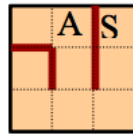




Grado en Ingeniería Informática en Sistemas de Información  
Inteligencia Artificial - Curso 2016/17  
EXAMEN NOVIEMBRE-22/11/2016

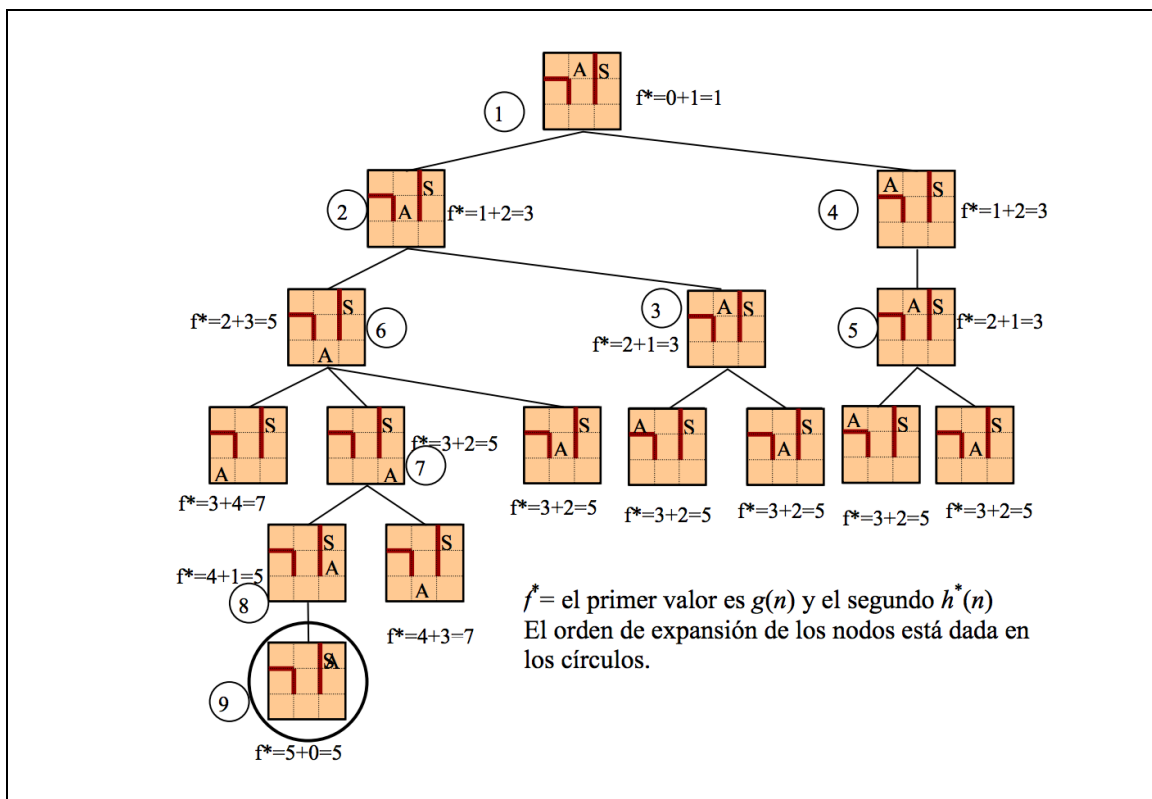
APELLIDOS: \_\_\_\_\_ NOMBRE: \_\_\_\_\_ D.N.I.: \_\_\_\_\_

**EJ1.- (3 puntos)** Aplicar el algoritmo A\* al laberinto de la figura, donde el robot inicialmente está en A y su meta es llegar a S. Las únicas acciones posibles son los movimientos (derecha, arriba, abajo, e izquierda) a un cuadrado adyacente. Sin embargo, cada una de estas acciones sólo es posible si en la dirección correspondiente no existe una barrera ni se sale del tablero. Cada acción tiene un coste de una unidad. Use como función heurística  $h$  la distancia desde A hasta la meta S. En el caso de la figura, el valor de  $h$  es 1.



a) Dibujar el árbol de búsqueda suponiendo que **no se evitan estados repetidos**. Si al expandir un nodo hay que elegir aleatoriamente entre varios, expanda preferiblemente primero un nodo que está más cerca de un nodo meta.

