

# CHULETA-EXAMEN-IA.pdf



Anónimo



Inteligencia Artificial



3º Grado en Ingeniería Informática



Escuela de Ingeniería y Arquitectura  
Universidad de Zaragoza



**Todas tenemos una amiga  
experta en recorrer  
kilómetros en discotecas.  
Y si no la tienes eres tú.**

Para ser una experta en kilómetros  
de verdad, déjate guiar por **coches.net**



Compra  
o vende  
tu coche

✓ fácil  
✓ rápido

**coches.net**



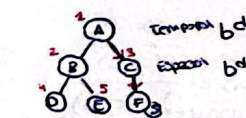
Tu colega "el experto en Erasmus": se fue pensando en aprobarlo todo.  
No aprobó nada. Lo probó todo.

Cuando necesites un auténtico experto, déjate guiar por coches.net

## RESUMEN IA

### BÚSQUEDAS NO INFORMADAS

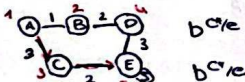
#### PRIMERO EN ANCHURA



- Test cuando es necesario
- Óptimo en pesos
- frontera con FIFO

Complejidad  $b^d$  b: sucesores d: profundidad

#### BÚSQUEDA COSTE UNIFORME

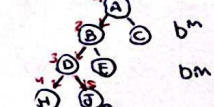


- Óptimo en coste
- Test cuando el expandido

#### BÚSQUEDA

grafo = frontera, error  
árbol = frontera

#### PRIMERO EN PROFUNDIDAD



- frontera con LIFO
- en caso de árbol no es completo

### BÚSQUEDA BIDIRECCIONAL



BÚSQUEDA EN PROFUNDIDAD  
BÚSQUEDA LIMITADA  
BÚSQUEDA EN PROFUNDIDAD  
BÚSQUEDA LIMITADA

### BÚSQUEDAS INFORMADAS

#### ALGORITMO USORAE

$f(n) = h(n) + g(n)$   
- No es completo si se busca en árbol

$f(n)$  función de valoración de la frontera

$h(n)$  heurística  $g(n)$  coste del nodo n

#### BÚSQUEDA A\*

$f(n) = h(n) + g(n)$

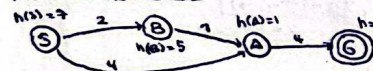
- Situación óptima si  $h(n)$

$h(n) = 0$  objetivo

$h_{\text{real}}(n) = \text{distancia real}$

es admisible en árbol  $\rightarrow \forall \text{ nodo } n \quad h(n) \leq C_{\text{obj}}(n)$   
es consistente en grafo  $\rightarrow \forall \text{ nodo } n \quad h(n) \leq c(n, a, n') + h(n')$

#### EJEMPLO HEURÍSTICAS



¿es Admisible? si  
 $h(S) = 3 \leq 2 + 1 + 4 \checkmark$   
 $h(B) = 5 \leq 4 + 1 \checkmark$   
 $h(A) = 1 \leq 4 \checkmark$

¿es consistente? NO  
 $h(S) = 3 \leq 2 + h(B) = 2 + 5 \checkmark$   
 $h(S) = 3 \leq 4 + h(A) = 4 + 1 \checkmark$

### BÚSQUEDA LOCAL

camino no relevante, parten de un estado completo

#### BÚSQUEDA ESCALADA

- Encuentra un mínimo
- Sigue buscando mínimos que haya vecinos con mejor valoración



#### ENTRAMAMIENTO SIMULADO

- Simula por permitiendo hacer movimientos erróneos de forma aleatoria o controlada

