



DEPARTAMENTO DE INFORMÁTICA
UNIVERSIDAD CARLOS III DE MADRID

Grado en Ingeniería en Informática

Inteligencia Artificial
Examen parcial
Abril 2016

Normas generales del examen

- El tiempo para realizar el examen es de **2 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

Ejercicio 1 (3p)

Se pretende construir un detector de *spam* que considera únicamente tres variables: el momento del día en que se recibió el email, $M \in \{\text{mañana, tarde, noche}\}$; si contiene la palabra *viagra*, $W \in \{\text{si, no}\}$; y si la persona que envió el correo está en la lista de contactos, $K \in \{\text{si, no}\}$. Se pide:

1. (1p) Diseña una red bayesiana para resolver el problema planteado.
2. (1p) Imagina que se cuenta con el siguiente histórico de datos:

spam	H	W	K
si	noche	si	no
no	mañana	no	si
no	tarde	no	si

Utiliza el histórico de datos para definir las probabilidades de la red bayesiana diseñada.

3. (1p) En un momento dado se observa que nuevo email recibido por la noche que no contiene la palabra *viagra*. La lista de contactos no se puede consultar porque el servidor que la contiene está caído en este momento. Utilizando la red bayesiana planteada, determina si ese email se clasificaría como *spam* o no.

Ejercicio 2 (4p)

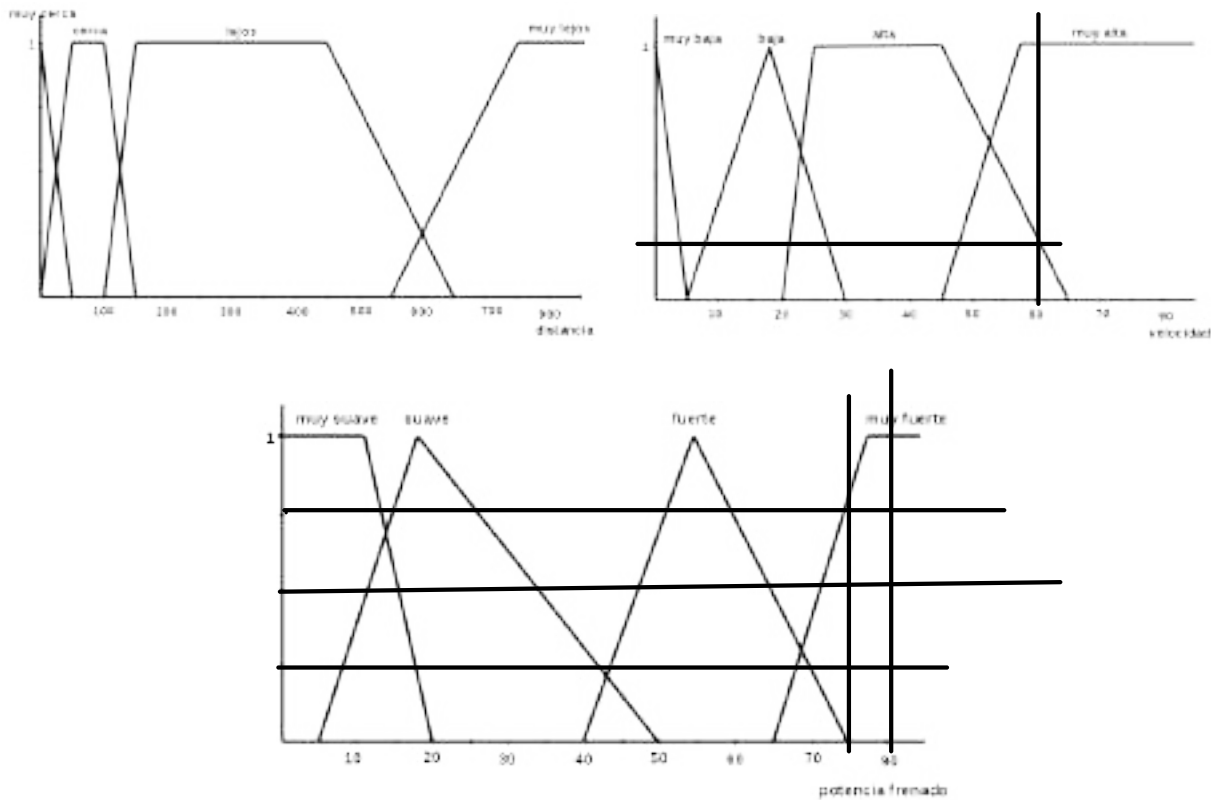
Un taxista que trabaja en una ciudad se puede encontrar en dos situaciones: con o sin pasajero. Cuando está sin pasajero, tiene dos opciones: (a) dar vueltas por las calles de la ciudad, o (b) ir a la parada de taxi. La experiencia previa le dice que cuando recorre las calles de la ciudad, la probabilidad de encontrar pasajero es 0,7, mientras que cuando va a la parada, la probabilidad de encontrar pasajero es 0,4. Dar vueltas por las calles tiene un coste de 20, mientras que ir a la parada tiene un coste de 15. Cuando tiene un pasajero debe realizar el servicio. Consideraremos que realizar servicios es positivo, por lo que está acción tiene un coste negativo de -20.