En el árbol debe indicar para cada nodo del árbol de búsqueda los valores de  $f$  desglosado como  $f = \text{coste} + h$ .

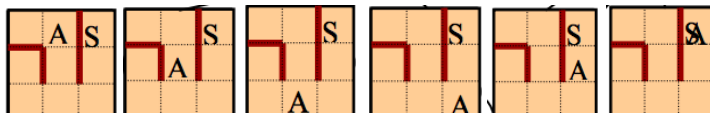


b) Indicar la solución obtenida, el coste de la solución y el número de nodos expandidos.

Coste: 5

Número de nodos expandidos: 9

Solución:



**EJ2.- (2 puntos)** a) Representa en lógica de primer orden el conocimiento siguiente:

**Los caballos son mamíferos. La descendencia de un caballo es un caballo. Cheroki es un caballo. Cheroki es el padre de Charly.**

Para ello, solo puede usar los siguientes predicados:

$\text{caballo}(X) \rightarrow X \text{ es un caballo}$ $\text{mamifero}(X) \rightarrow X \text{ es mamífero}$ $\text{desciende}(X,Y) \rightarrow X \text{ desciende de } Y$
--

$\text{caballo}(\text{Cheroki})$ $\forall x \text{ caballo}(x) \rightarrow \text{mamifero}(x)$ $\text{desciende}(x,y) \wedge \text{caballo}(y) \rightarrow \text{caballo}(x)$ $\text{desciende}(\text{Charly}, \text{Cheroki})$
--

b) Transforme el conocimiento a Forma Normal Conjuntiva y responda a la siguiente pregunta: ¿son todas las cláusulas de Horn?

$\text{caballo}(\text{Cheroki})$ $\neg \text{caballo}(x) \vee \text{mamifero}(x)$ $\neg \text{desciende}(x,y) \vee \neg \text{caballo}(y) \vee \text{caballo}(x)$ $\text{desciende}(\text{Charly}, \text{Cheroki})$
--

c) En caso afirmativo, transforme el conocimiento en un lenguaje de programación lógica, guárdelo en un archivo llamado TusApellidos.pl y use la herramienta Deduction para responder a la query: ¿Charly es un mamífero?

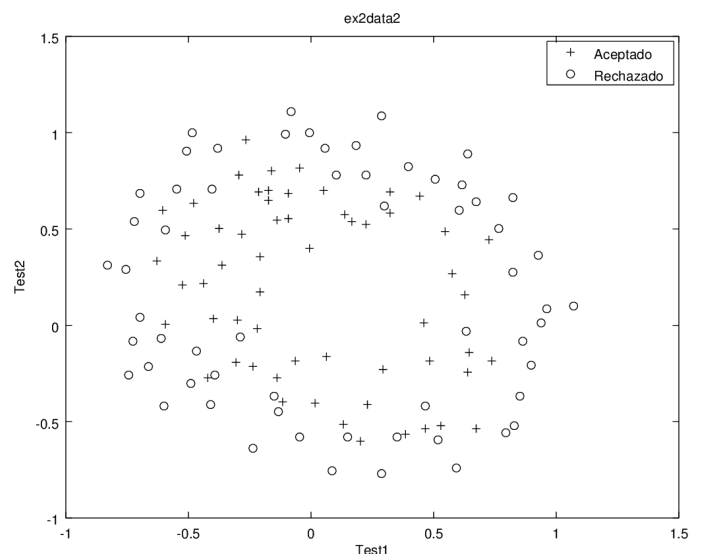
**Nota:** Tenga en cuenta que las variables deben empezar por una letra mayúscula y las constantes deben empezar por una letra minúscula.

