Flight Delay Prediction

Katherine Erdman (kerdman@stanford.edu)
Nicholas Benavides (nbenav@stanford.edu)

Introduction

Flight delays affect both passengers and airlines negatively. For passengers, delayed arrival is associated with increased stress, missed connections, inconvenience and potential expenses for food and hotel stays. In 2007, it's estimated that flight delays cost the airline industry $8.3 billion dollars on account of lost revenue from discontent passengers and increased costs due to scheduling changes that underutilized crew and planes[1]. The Federal Aviation Administration (FAA) updated their system, Time-Based Flow Management, which estimates flight delays, in 2013. This update was projected to save $88 million dollars for the airlines and $85 million dollars for the FAA by increasing prediction accuracy by just 2%[2]. Thus, small improvements in delay prediction can generate significant economic benefits. While even small improvements are impactful, they are difficult to achieve. Flight delays are often caused by variables that are hard to predict, such as maintenance issues. In addition, airlines have a business incentive to minimize the number of delays and, most likely, will have already mitigated any systemic patterns.

Related Work

Arrival delays have been modeled as a simple Normal or Poisson distribution based on aircraft and time-of-day for ten of the busiest airports in the United States by researchers at NASA Ames Research Center[3]. Specific to Newark airport, arrival delays have also been modeled by linear correlations based on weather conditions, specifically flight ceiling and visibility[4]. A more sophisticated model, which uses a combination of decision trees and neural networks, has been built to predict delays using current information from the airport as a whole, such as the total number of delayed passengers and the average flight delay[5]. What differentiates the work in this paper is the combination of data sources. Instead of solely relying on aircraft or weather information, these models have access to aircraft, flight and weather data. In addition, all of this information is available at least a few hours prior to take-off. As such, delays can be estimated, and ultimately planned for, in advance, unlike some previous work that relies on variables that are measured at the time of the flight, such as the current number of delayed passengers in the airport.

Task Definition

We will be predicting airline arrival delays, as well as understanding the impact of variables hypothesized to impact delays, such as weather, for flights departing Chicago O'Hare (ORD). Arrival delays will be predicted in three buckets: one for less than 0 minutes (early arrival), one for less than 15 minutes (on-time arrival) and one for greater than 15 minutes (delayed arrival).

---

[1] Ball, M., Barnhart, C., Dresner, M., Hansen, M., Neels, K., Odoni, A. R., ... & Zou, B. (2010). Total delay impact study: a comprehensive assessment of the costs and impacts of flight delay in the United States.

[2] Bellamy, W. (2013, August 29). FAA Deploys Next Generation Time-Based Air Traffic System.

[3] Mueller, E., & Chatterji, G. (2002). Analysis of aircraft arrival and departure delay characteristics. In *AIAA's Aircraft Technology, Integration, and Operations (ATIO) 2002 Technical Forum* (p. 5866).

[4] Allan, S. S., Beesley, J. A., Evans, J. E., & Gaddy, S. G. (2001, December). Analysis of delay causality at Newark International Airport. In *4th USA/Europe Air Traffic Management R&D Seminar* (pp. 1-11).

[5] Zonglei, L., Jiandong, W., & Guansheng, Z. (2008, December). A new method to alarm large scale of flights delay based on machine learning. In *Knowledge Acquisition and Modeling, 2008. KAM'08. International Symposium on* (pp. 589-592). IEEE.

Flight Delay Prediction

Katherine Erdman (kerdman@stanford.edu)
Nicholas Benavides (nbenav@stanford.edu)

Since a Support Vector Machine (SVM) ended up being our best model, we will describe the task formulated as an SVM. To create a multi-class SVM classifier, we create one binary SVM classifier for each class, and then compare the results between the various binary classifiers for each data point. The prediction generated by the multi-class classifier corresponds to the binary classifier that produces the lowest error for that data point.

Given training vectors $x_i \in \mathbb{R}^p$, i=1,…, n, in two classes, and a vector $y \in \{1, -1\}^n$, a binary SVM classifier solves the problem below[6]. In the case of this project, p = 251 and n = 70,800, or 80% of the 85,000 flights we used in our models.

