



DEPARTAMENTO DE INFORMÁTICA  
UNIVERSIDAD CARLOS III DE MADRID

# Grado en Ingeniería en Informática

Inteligencia Artificial  
Convocatoria ordinaria  
Mayo 2016

## Normas generales del examen

- El tiempo para realizar el examen es de **2.5 horas**
- No se responderá a ninguna pregunta sobre el examen
- Si se sale del aula, no se podrá volver a entrar durante el examen
- No se puede presentar el examen escrito a lápiz

## Preguntas de teoría (4 puntos)

Responda a las preguntas explicando brevemente la respuesta.

- (2.5 puntos) Se quiere controlar el comportamiento de un robot mediante un sistema de producción. El robot tiene cuatro sensores (S1, S2, S3 y S4) y dos ruedas (W1 y W2). Cada sensor devuelve un valor entero entre 1 y 10. Las acciones fijan la velocidad de cada rueda a valores enteros entre 1 y 5. Las reglas deben chequear el valor de todos los sensores y determinar la velocidad de las dos ruedas.
  - ¿De qué estaría compuesta la base de hechos (o memoria de trabajo)?
  - ¿Cuántos estados diferentes se podrían crear con esa memoria de trabajo?
  - ¿Cuántas reglas diferentes tendría el sistema?
  - ¿Cuántas instancias de reglas contendría el conjunto conflicto dada una base de hechos concreta?
- (2.5 puntos) Rachel quiere ir de un lugar a otro en una ciudad, utilizando con conjunto de  $L$  líneas de autobús. Por simplicidad, asumiremos que los lugares se corresponden exactamente con las paradas de autobús. Cada línea de autobús tiene un conjunto de  $S$  paradas y un conjunto de  $B$  autobuses. Cada autobús está en una parada de autobús en cada instante de tiempo. Varios autobuses pueden coincidir en la misma parada sólo si pertenecen a distintas líneas. Los autobuses se mueven de una parada a la siguiente en su línea en cada instante de tiempo. Las líneas son circulares (no tienen principio ni final). Los instantes de tiempo están discretizados de 10 en 10 minutos desde las 6am hasta las 11pm (6:00, 6:10, 6:20, ..., 23:00).
  - Defina el espacio de problemas correspondiente, es decir, cuáles son los estados, acciones, estado inicial y metas.
  - ¿Cuál es aproximadamente el tamaño del espacio de estados?
  - Si Rachel no tiene ninguna restricción sobre la calidad de la solución (tiene dinero suficiente y no tiene prisa), **seleccione** tres técnicas de búsqueda diferentes que proporcionen una solución correcta.
  - Si Rachel quiere llegar al destino tan rápido como sea posible, **seleccione** dos técnicas de búsqueda que proporcionen una solución correcta.
  - Si cada línea de autobús tiene un precio distinto y Rachel quiere obtener la solución más barata, **seleccione** dos técnicas de búsqueda que proporcionen una solución correcta.
- (2.5 puntos) Explique qué es un Modelo Oculto de Markov (HMM) y su relación con las Redes Bayesianas ¿Cuál es la propiedad de Markov? ¿Cómo se aplica el concepto de independencia condicional en los HMMs?
- (2.5 puntos) Explique qué es una neurona artificial y una red de neuronas artificiales. Explique en qué consiste el aprendizaje en redes de neuronas. Ponga un ejemplo de una aplicación práctica de las redes de neuronas artificiales.

## Preguntas sobre la práctica (1 punto)

- (0.5 puntos) Suponga que en el problema del *gridworld* se define el grid de la Figura 1 de nombre examGrid y luego se ejecuta el comando: `python gridworld.py -a value -i 1 -g examGrid --discount 1 --noise 0.1`

2			<b>+10</b>
1	<b>-1</b>		
Y=0			<b>+1</b>
	X=0	1	2

Figure 1: Definición del grid examGrid.

Tras la finalización del algoritmo de iteración de valor (con el número de iteraciones indicado en la línea de comandos con -i), responda a las siguientes preguntas, explicando brevemente las respuestas:

- ¿Qué contendrá la variable `self.values` de la clase `ValueIterationAgent`?
  - ¿Qué devolverá el método `computeQValueFromValues` de la clase `ValueIterationAgent` si se le pasa como argumentos el estado (0,0) y la acción `east`?
  - ¿Qué devolverá el método `computeActionFromValues` de la clase `ValueIterationAgent` si se le pasa como argumentos el estado (2,1)?
- (0.5 puntos) Explique detalladamente la/s variable/s que ha usado para hacer el aprendizaje de la función de transición en el MDP del Pacman.

## Solución

### Pregunta 1

- La base de hechos debe contener el estado de los cuatro sensores S1, S2, S3 y S4 (un valor entre 1 y 10 para cada uno de ellos) y las velocidades de las ruedas W1 y W2 (un valor entre 1 y 5 para cada una) en el momento actual.
- Se podrían crear  $10^4 \times 5^2$  estados diferentes. Todas las combinaciones posibles de valores de sensores y velocidades.
- Las reglas tendrían la forma:  
If S1=s1 AND S2=s2 AND S3=s3 AND S4=s4 THEN W1=w1 and W2=w2  
El sistema tendría  $10^4$  reglas. Estas reglas chequean el valor de los sensores para cada combinación posible y determinan la velocidad de las ruedas. El número de reglas total que se podrían crear sería  $10^4 \times 5^2$ .
- Dado que cada regla chequea el valor de los sensores para una combinación de valores de los mismos, sólo podría haber una instancia en el conjunto conflicto en cada ciclo.

