CSCI567 Fall 2015

Homework #2

Vaibhav Behl

vbehl@usc.edu

1. Linear Regression (Short Answer)

a) Linear Regression assumes uncertainty only in the measurement of dependent variable(y) and assumes that independent variables have been measured without errors. We have a model called "Noisy Observation Model" which maps this uncertainty as a Gaussian random variable η.

$$Y = w_0 + w_1 X + \eta$$

b) Outliers can impact our Linear Regression model severely. The impact is more severe when we use Least Square Distances (L_2) to measure our errors, thus one approach could be to use absolute distances to measure errors. This will decrease the impact an outlier has on our cost function. Another advance approach to deal with outliers is to use Random sample consensus (RANSAC) method. This method assumes that data consist of outliers and inliers, and that the number of outliers is not significantly high. In this method, we first randomly select a subset of data and fit a model to it. Then, we check whether all data points are consistent to this model or not. A data point will be considered an outlier if it doesn't fit to the set of estimated models within an error threshold. These steps are then repeated until we have a certain number of confirmed inliers. Thus this algorithm can detect outliers and reduce their impact on the model.

c) It is wrong to assume that magnitude of a regression coefficients have any relation to the importance of its corresponding feature. In fact, to deal with this and reduce the impact of a feature may have of the model because of its value, we Normalize the data. An example could be a feature measuring birth weight of a subject and predicting his life expectancy. Now this weight, whether measured in Pounds or Kgs, should not have an impact on the importance of this feature at all.

d) If one independent variable is a perfect linear combination of another independent variable, then we cannot take inverse of the predictor matrix (X). This will prevent us from getting a unique solution to the below equation.

$$w^{LMS} = (X^T X)^{-1} X^T y$$

e) We know that with 1-of-k encoding, we get a matrix where one columns can be represented as a linear combination of another columns, which means that the matrix is not full rank and thus cannot be inverted. This can be remedied by using (k-1) encoding, where k is the number of categories for the feature we want to encode. This approach will prevent a column from being a linear combination of another column and still allow us to use constant intercept term and normalization. For

example, to encode A,B,C. We'll encode only two categories and the third one will be reference. A(1 0) B(0 1) C(0 0).

f) With the help of regression coefficients of a linear regression model, we can estimate the impact an independent variable X_i may have on the dependent variable Y, given that others independent variables are kept same. But if the variables are not linearly independent, than we cannot have a precise estimate of the impact its change has on the dependent variable Y. This makes it tough to understand change in coefficient values on changing one variable.

g) Logistic regression uses a sigmoid function through which we classify into binary classes. With linear regression we could possibly achieve the same thing by using a cut-off point of 0.5. But in linear regression we fit a line to the data points, so the slope of this line could change based on the presence of outliers, which would make the points previously classified as more than 0.5 to now be less than 0.5, thus increasing the error rate. But, because we use sigmoid function with logistic regression, the outliers don't impact the classification severely.

h) Linear regression is usually an over-determined system of linear equations, since for most real world cases, the number of examples exceeds the number of features/columns present. But for some cases we can have feature size exceed the number of examples. This will cause this system of equations to be under-determined, since there is no way to get a unique solution to them via analytical approach. There are several approaches to solve under-determined equations, but taking an example from this assignment, in part (f) Sequential Feature Selection, we have a matrix X, which after basis expansion, has more columns than rows. In that part we try to select the best features one at a time, and try to reach convergence.

3. Perceptron and Online Learning

At i^{th}step we have the classifier as w_i and we want to update this to w_{i+1} based on points (x_i, y_i). The condition is that we want to minimize - $\left|\left|w_{i+1} - w_i\right|\right|_2$, given that w_{i+1} also classifies correctly on the current sample point (x_i, y_i).

