Homework Assignment 3: Logistic Regression

Due Monday, February 3rd, 2020 at 11:59pm

Description

In class we learned logistic regression and how to solve for model parameters using libraries for advanced optimization. In this problem set, you will implement such approaches and evaluate it on data.

What to submit

Download and unzip the template ps3_matlab_template.zip. Rename it to ps3_LastName_FirstName, and add in your solutions:

ps3_LastName_FirstName /

- input/ - input data, images, videos or other data supplied with the problem set
- output/ - directory containing output images and other generated files
- ps3.m - your Matlab code for this problem set
- ps3_report.pdf - A PDF file that shows all your output for the problem set, including images labeled appropriately (by filename, e.g. ps0-1-a-1.png) so it is clear which section they are for and the small number of written responses necessary to answer some of the questions (as indicated). Also, for each main section, if it is not obvious how to run your code please provide brief but clear instructions (no need to include your entire code in the report).
- *.m - Any other supporting files, including Matlab function files, etc.
- ps3_LastName_FirstName_debugging.m – one m-file that has all of your codes from all the files you wrote for this assignment. It should be a concatenation of your main script and all of your functions in one file (simply copy all the codes and pate them in this file). In fact, this file in itself can be executed and you can regenerate all of your outputs using it.

Zip it as `ps3_LastName_FirstName.zip`, and submit on canvas.

Guidelines

1. Include all the required images in the report to avoid penalty.
2. Include all the textual responses, outputs and data structure values (if asked) in the report.

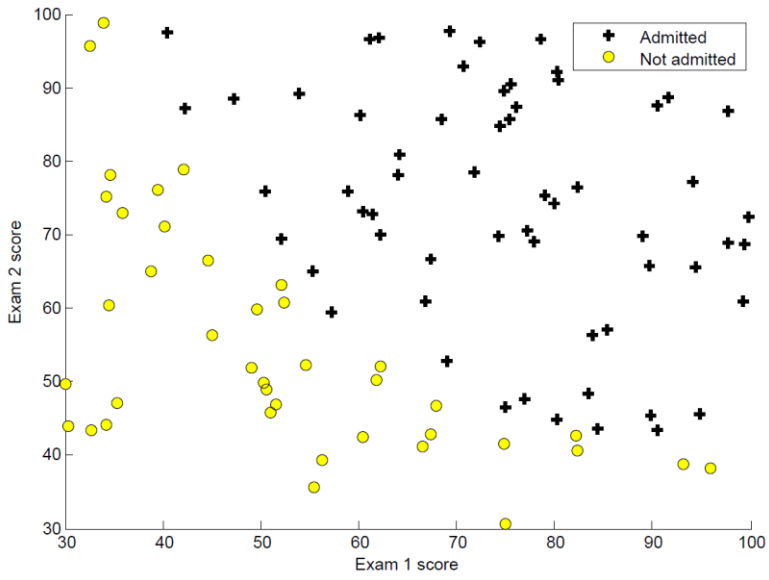
3. Make sure you submit the correct (and working) version of the code.
4. Include your name and ID on the report.
5. Comment your code appropriately.
6. Please avoid late submission. Late submission is not acceptable.
7. Plagiarism is prohibited as outlined in the Pitt Guidelines on Academic Integrity.

Questions

1- **Logistic regression**: In this part, you will build a logistic regression model to predict whether a student gets admitted into a university based on their results on two exams. You have historical data from previous applicants that you can use as a training set for logistic regression. For each training example, you have the applicant's scores on two exams and the admissions decision (0 = not admitted, and 1 = admitted). Load the training data from 'hw3_data1.txt' into MATLAB and answer/implement the following question toward building your classifier.

a. Define the feature matrix X, where each row corresponds to one feature example, and the labels vector y. Do not forget to append 1 for each feature vector, which will correspond to the bias (θ_0) that our model learns.
Text output: the size of the feature matrix X and the size of the label vector y.

