CS229 Term Project Report

Learning to Predict Flight Delay

Lei Sun
sunlei@stanford.edu

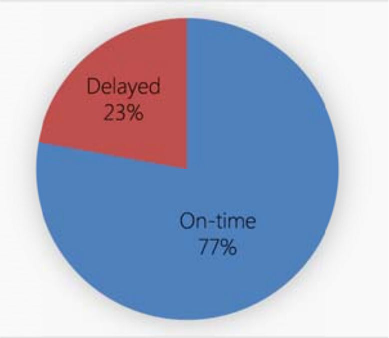
Abstract

Flight Delay is a very common risk when people take airplane. It is a frustrating experience not only to the people who have scheduled events after it, but also to the people who have connecting flights, and even to the people who meet arrival at the airport terminals. But with reliable prediction, people can have a head up of how likely their flights are going to be delayed such that they can mitigate the risk days before the flight. In this project, three different machine learning models are developed to predict the likelihood of a flight delay. The result from these three models will be compared.

Introduction

A flight delay is a delay in which an airline flight takes off and/or lands later than its scheduled time. The Federal Aviation Administration (FAA) considers a flight to be delayed when it is 15 minutes later than its scheduled time [1].

Unfortunately, air traffic control is a large and complicated system, which depends on many factors. Especially, flight safety is a factor that always has the highest priority, and should never be compromised. In year 2011, 23% of the 4,608,956 flight operations in USA are delayed [2]. In another word, one flight out of five is delayed.



Flight delays are a frustrating inconvenience to passengers. A delayed flight might be costly to passengers by making them late to their personal scheduled events. A passenger who is delayed on a multi-plane trip could miss a connecting flight.

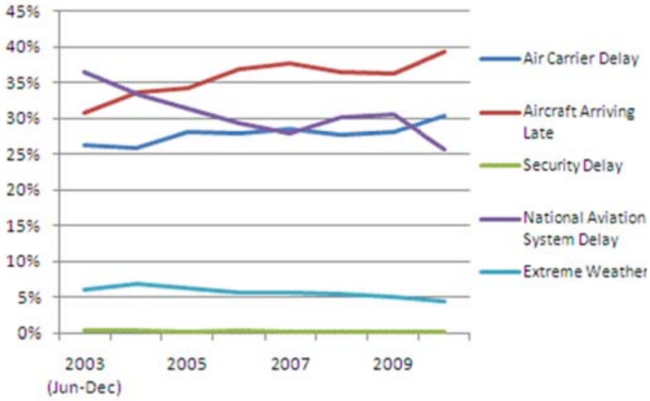
According to risk management theory, a risk is quantified by its probability and its consequence. Flight delay has both high probability and high consequence, thus should be definitely considered as high risk.

In this project, I am going to develop some machine learning models to predict the flight delay by analyzing some historical data. Since all the flight information and weather forecast is known before a flight, then it is possible that some prediction can be made to see how likely a flight is going to be delayed. Then people can make some migration plan in advance.

Data Collection and Overview

According to the statistics from The Research and Innovative Technology Administration

(RITA), the 5 top reasons that cause flight delay are: Air Carrier reason, Extreme Weather, National Aviation System (NAS) reason, Late-arriving aircraft and Security [3]



The figure above shows the percentage of the reason which caused the flight delay. At the first glance, surprisingly, weather causes only 4 percent of flight delays. But there is another category of weather within the NAS category. This type of weather slows the operations of the system but does not prevent flying. During 2011, 75.5% of NAS delays were due to weather [3].

RITA also provides detailed on-time performance data within some time range for research purpose. It contains most of the flight information. More importantly, the data set is large enough for both training and testing.

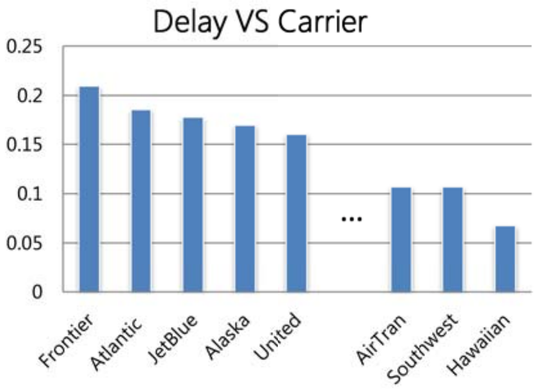
Unfortunately, the weather data is not provided in the data set. But due to its direct and strong relationship with flight delay, I decided to collect all the weather data for each flight record. After trying a couple of approaches, I finally chose to use the API on wunderground.com, although for some small airports, only some less accurate historical data is available, but they should be sufficient for us to build the model.