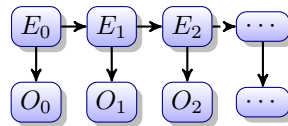
### Pregunta 2

- El espacio de problemas estaría compuesto por:
  - Un conjunto de estados. Cada estado debe contener la siguiente información:
    - Instante de tiempo.
    - Situación de Rachel en ese instante de tiempo. Rachel podría estar o bien en una parada, o bien subida a un autobús, que está en una parada.

- Situación de los autobuses: posición (parada) de cada autobús de cada línea. Hay un total de  $L \times B$  autobuses.
- También habría que representar el mapa de paradas de las líneas, aunque esto no es necesario que se introduzca en los estados, ya que es estático.
- Un conjunto de acciones: las acciones podrían ser **subir-a-autobús** (Rachel se sube a un autobús concreto que está en su parada), **bajar-de-autobús** (Rachel se baja de un autobús) y **avanzar** (el sistema avanza un instante de tiempo, con lo que los autobuses avanzan una posición hacia delante).
- Un estado inicial es un estado concreto en el instante inicial, que define la situación de Rachel y la de los autobuses.
- Las metas definen la situación final de Rachel (parada a la que quiere ir).
- El tamaño del espacio de estados sin considerar el mapa sería aproximadamente  $T \times S^{L \times B} \times (L \times (S + B))$  donde  $T$  es el número de instantes de tiempo posibles,  $S^{L \times B}$  es el número de posiciones posibles de autobuses para todos ellos,  $L \times (S + B)$  es el número máximo de posiciones de de Rachel. El tamaño del mapa sería  $L \times S$ .
- Cualquier técnica de búsqueda podría resolver esta tarea. Preferiblemente una completa. Por ejemplo amplitud, profundidad y búsqueda heurística como  $A^*$ .
- Únicamente técnicas admisibles para costes no uniformes (tiempo) como Dijkstra y  $A^*$ .
- Únicamente técnicas admisibles para costes no uniformes (precio) como Dijkstra y  $A^*$ .

### Pregunta 3

- Un modelo oculto de Markov (HMM) es un modelo probabilístico que representa cómo los estados de un sistema cambian con el tiempo, o bien secuencias de estados. En un HMM se considera que el estado del sistema en un momento dado no se conoce (está oculto) pero se puede observar parcialmente o de forma imperfecta, es decir, hay incertidumbre en la relación entre lo que se observa y el estado real. Un HMM es una red bayesiana con el siguiente aspecto:



donde los  $E_i$  representan estados (ocultos) y los  $O_i$  representan observaciones. Concretamente un HMM se define por:

- Un conjunto de estados
- Un conjunto de observaciones
- La probabilidad a priori del estado inicial  $P(E_0)$
- El modelo de transiciones  $P(E_{t+1}/E_t)$
- El modelo de observaciones  $P(O_t/E_t)$

La propiedad de Markov se refiere a que el estado del sistema en un momento dado depende del estado anterior, pero no de otros estados previos a éste:  $P(E_{t+1}/E_t, E_{t-1}, \dots, E_0) = P(E_{t+1}/E_t)$

La independencia condicional en HMM's se aplica de la siguiente forma:

- Cada estado  $E_{t+1}$  es condicionalmente independiente de todos los nodos que no son descendientes suyos, dado el estado anterior  $E_t$ .
- Cada observación  $O_t$  es condicionalmente independiente de todos los demás nodos de la red dado el estado en ese instante de tiempo  $E_t$ .

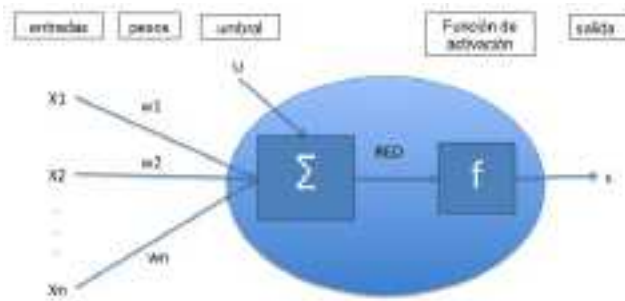


Figure 2: Estructura de una neurona.

## Pregunta 4

Una neurona artificial es una unidad de cómputo que dadas unas entradas  $x_i$  y unos pesos  $w_i$  para cada una de ellas genera una salida  $s$ . Concretamente la salida se genera mediante la aplicación de una función de activación  $f$  a la combinación lineal de las entradas ponderadas por los pesos:  $s = U + w_1 * x_1 + w_2 * x_2 + \dots + w_n * x_n = U + \sum_{i=0}^n w_i x_i$ . Donde  $U$  es el umbral o término independiente.

Una neurona se suele representar mediante un modelo gráfico como el que muestra la Figura 2.

Existen distintos tipos de funciones de activación, las más habituales y conocidas son las funciones sigmoidales y las funciones escalón.

Una red de neuronas es un conjunto de neuronas interconectadas entre si, de manera que las salidas de unas neuronas son entradas de otras neuronas. Las redes de neuronas se organizan en capas. Las conexiones se suelen dar entre neuronas de una capa y la siguiente o entre neuronas de la misma capa.

El aprendizaje en redes de neuronas consiste en ajustar los pesos para que la red produzca las salidas deseadas ante unas determinadas entradas.

Las redes de neuronas se utilizan principalmente para tareas de predicción y tareas de clasificación. Un ejemplo de aplicación práctica visto en clase, sería el reconocimiento de dígitos/letras escritos a mano. Esta es una tarea de clasificación en la que las entradas serían los píxeles de la imagen con la letra escrita a mano a reconocer y las salida sería la letra real que se ha escrito. Normalmente existen diversas formas de codificar las entradas y salidas. Las decisiones sobre cómo realizar esta codificación, así como el número de capas ocultas de la red forman parte del diseño de la misma.