Laboratory Exercise 3: Logistic Regression and Newton's Method

Goal

- To be able to implement logistic regression on a classification problem using Newton's Method.
- To build a binary classification model that estimates college admission chances based on a student's scores on two exams.

Data

- Suppose that a high school has a dataset representing 40 students who were admitted to college and 40 students who were not admitted.
- Each (x^{(i)}, y^{(i)}) training example contains a student's score on two standardized exams and a label of whether the student was admitted.

Data

In your training data,

a. The first column of your x array represents all Test 1 scores, and the second column represents all Test 2 scores.

b. The y vector uses '1' to label a student who was admitted and '0' to label a student who was not admitted.

Procedure 3.1

- Plot the data.
- Load the data for the training examples into your program and add the x_0 = 1 intercept term into your x matrix.
- Before beginning Newton's Method, we will first plot the data using different symbols to represent the two classes. In Matlab/Octave, you can separate the positive class and the negative class using the find command.

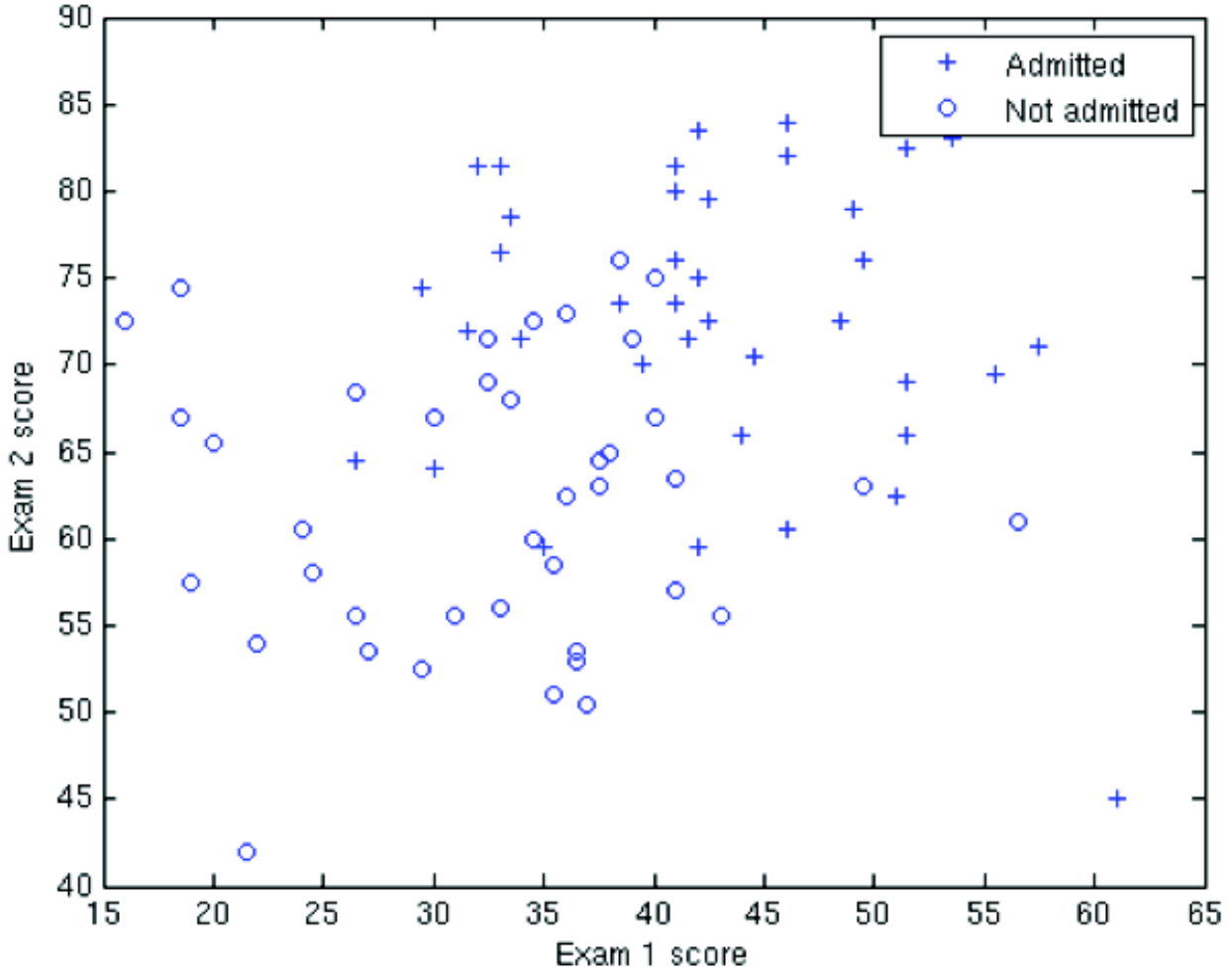
Octave

```
% find returns the indices of the
% rows meeting the specified condition
pos = find(y == 1); neg = find(y == 0);

% Assume the features are in the 2nd and 3rd
% columns of x
plot(x(pos, 2), x(pos,3), '+'); hold on
plot(x(neg, 2), x(neg, 3), 'o')
```

Example

- Your plot should look like the following:



Newton's Method

- In logistic regression, the hypothesis function is

$$h_\theta(x) = g(\theta^T x) = \frac{1}{1 + e^{-\theta^T x}}$$

$$= P(y = 1 | x; \theta)$$

Inline Function

- Matlab/Octave does not have a library function for the sigmoid, so you will have to define it yourself.
- The easiest way to do this is through an inline expression:

```
g = inline('1.0 ./ (1.0 + exp(-z))');
```