Se pide:

1. (1p) Define formalmente el problema descrito como un MDP (proceso de decisión de Markov).
2. (1p) Expresa las ecuaciones de Bellman para cada uno de los estados de este problema.
3. (1p) Ejecuta dos iteraciones del algoritmo *Iteración de valor* (value iteration).
4. (1p) ¿Cuál es la política después de las dos iteraciones ejecutadas en el apartado anterior? Explica por qué. ¿Crees que la política del apartado anterior es óptima? Razona la respuesta.

Ejercicio 3 (3p)

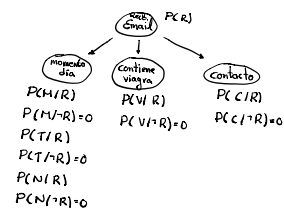
Se ha diseñado un controlador borroso para definir un sistema de frenado automático en un tren. Para ello se considera la distancia del tren a la próxima estación y la velocidad del mismo. La potencia con la que se debe realizar el frenado depende de estas dos variables. La velocidad del tren puede ser muy baja, baja, alta o muy alta. La distancia a la siguiente estación puede ser muy cerca, cerca, lejos o muy lejos. La potencia de frenado puede ser muy suave, suave, fuerte o muy fuerte.



Las reglas del controlador borroso vienen dadas por la siguiente tabla:

Distancia	Velocidad			
	muy baja	baja	alta	muy alta
muy cerca	suave	fuerte	muy fuerte	suave
cerca	suave	suave	fuerte	muy fuerte
lejos	suave	muy suave	suave	fuerte
muy lejos	muy suave	muy suave	suave	suave

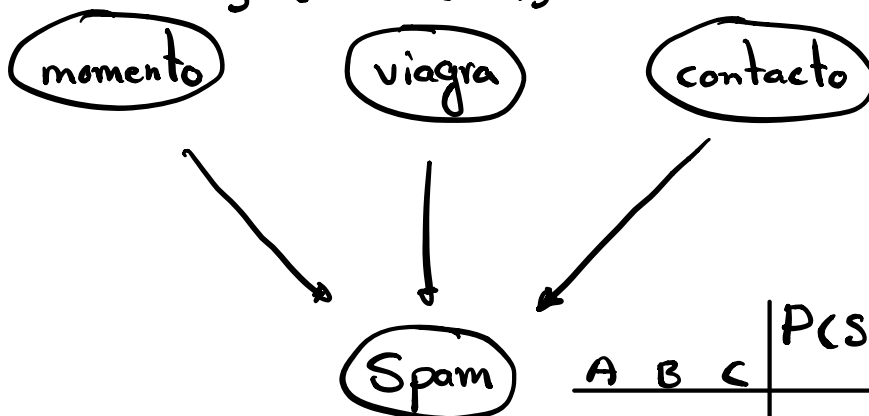
Actualmente el tren está a una distancia de 50 m. de la próxima estación con una velocidad de 60 km/h. Se pide plantear como ejecutaría el sistema de reglas borrosas, aplicando inferencia Mamdani, para determinar la potencia de frenado.



1.) $P(M) = \frac{1}{3}$ cada uno $P(V) = \frac{1}{3}$ $P(C) = \frac{2}{3}$

1.

2.



			$P(S/A, B, C)$
A	B	C	
M	V	C	0
M	V	$\neg C$	0
M	$\neg V$	C	0
M	$\neg V$	$\neg C$	0
T	V	C	0
T	V	$\neg C$	0
T	$\neg V$	C	0
T	$\neg V$	$\neg C$	0
N	V	C	0
N	V	$\neg C$	1
N	$\neg V$	C	0
N	$\neg V$	$\neg C$	0

$$3. P(S/N, \neg V, C) =$$

$$= \alpha P(S, N, \neg V, C) =$$

$$= \alpha \sum_c P(S/N, \neg V, C) P(N) P(\neg V) P(C) =$$

$$= \alpha P(N) P(\neg V) \sum_c P(S/N, \neg V, C) P(N) P(C) =$$

= ...

No se consideraria spam ya que
esté o no en la lista de contactos no es spam.

2.)

