Predicting airline delays

Raj Bandyopadhyay, Rafael Guerrero

12/14/2012

Introduction

In this project, we use publicly available data originally from the Bureau of Transportation Statistics to analyse and predict flight departure delays for a subset of commercial flights in the United States. We have three goals in mind. First, we would like to identify the factors which are most likely to cause flight delays. Second, we want to predict whether an individual flight will be delayed. Finally, if there is a delay, we would like to estimate its magnitude.

In this paper, we first describe the important features of the data, along with our preliminary pre-processing and analysis. As part of this step, we use linear regression to identify the most important factors affecting delays. Subsequently, we use a classifier (SVM) to predict if there will be a delay. To estimate the magnitude of delays, we use a non-parametric quadratic regression algorithm.

The airline delay data set

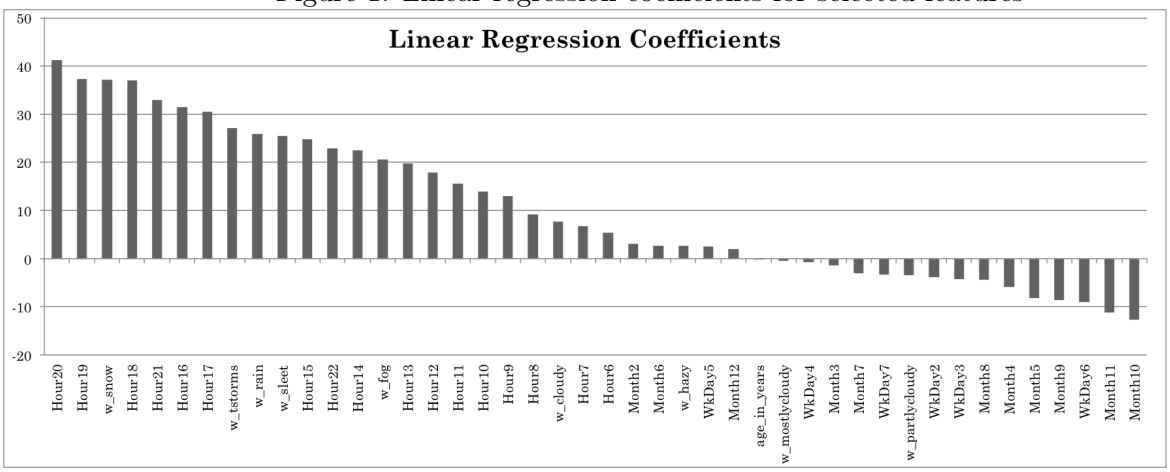
The original data set [1] contains information for all commercial flights in the US from 1987 to 2008. Since the data set is extremely large (several million records) we extracted a reasonable subset of the data as follows:

- Two years: 2007 and 2008.
- One airport of origin: O'Hare (Chicago), since it is one of the busiest airports and is exposed to a wide range of weather conditions.
- One airline: AA (American Airlines), since it is the largest airline using the O'Hare airport.

This reduces the size of our data set to around 80,000 records. For each record, we use secondary data sources to enrich it with information about airplanes and historical weather statistics. At the end of the pre-processing steps, our data set has the following features:

- Flight features: These include the times of departure and arrival, the destination airport, and the distance covered by the flight. Except for the distance, we convert all variables to categorical form.
- Airplane features: These include the make and model of the plane, its age and engine type.
- Weather features: We use the Weather Underground API [2] to obtain historical weather information for the time closest to the departure time. The weather data includes several categorical features indicating the presence of snow, hail, thunder, rain and tornado warnings. It also contains a few numeric features such as wind speed, temperature and humidity.

Figure 1: Linear regression coefficients for selected features



Preliminary analysis using Linear Regression

As our first pre-processing step, we use histograms and basic plots to eliminate features that have no impact on the delays, such as the airplane manufacturer and temperature. We then use linear regression (as implemented in R [3]) as a quick-and-dirty first technique to examine the data. Our goal with linear regression is threefold: first, we want to use the regression parameters to identify prominent factors affecting delays. Second, we would like to get a baseline estimate of how accurately delays can be predicted. Third, we want to analyse how the prediction error changes with the size of the training set.

Features affecting delays

The coefficients of the linear regression (Figure 1) give us a good idea of the relative importance of various features. We observe that the features most correlated with delays are the flight departure times, followed by certain bad weather conditions. For example, flights departing during evening hours (around 8 pm) tend to have more delays. Also, adverse weather conditions such as snow, thunderstorms, rain, sleet and fog show a high correlation with delays. Somewhat surprisingly, the month has little effect on the delay. In fact, certain months (such as October) even show a small negative correlation, indicating that flights tend to be early in those months.

Influence of training set size

In order to study the effect of training set size, we run cross-validated linear regression on data sets ranging from 500 to 80,000 samples and measure the root mean residual error. The data sets are generated by randomly sampling the original data. We find that beyond 1,000 samples, this error remains practically the same for all sample sizes. This leads us to conclude that for a well-shuffled data set, the size of the training set has little influence beyond a relatively small size of 1,000 points.