Now we define our function that we have to minimize as: $f(w) = \left|\left|w - w_i\right|\right|_2$; and the constraint is that - $y_i(w^T x_i) = 0$

To solve this we make use of Lagrange multipliers. We define a function $L(w, \lambda)$ as:

$$L(w, \lambda) = f(w) + \lambda g(w)$$

Putting values into this equation we get:

$$= \left|\left|w - w_i\right|\right|_2^2 + \lambda(y_i w^T x_i)$$

$$= \frac{1}{2}(w - w_i)^T(w - w_i) + \lambda(y_i w^T x_i)$$

Now we'll take its derivative and equate it to zero:

$$\frac{dL}{d\lambda} = (w - w_i) - \lambda x_i y_i = 0$$

Solving this equation we get:

$$w = w_i + \lambda x_i y_i$$

Now, taking transpose on both sides :

$$w^T = w_i^T + \lambda(x_i y_i)^T$$

Now, simplifying this equation so that we get the constraint term in it which we'll equate to zero:

$$w_i^T(x_i y_i) + \lambda(x_i y_i)^T(x_i y_i) = w^T(x_i y_i) = 0$$

Simplifying for λ we get:

$$\lambda = -\frac{w_i^T(x_i y_i)}{\left|\left|x_i\right|\right|_2^2}$$

Putting this value of λ back in the previous equation, we can now get a value for w as w_{i+1} :

$$w_{i+1} = w_i - \frac{w_i^T(x_i)}{\left|\left|x_i\right|\right|_2^2} x_i$$

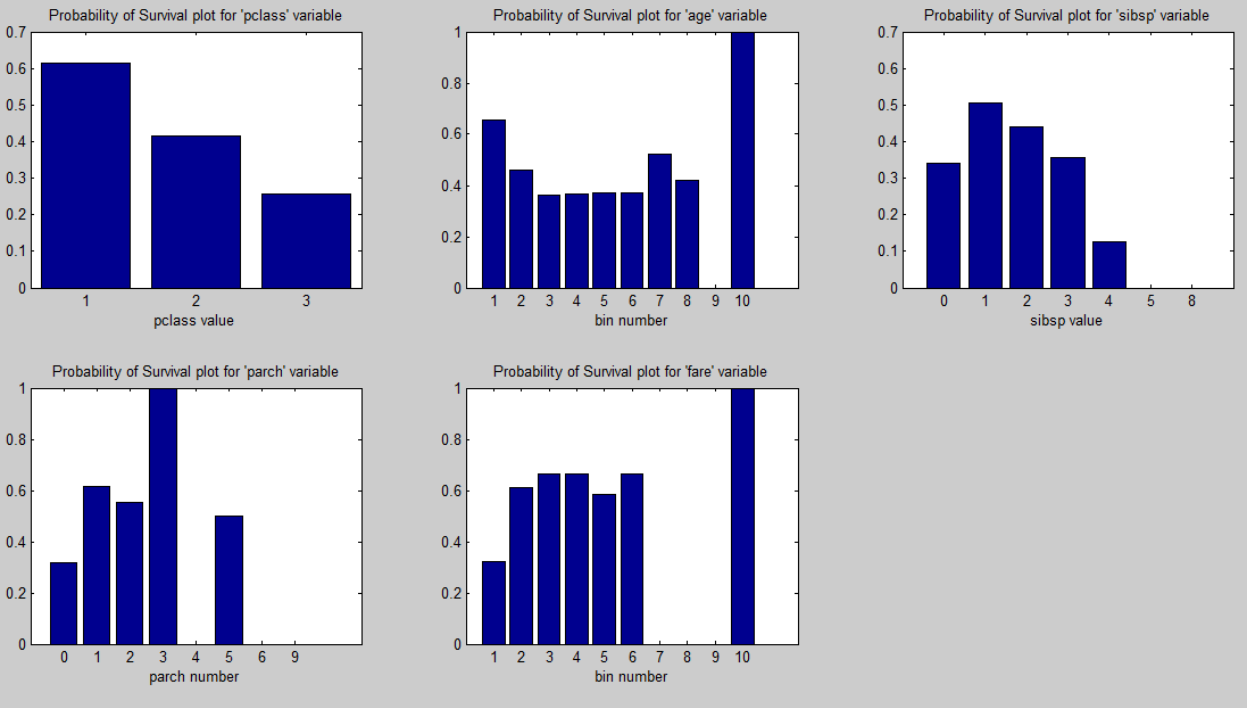
4. Programming - Logistic Regression (Matlab Version used: R2013b)

a) Load Data

Data was loaded using xlsread. Below are the number of missing values in each column:

