Evaluation of Algorithms for a Miles-in-Trail Decision Support Tool

Michael Bloem
Aviation Systems Division
NASA Ames Research Center
Moffett Field, CA, USA
Email: michael.bloem@nasa.gov

David Hattaway
Xbox Console Architecture
Microsoft
Mountain View, CA, USA
Email: dave.hattaway@microsoft.com

Nicholas Bambos
Department of Electrical Engineering and
of Management Science & Engineering
Stanford University
Stanford, CA, USA
Email: bambos@stanford.edu

***Abstract*—Four machine learning algorithms were prototyped and evaluated for use in a proposed decision support tool that would assist air traffic managers as they set Miles-in-Trail restrictions. The tool would display probabilities that each possible Miles-in-Trail value should be used in a given situation. The algorithms were evaluated with an expected Miles-in-Trail cost that assumes traffic managers set restrictions based on the tool-suggested probabilities. Basic Support Vector Machine, random forest, and decision tree algorithms were evaluated, as was a softmax regression algorithm that was modified to explicitly reduce the expected Miles-in-Trail cost. The algorithms were evaluated with data from the summer of 2011 for air traffic flows bound to the Newark Liberty International Airport (EWR) over the ARD, PENNS, and SHAFF fixes. The algorithms were provided with 18 input features that describe the weather at EWR, the runway configuration at EWR, the scheduled traffic demand at EWR and the fixes, and other traffic management initiatives in place at EWR. Features describing other traffic management initiatives at EWR and the weather at EWR achieved relatively high information gain scores, indicating that they are the most useful for estimating Miles-in-Trail. In spite of a high variance or "over-fitting" problem, the decision tree algorithm achieved the lowest expected Miles-in-Trail costs when the algorithms were evaluated using 10-fold cross validation with the summer 2011 data for these air traffic flows.**

***Keywords*— Miles-in-Trail; machine learning**

I. Introduction

Miles-in-trail (MiT) restrictions require that flights in a flow of air traffic crossing a certain point must be separated by a certain number of miles. These restrictions are one way that the volume of air traffic in airspace and at airports is maintained at a safe level. Traffic managers may set better MiT restrictions if they could consult an estimation of the MiT restrictions that should be put in place in the current conditions. Previous research by Evans et al. modeled how MiT restrictions are set and also the impact of MiT restrictions on flights [1]. The purpose of that MiT modeling was to enable simulations comparing MiT and "time-based metering," an alternative congestion-management technique [2]. The MiT prediction model developed by Evans et al. uses a variety of rules based on current operations and a single-feature linear regression to predict what MiT restrictions would be used in a given situation. The quality of the Evans et al. MiT prediction model was judged based on how well it predicts average aircraft delays when coupled with a queuing-based MiT impact model. While this is appropriate for the purpose of that research, it is unclear how well the Evans et al. MiT model predicts actual MiT restriction values. Furthermore, the Evans et al. model is not suited to real-time decision support because it makes use of future data that would not be available to traffic managers when they are setting MiT restrictions. Finally, a variety of machine learning techniques not evaluated in the development of the Evans et al. model have proven useful for other air traffic management research, as described in Section II, and might also be helpful for estimating MiT restrictions.

The objective of this project is to prototype and investigate algorithms that could be used in a tool to assist traffic managers by estimating what MiT restrictions should be put in place. This tool would not predict MiT in the future but rather estimate what MiT should be used now based on current conditions. Based on discussions with a traffic manager who sets MiT restrictions in airspace near New York City, we propose a tool that presents the traffic manager with the probability that each possible MiT value should be used in the current conditions. A sample of the type of information that would be displayed to traffic managers is shown in Table I. By assigning a high probability to an MiT value, the tool could indicate that it is very likely that this MiT value should be used in the current conditions. By assigning no single MiT value a high probability and instead assigning several MiT values roughly the same lower probability value, the tool could indicate that it is not clear which of these MiT values should be used in the current conditions. Although more input from potential tool users is needed to determine how useful this tool would be and how it could be improved, this paper proposes and evaluates algorithms that could be used in such a tool. The probability values presented to the manager could be computed by a machine learning algorithm that takes as inputs current air traffic demand, current weather conditions, current airport runway configurations, and other air traffic restrictions currently in use. In this research, four candidate algorithms for this purpose will be compared.

Applications of machine learning to air traffic management problems such as runway configuration selection and Ground Delay Program rate selection are discussed in Section II. The problem objective, which is to minimize an expected

TABLE I
SAMPLE TOOL OUTPUT: SUGGESTED MILES-IN-TRAIL.

Miles-in-Trail	Probability
No restriction	0.00
10	0.10
15	0.80
20	0.10

MiT cost, is defined in Section III. In Section IV, the four investigated algorithms are described and Section V discusses the Newark Liberty International Airport arrival flow data used to analyze the algorithms. Implementation details such as cost and algorithm parameter values are documented in Section VI. The results in Section VII describe how the algorithms perform with respect to the problem objective, as well as the results of some feature scoring and feature selection work. Finally, some possibilities for future work are described in Section VIII and conclusions are in Section IX.

