**EX-2**

1. **costfunction**

hyp = sigmoid(X\*theta);

J = (1/m)\*sum(-y.\*log(hyp)-(1-y).\*log(1-hyp));

grad = (1/m)\*(X')\*(hyp-y);

1. **costFunctionReg**

hyp = sigmoid(X\*theta);

J = (1/m)\*sum(-y.\*log(hyp)-(1-y).\*log(1-hyp)) + (lambda/(2\*m))\*(sum(theta(2:length(theta)).^2));

grad(1) = (1/m)\*(X(:,1)')\*(hyp-y);

grad(2:end) = (1/m)\*(X(:,2:end)')\*(hyp-y)+(lambda/m)\*(theta(2:end));

1. **plotData**

pos = find(y==1);

neg = find(y==0);

% Plot Examples

plot(X(pos, 1), X(pos, 2), 'k+','LineWidth', 2, 'MarkerSize', 7);

plot(X(neg, 1), X(neg, 2), 'ko', 'MarkerFaceColor', 'y','MarkerSize', 7);

1. **plotDecisionBoundary**

function plotDecisionBoundary(theta, X, y)

%PLOTDECISIONBOUNDARY Plots the data points X and y into a new figure with

%the decision boundary defined by theta

% PLOTDECISIONBOUNDARY(theta, X,y) plots the data points with + for the

% positive examples and o for the negative examples. X is assumed to be

% a either

% 1) Mx3 matrix, where the first column is an all-ones column for the

% intercept.

% 2) MxN, N>3 matrix, where the first column is all-ones

% Plot Data

plotData(X(:,2:3), y);

hold on

if size(X, 2) <= 3

% Only need 2 points to define a line, so choose two endpoints

plot\_x = [min(X(:,2))-2, max(X(:,2))+2];

% Calculate the decision boundary line

plot\_y = (-1./theta(3)).\*(theta(2).\*plot\_x + theta(1));

% Plot, and adjust axes for better viewing

plot(plot\_x, plot\_y)

% Legend, specific for the exercise

legend('Admitted', 'Not admitted', 'Decision Boundary')

axis([30, 100, 30, 100])

else

% Here is the grid range

u = linspace(-1, 1.5, 50);

v = linspace(-1, 1.5, 50);

z = zeros(length(u), length(v));

% Evaluate z = theta\*x over the grid

for i = 1:length(u)

for j = 1:length(v)

z(i,j) = mapFeature(u(i), v(j))\*theta;

end

end

z = z'; % important to transpose z before calling contour

% Plot z = 0

% Notice you need to specify the range [0, 0]

contour(u, v, z, [0, 0], 'LineWidth', 2)

end

hold off

end

1. **predict**

hyp = sigmoid(X\*theta);

p=(hyp>=0.5)

1. **sigmoid**

g = 1./(1+exp(-z))