Predicting delay with classifiers (Naive-Bayes and SVMs)

In order to predict whether a flight will be delayed or not, we model the problem as a classification with two classes: *delayed* for flights with delays above 15 minutes, and *non-delayed* otherwise. Using the

Table 1: Performance of classifiers in predicting non-delayed flights

	Accuracy %	Precision	Recall	F-score
Naive-bayes	70.8	0.752	0.81	0.78
SVM (unweighted)	71.65	0.737	0.864	0.796
SVM (weighted, 1 FN = 10 FP)	49.2	0.909	0.227	0.363
Random Forests (unweighted)	70.94	0.784	0.813	0.798
Random Forests (weighted)	58.01	0.839	0.502	0.628

Table 2: Performance of classifiers in predicting delayed flights

	Accuracy %	Precision	Recall	F-score
Naive-bayes	70.8	0.612	0.527	0.566
SVM (unweighted)	71.65	0.655	0.455	0.538
SVM (weighted, 1 FN = 10 FP)	49.2	0.413	0.86	0.577
Random Forests (unweighted)	70.94	0.507	0.461	0.483
Random Forests (weighted)	58.01	0.391	0.769	0.519

Weka machine learning toolkit [4], we first apply a Naive-Bayes algorithm with ten-fold cross-validation on the entire training set. The Naive-Bayes algorithm runs extremely fast and provides some baseline results, as shown in Tables 1 and 2.

The Naive-Bayes results show us that the classifer performance is far better in predicting non-delayed flights than delayed ones. The F-score on predicting on-time flights is 0.78, while that for delays is only 0.566. So, how can we improve the performance of the classifier, particularly in predicting delays?

We train an SVM, which is reputed to be the best out-of-the-box classifer. Our initial attempts at using the SVM (with a gaussian kernel) does not greatly improve performance. In fact, the SVM performs slightly worse than Naive-Bayes in predicting delays (F-score 0.538).

Since it's more important, in our opinion, to predict a delay correctly than an on-time flight, we run an SVM again, this time setting a 10:1 cost penalty for delays. That is, predicting a false negative incurs 10 times the cost of a false positive. This step improves our classification result significantly for delays, pushing the F-score slightly above Naive-Bayes, but it adversely impacts the classification of on-time flights.

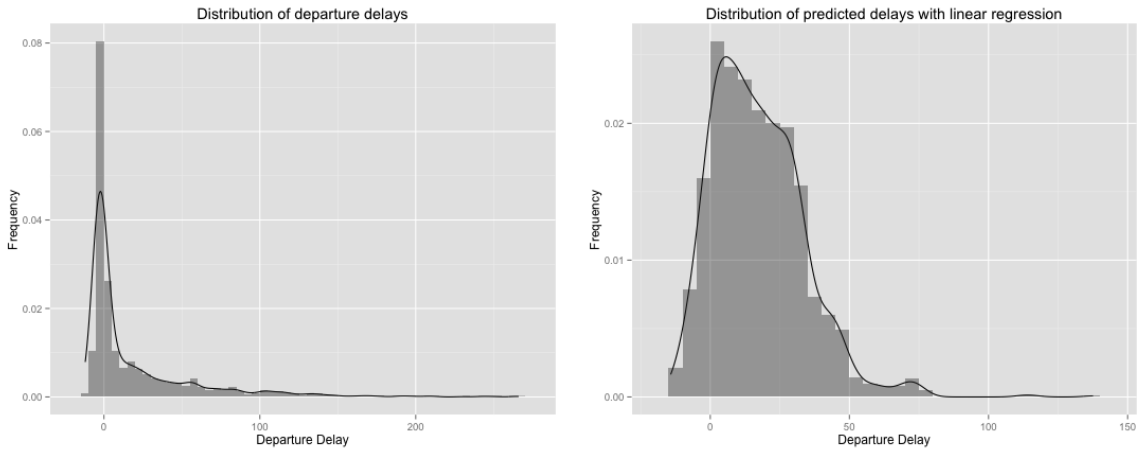
As a digression, we also show some classification results with Random Forests [5]. While they perform worse in general than Naive-Bayes and SVMs, running Random Forests with a weighted penalty improves classification on delays without significantly hurting that on non-delayed flights.

Estimating delay magnitude with non-parametric regression

Why does linear regression perform so poorly? In order to investigate this question, we perform some experiments on a smaller training set (4,500 samples) and test set (1,500) samples. We first plot the distributions of the actual and predicted departure delays and notice a very large discrepancy. While the distribution of predicted delays looks 'spread-out', the actual delays are drawn from an extremely skewed and narrow distribution (Figure 2). This results in a high RMS error (37 minutes).

