Lab Assignment 10 Report

Alekhya Gupta

Soham Panigrahi

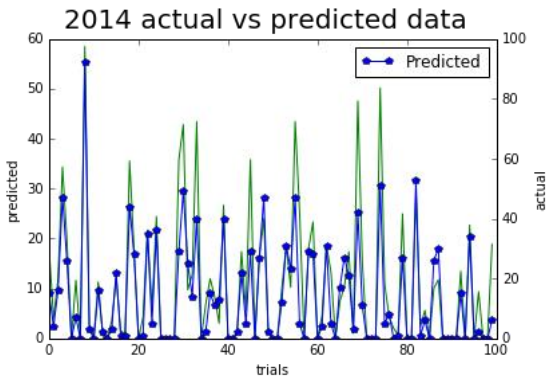
ValError : **77.53**

Normalised RMS: 0.0527 (5.27%) which is (RMS/(ytruemax-ytruemin)) where ymax-ymin= 3019

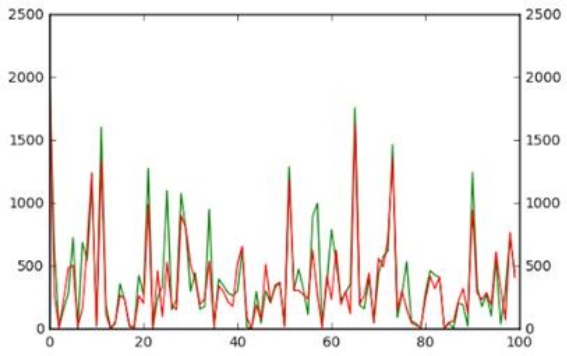
Calculate the error of prediction: E = (gt_count - predicted_count)^2
Calculate the averaged error AvgE for E over all trials and event types. The final model validation error is:
ValError = sqrt(AvgE).
However, the measure we use here is Normalised root mean square error which is: Validation error/ (ytruemax -ytruemin)

Evaluation:

We plotted the graph between actual and predicted 2014 values for Ridge regression obstruction dataset.



Ridge Regression



Polyfit regression

Green graph: predicted 2014 values

Red: actual 2014 counts

The predicted values also look well correlated with the actual values.

We see that the data has a lot of zero counts and therefore spent time selecting Ridge regression as the appropriate model which assigns low weightage to values the model considers as outliers (insignificant). Ridge regression also helps to avoid overfitting.

This was why we decided **against** Decision tree regression because it seems to overfit the data.

Introduction:

In this assignment we use the information on the occurrence of the following events (Accidents and Incidents (A), Roadwork (R), Precipitation (P), Device Status (D), Obstruction (O), Traffic Conditions (T)) in different years. We use this data to make a prediction on the future occurrences of the events.

Approach: Analytics and statistical approaches

We make a count of the individual events as per their occurrences and divide them into boxes in terms of years. Now we use this data of type Event{year,count} to fit into various regression models and make a prediction for the counts on year 2014. We compare the predicted values and analyze them against the actual event counts for the year 2014 to examine the validity of the regression model. We also check for overfitting of the data into models while making predictions on the data and discard such models.

Salient implementation features:

Bounding boxes:

The first part involved getting the {year,count} tuples for all the six different event types for all the prediction trials. This was very computationally expensive, hence we implemented it using Python multiprocessing (Pool) and avoided for loop by using numpy.all function to apply the bounding box conditions to all rows of the event type. Therefore, we were able to complete this part in less than 30 minutes.

- Shapely : Our test results using this library gave the slowest results and hence was discarded.
- matplotlib.path :This performed very well for test sets. But given the scale of the data this too failed to provide a practical solution.

After finalizing the method, we ran our data for all geo boxes and utilized pythons multiprocessing to speed up the method.

```
from multiprocessing import Pool
p = Pool(3)
```

Regression models tried:

- **Linear Regression:** The results were unsatisfactory with the RMS(root-mean-square) value close to 92.7
- **Polyfit Regression:**
  The results gave validation error value around **83.21** when we use polyfit of a degree of 4. This model generated some promising predicted values for some of the events. The normalized RMS is 0.0572 which is 5.72%.
- **Decision tree Regression:** This model generated RMS of 79.5.
- **Ridge regression:**
  Linear least squares with l2 regularization.
- This model solves a regression model where the loss function is the linear least squares function and regularization is given by the l2-norm. Also known as Ridge Regression or Tikhonov regularization. This estimator has built-in support for multi-variate regression
- This model generated validation error of 65.67 and Normalised RMS of
- We check for overfitting by running with different test data like 2013 (comparing actual vs predicted) and for 2012.

This part of the code runs in less than 15 minutes

Methodology:

We tried various regression models like Linear regression, Curve fitting, Polyfit (polynomial regression), Ridge regression, Decision trees regression. Linear regression generated validation error value of 92.74 for roadworks event type, polyfit generated validation error of 83.3, Ridge regression (with alpha of 15 with rows which have greater than 5 non-zero values and alpha=30 with rows which have many zeroes) gives validation error of 83.16.

We finally decided to go with Ridge regression because it generated an overall validation error of 65.67. The normalized rms for it is:
Here we set alpha as 15 for rows which have less than 5 zero values and 30 for the other rows.

Challenges:

- Improving efficiency of the loops in order to generate the counts of the events.
- Making sure that the regression models did not over fit the data and mislead with accurate predictions.
- We do so by checking for the percentage of the data that has zero difference with the actual event counts from the year 2014.
- We also compute the root mean square error on the predicted event count for the year 2014.

Implementation steps:

- Generate the count of the individual events that occurred in a range of the geo boxes and compute their occurrences by years.
- We fit the computed data with the counts of the events as per years on different regression models and verify the predicted values for the year 2014.
- We use linear and polynomial regression models along with Decision trees and ridge regression to make a prediction of the event counts for 2015 and 2016.
- We compute the root mean squared value for the predicted Value and evaluate the models.
- We compute the final valError from the combined RMS values obtained from all the events.
- We write out the predicted values dataframe into a prediction.tsv.
- We divide the predicted values generated by the models by 12 to get predicted values for month.

Summary (Improvement of the final results):

The required output was obtained and hence we predicted the traffic events of the year 2015 and 2016 using the training data from the previous decade of traffic data.