



DEPARTAMENTO DE INFORMÁTICA
UNIVERSIDAD CARLOS III DE MADRID

Ingeniería en Informática

Aprendizaje Automático

Junio 2006

Normas generales del examen

- El tiempo para realizar el examen es de **3 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

Pregunta 1/3 (4 puntos)

Se dispone de la siguiente base de datos, compuesta por las tablas 1 y 2:

Empleado	Compañía	Sueldo	Ascendido
juan	ibm	bajo	sí
luis	microsoft	alto	sí
pepe	telefónica	alto	no
carmen	microsoft	alto	sí
pedro	amena	medio	no
rocío	telefónica	alto	sí
maría	amena	bajo	no
raquel	telefónica	bajo	no
fernando	intel	bajo	sí

Cuadro 1: Tabla describiendo la compañía en la que trabaja cada empleado, el sueldo que recibe, y si ha ascendido en el último año o no.

Compañía	Empleados	I+D
ibm	340	sí
telefónica	200	no
amena	100	no
intel	200	sí

Cuadro 2: Tabla describiendo el número de empleados de cada compañía, y si la compañía desarrolla actividades de I+D.

A partir de dicha base de datos, se desea construir un árbol de decisión que modele cuándo un empleado es ascendido o no en su compañía. Para ello, se pide:

- Describir la representación utilizada para describir los ejemplos de entrenamiento
- Representar el conjunto de entrenamiento
- Generar un posible árbol de decisión para ese conjunto de entrenamiento. ¿Qué algoritmo podría utilizarse para generarlo?

- Dado el árbol de decisión generado, aplicamos un método para obtención de reglas de decisión. ¿Cuántas reglas obtenemos? ¿Cuáles son?

Pregunta 2/3 (2 puntos)

Muchos métodos de análisis de datos se basan en una medida de distancia entre las distintas instancias. Dado el conjunto de datos de la tabla :

Instancia	A1	A2	A3	A4
1	0,1	87	azul	13
2	0,12	14	azul	70
3	0,8	3	verde	12
4	0,6	90	verde	41
5	0,23	45,3	azul	93,8
6	0,97	62	verde	91

Cuadro 3: Conjunto de Datos

- Definir una medida de distancia adecuada para realizar agrupación o *clustering* sobre el conjunto de datos.
- ¿Qué método que utilice la medida de distancia anterior (o una similar) podrías usar para calcular el valor del atributo A3 para un ejemplo en el que dispones de los valores de los atributos A1, A2, y A4? ¿Sería útil/necesario algún tipo de preprocesado de los datos?
- Utiliza la medida de distancia y el algoritmo definidos en los apartados anteriores para calcular el valor del atributo A3 de una instancia que tiene los siguientes valores para el resto de atributos: A1 = 0,1, A2 = 56, A4 = 71.

Pregunta 3/3 (4 puntos)

Una de las habilidades más básicas que un jugador de fútbol en un simulador debe tener es la de interceptar el balón. La importancia de esta habilidad viene dada por la dependencia que otras habilidades tienen con esta. Habilidades como golpear el balón o que el portero lo atrape necesitan que el jugador esté suficientemente cerca del balón para poder ser ejecutadas y obtener el resultado deseado. La figura 1 muestra una imagen con el escenario. En ella aparece un delantero a punto de golpear la pelota. Una vez que la golpea, el portero debe ejecutar acciones que le permitan colgarse a una distancia de menos de un metro del balón. En ese momento, se considera que el objetivo ha sido satisfecho, y la tarea resuelta. Si el balón sale fuera del terreno de juego, se asume que es un episodio nulo. Si el balón entra en la portería se considera que el portero ha fallado.

El objetivo de la habilidad es interceptar el balón. Para conseguirlo, el portero debe analizar la información visual que recibe del simulador y obtener la información que necesitan. Esta información no es más que la información relativa del balón con el agente:

- Distancia relativa del balón al jugador (dis).
- Dirección relativa del balón al jugador (dir).
- La variación de la distancia, que da una idea de cómo está cambiando el parámetro distancia (Δ dis).
- La variación de la dirección, que da una idea de cómo está cambiando el parámetro dirección (Δ dir).

