Tarea 1 - Aprendizaje por Refuerzo
Fecha de entrega: Lunes 6 de noviembre de 2023

Para esta tarea utilizaremos OpenAI Gym con el ambiente GridWorld, adaptado de [1], que ud. debe descargar e instalar. El GridWorld es un ambiente creado para OpenAI Gym, que se basa en mapas 2D. En este ambiente, un agente debe aprender a llegar hasta la salida evitando las bombas. El estado está definido por un número de 0 a n (donde n = (número de filas x número de columnas)-1). Cada mapa se encuentra en un archivo .txt (ej. map1.txt) que contiene las siguientes características:

- 0 : celda vacía
- x: posición inicial del agente
- G : posición objetivo
- 1 : obstáculo/pared
- B : bomba (conlleva penalización)

Para esta tarea usaremos 2 mapas del GridWorld, uno abierto de 10x10 y otro con obstáculos de 11x10. Se puede escoger el mapa en la línea 17 del archivo GridWorld.py. Utilice el código incluído en tarea1.py (que cuenta con los métodos de RL tabular según visto en las ayudantías) junto con los mapas del ambiente Gridworld para responder las siguientes preguntas:

1) Entrene a un agente Q-learning y SARSA para map1.txt. Muestre la curva de aprendizaje (recompensas vs episodio) en cada caso. Intente hacer lo mismo en map2. Como verá, en este mapa el agente no es capaz de aprender a llegar al objetivo. Para resolver ésto, pruebe modificando algún hiperparámetro en el archivo tarea1.py. Modificando distintos hiperparámetros el agente toma diferentes caminos para el objetivo. Indique qué hiperparámetros se deben ajustar (y con qué valores) para que el agente escoja diferentes caminos al objetivo con cada algoritmo, y explique por qué ocurre lo observado en cada caso. ¿En qué caso los algoritmos encuentran la política óptima?

2) Cambie el código de GridWorld.py para que la transición de acciones sea estocástica: con probabilidad 0.7 el agente se mueve según su intención, con prob. 0.15 se mueve a la izq. de la dirección deseada, y con prob. 0.15 se mueve a la derecha de la dirección deseada. Entrene nuevamente Q-learning y SARSA. ¿Se mantiene la relación entre el desempeño de ambos algoritmos en el ambiente estocástico? ¿por qué?

3) Haga una copia del método Q-learning en tarea1.py y modifíquela para convertir Q-learning en Double Q-learning. Grafique la curva de aprendizaje en map2 y compárela con la de Q-learning. En base a lo visto en clase sobre Double Q-learning, explique el por qué de la diferencia (o similitud) entre ambos resultados. En caso de ser similares, ¿qué cambio del ambiente cree ud. que llevaría a resultados distintos entre Q-learning y Double Q-learning? Justifique.

4) **TD(lambda)**. Para realizar lo que se solicita en esta pregunta, comience leyendo el capítulo 7 de la primera edición del libro guía del curso (Reinforcement Learning: and introduction): “Eligibility Traces”, hasta la subsección 7.6 (incluída): http://www.incompleteideas.net/book/ebook/node72.html

   Las eligibility traces son una forma de combinar los métodos TD y Monte Carlo, permitiendo actualizar más de un estado en la trayectoria actual por cada recompensa recibida. Puede ayudarse también con estas referencias:

   https://people.cs.umass.edu/~barto/courses/cs687/Chapter%207.pdf
   https://www.youtube.com/watch?v=xc0IwNI3NHU

   Implemente Q(lambda) y SARSA(lambda) en tarea1.py, con lambda de 0.5, y compare los resultados con los métodos evaluados en la pregunta 2) en map2. Grafique la curva de aprendizaje en cada caso.

Se debe entregar, mediante enlace a habilitar en canvas, un archivo comprimido que contenta:

a) el código desarrollado en el archivo tarea1.py (con comentarios)
b) Un informe con los gráficos y los análisis respectivos, junto con instrucciones de ejecución del código.

---

[1] GitHub de Jacopo Castellini: https://github.com/opocaj92/GridWorldEnvs

Para instalar desde este github, descargue o clone el repositorio e instale con:

pip3 install -e .
(el punto es parte del comando)

Luego de instalar, reemplace el archivo GridWorld.py con el archivo adjunto a esta tarea, y los archivos de mapas map1.txt y map2.txt en la carpeta de los ambientes (env).
Al llamar al método playgames en el archivo tarea1.py se abre una ventana de visualización (arriba y a la izquierda), que se puede agrandar manualmente para ver paso a paso la ejecución.