$\text{caballo}(\text{cheroki}).$ $\text{mamifero}(X) \leftarrow \text{caballo}(X).$ $\text{caballo}(X) \leftarrow \text{desciende}(X,Y) \ \& \ \text{caballo}(Y).$ $\text{desciende}(\text{charly}, \text{cheroki}).$
---

**EJ3.- (5 puntos)** Realizar diversos estudios de regresión sobre el conjunto de datos `ex2data2.txt` de piezas de microchips, que contiene un histórico con los resultados de dos test realizados y la indicación de si fueron aceptadas o rechazadas.

Descripción de los ficheros:

- `M1201611`: programa principal a utilizar.
- `computeCost(X, y, theta)`: calcula el coste en la regresión lineal.
- `gradientDescent(X, y, theta, alpha, num_iters)`: obtiene el modelo de una regresión lineal.



**Figura 1. Resultado de la aceptación de microchips**

- a) Representar los datos gráficamente como aparece en la figura 1 anterior implementando la función `dibujarClases(data)` llamada desde el programa principal.

```
function dibujarClases (data)

    X = data(:,1:2);
    y = data(:, 3);

    pos = find(y == 1);
    neg = find(y == 0);

    plot(X(pos, 1), X(pos, 2), 'k+');
    hold on;

    plot(X(neg, 1), X(neg, 2), 'ko');
    ylabel('Test2')
    legend('Aceptado', 'Rechazado')
    title('ex2data2')

end
```

- b) Se quiere saber cómo de bien determina un atributo al otro, es decir, cómo los valores del atributo test1 valdrían para predecir los valores del atributo 2, y viceversa. Implementar la función `regresionEntreAmbos(data, iteraciones, alpha)` de manera que muestre los valores de theta y el coste para cada uno de los dos modelos obtenidos. Utilizar los valores de iteraciones y alpha indicados en el programa principal. Mostrar en pantalla el resultado, como por ejemplo:

```
Theta12: 0.190776 -0.141335 --> Coste: 0.131347
Theta21: 0.078349 -0.129704 --> Coste: 0.119936
```

- c) Mostrar gráficamente las dos rectas de regresión como aparecen en la figura 2, y predecir los respectivos valores para el primer ejemplo del conjunto de datos. Mostrar en pantalla el resultado, como por ejemplo:

```
Para un valor 0.051267 de la prueba 1 la predicción de la prueba 2 es 0.183530,
siendo el valor real 0.699560, y la diferencia es 0.516030
Para un valor 0.699560 de la prueba 2 la predicción de la prueba 1 es -0.012387,
siendo el valor real 0.051267, y la diferencia es 0.063654
```

- d) Mostrar la curva de convergencia como la que aparece en la figura 3 para el mejor modelo obtenido en el apartado anterior.