Una vez que el jugador obtiene estos parámetros sobre la bola, debe ejecutar la secuencia de acciones más adecuada que le permita alcanzar el balón. Las acciones posibles son dos:

- *Girar(ángulo)*. Cambiar la dirección del jugador de acuerdo a un ángulo entre -180 y 180 grados.
- *Avanzar(fuerza)*. Incrementar la velocidad del jugador en la dirección en la que se encuentra, con una potencia que puede oscilar entre -30 y 100.

En este ejercicio, se pide:

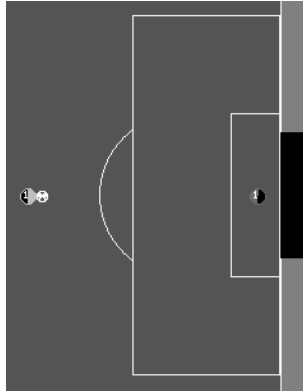


Figura 1: Imagen de la situación inicial para el problema de interceptar el balón

1. Definir qué técnica de Aprendizaje Automático es la más adecuada para resolver este problema.
2. Realizar el diseño en detalle de todos los elementos requeridos para implementar el sistema mediante el sistema de aprendizaje escogido
3. Definir cómo se realizaría el proceso de aprendizaje
4. Definir cómo se realizaría el proceso de test
5. Definir aquellos elementos del proceso de aprendizaje que pudieran ser problemáticos, y plantear distintas alternativas.

Solución Pregunta 1/3

En este problema, se pueden representar los datos siguiendo dos aproximaciones, una basada en atributo-valor, y la otra relacional. Por ejemplo, utilizando una representación atributo-valor, podríamos tener las siguientes instancias:

Empleado	Compañía	Sueldo	Ascendido	empleados	I+D
juan	ibm	alto	sí	340	sí
luis	microsoft	alto	sí	¿?	¿?
pepe	telefónica	alto	no	200	no
carmen	microsoft	alto	sí	¿?	¿?
pedro	ibm	medio	no	340	sí
rocío	telefónica	alto	sí	200	no
maría	amena	bajo	no	100	no
raquel	telefónica	bajo	no	100	no
fernando	intel	bajo	sí	200	sí

Cuadro 4: Representación atributo-valor

Sin embargo, y tal y como se observa en la tabla, una representación atributo-valor tiene varios inconvenientes. El principal es que nos llevaría a ejemplos que no disponen de todos los valores de los atributos. Este problema, no obstante, puede resolverse con distintas técnicas más o menos eficientes, que deberían tratarse en la solución del ejercicio. Otro inconveniente es que el árbol obtenido tendría demasiadas hojas, lo que daría idea de poca capacidad de generalización.

El segundo tipo de representación estaría basado en una representación en lógica de primer orden. El predicado objetivo podría ser un predicado *asciende(nombre-empleado: término, asciende: término)*, que indica para cada empleado si asciende o no. Los datos de entrenamiento se representarían con dos predicados principales:

- empleado(nombre: término, compañía: término, sueldo: numérico)
- compañía(nombre: término, empleados: numérico, i+d: término)

Con lo que que el conjunto de datos quedaría como:

- Ejemplos positivos:
 - *asciende(juan, sí)*
 - *asciende(luis, sí)*
 - *asciende(carmen, sí)*
 - *asciende(rocío, sí)*
 - *asciende(fernando, sí)*
- Ejemplos negativos:
 - *asciende(pepe, no)*
 - *asciende(pedro, no)*
 - *asciende(maría, no)*
 - *asciende(raquel, no)*
- Conocimiento de Fondo:
 - *empleado(juan, ibm, bajo)*
 - *empleado(luis, microsoft, alto)*
 - *empleado(carmen, microsoft, alto)*
 - *empleado(rocío, telefónica, alto)*
 - *empleado(fernando, intel, bajo)*
 - *empleado(pepe, telefónica, alto)*

- empleado(pedro, amena, medio)
- empleado(maría, amena, bajo)
- empleado(raquel, telefónica, bajo)
- compañía(ibm, 340, sí)
- compañía(telefónica, 200, no)
- compañía(amena, 100, no)
- compañía(intel, 200, sí)

El conjunto de entrenamiento se compondría de los predicados sobre empleados, junto con los predicados de las compañías. Así, un algoritmo del tipo S-CART podría generar el árbol de decisión mostrado en la Figura 2.