1. Estados { Con pasajeros CP, Sin pasajeros SP }

Acciones { dar vuelta V, parada P, servicio S }

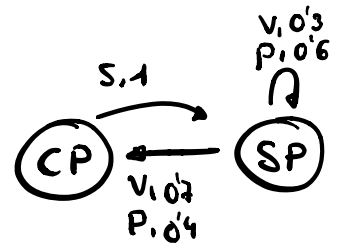
Costes { $C(V) = 20$, $C(P) = 15$, $C(S) = -20$ }

Probabilidades:

$$P_V(CP/SP) = 0.7 \quad P_V(SP/SP) = 0.3$$

$$P_P(CP/SP) = 0.4 \quad P_P(SP/SP) = 0.6$$

$$P_S(SP/CP) = 1$$



$$2. \quad V_{t+1}(SP) = \min_a [20 + 0.3 V_t(SP) + 0.7 V_t(CP), \\ 15 + 0.6 V_t(SP) + 0.4 V_t(CP)]$$

$$V_{t+1}(CP) = -20 + 1 \cdot V_t(SP)$$

$$3. \quad V_0(SP) = 0 \quad V_1(CP) = 0$$

$$V_1(SP) = \min[20, 15] = 15 \quad V_1(CP) = -20$$

$$V_2(SP) = \min[10.5, 16] = 10.5 \quad V_2(CP) = -5$$

4. La política que obtenemos es:

Para SP es buscar por la ciudad dando vueltas por las calles.

Para CP es hacer el servicio

Son los que menos costes generan.

La política es la óptima^{y que} para CP, es el único, y para SP

la probabilidad de encontrar a alguien dando vueltas es bastante mas alta y la diferencia de costes no lo supera.

3.)

Distancia	Velocidad			
	muy baja	baja	alta	muy alta
muy cerca	0	0	muy fuerte	1
cerca	1	0	fuerte	muy fuerte
lejos	0	muy fuerte	0	fuerte
muy lejos	muy fuerte	muy fuerte	0	fuerte

Entrada: Distancia 50 m Velocidad 60 km/h

Borrosificar: Pasar los valores al grado en el que son ciertos.

Distancia:

Velocidad:

muy cerca = 0

muy baja = 0

cerca = 1

baja = 0

lejos = 0

alta = 0.25

muy lejos = 0

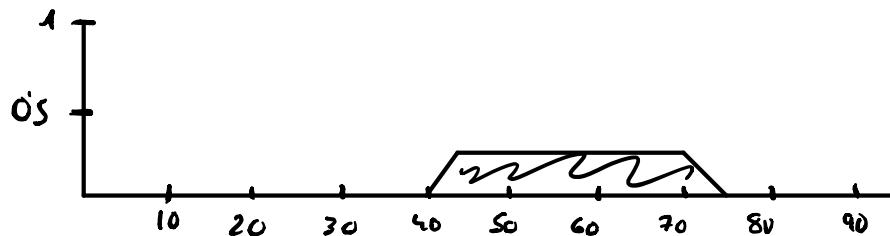
muy alta = 1

Analizar las reglas: Hallamos la similitud y la salida correspond.

Se han eliminado las que dan 0

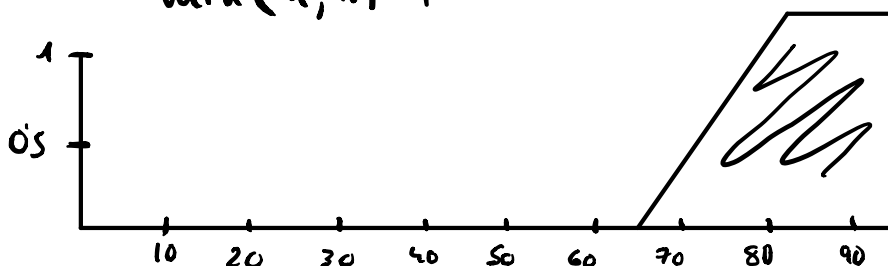
$D(c)$ y $V(a) \Rightarrow$ fuerte

$\min(1, 0.25) = 0.25$

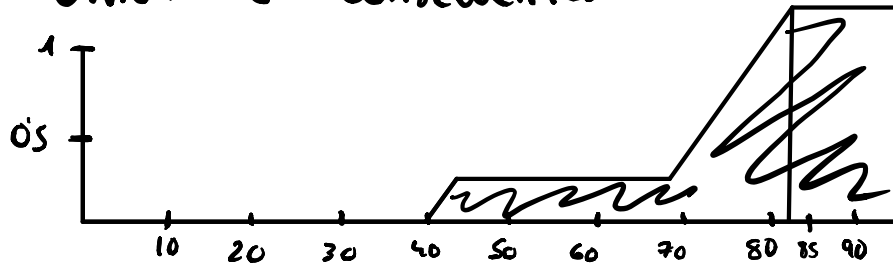


$D(c)$ y $V(ma) \Rightarrow$ muy fuerte

$\min(1, 1) = 1$



Unión de consecuentes:



Desborrosificar: Transformar la función en valor numérico, mediante por ejemplo el mínimo del máximo.

Salida: 82'5