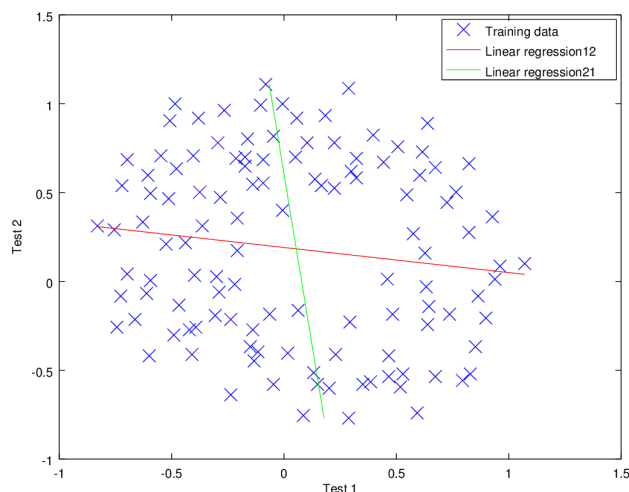


Figura 3. Rectas de regresión lineal entre atributos

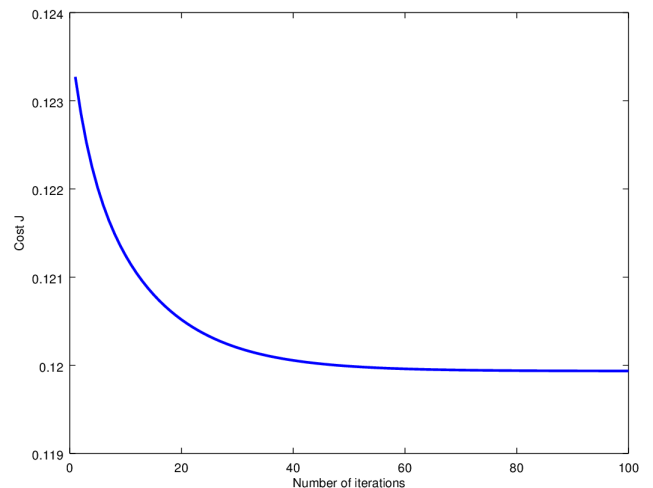


Figura 2. Curva de convergencia

Apartados b), c) y d)

```
function regresionEntreAmbos (data, iterations, alpha)

%% b) Regresion entre atributos =====

fprintf('\n\nApartado b): calculando modelos de
regresion...\n')

m = size(data,1);
theta = zeros(2, 1); % initialize fitting parameters

X1 = [ones(m, 1), data(:,1)]; % Add a column of ones to x
y1 = data(:, 2); % La clase es el segundo atributo

[theta12 J_history12]= gradientDescent(X1, y1, theta, alpha,
iterations);
coste12 = computeCost(X1, y1, theta12);

X2 = [ones(m, 1), data(:,2)]; % Add a column of ones to x
y2 = data(:, 1); % La clase es el segundo atributo

[theta21 J_history21]= gradientDescent(X2, y2, theta, alpha,
iterations);
coste21 = computeCost(X2, y2, theta21);

% print theta12 to screen
fprintf('Theta12: %f %f --> Coste: %f \n', theta12(1),
theta12(2), coste12);

% print theta21 to screen
fprintf('Theta21: %f %f --> Coste: %f \n', theta21(1),
theta21(2), coste21);

% c) Rectas de regresion y prediccion =====

fprintf('\n\nApartado c): mostrando rectas de regresión y
prediccion...\n')

plot(data(:,1),data(:,2),'x', 'MarkerSize', 10);
xlabel('Test 1');
ylabel('Test 2');
% Plot the linear fit
hold on; % keep previous plot visible
plot(X1(:,2), X1*theta12, 'r-')

plot(X2*theta21,X2(:,2), 'g-');

legend('Training data', 'Linear regression12', 'Linear
regression21')
%print('rectasRegresion.png','-dpng')
```

```

hold off % don't overlay any more plots on this figure

predict1 = [1, data(1,1)] *theta12;
fprintf('Para un valor %f de la prueba 1 la predicción de la
prueba 2 es %f, siendo el valor real %f, y la diferencia es
%f\n',...
    data(1,1),predict1,data(1,2),(data(1,2)-predict1));
predict2 = [1, data(1,2)] *theta21;
fprintf('Para un valor %f de la prueba 2 la predicción de la
prueba 1 es %f, siendo el valor real %f, y la diferencia es
%f\n',...
    data(1,2),predict2,data(1,1),(data(1,1)-predict2));

% d) curva de convergencia para el de menor coste

fprintf('\n\nApartado d): mostrando la curva de convergencia
para el de menor coste');

figure;
if (coste12 <= coste21)
    fprintf('Regresión 12');
    plot(1:numel(J_history12), J_history12, '-b', 'LineWidth',
2);
else
    fprintf('Regresión 21');
    plot(1:numel(J_history21), J_history21, '-b', 'LineWidth',
2);
end
xlabel('Number of iterations');
ylabel('Cost J');
print('convergencia.png','-dpng')

end

```