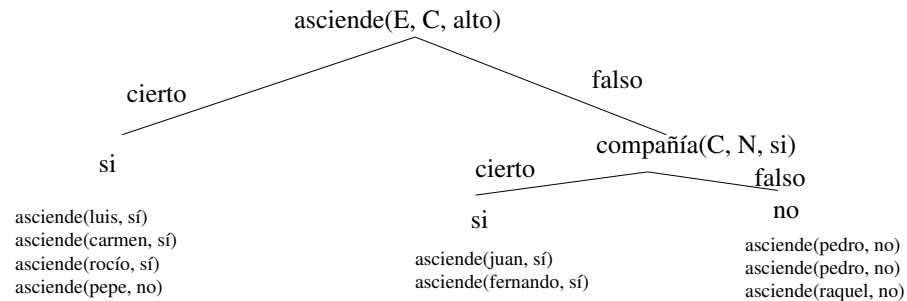


Figura 2: Árbol de Decisión Relacional

En las hojas del árbol se han incluido también los ejemplos que quedan clasificados en esa hoja. Se observa, no obstante, que en la rama de la izquierda hay una tupla clasificada de forma incorrecta. Esta incorrección es ruido del dominio (puesto que hay una tupla con exactamente los mismos valores clasificada de distinta forma), por lo que seguir profundizando en esa rama no tiene sentido.

A partir de ese árbol de decisión, se generan 3 reglas (una por cada nodo), del siguiente modo (en lenguaje *prolog*):

```

asciende(E, sí) :- empleado(E, C, alto), !.
asciende(E, sí) :- compañía(C, _, si), !.
asciende(E, no).

```

Solución Pregunta 2/3

Para realizar clustering, sería necesario utilizar una medida de distancia que tenga en cuenta la distancia individual entre cada par de atributos, como la distancia euclídea:

$$d(\vec{x}_i, \vec{x}_j) = \sqrt{\sum_{r=1}^n (\vec{x}_i[r] - \vec{x}_j[r])^2} \quad (1)$$

No obstante, esta medida no se puede aplicar sobre el atributo $A3$, dado que es nominal. Para dicho atributo, se puede aplicar una distancia que devuelva 0 si los dos atributos son iguales, y 1 si son distintos.

Por último, hay que tener en cuenta que el rango de los distintos atributos es muy diferente. En este caso, convendría utilizar una medida de distancia ponderada.

$$d(\vec{x}_i, \vec{x}_j, \vec{w}) = \sqrt{\sum_{r=1}^n \vec{w}[r] (\vec{x}_i[r] - \vec{x}_j[r])^2} \quad (2)$$

Un posible vector de pesos podría ser $\vec{w} = (1, 1, 1/1000, 1/1000)$. Otra opción más cómoda en este caso sería normalizar los atributos para que todos sus valores estén en el rango $[0, 1]$. Para calcular el valor del atributo $A3$ utilizando la distancia euclídea, y normalizando los atributos $A1$ y $A3$ al rango $[0, 1]$, se podría utilizar aprendizaje basado en instancias, por ejemplo, IBK (con $K = 1$). En ese caso, los ejemplos y las distancias quedarían como define la tabla .