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i$$

$$\text{subject to } y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i,$$

$$\zeta_i \geq 0, i = 1, ..., n$$

Input
Each flight, or training vector, is a sequence of 13 variables, as shown below. After converting the categorical variables to binary dummy variables, each flight is represented by 251 variables, which is the size of input given to all of the models with the exception of the Bayesian networks. The output for each flight is a single number (either 0, 1, or 2), which corresponds to a predicted class for the flight (early on-time, and late, respectively). We used 80% of our dataset of 88,500 flights for training, so we used 70,800 training vectors to train our models, resulting in the dimensions of the feature matrix being 70,800 x 251. The output had dimensions 70,800 x 1.

Month	**Day of Week**	**Destination**	**Airline**	**Scheduled Departure**	**Temperature**	**Humidity**
7	1	CLT	AA	7	3	3
Plane Year	**Model**	**Wind Speed**	**Visibility**	**Precipitation Probability**	**Precipitation Intensity**	
1994	757-2B7	6	5	1	1	

Output 0 (corresponding to delay < 0 minutes -- arriving early)

Approach

Data

We started with a dataset from Kaggle, which had information on more than 5 million domestic flights in the United States in 2015[7]. This dataset included features such as the month, day of the week,

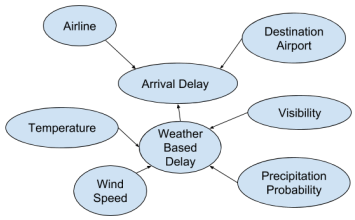
[6] http://scikit-learn.org/stable/modules/svm.html#svm-mathematical-formulation
[7] https://www.kaggle.com/usdot/flight-delays

destination airport, airline, and scheduled departure, in addition to other features that we did not use in our model. Then, we scraped the FAA registry to obtain the model and year for each plane and augmented our dataset with these features. Using on the latitude and longitude of the airport and the scheduled departure, we queried the Dark Sky API to obtain data about the weather at the time of takeoff. The features from Dark Sky included temperature, humidity, wind speed, visibility, precipitation probability, and precipitation intensity, as well as other features that we decided not to include in our models.

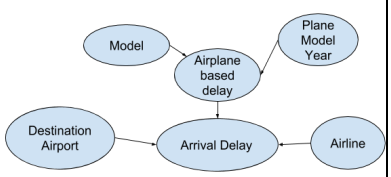
From the initial 5 million flights, we extracted a reasonable subset for training a Bayesian network, as well as other classifiers, that had 88,500 records. The origin airport was limited to Chicago O'Hare as it is one of the nation's busiest airports and susceptible to variables like weather. We also limited the airlines to the top five with the most flights out of Chicago O'Hare. The dataset was initially very skewed: only 20% of flights were delayed by more than 15 minutes. As a result, we further limited the data by sampling such that early, on-time and delayed flights occur with equal probability. From the limited dataset, 20% was reserved for testing.

Proposed Methods

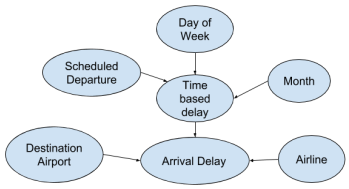
Multiple Bayesian networks were employed to predict arrival flight delay, as well as demonstrate the relationship flight delay has with factors like weather and airline.



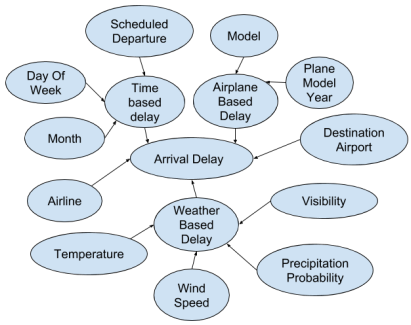
Model 1: Hidden weather delay node



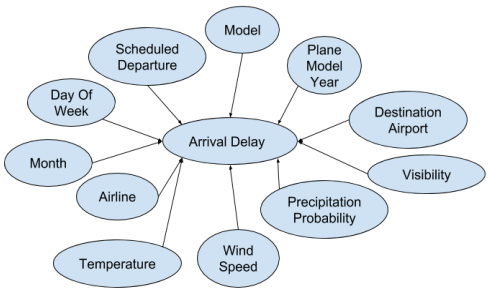
Model 2: Hidden airplane delay node



Model 3: Hidden time delay node



Model 4: Hidden time, airplane and weather delay nodes



Model 5: Flat Bayesian Network

Katherine Erdman (kerdman@stanford.edu)
Nicholas Benavides (nbenav@stanford.edu)

For the flat Bayesian network, the value of all nodes is known. Thus, to determine the probabilities, θs, for this network, maximum likelihood estimation with Laplace smoothing was implemented. For the Bayesian networks that had hidden variables, expectation maximization was used to determine the probabilities, θs, that included the hidden variables. θs that only relied on known variables were initialized to the value as determined by the training data. Hidden variables were randomly initialized.

