ME5194: Applied Machine Learning for MAE and Robotics
Homework 2, Part 1
Outliers / Classification / Logistic Regression etc.

Due: **Feb 17, Tuesday.** Total points = 25.
Prof. Manoj Srinivasan

Guidelines. As noted in syllabus, it is fine to submit the HW a couple of days later without asking for an extension. Look at the **Course Guidelines** document to see how exactly to submit the HW.

All HW submissions will have two parts: (1) a PDF file and (2) a zip file (named as per instructions) containing sub-folders called Q1, Q2 etc. containing all the data files needed, .m files corresponding to each question: Q1.m, etc. The submitted PDF will have screen-shots of the question, any written answers, MATLAB figures, etc. to produce a coherent narrative response to the HW. The PDF should not contain the code.

On the PDF submission, start each question on a new page.

When uploading the PDF on gradescope, it will ask you to enter the pages on your PDF corresponding to each question. Please fill that information in.

We encourage working together. But to best learn coding, I suggest trying to solve the questions by yourselves first and then comparing notes and discussing with your classmates. Also, rather than just starting with our in-class examples and modifying them to answer the questions below, I think you will learn more and remember more if you either try to re-create the code and/or even re-typing the code (rather than just copy-paste). The benefits of this is like why writing your own notes of lectures may be helpful in learning better.

Solve Q1, Q2, Q5, and solve at least one of Q3 and Q4.

Q1. Some real life data sets

The following datasets are derived from the **UCI Machine Learning datasets**, hosted here:
`https://archive.ics.uci.edu/ml/index.php`.

i) APS Failure at Scania Trucks Data Set:
`https://archive.ics.uci.edu/ml/datasets/APS+Failure+at+Scania+Trucks`

ii) Gas sensor array under dynamic gas mixtures Data Set:
`https://archive.ics.uci.edu/ml/datasets/gas+sensor+array+under+dynamic+gas+mixtures`

iii) Pittsburgh Bridges Data Set:
`https://archive.ics.uci.edu/ml/datasets/Pittsburgh+Bridges`

iv) Robot Execution Failures Data Set (just comment on lp1.dat):
`https://archive.ics.uci.edu/ml/datasets/Robot+Execution+Failures`

v) Any other dataset from the UCI Machine Learning Repository.

Read through the descriptions of each of the datasets. If necessary, you should be able to download and open most datasets either using excel (if a csv file) or using a simple text editor. For **at least three out of the above five datasets**, answer the following questions.

a) You've so far learned about (A) least squares regression for fitting linear or polynomial models from continuous inputs to continuous outputs (B) logistic regression for classification: that is, fitting models that

map continuous inputs to 'categories' or 'classes'. Which of the two Machine Learning methods would you use, (A) or (B)? (or perhaps neither!) And if you select logistic regression, you can say whether you will do 'binary classification' (as in the last HW) or 'multi-class classification' (as in Q4 below).

b) In these models for each of the datasets, what will be the input variables and what will be the predicted variables? Roughly, how many input variables do you have and how many output variables?

Q2. Identifying and removing outliers

Given dataset `DatasetHW2Q1` with real valued input x_i and real-valued output y_i, with $i = 1$ to N. In addition to the plots below, report the values for b_0, b_1, and b_2.

a) Fit a quadratic function $y = b_0 + b_1 x + b_2 x^2$ to all the available data. Overlay the best fit curve on the plotted data.

b) Remove y values that are more than $2\sigma_y$ from the mean μ_y of y. Re-do the regression and plot the best fit curve.

c) Fit without outlier removal and find the error for all data points ($E = Y - XB$). Remove 10 data points with the highest absolute errors: `abs(E)`.

d) Use `robustfit` to do the regression. Note that this is a standard MATLAB function and the syntax for `robustfit` is different from `regress`, so make this appropriate syntax change by looking up `help robustfit`. Specifically your input matrix does not need a column of ones, and the order of arguments may be different (check). Explain in 2-3 sentences how the calculation in `robustfit` is different from that in `regress`.

e) Use `L1regression` code provided in lecture to do the regression. Based on the lectures, explain in 2-3 sentences how the calculation in `L1regression` is conceptually different from that in `regress`.

f) Add one 'far away' outlier to the data. Show that the coefficients from the usual least squares regression (minimizing MSE, say using the backslash or pseudoinverse) is affected by the outlier. But using L1 regression, show that adding the outlier does not very affected the obtained coefficients.

Q2. Logistic regression in 1D

a) You are given `DatasetHW2Q2a`, which has 1-dimensional input data σ_x and a binary label y with value 0 or 1 associated with each data point. Plot the data to visualize. Build a logistic regression model for classifying the data. Based on your program output, what is the approximate condition on x for getting $y = 1$?

b) You are given `DatasetHW2Q2b`, which has 1-dimensional input dataσ_x and a binary label y with value 0 or 1 associated with each data point. Plot the data to visualize. Build a logistic regression model for classifying the data. Based on your program output, what is the approximate condition on x for getting $y = 1$?

c) Report the accuracy of the classifiers on the provided training set.

Hint: Try quadratic instead of linear for Q2b, so you are solving for 3 variables.

Q3. Classification in 2D

You are given `DatasetHW2Q3`, which has 2-dimensional input data (variables σ_x and σ_y) and a binary output label y (which is 0 or 1) associated with each data point. Visualize the data in 2D. Build a logistic regression model for classifying the data. Report the accuracy of the classifier on the provided training set.

Hint: Try models of a few different complexity: $z = b_0 + \ldots$ up to cubic, until you get a good separation of the two categories.

Q4. Logistic regression / Classification in 3D

A material in a 3D stress state has been classified as yielded (1) or not (0). You are given `DatasetHW2Q4`, which has 3-dimensional input data, variables σ_x, σ_y, and σ_z and a binary output label y (which is 0 or 1)

associated with each data point. Visualize the data in 3D using `plot3`. Build a logistic regression model for classifying the data. Report the accuracy of the classifier on the provided training set. No need to plot the boundary in 3D.

Hint: $z = \ldots$ a quadratic function of the σ's should work well.

Q5. MNIST database.

Use the MNIST database file in the logistic programs folder.

a) Modify the provided programs to determine whether the image contains a number greater than 5 or not. What is the training set accuracy and the test sect accuracy?

b) Modify the provided programs to determine whether the image contains a number is even or odd. Consider 0 to be an even number. What is the training set accuracy and the test set accuracy?

c) Modify the provided programs to determine whether the image contains a number is 7 or not.

Just use a linear model for the decision boundary. Can you think of any rationale for why the accuracies are high or low for each problem?

Further useful notes

- For initial guesses Binput in the logistic regression code, it is generally best to leave them at their zero value, rather than use the randn value. (Even though random initial guesses should work theoretically, there are sometimes numerical difficulties related to computers' inability to deal with very large numbers.)

- When you are asked to find 'accuracy' of a logistic regression classifier, what you do is take your data set, and use Xmatrix*B on your dataset to get z. Then, if $z >= 0$, $y_{\text{predicted}} = 1$, and if $z < 0$, $y_{\text{predicted}} = 0$. You can then compare this with the yList in the data, and see what fraction the model gets right. This is done in the example code given for MNIST and you can modify that for any of the other questions. Again, $z = 0$ corresponds to the 'decision boundary'.

- a cubic model in 2 variables x and y has the following terms: constant, $x, y, x^2, xy, y^2, x^3, x^2y, xy^2, y^3$. That is, all the terms you will find (plus a constant) when you do $(x + y)^3$.