Missing values for pclass - 0
Missing values for survived - 0
Missing values for name - 0
Missing values for sex - 0
Missing values for age - 263
Missing values for sibsp - 0
Missing values for parch - 0
Missing values for ticket - 0
Missing values for fare - 1
Missing values for cabin - 1014
Missing values for embarked - 2
Missing values for boat - 823
Missing values for body - 1188
Missing values for home.dest – 564

b) Monotonic Relationship



Relation of numeric variable with probability of survival-

- pclass- this variable shows a perfect decreasing monotonic relationship with probability of survival.
- age- this variable shows a non-linear(U-shaped) relation with probability of survival.
- Sibsp- except for one bin/value, this variable shows a decreasing monotonic relationship, although not as strongly as pclass.
- Parch- this variable shows an increasing monotonic relationship, except at one value.
- Fare- this variable also shows an increasing monotonic relationship, except for one value in the middle.

c) Mutual Information

List of independent variables and their mutual information scores sorted in descending order by mutual information:

homeDest = 1.351319
name = 1.336810
age = 1.333718
embarked = 1.325473
sibsp = 1.321358
parch = 1.319019
ticket = 1.295110
pclass = 1.280851
fare = 1.272010
cabin = 1.269095
sex = 1.183012
boat = 0.746621
body = 0.668440

d) Missing Values

- Multiple Model
  - Train Accuracy: 80.02%
  - Test Accuracy: 77.98%
- Substituting Values
  - Train Accuracy: 79.54%
  - Test Accuracy: 77.68%

The accuracy shown by both methods doesn't differ by much, but using Multiple Models does give a higher accuracy than using a mean value to substitute the missing Age values.