II. APPLICATIONS OF MACHINE LEARNING TO AIR TRAFFIC MANAGEMENT

Machine learning has been applied to various problems related to Ground Delay Programs (GDPs). Wolfe and Rios use similarity measures from machine learning and other disciplines to find "relevant" historical days that might be useful for traffic managers as they decide how to use GDPs to manage congestion on a given day [3]. Work by Smith and Sherry attempts to assist traffic managers by using a Support Vector Machine (SVM) algorithm to predict future airport capacity from features that describe a weather forecast at the airport [4], [5]. Wang considered a similar airport capacity prediction problem and found that ensemble bagging decision trees out-performed SVMs [6]. Finally, Buxi and Hansen use a response surface methodology approach to produce a set of possible airport capacity profiles with associated probabilities from a weather forecast for an airport [7]. The forecasts were clustered, a set of capacity profiles and corresponding probabilities were generated for each cluster, and these were then used for all days with forecasts that fell into the cluster. This set of capacity profiles and corresponding probabilities served as inputs when solving a stochastic optimization problem instance from [8] to determine how to set GDP parameters. This approach performs better than an approach proposed by Liu et al. that does not use weather forecasts but instead clusters historical airport capacity profiles to generate the capacity profiles; profile probabilities were set equal to the historical frequencies in that work [9].

Machine learning algorithms have also been applied to the problem of predicting airport runway configurations. Wang found that ensemble bagging decision tree models outperformed SVMs when predicting runway configurations (just as they did when predicting airport capacities) [6]. Ramanujam and Balakrishnan used historical data to tune a discrete-choice model that predicts runway configuration changes [10]. This model also identifies the importance of factors in runway configuration decision-making.

Pfeil and Balakrishnan apply machine learning algorithms like SVMs, ensembles of SVMs, decision trees, and random forests to the problem of predicting the likelihood that a terminal-area route will be blocked by weather based on features describing forecasted weather [11]. This information is then used to increase the number of available routes by moving arrival and departure routes.

III. PROBLEM OBJECTIVE

The objective of this problem is to minimize the expected value of a MiT cost, assuming that the traffic manager selects MiT restrictions according to the probabilities presented by the tool (as in Table I). Let Y denote the possible MiT values; in the example in Table I, $Y = \{0, 10, 15, 20\}$. The current air traffic demand, current weather conditions, current airport runway conditions, and other air traffic restrictions currently in use are quantified in x, a $n \times 1$ vector of features. The proposed tool requires an algorithm that outputs $p(y = \hat{y}|x)$ for each $\hat{y} \in Y$, the probability that MiT level $\hat{y}$ is the ideal MiT level y in the current conditions. For example, $p(y = 15|x) = 0.80$ for the example in Table I. Algorithms learn how to estimate these probabilities from feature data by making use of a training data set containing samples of features describing a situation $x^{(i)}$ and a corresponding ideal MiT value $y^{(i)}$. The ideal MiT achieves the appropriate balance between safety and delay; the lack of historical ideal MiT value data is discussed in subsection V-A.

For a data set $\left\{(x^{(i)}, y^{(i)})\right\}_{i=1}^{m}$, the expected MiT cost is

$$\begin{aligned} J &= \mathbf{E}\left[\sum_{i=1}^{m} C(\hat{y}, y^{(i)})\right] \\ &= \sum_{i=1}^{m} \sum_{\hat{y}\in Y} p(y = \hat{y}|x^{(i)}) C(\hat{y}, y^{(i)}), \end{aligned} \tag{1}$$

where $C(\hat{y}, y^{(i)})$ is the MiT cost when the MiT $\hat{y}$ is used and when the ideal value of the MiT is $y^{(i)}$. No MiT restriction is treated as if the MiT were 0, even though aircraft are required to remain 5 miles apart at all times. This imposes additional penalties for cases when the ideal MiT is to have no restriction but an MiT restriction is used and vice versa.

The MiT cost is

$$C(\hat{y}, y^{(i)}) = \beta_{\text{safety}}[y^{(i)} - \hat{y}]_{+} + \beta_{\text{delay}}[\hat{y} - y^{(i)}]_{+}, \tag{2}$$

where $[a]_{+}$ equals a if $a \geq 0$ and equals 0 otherwise. Here the first term is the safety cost and β_{safety} is the cost per mile-in-trail under the ideal MiT. Using too few MiT can lead to unsafe congestion. Similarly, the second term is the delay cost and β_{delay} is the cost per mile-in-trail above the ideal MiT. Using too many MiT imposes unnecessary delays on flights. Tool users could tune the ratio of these parameters in order to achieve an appropriate tradeoff between these safety and delay costs. Even different functional forms could be used for the cost function, but only this simple form was studied in this research. A sample MiT cost is shown in Fig. 1.