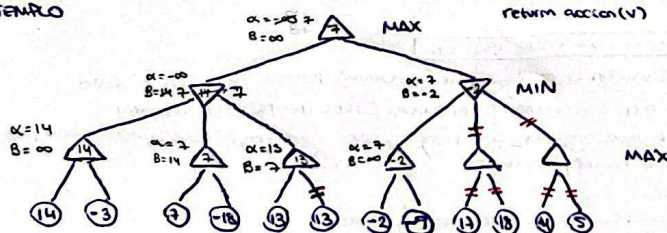
#### ALGORITMO GENÉTICO

- Simula a mutaciones aleatorias para poder encontrar una solución

### JUEGOS

#### ALGORITMO MINIMAX Y POEA ALFA-BETA

#### EJEMPLO



#### función evaluate-search

$V = \text{Max-Value}(\text{estado}, -\infty, \infty)$   
return acción(V)

función  $\text{Max-Value}(\text{estado}, \alpha, \beta)$  return valor  
if best-terminal (estado) return valor(estado)  
 $V = -\infty$

for each a in Acciones(estado) do  
 $V = \text{Max}(V, \text{Min-Value}(\text{resultado}(a), \alpha, \beta))$   
if  $V \geq \beta$  then return  $\beta$   
return  $V$

función  $\text{Min-Value}(\text{estado}, \alpha, \beta)$  return valor  
if best-terminal (estado) return valor(estado)  
 $V = \infty$   
for each a in Acciones(estado) do  
 $V = \text{Min}(V, \text{Max-Value}(\text{resultado}(a), \alpha, \beta))$   
if  $V \leq \alpha$  then return  $\alpha$   
return  $V$

### LINGÜAJES REGLAS

#### Representación

(define template estado (slot garrapa (type INTEGER)))

#### Estado inicial y final

(estado (garrapa 0)) (estado (garrapa 3))

#### Operador

(define Agregar-un-litro  
?estado (- (estado (garrapa ?cantidad))  
=>  
(modify ?estado (garrapa (+ ?cantidad 1))))  
assert -> estado nuevo declaración  
retract

#### Restricciones

(persona (nombre ?nombre) (edad ?edad) (> ?edad 18))

(test (> ?edad 18)) (edad = (+ 18 2))

(bird ?suna (+ ?a ?b))

#### Métricas para clasificadores

$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}$

$\text{Precision} = \frac{TP}{TP + FP}$

$\text{Recall} = \frac{TP}{TP + FN}$

$\text{F1-Score} = \frac{2 \cdot \text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Close Real

	A	B
A	TP	FP
B	FN	TN

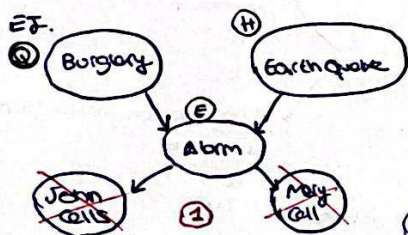




# Razonamiento Probabilístico sobre Redes Bayesianas

1. IDENTIFICAR: nodos las variables que no sean ancestros de A o E
2. INSTANCIAR: tablas de prob de las evidencias
3. ELIMINAR: variables ocultas H, juntar todos los factores y eliminar H (sumando)
4. JUNTAR: todas las factores restantes y NORMALIZAR

Nodos: Q Query  
E Evidencia  
H Resto



B	P(B)
0	0.001
E	P(E)
0	0.002

A	B	E	P(A B,E)
+	+	+	0.95
+	+	-	0.94
+	-	+	0.29
+	-	-	0.001

$$P(A,B) = P(A|B) \cdot P(B) = 0.94 \cdot 0.001 = 0.00094$$

$$P(A,-B) = 0.0611 \cdot 0.999 = 0.0610$$

NORMALIZAR

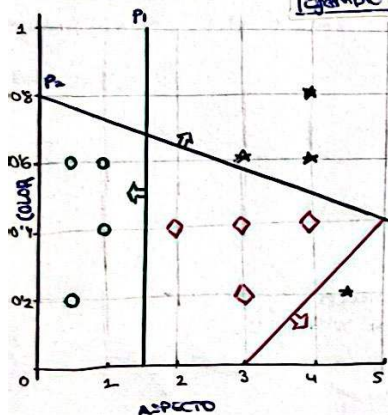
$$P(B|A) = \frac{0.0009}{0.0009 + 0.0610} = 0.0145$$

$$P(B|+A) = 0.9855$$

A	B	E	P(E)	P(A B,E)	P(A,E B)	P(A B)
+	+	+	0.002	x 0.95	0.0019	0.014
+	+	-	0.998	x 0.94	0.9331	>
+	-	+	0.002	x 0.29	0.0012	>
+	-	-	0.998	x 0.001	0.0009	>