Once the θs are determined, the arrival delay is predicted by the argmax P(Arrival Delay = d | E=e; θ) where the different values of arrival delay are 0, 1, and 2.

In addition to Bayesian networks, we predicted flight delay using SVM, Decision Tree, and Random Forest classifiers from the scikit-learn Python package. The SVM formulation is described in the task definition, and we chose to use an SVM classifier because SVM's tend to perform well in high-dimensional spaces and they are commonly used classifiers for machine learning problems.

A decision tree classifier takes training vectors $x_i \in R^n$, i=1,…, l and a label vector $y \in R^l$ as given and recursively partitions the feature space to group samples with the same label together[8]. The objective function for the decision tree is $\theta^* = \operatorname{argmin}_\theta G(Q, \theta)$, where

$$G(Q,\theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta)) + \frac{n_{right}}{N_m} H(Q_{right}(\theta))$$
$$Q_{left}(\theta) = (x,y)|x_j <= t_m$$
$$Q_{right}(\theta) = Q \setminus Q_{left}(\theta)$$

We chose to use a decision tree model to predict flight delays because they are fairly simple to interpret and can incorporate both numerical and categorical data, in addition to being a common and effective model in machine learning.

A random forest classifier generates multiple decision tree classifiers on different portions of the training dataset, using the mode predicted class of these decision tree classifiers to improve prediction accuracy and limit overfitting.[9] Thus, the formulation of a random forest is very similar to the formulation of the decision tree classifier described above. We chose to implement a random forest because we noticed that our decision tree was heavily overfitting on the training data, and we thought that a random forest would limit overfitting and perform better on our test set.

Evaluation

Baseline

Our baseline is a multi-class logistic regressor and solely uses flight information as features. Our oracle is also a multi-class logistic regression, which includes these features, as well as the departure delay,

[8] http://scikit-learn.org/stable/modules/tree.html
[9] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

aircraft information and weather data. Both predict the class of arrival delay, the difference between actual arrival time at the gate and expected arrival time.

Metric

Industry standard for accuracy based on 20-minute bucket intervals is around 70%. When there is a binary classification based on a delay threshold of 60 minutes, the accuracy is around 80%[10]. None of the literature that contained these industry standards included precision or recall. In comparison, our baseline has an accuracy of 40%. The precision and recall of the baseline is also only 40%. In comparison, the accuracy of our oracle is 65%. Similarly, both precision and recall for the baseline is 65%. Note that our baseline and oracle are based off of unskewed data: all three classes, early, on-time and delayed, are equally likely. Naturally, flight data is skewed such that on-time and early flights are three to four times more likely than delayed flights.

Experimental Procedures

Expectation maximization defined convergence as the difference between every θ before and after an iteration be less than 0.015. Alternatively, the algorithm terminated after 50 iterations. When training, none of the models reached convergence. Model 2 was run for 50 iterations and then 100 iterations, but the classification of training data did not change after more iterations of expectation maximization. As such, 50 was considered the standard for all other Bayesian models as well. The time delay variable could take on 5 different values, 0-4 inclusive. Similarly, the airplane delay variable and the weather delay variable could take on 5 different values, 0-4 inclusive.