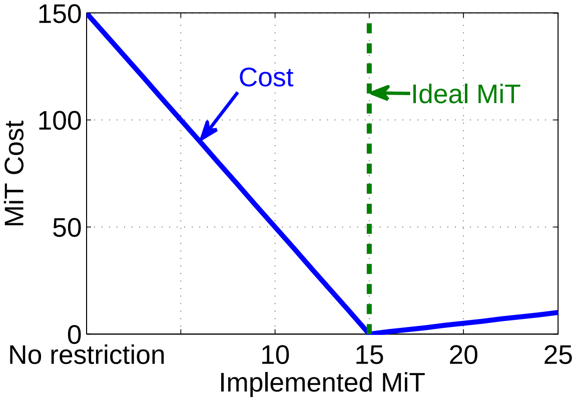


Fig. 1. Sample MiT cost when $\beta_{\text{safety}} = 10$, $\beta_{\text{delay}} = 1$, and the ideal MiT is 15.

IV. Algorithms

A. Classification Algorithms

SVM, decision tree, and random forest classification algorithms were investigated for this problem. The SVM algorithm was implemented with a radial basis function kernel. Multi-class estimation is achieved with the "one-against-one" method, in which $k(k-1)/2$ binary classifiers are built, one for each pair of the k classes. If the tool required the estimation of a single MiT value instead of estimates of probabilities that each possible value is ideal, the MiT value output by this SVM algorithm would be the one that is estimated most often when all of these $k(k-1)/2$ binary classifiers are run. Cross validation is used to enable the SVM algorithm to output ideal MiT value probabilities.

The decision tree split nodes by selecting the split for a single feature that produces the highest information gain ratio. No tree pruning techniques were used. The probability of each MiT value output by the decision tree is the fraction of each MiT value in the leaf corresponding to the feature data. The random forest algorithm consists of 100 decision trees trained with different bootstrap samples (random samples drawn with replacement from the training dataset) and in which the best feature to split each node is selected from a random sub-set of the features. The probability of each MiT value is the average of the probabilities returned by each of the trees in the forest.

B. Modified Softmax Regression and Stochastic Gradient Descent

The classification algorithms described in sub-section IV-A do not know the objective function for this problem, so it seems possible that an algorithm that explicitly reduces the objective function would perform better with respect to the objective function. To investigate this possibility, a modified softmax regression algorithm was implemented and its parameters were tuned with gradient descent to explicitly reduce the expected MiT cost objective function.

In this algorithm, the model for $p(y = \hat{y}|x)$ is parametrized by θ, which is made up of a $n \times 1$ vector θ_y for each $y \in Y$. The assumed model for $p(y = \hat{y}|x)$ is denoted as $h_\theta(\hat{y}, x)$ and equal to

$$p(y = \hat{y}|x; \theta) = h_\theta(\hat{y}, x) = \frac{\exp(\theta_{\hat{y}}^T x)}{\sum_{y' \in Y} \exp(\theta_{y'}^T x)}. \tag{3}$$

Given this model for the probability of each MiT value $\hat{y} \in Y$, the expected MiT cost is

$$J(\theta) = \sum_{i=1}^{m} \sum_{\hat{y} \in Y} \frac{\exp(\theta_{\hat{y}}^T x^{(i)})}{\sum_{y' \in Y} \exp(\theta_{y'}^T x^{(i)})} C(\hat{y}, y^{(i)}). \tag{4}$$

The gradient of this objective function with respect to each θ_y vector is

$$\begin{aligned}
\nabla_{\theta_y} J(\theta) &= \sum_{i=1}^{m} \sum_{\hat{y} \in Y} C(\hat{y}, y^{(i)}) \frac{\partial}{\partial \theta_y} h_\theta(\hat{y}, x) \\
&= \sum_{i=1}^{m} \sum_{\hat{y} \in Y} C(\hat{y}, y^{(i)}) \Bigg[1\{y = \hat{y}\} h_\theta(\hat{y}, x) \left(1 - h_\theta(\hat{y}, x)\right) \\
&\qquad + 1\{y \neq \hat{y}\} \frac{-\exp((\theta_{\hat{y}} + \theta_y)^T x)}{\left(\sum_{y' \in Y} \exp(\theta_{y'}^T x)\right)^2} \Bigg] x^{(i)},
\end{aligned} \tag{5}$$

where $1\{a\}$ evaluates to 1 when a is true and 0 when a is false. This gradient can be used by a gradient descent algorithm to update θ in a way that reduces the objective function. A gradient descent update rule for the k^{th} update of θ_y is, for each $y \in Y$,

$$\theta_y[k+1] := \theta_y[k] - \alpha[k] \nabla_{\theta_y} J(\theta[k]), \tag{6}$$

where $\alpha[k]$ is a learning rate parameter that can vary with k. The objective function is not convex, so a gradient descent algorithm may only converge to locally optimal values for θ.

V. Data

The flows selected for this study are those on the arrival routes into Newark Liberty International Airport (EWR) from the south over the ARD fix, from from the west over the PENNS fix, and from the north over the SHAFF fix. These routes are depicted in Fig. 2.