b. Plot the training date to visualize the problem. Your output should look similar to the figure below, where the axes are the two exam scores, and the positive and negative examples are shown with different markers.
Output: Scatter plot of training data as ps3-1-b.png



c. Write a function, `g = sigmoid(z)` that computes the sigmoid function $g(z) = \frac{1}{1+e^{-z}}$. Test your function by calling `gz = sigmoid(z);` in your main script where `z = [-10:10]`. Plot gz versus z.
Function file: `sigmoid.m` containing function `sigmoid` (identical name)
Output: save the testing figure as ps3-1-c.png

d. Now you will implement the cost function and gradient for logistic regression. Write the function `[J, grad] = costFunction(theta, X_train, y_train)` where `J` contains the value of the cost function computed for a given parameter vector `theta`, and `grad` is the partial derivative of the cost w.r.t each parameter in `theta`. Recall that the cost function in logistic regression is

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

and the gradient of the cost is a vector of the same length as θ where the j^{th} element (for $j = 0, 1, \ldots, n$) is defined as follows

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

Note that while this gradient looks identical to the linear regression gradient, the formula is actually different because linear and logistic regression have different definitions of $h_\theta(x)$. To test your implementation, use the given training data, test your function when $\theta = [0, 0, 0]^T$.
Function file: `costFunction.m` containing function `costFunction` (identical name)
Text output: the value of the cost J when $\theta = [0, 0, 0]^T$.

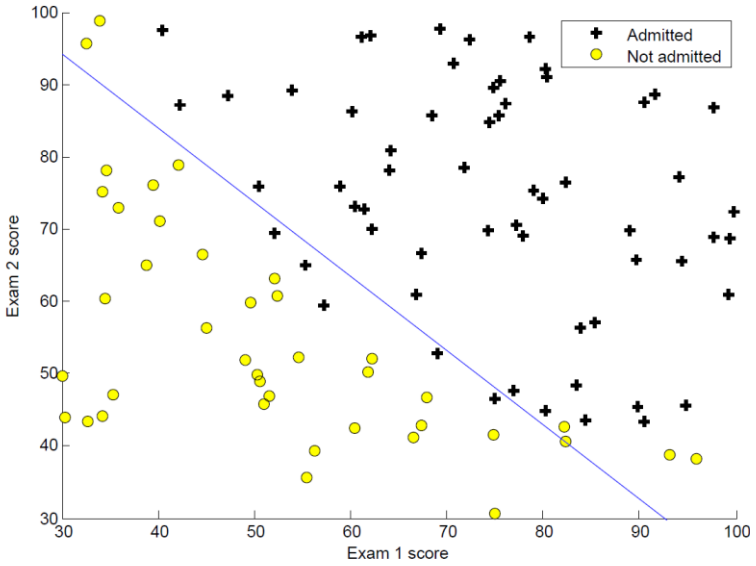
e. For logistic regression, you want to optimize the cost function $J(\theta)$ with parameters θ. To do so, you are going to use the MATLAB built-in fminunc function. fminunc is an optimization solver that finds the minimum of an unconstrained function. In this part we will use a linear model , $h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$. Concretely, you are going to use fminunc to find the best parameters $\theta = [\theta_0, \theta_1, \theta_2]^T$ for the logistic regression cost function, given a training dataset (of X and y values). Use the following options line and use zeros as initial value of theta.

```
%  Set options for fminunc
options = optimset('GradObj', 'on', 'MaxIter', 400);

%  Run fminunc to obtain the optimal theta
%  This function will return theta and the cost
[theta, cost] = ...
    fminunc(@(t)(costFunction(t, X, y)), initial_theta, options);
```

Text output: the optimal parameters θ.
Text output: the value of the cost function at convergence.

f. Once you have the optimal θ, you can plot the decision boundary. With the help of the line function in Matlab (or any other technique of your choice), generate a figure that contains the training samples and the decision boundary. Your figure should be similar this one.
Output: save the figure as ps3-1-f.png



g. Use your model, compute the admission probability of a student whose scores are as following: test1 = 45 and test2 = 85.
Text output: the admission probability
Text output: what should be the admission decision? (admitted or not admitted)