Assignment 6, Urban Informatics

Ravi Shroff

December 8, 2013

Note that the code for each part of the assignment (a through d) is in files named problem_a.py, etc. There are also three plots from part a and a bar chart from part c. Part d includes a text file, and explanation of procedures is included both in this document and in comments in the code.

(a) We make three simple scatterplots of the labeled data, with population on the x-axis, number of incidents on the y-axis, and each point representing one zip code. In our first scatterplot, problem_a_graph_1.png , we plot all zip codes and notice that while there are a decent number of points which have incident counts positively correlated with population, there are also many points with very low incident counts (for both small and large populations). This observation is confirmed by our second and third plots, problem_a_graph_2.png and problem_a_graph_3.png, which just plot the zip codes which have more than 10 and 50 incidents, respectively. When we just plot the zip codes with more than 50 incidents, we notice that now population and incident count seem to have a noticeable positive correlation.

This suggests that we should somehow split the zip codes into two chunks, those with few incident counts (to be dealt with in one model), and those with many incident counts. We also notice (this observation is due to Alex Chohlas-Wood) that many (perhaps all) zip codes with few incident counts lie *outside* New York City itself (but in New York State). Therefore when we come up with our final model in part d, we will consider another dataset that just has zip codes from NYC, and use one model for NYC zip codes and another for non-NYC zip codes.

(b) As noted in the comments in the file problem_b.py, we choose the degree three polynomial to model our data since it minimizes root mean square error (RMSE) and (almost always) maximizes R^2 scores.

(c) For this graph we use values from one run of problem_b.py (the values for 10-fold cross validation RMSE and the corresponding standard deviations will vary depending on what the folds are). We graph the RMSE values *without* cross validation (i.e. on the whole training set) in yellow, and the RMSE values *with* cross validation in blue, with the standard error bars for cross validation in red. Note that we have elected to display just the graph from $11,000$ upward on the y-axis for clarity, and hence have scaled the error bars accordingly.

The main thing we notice is that without cross-validation, the degree 5 polynomial model minimizes RMSE, rather than the degree 3 polynomial with cross-validation. This is perhaps unsurprising, as we may expect RMSE to decrease as model complexity increases, indicating overfitting. We also notice that the size of the error bars seems significantly larger in degree 4 and degree 5 (with cross validation). This makes sense to me because as the degree of the polynomial increases (as we overfit more and more), the model is better for each fold, but there is more variation as we switch to different folds, so larger standard deviation.

(d) For our final OLS model we use an external data set, boroughs.csv, which only has the zip codes from New York City itself. We use this external data set to partition our labeled training data into two chunks, one of those zip codes and populations from NYC, and the other with those zip codes and populations outside NYC. We perform 10-fold cross-validation on both chunks of the training data, and find that a linear model suits the NYC data best, while a quadratic model suits the outside-NYC data best.

Next, we partition the unlabeled test data into two chunks based on the same criteria as before. We plug the unlabeled population data from NYC zip codes into the linear model from above, then plug the unlabeled population data from outside-NYC zip codes into the quadratic model, to get our predicted values. Finally, we (awkwardly) reassemble all population and predicted incident count information into one list in the same order as the original unlabeled data file. We print the output in the form (zip, population, predicted incident count) in the file test_data_predictions.txt.

(e) I don't really know how to estimate the expected RMSE from our application of the OLS model in part (d) to the unlabeled data. I'd hope it would be at most the same order of magnitude as the RMSEs computed in part (b). If I had more time I'd mainly try to teach myself more machine learning techniques to perhaps choose a more sophisticated model! I would also look for additional data sets (for example, maybe income by zip code) to include as independent variables in the model. However, I don't feel like I understand the process of systematically adding new data sets to improve an OLS model enough to comment on it at length. I really need to just learn more about ML in general.