For each of these flows, historical MiT value and feature data were parsed and loaded into a table with more than 44,000 rows, one for every 5-minute interval in the summer months of May–September, 2011. The MiT values were from a database containing National Traffic Management Log (NTML) data. While [15] points out that NTML data is not always accurate, it was the only available source of historical MiT values. The distributions of MiT values for each fix are shown in Table II. Many of the intervals are in periods of good weather or in the middle of the night when there is low traffic volume. Accordingly, just over 5,000 of these 44,000 intervals (around 12% of them) have any MiT restriction for each fix. There were 3 or 4 possible MiT values for each fix, some used in less than 1% of the intervals and others used in as many as 10% of the intervals.

The features used in this study were selected based on previous research (such as [1]) and discussions with a traffic

TABLE III
FEATURES.

Feature Description	Data Source
Ceiling at EWR	ASPM
Meteorological condition at EWR (“Instrument” or “Visual”)	ASPM
Visibility at EWR	ASPM
Windspeed at EWR	ASPM
Runway configuration at EWR	ASPM
Ground Delay Program (GDP) planned in future at EWR	NTML
GDP in place at EWR	NTML
Planned remaining minutes of GDP at EWR	NTML
Duration of GDP so far at EWR	NTML
Airport rate for GDP at EWR	NTML
Ground Stop (GS) in place at EWR	NTML
Planned remaining minutes of GS at EWR	NTML
Duration of GS so far at EWR	NTML
Scheduled arrivals at EWR	ASPM
Scheduled plus queued arrivals at EWR	ASPM
Scheduled number of flights over ARD fix	ASDI
Scheduled number of flights over PENNS fix	ASDI
Scheduled number of flights over SHAFF fix	ASDI

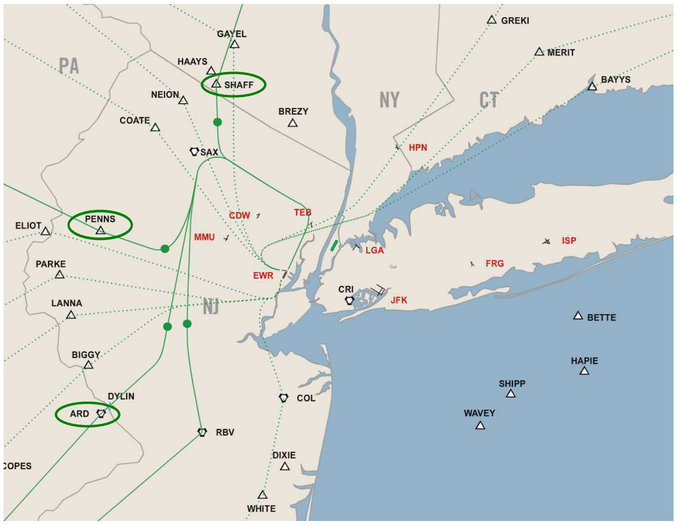


Fig. 2. Arrival and departure routes for EWR for a particular runway configuration. Arrival routes into EWR are depicted with solid green lines. Departure routes out of EWR are dashed green lines. The ARD, PENNS, and SHAFF fixes are circled. This figure was generated with the “N90 flows visualizer” [14].

TABLE II
DISTRIBUTIONS OF MIT VALUES.

Fix	No restriction	MiT=10	MiT=15	MiT=20	MiT=25
ARD	88.45%	0.20%	10.26%	0.96%	0.12%
PENNS	88.54%	0.33%	10.07%	1.06%	0.00%
SHAFF	87.75%	1.22%	9.10%	1.93%	0.00%

manager who sets MiT for traffic flows including those considered in this study. The number of features used in this study was 18, although some algorithms required slightly different sets of features. These features include current EWR weather conditions, scheduled and queued arrival traffic demand at EWR, scheduled traffic demand at each fix, the current EWR runway configuration, and data related to other traffic management initiatives currently in place at EWR. Of these features, 12 were continuous-valued and 6 were discrete or enumerated. Discretized versions of the continuous-valued features were created and added to the feature set to enable algorithms that require discrete features to make use of these features and also to see if such features might be more useful to the algorithms. After the discretized versions of continuous-valued features were added, there were 30 features, so $n = 30$ for the random forest, SVM, and decision tree algorithms. The modified softmax algorithm could not use one enumerated feature (runway configuration at EWR) and did not use the discretized features, so $n = 17$ for that algorithm. Table III shows the 18 features and the data source for each feature. Data were acquired from the FAA’s Aviation System Performance Metrics (ASPM) data, the National Traffic Management Log (NTML), and the Aircraft Situational Display to Industry (ASDI). Flight plans from ASDI were simulated with an air traffic management simulator (FACET) to determine the scheduled number of flights at the fixes [16].

A. *Ideal vs. Historical Miles-in-Trail*

There is no record of the ideal MiT value that achieves the appropriate balance between safety and delay in each time interval. Therefore, the algorithms were trained with the historical MiT values. If these historical MiT values are not ideal, then any machine learning algorithm trained with them will be biased accordingly.