Procedure 3.2

- Plot the Cost function J(θ).
- The cost function J(θ) is defined as

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right]$$

Newton's method

- Our goal is to use Newton's method to minimize the cost function.
- The update rule for Newton's method is

$$\theta^{(t+1)} = \theta^{(t)} - H^{-1}\nabla_\theta J$$

Hint: `theta = theta - H \ gradient`

- In logistic regression, the gradient is

$$\nabla_\theta J = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x^{(i)}$$

Hessian

- The Hessian matrix of f is a nxn-matrix-valued function with domain a subset of the domain of f, defined as follows:

  “the Hessian matrix at any point in the domain is the Jacobian matrix of the gradient vector of at the point”.

$$H = \frac{1}{m} \sum_{i=1}^{m} \left[h_\theta(x^{(i)}) \left(1 - h_\theta(x^{(i)})\right) x^{(i)} \left(x^{(i)}\right)^T \right]$$

Implementation Notes

- Start with the initial value of $\theta = \vec{0}$.
- To determine how many iterations to use, calculate J(θ) for each iteration and plot your results as you did in Laboratory Exercise 1.
- Note: Newton's method often converges in 5-15 iterations. If you find yourself using far more iterations, you should check for errors in your implementation.

Implementation Notes

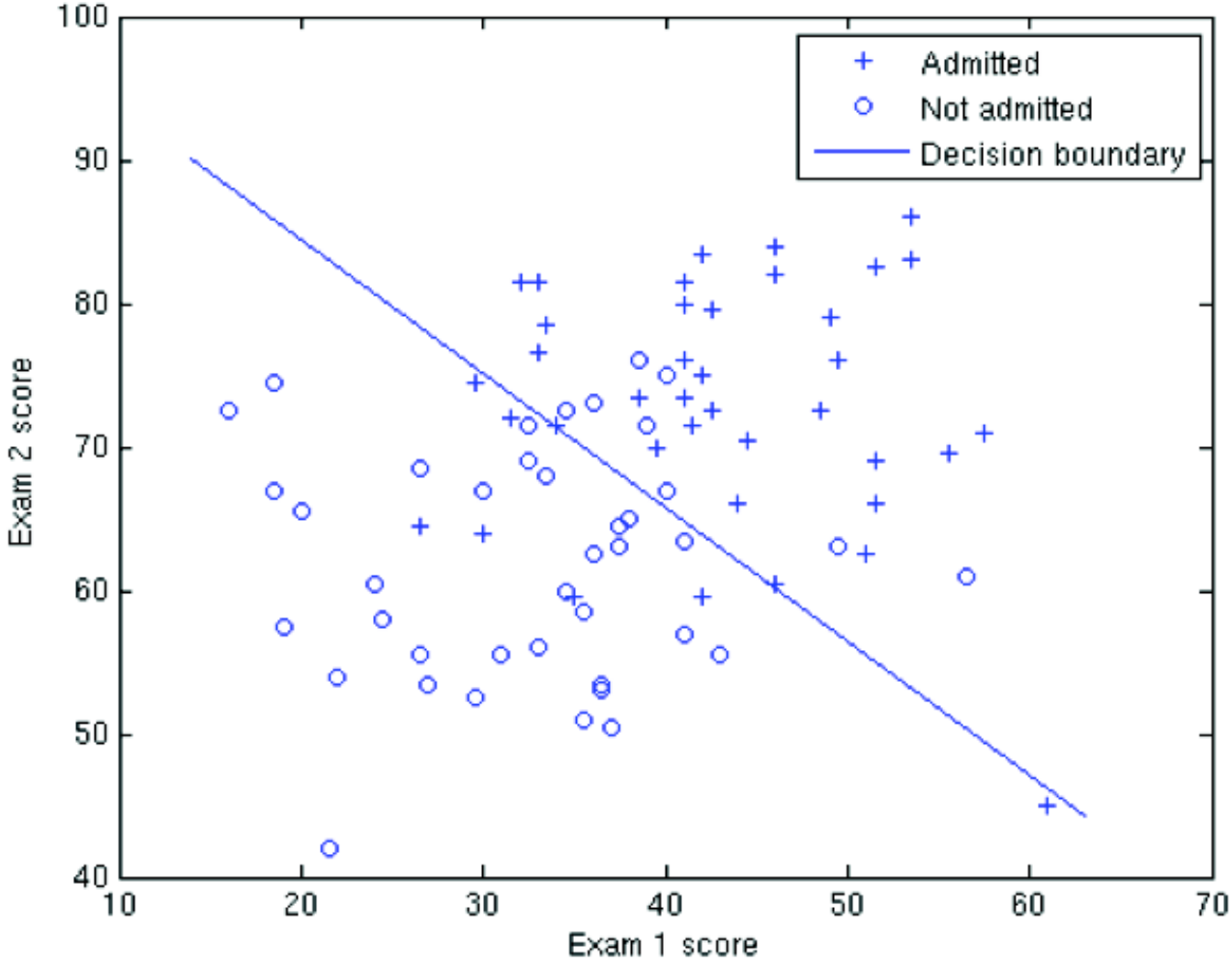
- After convergence, use your values of theta to find the decision boundary in the classification problem.
- The decision boundary is defined as the line where

$$P(y = 1|x; \theta) = g(\theta^T x) = 0.5$$

$$\theta^T x = 0$$

Implementation Notes

- Plotting the decision boundary is equivalent to plotting the $\theta^T x = 0$ line.



Procedure 3.3

Answer thr ff. questions:

1. θ = _____

2. How many iterations were required for convergence? _____

3. What is the probability that a student with a score of 20 on Exam 1 and a score of 80 on Exam 2 will not be admitted?

_____

End