Instancia	A1	A2	A4	clase	d(I,J)
1	0,1	0,87	0,13	azul	0,59
2	0,12	0,14	0,70	azul	0,13
3	0,8	0,3	0,12	verde	0,91
4	0,6	0,9	0,41	verde	1,06
5	0,23	0,453	0,938	azul	0,51
6	0,97	0,62	0,91	verde	1,08

Cuadro 5: Datos normalizados y distancia a la instancia con los siguientes valores para sus atributos: $A1 = 0, 1$, $A2 = 56$, $A4 = 71$.

La instancia más cercana en este caso sería la instancia 2, por lo que le asignamos el mismo valor para el atributo $A3$, es decir, *azul*.

Solución Pregunta 3/3

El problema consiste en minimizar el tiempo o número de acciones ejecutadas para interceptar el balón. Para ello, se debe generar una política de acción que para cada estado o situación del portero respecto al balón, defina cuál es la mejor que el portero puede ejecutar. Por tanto, se trata de un problema de Aprendizaje por Refuerzo.

En el diseño de un problema de Aprendizaje por Refuerzo hay que definir:

- Espacio de estados, S : compuesto por 4 características, que pueden considerarse continuas: $dist$, dir , $\Delta dist$, y Δdir .
- Espacio de acciones, A : compuesto por la acción de avanzar(f) y girar(a). Por tanto, puede considerarse como un conjunto finito de acciones, pero con un alto número de ellas: $360+130=490$ acciones.
- Función de transición de estados: desconocida
- Función de refuerzo: se recibe un refuerzo positivo cuando se ejecuta una acción que coloca al portero a menos de 1 metro del balón. Se recibe un refuerzo nulo en cualquier otro caso

Se trata de un problema de aprendizaje por refuerzo, con un refuerzo retardado (sólo se puede recibir un refuerzo positivo al final de cada intento de interceptar el balón). La función de transición de estados es desconocida, por lo que no se pueden utilizar métodos de programación dinámica directamente. Por tanto, se requiere un método basado en el modelo, o libre de modelo, como el algoritmo Q-Learning. Sin embargo, el espacio de estados y de acciones es demasiado grande, por lo que es necesario utilizar algún método de generalización:

- Espacio de estados: seguir algún método de discretización para utilizar una representación tabular de Q, o un método de aproximación de funciones para aproximar la función Q.
- Espacio de acciones: discretizar a un conjunto de acciones reducido. Por ejemplo, se puede limitar el avance a un avance medio ($avanzar(50)$) y un avance máximo ($avanzar(100)$) pero no tiene sentido avances negativos. Para el giro, se puede también discretizar en, por ejemplo, 5 acciones: ($girar(-60)$, $girar(-30)$, $girar(0)$, $girar(30)$, $girar(60)$).

Dado este diseño, habría que comenzar el aprendizaje de la función Q. Para ello, se ejecutaría un número alto de episodios de aprendizaje, consistentes en que el delantero lanza el balón, y ejecuta una secuencia de acciones para intentar interceptarlo. Siguiendo ese proceso, se generan tuplas de experiencia que permiten actualizar Q siguiendo el algoritmo Q-Learning para dominios estocásticos. El proceso se repite hasta que se converja a una política. Para ese proceso se debe elegir una estrategia de exploración adecuada. Posteriormente, se debe testear la política obtenida siguiendo un proceso equivalente, en el que se sigue la política de forma totalmente avariciosa para obtener las acciones. Repitiendo un conjunto de test suficientemente grande, se puede evaluar la calidad de la política obtenida, teniendo en cuenta el número de veces que se ha logrado alcanzar el balón.

El principal inconveniente de la aplicación del algoritmo Q-Learning a este problema es cómo representar la función Q. Una aproximación tabular no es posible, puesto que el espacio de estados y de acciones son continuos/muy grandes. Las dos posibles soluciones pasan por una discretización de los espacios, o por utilizar un aproximador de funciones.

En caso de discretización:

- Espacio de Estados: se puede optar por una discretización manual uniforme, o por utilizar un método automático para realizar la discretización.
- Espacio de Acciones: una discretización manual, por ejemplo, en 5 acciones: $avanzar(100)$, $girar(-60)$, $girar(-30)$, $girar(30)$, $girar(60)$.