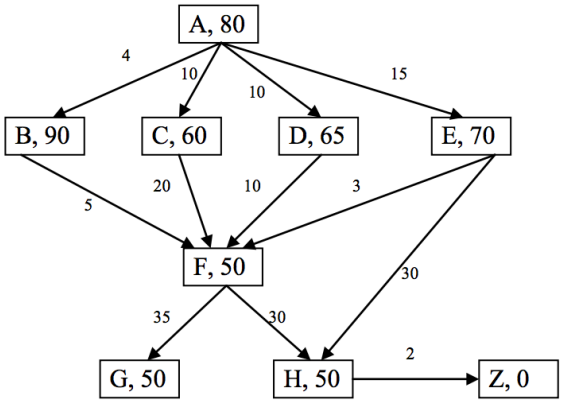
Grado en Ingeniería Informática en Sistemas de Información
Inteligencia Artificial - Curso 2015/16
EXAMEN JUNIO-17/06/2016

APELLIDOS:_____ **NOMBRE:**_____ **D.N.I.:**_____

EJ1.- **(3 puntos)** Usar la herramienta *Search* para aplicar el algoritmo A* al árbol de la figura, donde el nodo inicial es A y el nodo meta es Z. En los arcos se muestra el coste y en cada nodo se muestra el valor de la función heurística ***h*** para dicho nodo.



Rellena las siguientes tablas con los resultados obtenidos:

Iteraciones	Contenido de la frontera	Función heurística h	h + coste
Iter=0	A	80	80
iter=1	CDEB	60-65-70-90	70-75-85-94
iter=2	DFEB	65-50-70-90	75-80-85-94
iter=3	FFEB	50-50-70-90	70-80-85-94
iter=4	FEBHG	50-70-90-50-50	80-85-94-100-105
iter=5	EBHGHG	70-90-50-50-50-50	85-94-100-105-110-115
iter=6	FBHHGHG	50-90-50-50-50-50-50	68-94-95-100-105-110-115
iter=7	BHHHGGHG	90-50-50-50-50-50-50-50	94-95-98-100-103-105-110-115
iter=8	FHHHGGHG	50-50-50-50-50-50-50-50	59-95-98-100-103-105-110-115
iter=9	HGHHHGGHG	50-50-50-50-50-50-50-50-50	89-94-95-98-100-103-105-110-115
iter=10	ZGHHHGGHG	0-50-50-50-50-50-50-50-50	41-94-95-98-100-103-105-110-115

CAMINO ENCONTRADO	A-B-F-H-Z
COSTE	41
NODOS EXPANDIDOS	11

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

Paco es un científico. Todos los científicos están locos. Ningún vegetariano está loco.

Para ello, solo puede usar los siguientes predicados:

cientifico(X) -> X es científico
vegetariano(X) -> X es vegetariano
loco(X) -> X está loco

cientifico(Paco)
∀ x cientifico(x)→ loco(x)
∀ x vegetariano(x)→ ¬loco(x)

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

cientifico(Paco) ¬ cientifico(x) ∨loco (x) ¬ vegetariano(x) ∨ ¬loco (x)

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: ¿Paco es vegetariano?

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

científico(paco). loco(X)<- cientifico(X). false <- vegetariano(X) & loco(X).

EJ3.- (5 puntos) Una empresa tiene un conjunto de datos con información de sus clientes y desea explotarlos para poder hacer ofertas personalizadas. Los datos contienen 25 valores: 24 atributos que corresponden con el consumo para cada una de las horas del día y un último valor que indica el tipo de cliente (variable binaria 0/1). Implementar una regresión logística sin regularización para predecir el tipo de cliente a partir de su consumo horario.

Apartado A)

Implemente el programa principal **main_holdout.m** para que calcule un modelo de predicción y lo evalúe mediante holdout, usando una división del 70% de las instancias para el conjunto de entrenamiento y el 30% restante para el conjunto de test.

```
%% Initialization
clear ; close all; clc

%% Load Data

data = load('consumos.txt');

[fil col] = size(data);
X = data(:, 1:col-1);
y = data(:, col);
m = length(y);

%holdout
SPLIT = 0.7;
[X_train y_train X_test y_test] = holdout(X,y,m,SPLIT);


%% ============ Part 2: Compute Cost and Gradient ============
%   In this part of the exercise, you will implement the cost and
gradient
%  for logistic regression. You neeed to complete the code in
%  costFunction.m

%  Setup the data matrix appropriately, and add ones for the intercept
term
[m, n] = size(X_train);

% Add intercept term to x and X_test
X_train = [ones(m, 1) X_train];

%  Set options for fminunc
```

```
options = optimset('GradObj','on','MaxIter', 50);

% Initialize fitting parameters and run fminunc to obtain the optimal
theta
initial_theta = zeros(n + 1, 1);
[theta,  cost]  =  fmincg(@(t)(costFunction(t,  X_train,  y_train)),
initial_theta, options);

% Plot the cost
plot(cost);

% Predict the test set
[m, n] = size(X_test);
X_test = [ones(m, 1) X_test];
p = predict(theta, X_test);

%Compute the error on our test set
fprintf('Error: %f\n', mean(double(p ~= y_test)) * 100);
```

Apartado B)

Implemente el programa principal **main_cv.m** para que calcule un modelo de predicción y lo evalúe mediante cross-validation (K=5).

```
%% Initialization
clear ; close all; clc

%% Load Data

data = load('consumos.txt');

[fil col] = size(data);
X = data(:, 1:col-1);
y = data(:, col);
m = length(y);

%  Set options for fminunc
options = optimset('GradObj','on','MaxIter', 150);

%Elegimos  el  n?mero  de  particiones  para  cross-validation  e
inicializamos error
K = 10;
error = zeros(K,1);

%Hago particion para crossvalidation
indices = cv(m,K);

for i=1:K
  fprintf('Ejecutando...K=%d\n',i);
  %Conjunto de entrenamiento para cada particion
  test = (indices == i);
  train = ~test;
  X_train = X(train,:);
  y_train = y(train);


  % Add intercept term to x and X_test
  [m, n] = size(X_train);
  X_train = [ones(m, 1) X_train];

  % Init Theta and run optimization method
```

```
  initial_theta = zeros(n+1, 1);
  [theta, cost] = fmincg(@(t)(costFunction(t, X_train, y_train)),
initial_theta, options);

  %Plot the cost
  plot(cost);

  % Compute accuracy on our test set
  X_test = X(test,:);
  [m, n] = size(X_test);
  X_test = [ones(m, 1) X_test];
  y_test = y(test);

  p = predict(theta, X_test);

  % Compute accuracy on our test set
  error(i) = 100 * mean(double(p ~= y_test));
end

fprintf(['Error:\n %.2f\n'], sum(error)/K);
```