## REDES NEURONALES

### Ejemplo 1



### Neurona 1

$$W_1 = (-1, 0)$$

$$b_1 = -W_{11} \cdot P_{11} - W_{12} \cdot P_{12}$$

$$b_1 = (-1) \cdot 1.5 - 0 \cdot 0 = -1.5$$

### Neurona 2

$$W_2 = P_2 - P_3 = (0, 0.5) - (3, 0.9)$$

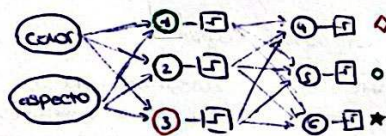
$$W_2 = (-3, -0.4)$$

$$b_2 = -4$$

### Neurona 3

$$W_3 = (0.2, 1)$$

$$b_3 = -0.6$$



$$\diamond = (0, 0, 0)$$

$$\circ = (1, 0, 0)$$

$$\star = (0, 1, 0), (0, 0, 1)$$



### Neurona 4

$$W_4 = (-1, -1, -1) \quad b_4 = 10.5$$

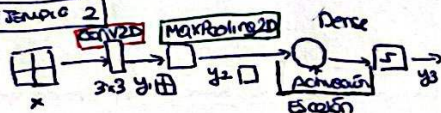
### Neurona 5

$$W_5 = (1, 0, 0) \quad b_5 = -0.5$$

### Neurona 6

$$W_6 = (0, 1, 1) \quad b_6 = -0.5$$

### Ejemplo 2



$$\text{kernel} = \begin{pmatrix} 0 & -1 & -0.3 \\ -1 & 2 & 0.8 \\ -0.2 & 0.5 & 0 \end{pmatrix}$$

$$W_1 = 0.5$$

$$b_1 = -5$$

$$x = \begin{pmatrix} 3 & 5 \\ 1 & 4 \end{pmatrix} \rightarrow y_1 = \begin{pmatrix} 10.5 & 8.8 \\ 0.4 & 2 \end{pmatrix} \rightarrow y_2 = 10.7 \rightarrow y_3 = \frac{10.5 \cdot 0.5}{-0.25} = 1$$

## FUNCIONES DE ACTIVACIÓN

Función lineal  $f(x) = x$

Función sigmoid  $f(x) = \frac{1}{1 + e^{-x}}$

Función escalon

$$f(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Función Relu Rectified Linear Unit

$$f(x) = \begin{cases} x & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

Función Softmax

para cada salida i  $f(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}}$

Ejemplo para salidas

$$(-0.18, 0.76, -0.52)$$

$$e^{-0.18} = 0.83 \quad \frac{0.83}{3.55} = 0.23$$

$$e^{0.76} = 2.13 \quad \frac{2.13}{3.55} = 0.60$$

$$e^{-0.52} = 0.59 \quad \frac{0.59}{3.55} = 0.17$$

## ALGORITMO BÚSQUEDA COSTE UNIFORME

función BÚSQUEDA-COSTE-UNIFORME (problema) returns solución o fallo

nodo ← un nodo con ESTADO = problema.ESTADO-INICIAL, COSTE-CAMINO = 0  
frontera ← cola ordenada por COSTE-CAMINO, con nodo como único elemento  
explorados ← un conjunto vacío

local do

IF frontera.VACIO() then return fallo

nodo ← Pop(frontera)

IF problema.TEST-objetivo (hijo.ESTADO) then return solución (hijo)

add.nodo a explorados

for each accion un problema.ACCIONES (nodo.ESTADO) do

hijo ← HIJO-NODO (problema, nodo, accion)

IF hijo no esté en explorados, frontera then

frontera.INSERT (nodo)

else if hijo está en frontera con mayor COSTE-CAMINO then

reemplazar el nodo en la frontera por el hijo