Initially, we select 11 features that are related to the 5 top reasons listed by RITA:

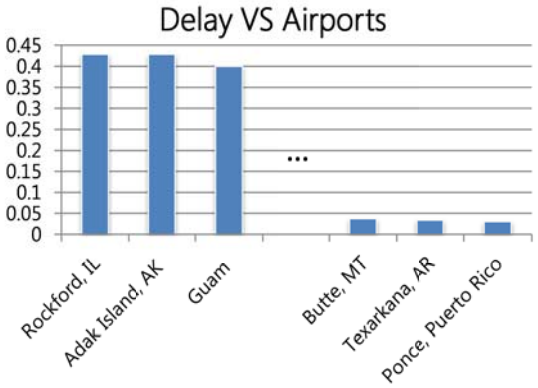
	Day of week	*
	Carrier	
Departure	Airport	
	Visibility	
	Wind Speed	
	Condition (Snow, Rain…)	
Arrival	Airport	
	Visibility	
	Wind Speed	
	Condition (Snow, Rain…)	
	Distance of flight	*

In the following four figures, I showed the percentage of delayed flight versus some selected feature data to visualize the patterns in it, and also we can roughly tell that if it is a strong indicator of a flight delay. Due to the high dimension of feature data, I plotted them individually in each figure.

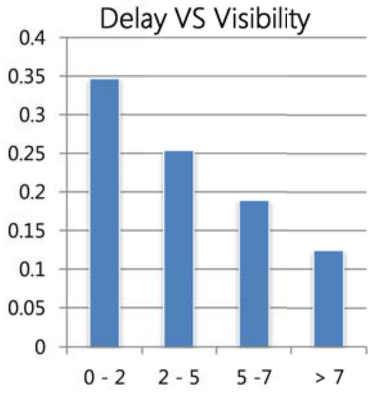
According to RITA's study, air carrier is one of the main reasons for delay. Although we have 15 carriers in our data set. For space reason, only the top 5 carriers with most delay and the top 3 with least delay are shown in the figure below. The worst airlines for flight delay are Frontier, Atlantic, JetBlue, Alaska, and United, while the best airlines are Hawaiian, Southwest and AirTran. This result is very similar to the statistics that I can find from other sources [4].

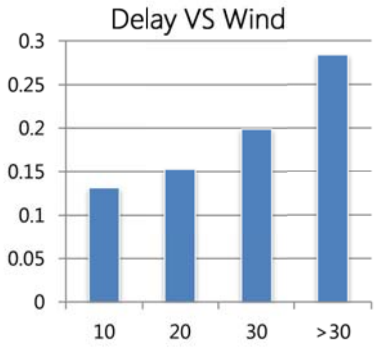


From our intuition, airport should be another factor that affects the flight delay a lot, because larger airport tends to have heavier traffic thus is more likely to cause delay. Our data verifies this assumption that busier airport have higher delay rate, but on the other hand, the worst airports in our data set are not one of the busiest airports in the US, such as New York Airport (JFK) or Los Angeles Airport (LAX).



In addition, even common sense can tell us that bad and extreme weather contributes greatly to flight delay. From the historical weather database, I picked three features that have strong tie with delay: visibility (mile), wind speed (mph) and weather condition (such as rain, snow, cloudy, or hail). The following two figures show the relationship between flight delay versus visibility, and relationship between the delay percentages versus wind speed. For these two weather features, the relationship is almost perfectly linear.





On the other hand, however, two features surprisingly show no indication of flight delay. Day of week (Monday, Tuesday…) is initially selected due to the reason that the air traffic should be heavier during weekend, which is not, however, supported by the data. Also the flight distance is another feature that does not show any pattern and is finally removed from the feature set.

Model 1: Naïve Bayes

The first model I built was Naïve Bayes, since it is the most straightforward model and easy to be implemented.

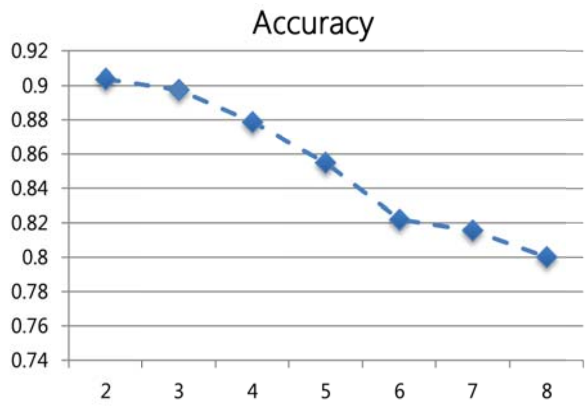
In order to build the model, I first discretized some continuous valued input, such as visibility and wind speed. After some experiments, the buckets that achieve best result are shown below, which is the same as what is used as variables in the figures in the last section:

Visibility (Mile)	0-2	2-5	5-7	>7
	1	2	3	4

Wind (MPH)	0-10	10-20	20-30	>30
	1	2	3	4

There are only 2 classes in output, delayed or not delayed. So for training data, delayed minutes will be discretized using the official definition: 15 minutes or more is considered as delayed.