Our first thought is to use a GLM to model the delay distribution. The GLM is based on a gamma distribution, whose shape is close to the actual delay distribution. In order to get this to work, we add

Figure 2: Distribution of actual delays, *left*, predicted by Linear Regression, *right.*



a positive number to the delay, since the gamma distribution is only defined for non-negative numbers. However, when we use a GLM, our RMS error actually increases to 41 minutes. This is because while the GLM models the peak of the delay distribution well, it fails to fit the tail, thereby performing poorly on large delays (Figure 3).

Our next approach is to use a non-parametric regression model, based on the following justification:

1. We need a way to model the skewness of the distribution.
2. Small training sets do as well as large ones.
3. In real life, airplane delays often occur in 'clumps', i.e., flights get delayed around the same time. This is borne out by some simple time series plots. Can we use this fact to our benefit?

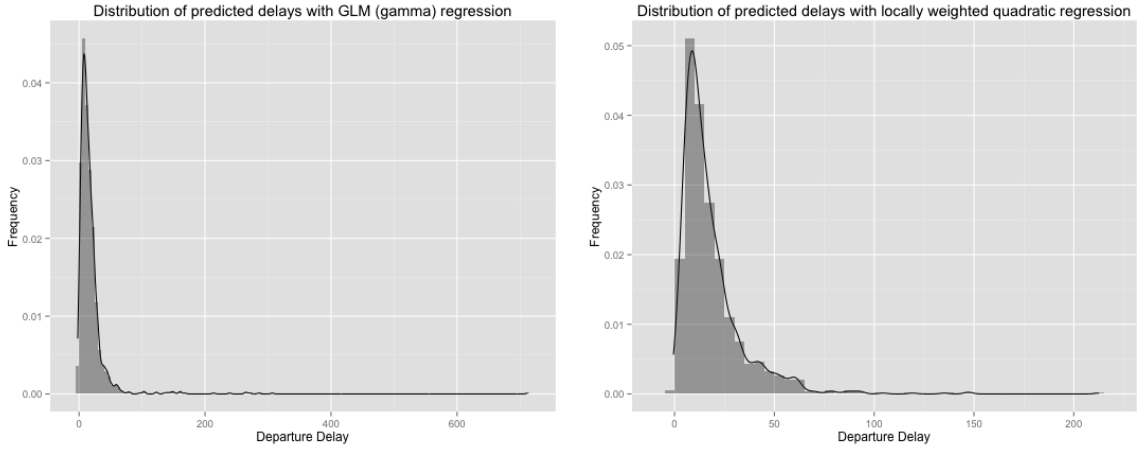
Since we have several features in our data set involving flight departure time (month, day, hour), flights closer in time to a given sample flight are more influential on the prediction for that sample. Also, the slow execution time of non-parametric regression can be mitigated, in this case, by using a smaller training set without losing much in terms of predictive power.

Using a non-parametric regression implementation in R and cross-validation, we train several models on randomly constructed sample data sets and evaluate the best parameters for the model. We find that a locally weighted quadratic regression gives us the best results that we have attained so far in terms of modeling the delay distribution. However the RMS error decreases only slightly to 35 minutes.

Discussion

We have had mixed success in attaining our three goals for this project. We have achieved the first goal of identifying factors that influence delays using a simple linear regression. For the second goal of predicting whether a flight will be delayed, we have shown that overall, a simple Naive-Bayes does pretty well. A weighted SVM does a slightly better job in predicting delays, but takes a much longer time to train. Our efforts have been least successful in estimating the delay magnitudes. While a

Figure 3: Predicted delay distribution with GLM (gamma) regression, *left,* and non-parametric regression, *right.*



non-parametric regression produces a distribution of predicted delays that is closer to the actual delay distribution, it still has a high RMS error overall.

Why is the data so resistant to prediction? In our opinion, the highly skewed distribution has an important effect on the algorithms. The distribution has two main characteristics: 1) a very high peak around zero and 2) a long right tail. All of the algorithms we tried so far do very well on one of the two characteristics at the expense of the other. For example, the GLM gamma regression captures the peak but not the tail, while the non-parametric regression captures the tail but not the peak. As a result, each algorithm makes either a few big mistakes on larger delays, or many small mistakes on smaller delays, thereby keeping the RMS error high.

Future work: using a mixture model

Based on our experience with this data set so far, we would like to try a density estimation approach. Since each of our current models has difficulty modeling either the tail or the peak of the delay distribution, we would develop one model for the peak (small delays) and another for the tail (long delays). This is analogous to using a mixture of gaussians, but our models would not necessarily be gaussian. Assuming that our data is drawn from this mixture model, we would estimate the parameters governing the individual components using an Expectation Maximization (EM) approach.

References

[1] American Statistics Association, Data Expo 2009, `http://stat-computing.org/dataexpo/2009/`

[2] The Weather Underground API, `http://www.wunderground.com/weather/api/`

[3] The R project for statistical computing, `http://www.r-project.org/`

[4] Weka, `http://www.cs.waikato.ac.nz/ml/weka/`

[5] Random Forests, `http://www.stat.berkeley.edu/~breiman/RandomForests/cc_home.htm`