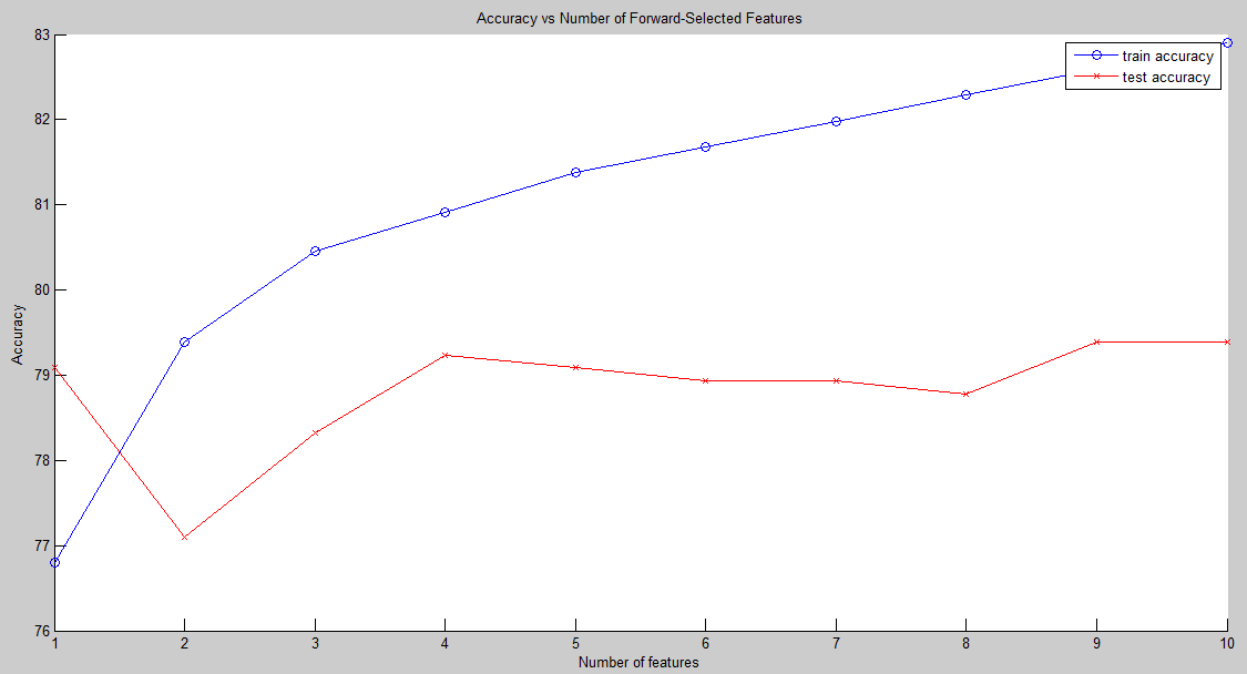
e) Basis Expansion

After completing the all the steps mentioned in this section, we got the number of columns in X (train and test) to be – 846, for one of the iterations. This value changes as the data in Train and Test is chosen randomly, but stays in the range from 830 to 848.

f) Sequential Feature Selection

Below is a plot of Training and Testing Accuracy vs the number of forward selected features. Answers to questions asked:

- Yes, this type of features selection worked well because we were able to extract key features from the set of many features we had, and still maintain a high accuracy.
- For this particular case, optimal features appear to be around 4, since after that even though the training accuracy increases, the testing accuracy falls, indicating an over-fitting (although the curve does stabilize around 9-10 feature count, but doesn't rise higher than 4)



g) Batch Gradient Descent

After running this algorithm for different step sizes over different number of iterations, following results are achieved (below). The accuracy gradually increases as we increase the step size, implying that we are converging. Also we are using the same dataset as used for part (f), so we can compare the numerical value from the graph.

The accuracy for step size 0.100000 :
Accuracy for iterations 500 :
Train accuracy - 76.946565 %
Test accuracy - 75.229358 %
Accuracy for iterations 1000 :
Train accuracy - 78.473282 %
Test accuracy - 76.299694 %
Accuracy for iterations 1500 :
Train accuracy - 77.709924 %
Test accuracy - 75.229358 %
Accuracy for iterations 2000 :
Train accuracy - 75.419847 %
Test accuracy - 74.311927 %
Accuracy for iterations 2500 :
Train accuracy - 77.251908 %
Test accuracy - 78.899083 %
Accuracy for iterations 3000 :
Train accuracy - 80.458015 %
Test accuracy - 79.911315 %

The accuracy for step size 0.300000 :
Accuracy for iterations 500 :
Train accuracy - 75.442748 %
Test accuracy - 76.510703 %
Accuracy for iterations 1000 :
Train accuracy - 77.251908 %
Test accuracy - 79.051988 %
Accuracy for iterations 1500 :
Train accuracy - 81.526718 %
Test accuracy - 79.051988 %
Accuracy for iterations 2000 :
Train accuracy - 77.251908 %
Test accuracy - 77.981651 %
Accuracy for iterations 2500 :
Train accuracy - 79.541985 %

Test accuracy - 77.370031 %
Accuracy for iterations 3000 :
Train accuracy - 82.900763 %
Test accuracy - 79.510703 %

h) Newton's Method

After running this algorithm for five iterations, we observe the below outputs. It can be seen that convergence is achieved in 2 iterations, after which there is no improvement in accuracy. Also the training accuracy is greater than the one achieved with glmfit(part-f), in which we had selected 10 features.

Accuracy for iteration 1 :
Train Accuracy = 0.841221
Test Accuracy = 0.776758

Accuracy for iteration 2 :
Train Accuracy = 0.842748
Test Accuracy = 0.776758

Accuracy for iteration 3 :
Train Accuracy = 0.842748
Test Accuracy = 0.776758

Accuracy for iteration 4 :
Train Accuracy = 0.842748
Test Accuracy = 0.776758

Accuracy for iteration 5 :
Train Accuracy = 0.842748
Test Accuracy = 0.776758