Three actions that might help resolve this issue are discussed here. With additional subject-matter expert input, it may be possible to build a data set with MiT values that are closer to ideal. A second resolution involves assuming that the historical MiT values are an unbiased noisy sample of the ideal MiT. If this is the case, we still may be able to achieve and even guarantee good expected MiT cost performance with respect to the ideal MiT, given enough training data. A third and final resolution would involve adjusting the objective function to reflect known problems with historical MiT restrictions. For example, if historical MiT were typically overly restrictive, the objective function could assume an ideal MiT of 5 miles less than the historical MiT. Then the algorithm would learn to set MiT that are less restrictive than historical MiT. This topic should be investigated in future research.

VI. Implementation Details

For this study, $\beta_{\text{safety}} = 10$ and $\beta_{\text{delay}} = 1$, reflecting a higher cost for unsafe congestion than for unnecessary delays. The shape of the MiT cost with these values for the β parameters is shown in Fig. 1. Determining an appropriate value for the ratio of these parameters is a topic for future research.

The Orange machine learning software package was used to train and test these algorithms, except for the modified softmax regression algorithm [17]. For the SVM algorithm with the radial basis function kernel, the kernel weighting parameter γ was set equal to $\frac{1}{n}$. The SVM was ℓ_1-regularized with a weight $C = 1$ on the regularization term.

Custom code was developed to implement, train, and test the modified softmax regression algorithm. Before running gradient descent, the feature data were shuffled and normalized to have zero mean and unit variance. The parameters were tuned with a stochastic gradient descent algorithm because stochastic gradient descent performed better than batch gradient descent on this problem. The stochastic gradient descent update rule for the k^{th} update of θ is the same as in (6) except that $\nabla_{\theta_y} J(\theta[k])$ is computed with just a single data point instead of with the sum over m in (5). Various values for the learning rate parameter and the number of iterations through the data were investigated and values that consistently performed well for this problem were used. The iteration in (6) was applied until each data point was visited 5 times. The learning rate parameter $\alpha[k]$ was set to $\frac{1}{500+k^{1.5}}$. Techniques for improving gradient descent, such as those described in [12] and [13], may lead to better results and could be studied in future research.

VII. Results

A. Feature Scoring

We used information gain (IG), a measure of the expected decrease of the entropy in the MiT values when conditioning on each input feature, to score the discretized features. The IG for the j^{th} feature is computed as

$$\text{IG} = -\sum_{y \in Y} \bar{p}_y \log \bar{p}_y - \sum_{x \in X_j} \bar{p}_x \left[-\sum_{y \in Y} \bar{p}_{y|x} \log \bar{p}_{y|x} \right], \quad (7)$$

where X_j is the set of possible values for the j^{th} feature, $\bar{p}_y$ is the frequency of MiT value y in the data set, $\bar{p}_x$ is the frequency of the value x for the j^{th} feature in the data set, and $\bar{p}_{y|x}$ is the frequency of MiT value y in entries in the data set where the j^{th} feature takes the value x. The entropy of the MiT values in the data (the first term in (7)) helps interpret the IG scores. The entropy is 0.407, 0.406, and 0.463 for the MiT values on the ARD, PENNS, and SHAFF fixes, respectively. IG scores quantify expected entropy decreases, so these entropy values provide an upper bound on IG scores for each fix. Relatively large IG scores mean that conditioning on the value of a feature is expected to produce a relatively large reduction in the entropy of the MiT values, suggesting that this feature might be useful for estimating MiT. Negative IG scores mean that conditioning on the value of a feature actually increases the entropy in the MiT values, suggesting that knowing the value of this feature will not help when estimating the MiT value.

Table IV shows the IG for each feature for the estimation of MiT at each fix. When the features are ranked by their IG for the SHAFF fix MiT (as they are in Table IV), features related to GDPs or GSs make up the top 4 features and 6 of the top 10 features. IG results for the other two fixes are similar. Features related to weather at EWR make up the other features in the top 10 for the SHAFF fix. Features related to traffic demand at EWR or at the fixes have relatively low IG scores, which is surprising because these features were found to be most useful for MiT prediction in previous research [1].

The IG ranking of features is consistent with how an FAA employee who sets MiT for EWR and nearby airports described the MiT selection process to us. A GDP or GS may be used when severe congestion is expected at the airport. According to this employee, if the congestion is severe enough to require a GDP or GS, it also may necessitate MiT to further reduce the demand from airborne flights, which are not impacted by GDPs or GSs. This could explain the high IG scores for features related to GDPs and GSs. Severe weather leads to reductions in airport capacity that are considered when MiT are set, so it is not surprising that weather-related features have high IG scores. The low IG scores for demand-related features are somewhat surprising, as scheduled demand was described as relevant to MiT restriction decisions. The demand features used in this model are not relative to capacity; such relative demand values may have higher IG scores.