Results

Table 1: Results of Bayesian Models for three-class output (early, on-time, delayed) on unbiased data

Bayesian Model	Accuracy	Precision	Recall
Model 1: Hidden weather delay node	34.9%	34.2%	40.1%
Model 2: Hidden airplane delay node	32.4%	33.7%	32.3%
Model 3: Hidden time delay node	34.6%	34.2%	34.1%
Model 4: Hidden time, airplane and weather delay nodes	33.4%	33.1%	33.4%
Model 5: Flat Bayesian Network	43.1%	42.9%	42.9%

Model 5, the flat Bayesian network, performed the best with an accuracy of 43% on the test data and a precision of 0.43. This is barely above baseline performance. Models 1, 2 , 3 and 4 actually performed below the baseline. The precision and recall for all of these models was around 0.34. We hypothesize that the unknown nodes which we add, the airplane delay node and the time delay node, abstract away

[10] Liu, Y. J., Cao, W. D., & Ma, S. (2008, October). Estimation of arrival flight delay and delay propagation in a busy hub-airport. In *Natural Computation, 2008. ICNC'08. Fourth International Conference on* (Vol. 4, pp. 500-505). IEEE.

too much. The minimal impact that the parents of those unknown variables can have on the classification is potentially lost when they do not directly affect the arrival delay.

Table 2: Results of other classifiers for three-class output (early, on-time, delayed) on unbiased data

Classifier	Accuracy	Precision	Recall
Baseline	40.3%	40.2%	40.3%
Logistic Regression	42.8%	42.5%	42.8%
Decision Tree	42.9%	42.9%	42.9%
Random Forest	43.9%	43.9%	43.9%
SVM	44.8%	44.7%	44.8%
Oracle	65.3%	65.1%	65.3%

Overall, the other classifiers had performance that is similar to the flat bayesian network, which has small improvement over the baseline. While small improvements can be impactful within the airline industry, these accuracies are far below the industry standard. However, the industry standard is probably based on realistic data, which is heavily biased towards on-time and delayed flights. Therefore, in order to get a more accurate comparison between our models and the industry standard, the top two models were re-trained on the biased dataset.

Table 3: Results of Random Forest and SVM for three-class output (early, on-time, delayed) on biased data.

Classifier	Accuracy	Precision	Recall
Random Forest	43.9%	43.9%	43.9%
SVM	55.8%	44.2%	55.8%

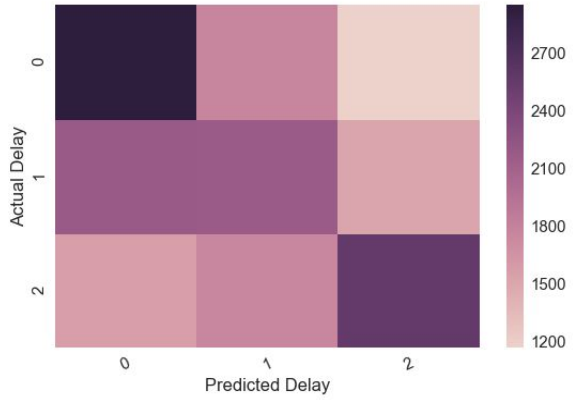
Even when the data is biased such that only 20% of flights are delayed, the best accuracy is 55.8%, far below the industry standard, which is around 70%.

Quantitative Analysis

To examine the predictions and errors made by our model, we plotted the confusion matrix as a heatmap, which is shown below.

Flight Delay Prediction

Katherine Erdman (kerdman@stanford.edu)
Nicholas Benavides (nbenav@stanford.edu)



From this, we see that our model was fairly good at identifying early flights as either early or on-time, with few early flights being predicted to be late. Similarly, our model did not predict that many late flights would be early. On-time flights were more difficult to predict, and our models had trouble distinguishing between early and on-time flights.

Given these results, we hypothesized that if we combined the early and on-time arrivals into a single class representing flights that were not late, then our models would perform significantly better due to the challenge of distinguishing between early and on-time flights. Binarizing our data in this fashion still kept the value of the task, as the primary benefit to airlines is identifying which flights will be late, not if a given flight will be early or on-time. After binarizing the dataset and resampling such that we had the same number of not-late (class 0) and late (class 1) flights, we obtained the following results.

