidad de los comentarios. La claridad del fichero README.txt también se valorará. En ningún caso se evaluará un trabajo con código copiado directamente de internet o de otros compañeros.
- El documento – artículo científico (hasta 0,75 puntos):
  Se valorará la claridad de las explicaciones, el razonamiento de las decisiones, el análisis y presentación de resultados, y el uso del lenguaje. Se valorará el uso de una plantilla de las que se facilitan. Igualmente, no se evaluará el trabajo si se detecta cualquier copia del contenido.
- La presentación y defensa (hasta 0,75 puntos):
  Se valorará la claridad de la presentación y la buena explicación de los contenidos del trabajo así como, especialmente, las respuestas a las preguntas realizadas por el profesor.

- Mejoras: Se valorarán hasta con 0.5 puntos extra sin superar el máximo de 3 puntos totales del trabajo.

IMPORTANTE: Cualquier plagio, compartición de código o uso de material que no sea original y del que no se cite convenientemente la fuente, significará automáticamente la calificación de cero en la asignatura para todos los alumnos involucrados. Por tanto, a estos alumnos no se les conserva, ni para la actual ni para futuras convocatorias, ninguna nota que hubiesen obtenido hasta el momento. Todo ello sin perjuicio de las correspondientes medidas disciplinarias que se pudieran tomar.

8. Referencias

[1] Documento inicial. http://mnemstudio.org/path-finding-q-learning-tutorial.htm
[2] Plantilla IEEE. https://www.ieee.org/conferences/publishing/templates.html