B. Learning Curves

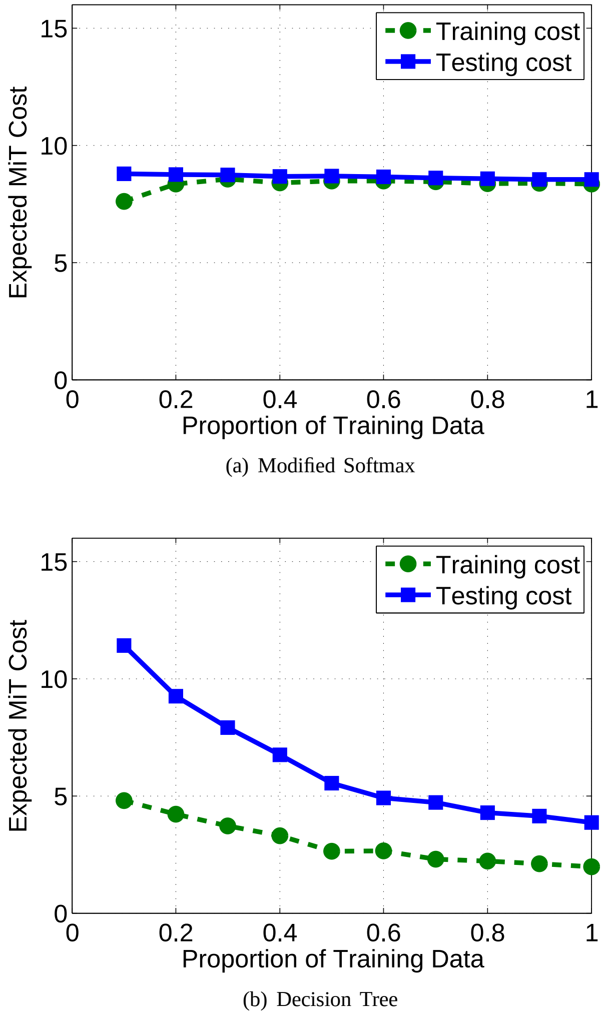
Learning curves were generated for the algorithms to help understand their performance on this problem. To generate these curves, a randomly-selected 70% of the summer data was used for training and the other 30% of the summer data was used for testing. The algorithms were given larger and larger proportions of the training data set and then tested on the test data set. Sample learning curves for the modified softmax and decision tree algorithms when estimating the MiT for the SHAFF fix are shown in Fig. 3.

Although it achieves low expected MiT costs on the test data set, the decision tree algorithm suffers from high variance (or "over-fitting"), as shown by the gap between training cost and test cost curves. This gap exists both when evaluating with the expected MiT cost and when evaluating with classification accuracy.

The training costs and test costs for the modified softmax algorithm, on the other hand, are nearly identical once the algorithm has been trained on just $\frac{3}{10}$ of the training data set, indicating that it does not suffer from high variance. This low variance may be caused in part by the relatively small number of parameters that need to be tuned for this algorithm. With few parameters, it may be difficult or unlikely

TABLE IV
FEATURE INFORMATION GAIN.

Feature Name	ARD	PENNS	SHAFF
Airport rate for GDP at EWR	0.290	0.277	0.272
Duration of GDP so far at EWR	0.210	0.164	0.210
GS in place at EWR	0.199	0.201	0.201
Duration of GS so far at EWR	0.211	0.213	0.197
Windspeed at EWR	0.199	0.197	0.195
Planned remaining minutes of GS at EWR	0.193	0.195	0.194
Ceiling at EWR	0.204	0.218	0.192
Visibility at EWR	0.185	0.174	0.172
Planned remaining minutes of GDP at EWR	0.181	0.167	0.170
Runway configuration at EWR	0.196	0.170	0.161
Scheduled arrivals at EWR	0.162	0.121	0.148
Scheduled plus queued arrivals at EWR	0.114	0.040	0.101
Meteorological condition at EWR (“Instrument” or “Visual”)	0.081	0.090	0.076
GDP in place at EWR	0.016	0.016	0.015
Scheduled number of flights over PENNS fix	0.014	0.001	0.005
Scheduled number of flights over SHAFF fix	-0.005	-0.014	-0.015
GDP planned in future at EWR	-0.027	-0.027	-0.027
Scheduled number of flights over ARD fix	-0.062	-0.063	-0.062



(a) Modified Softmax

(b) Decision Tree

Fig. 3. Learning curves for MiT estimation at the SHAFF fix.

that over-fitting will occur. The low variance may also be caused in part by the custom gradient descent algorithm that was developed to quickly tune the parameters on this type of problem. The modified softmax algorithm incurs considerably higher expected MiT costs on the test data set than the decision tree algorithm, indicating that it suffers from high bias.

C. *Feature Selection*

We pursued forward search feature selection in an attempt to resolve the variance problem for the decision tree algorithm. At each iteration of the feature selection process, the feature that led to the largest decrease in the expected MiT cost on a test data set was added to the feature set. The feature selection stopped after 15–19 of the 30 features were added to the set, depending on the fix, because the expected MiT cost was no longer decreasing with additional features.