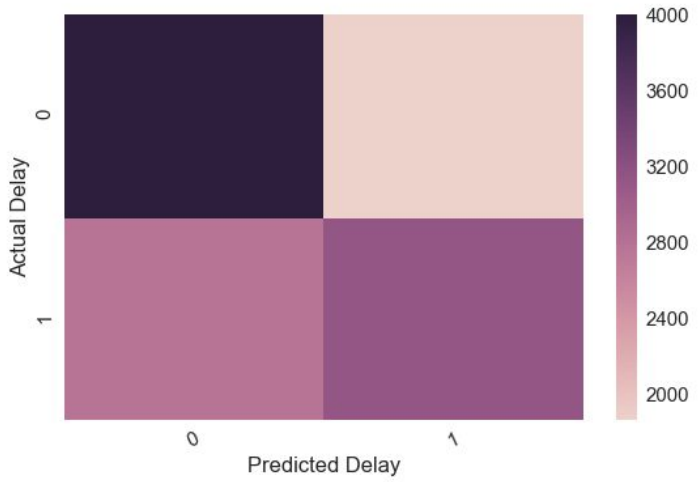
Table 4: Results of classifiers for binary output (not-late, late) on unbiased data

Binary Output (Not Late & Late)			
Model	**Accuracy**	**Precision**	**Recall**
Baseline	59.7%	40.3%	45.4%
Logistic Regression	63.9%	46.4%	48.1%
Oracle	86.0%	79.0%	78.6%
SVM	61.1%	63.5%	53.4%
Decision Tree	60.2%	60.6%	60.0%
Random Forest	60.9%	64.0%	51.7%

As predicted, the models trained on binarized data performed much better than the multiclass classifiers, since the challenge of distinguishing between early and on-time arrivals was eliminated. The SVM was still the best model considering its higher precision and recall than logistic regression, although the accuracy of the logistic regression model was slightly higher. Examining the confusion matrix for our SVM using binarized data, we found a fairly even distribution of errors.

Flight Delay Prediction

Katherine Erdman (kerdman@stanford.edu)
Nicholas Benavides (nbenav@stanford.edu)



While this model does predict that a flight will not be late more than half of the time, it is able to identify most of the late flights as late. Similar to the other models, flights that are not late are rarely predicted to be late, so most of the error of this model results from predicting late flights as not late.

Conclusion

Flights were classified into three categories: early, on-time and late, based on a multitude of Bayesian networks and other classifiers. Of the Bayesian networks that had hidden nodes, accuracy and recall were below the baseline. However, the flat Bayesian network, without any hidden nodes, performed above the baseline. This implies that there are not significant dependencies between features. While the flat Bayesian network was an improvement over the baseline, the improvement was minimal: accuracy increased from 40.3% to 43.1%. Therefore, other classifiers, such as logistic regression and random forests, were also trained to see if these models would have better performance. Of these classifiers, SVM had the best performance with an accuracy of 44.8% and a recall of 44.8%. While SVM was an improvement over the flat Bayesian network, none of these models performed close to the oracle, which had an accuracy of 65.3% and a recall of 65.3%. Though we did not to expect to find striking systemic patterns because airlines have a business incentive to minimize the number of delays and, most likely, will have already mitigated any patterns, we were surprised by the performance of the oracle.

The oracle, which knew the departure delay, the difference between the time that the flight should have left the origin airport and the time that the flight actually left, had a surprisingly low accuracy. It was a hypothesized that knowing the departure delay would inform the arrival delay, and while this was true, the oracle still performed below the industry standard of 70% classification accuracy for multiple buckets. However, when we binarized the dataset such that there were only two classes, delayed and not delayed, the oracle, performed above the industry standard of 80% classification accuracy.

One constraint on our models was data. There was obvious overfitting for some models, like random forests, which had a training accuracy of 99%. In order to have a reasonable number of variables for the Bayesian network, we limited the dataset to a single origin airport and five airlines. This allowed us to train our network with Laplace smoothing, but reduced our dataset to 90,000 rows. As this data was only from 2015, expanding the dataset to include flights in 2016 and 2017 from the same original airport and

on the same five airlines would minimally increase the number of values for variables, but greatly increase the dataset.

Delay propagation studies the dependency between a series of flights on the same aircraft. When a aircraft is scheduled for multiple sequential flights, a delay on one of those flights can propagate and affect later flights. By limiting the dataset to a single origin airport, delay propagation could not be accounted for, but would be interesting to study.

If we were to create a high-performing Bayesian network, the dependencies within the network can be examined to determine how these variables affect flight delays. Else, if other machine learning models are more accurate, those parameters can be examined and compared to the dependencies identified in academic literature.

Code

https://github.com/nnbenavides/CS221-Project