After tuning the model, the best result from 5-fold cross validation is 90.37%. For study purpose I also did some experiment on multi-class model, because it is also interesting to see how the model performs when we want it to predict the delay time more precisely, such as delayed 30 minutes or 60 minutes instead of just two categories: delayed or not. As shown in the figure below, with more number of output buckets, the performance becomes lower.



Model 2: Logistic Regression

As discussed in class, Naïve Bayes model makes a very strong assumption that x's are conditionally independent given y, which is also known as Naïve Bayes assumption. In our data, however, we are not confident about it, especially on weather data. For example, foggy days always have lower visibility than sunny days, so they are obviously not independent features. Moreover, since we have enough data, we don't have to make any other strong assumptions like $p(x|y)$ follows a Gaussian distribution, therefore, logistic regression should be more robust. As a second exercise, I developed a logistic regression model with stochastic gradient ascent rule to see if this model can provide us better prediction over Naïve Bayes.

It turned out that the accuracy from model 2 is 91.88%. It indeed is better than that from Naïve Bayes model, although the improvement is not as large as I expected. We can also safely draw the conclusion that not all the features in our feature set are perfectly, mutually independent. This also leads me to look for a better model to describe the data.

Model 3: Nonlinear with Kernel Trick

One of the limitation of logistic regression model is that its decision boundary is still linear. We can add some extra features and use a high order polynomial, but there are two difficulties. First, it takes too much time to generate new features due to the high volume of the training data set. Second, we are not sure about what high order polynomial is the best fit.

Kernel trick is a very powerful tool in machine learning domain, and more important, it can solve the two difficulties when using high order polynomial. So in the third model, I'd like to build a non-linear regression model with kernel trick.

From Problem Set 2, we derived an approach to kernelize the perceptron algorithm. Similarly, we will consider a stochastic gradient descent-like implementation of the classic linear algorithm where each update to the parameters θ is made using only one training example.

It is also worth mentioning that the first two models are classification models, while this model is a regression model which has continuous valued output. So when we get the continuous-valued late minutes, we can tell the flight is delayed (greater than 15) or not. At the same time we can easily evaluate how bad the delay is. After all, delay 20 minutes or delay 3 hours is totally different experience.

From the classic linear regression model, the update rule is:

$$\theta^{(i+1)} := \theta^{(i)} + \alpha[y^{(i+1)} - \theta^{(i)} \cdot \phi(x^{(i+1)})]\phi(x^{(i+1)})$$

Since $\theta^{(0)} = 0$

Then

$$\theta^{(i)} = \sum_{l=1}^{i} \beta_l \phi(x^l)$$

The prediction step is

$$y = \theta^{(i)}\phi(x^{(i+1)}) = \sum_{l=1}^{i} \beta_l K(x^l, x^{i+1})$$

Despite the high dimension of θ, we don't have to update θ explicitly; we just need to keep tracking the value of β_i

$$\beta_{i+1} = \alpha[y^{(i+1)} - \sum_{l=1}^{i} \beta_l K(x^l, x^{i+1})]$$

In this model, I used a degree-5 polynomial kernel. The above iteration keeps running until each training data is passed into the training once.

The way I evaluate the performance of this model is to calculate the average difference between the actual delay and the prediction on the test data.

After tuning, the performance is very satisfying. The average error is 5.37 minutes which means the prediction is only about 5 minutes off the actual value on average.

Furthermore, when I convert the continuous valued prediction result into two categories as in the first two models, the accuracy (93.2%) is the highest among the three models, which is also an expected result.

Conclusion

Flight delay is directly related to airport location and weather condition thus can be predicted by machine learning algorithm. Naïve Bayes model performs OK considering its simplicity. Logistic Regression model is slightly better because it removes the independence assumption. Kernel based non-linear regression model has the best performance but time complexity is higher especially when there are many training data.

Reference

1. http://en.wikipedia.org/wiki/Flight_delay

2. http://www.transtats.bts.gov/homedrillchart.asp

3. http:/bts.gov/help/aviation/html/understanding

4. http://www.travelandleisure.com/articles/best-and-worst-airlines-for-delays