Even though the forward search feature selection evaluated feature sets based on the expected MiT cost, the decision tree made best use of the features with high information gain scores (see Table IV). This is expected, as the information gain ratio used for splitting tree nodes is closely related to information gain.

Ultimately, feature selection did not resolve the decision tree’s high variance problem. Tree pruning techniques may reduce its variance and should be studied in future work. Also, the variance issue may be resolved with a larger training data set.

D. *Final Algorithm Evaluation*

Finally, the four algorithms were evaluated by running 10-fold cross validation with the summer data. In cross fold validation, the available data are repeatedly broken into different training and testing data sets and the reported algorithm performance is the average of the performance across all the testing data sets. These cross fold validation training and testing data sets are a different way of making use of the summer data than the 70%-30% split used to generate learning curves in sub-section VII-B. The expected MiT cost and accuracy results for each fix resulting from this cross fold validation are in Table V and VI, respectively.

TABLE V
EXPECTED MIT COST CROSS VALIDATION RESULTS.

Algorithm	ARD	PENNS	SHAFF
Random Forest	9.29	9.33	9.78
SVM	8.80	8.88	8.87
Modified Softmax	8.09	8.11	8.54
Decision Tree	3.12	3.10	3.18

TABLE VI
ACCURACY CROSS VALIDATION RESULTS.

Algorithm	ARD	PENNS	SHAFF
Random Forest	0.946	0.946	0.940
SVM	0.954	0.951	0.948
Modified Softmax	0.826	0.813	0.810
Decision Tree	0.968	0.968	0.966

The tree algorithm achieves the lowest expected MiT cost and highest accuracy for each fix by a considerable margin, so it seems to be the most promising algorithm for the proposed MiT decision support tool. The modified softmax algorithm achieves the next-best performance on the expected MiT cost, even with lower accuracies than the other algorithms. It achieves low expected MiT cost and low accuracy because it tends to estimate relatively high probabilities for higher MiT values. Estimating too many MiT is much cheaper than estimating too few MiT because $\beta_{\text{safety}} > \beta_{\text{delay}}$ in the expected MiT cost in (2). The modified softmax algorithm can take advantage of that because it uses the gradient to explicitly reduce the expected MiT cost. If the ratio of the β parameters were set differently or even if the cost function were changed to a completely different form, the parameters of the modified softmax algorithm would be adjusted accordingly by the stochastic gradient descent parameter tuning. The other algorithms are not aware of the relative costs of various types of wrong estimations and so cannot take advantage of these cost differences. Fig. 4 shows the weighted expected MiT cost breakdown for each algorithm for the SHAFF fix. The tendency of the modified softmax algorithm to avoid relatively expensive safety cost in favor of relatively cheap delay cost can be seen in the relatively large fraction of the total cost that is delay cost for the modified softmax algorithm.

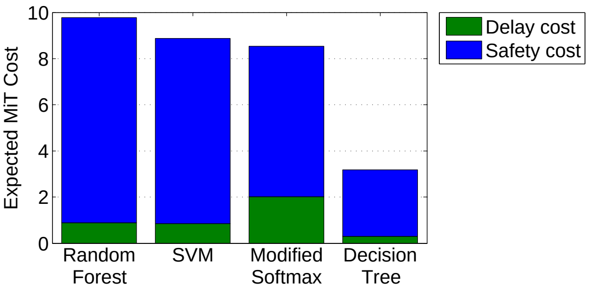


Fig. 4. Expected MiT cost, broken into weighted delay and safety costs, for estimations of MiT at the SHAFF fix.

VIII. FUTURE WORK

There are many possible extensions for this work. Feedback from more traffic managers should be used to determine if the proposed tool will be useful and whether or not the investigated algorithms are achieving acceptable performance. For example, do managers prefer to see the probability that each MiT value should be used in the current situation, or would a single determinstic suggestion be more useful? Would suggestions for future MiT values be more useful than estimates of the ideal MiT in the current situation? Do the probabilities suggested by the tool change too dramatically over time to be useful? The cost function form and parameters should also be set with traffic manager input.

Some features that traffic managers describe as important to how they set MiT are still not in the data set, such as weather forecasts, scheduled demand relative to expected capacity, and the time of the day, and algorithm performance may improve if these are considered. MiT restrictions for each of these fixes are primarily used to reduce congestion at EWR, so improvement may also be possible by estimating the MiT values for these fixes simultaneously, as was done in [1]. Pruning or a larger training data set may reduce the decision tree algorithm's high variance. Resolving the lack of ideal MiT value data should be investigated. Further fine-tuning might improve the performance of the stochastic gradient descent algorithm used for tuning the softmax regression parameters. New algorithms such as neural networks could be considered, and algorithms other than softmax regression may be modified so that they explicitly consider the expected MiT cost. Online algorithms would allow the tool to update its behavior based on new data. Reinforcement learning algorithms could consider the operational cost of changing MiT restrictions and respond to traffic manager feedback. Finally, the algorithms should be tested in other regions of airspace and airports where MiT are used.

IX. CONCLUSIONS

Miles-in-trail (MiT) restrictions help maintain safe air traffic volumes in airspace and at airports. A decision support tool that would output a probability that each of a set of possible MiT values should be used in a particular situation is proposed, and four algorithms that could be used in such a tool were prototyped and evaluated. The algorithms were evaluated based on an expected MiT cost that can impose different penalties for setting MiT too high or too low. Basic Support Vector Machine, random forest, and decision tree classification algorithms were prototyped, and a softmax algorithm was modified to explicitly reduce the expected MiT cost. The algorithms were evaluated with data collected from the summer of 2011 for flows of traffic bound to the Newark Liberty International Airport (EWR) over the ARD, PENNS, and SHAFF fixes. From NTML, ASPM, and ASDI data, 18 features were computed for each 5-minute time interval during this period. These features were used by the algorithms to estimate the probability that each possible MiT value was ideal for the current situation. The features describe the weather at EWR, the runway configuration at EWR, the scheduled traffic demand at EWR and the fixes, and current traffic management initiatives at EWR like Ground Delay Programs and Ground Stops. The features that describe other traffic management initiatives at EWR and the weather at EWR achieve higher information gain scores than the features describing the scheduled demand, indicating that they are more useful for estimating MiT values. In spite of a high variance (or "over-fitting") problem, the decision tree algorithm achieved considerably lower expected MiT costs than the other algorithms when they were evaluated using 10-fold cross validation with data from the summer of 2011. These results suggest that the decision tree algorithm would be the best choice for the proposed MiT decision support tool, but more research and testing should be done to confirm this conclusion.

Acknowledgements

We are thankful to the many individuals who have helped us with this project. Shon Grabbe suggested this research, and he along with William Chan, Kapil Sheth, Joey Rios, Yao Wang, and Shawn Wolfe provided helpful early feedback and suggestions. Cindy Hood (Supervisor Traffic Management Coordinator in the New York TRACON) taught us about how and why she sets MiT restrictions. William Chan connected us with Jim Evans, who shared his knowledge of the New York TRACON and also pointed us to [15]. Appendix D of [15] carefully explains how to parse NTML data. Joey Rios also assisted us as we acquired and made use of NTML data. Todd Farley made us aware of the earlier work on this topic by Tony Evans and others in [1], and Tony Evans answered our questions regarding that work and also regarding ASPM data. Michael Drew and Kapil Sheth assisted us as we ran FACET simulations to determine the air traffic demand at the fixes. Finally, teaching assistants for the CS229 course at Stanford University during the autumn 2011 quarter provided us with feedback as we investigated different approaches to this problem and CS229 Professor Andrew Ng taught us the machine learning theory used in this work.

References

[1] S. W. Evans, A. D. Evans, H. Idris, and D. Kozarsky, "Modeling distance based metering of arrival operations into major airports," in *Proc. of AIAA Aviation Technology, Integration and Operations Forum*, Chicago, IL, September 2004.

[2] H. R. Idris, A. D. Evans, S. W. Evans, and D. Kozarsky, "Refined benefits assessment of Multi-Center Traffic Management Advisor for Philadelphia and New York," in *Proc. of AIAA Aviation Technology, Integration and Operations Forum*, Chicago, IL, September 2004.

[3] S. R. Wolfe and J. L. Rios, "A method for using historical Ground Delay Programs to inform day-of-operations programs," in *Proc. of AIAA Guidance, Navigation, and Control Conference*, Portland, OR, August 2011.

[4] D. A. Smith, "Decision support tool for predicting aircraft arrival rates from weather forecasts," PhD Dissertation, George Mason University, 2008.

[5] D. A. Smith and L. Sherry, "Decision support tool for predicting aircraft arrival rates, Ground Delay Programs, and airport delays from weather forecasts," in *Proc. of International Conference on Research in Air Transportation*, Fairfax, VA, February 2008.

[6] Y. Wang, "Prediction of weather impacted airport capacity using ensemble learning," in *Proc. of AIAA/IEEE Digital Avionics Systems Conference*, Seattle, WA, October 2011.

[7] G. Buxi and M. Hansen, "Generating probabilistic capacity profiles from weather forecast: A design-of-experiment approach," in *Proc. of USA/Europe Air Traffic Management Research & Development Seminar*, Berlin, Germany, June 2011.

[8] M. O. Ball, R. Hoffman, A. R. Odoni, and R. Rifkin, "A stochastic integer program with dual network structure and its application to the ground-holding problem," *Operations Research*, vol. 51, no. 1, pp. 167–171, January-February 2003.

[9] P. B. Liu, M. Hansen, and A. Mukherjee, "Scenario-based air traffic flow management: From theory to practice," *Transportation Research Part B*, vol. 42, pp. 685–702, 2008.

[10] V. Ramanujam and H. Balakrishnan, "Estimation of maximum-likelihood discrete-choice models of the runway configuration selection process," in *Proc. of American Control Conference*, June 2011.

[11] D. M. Pfeil and H. Balakrishnan, "Identification of robust terminal-area routes in convective weather," *Transportation Science*, 2011.

[12] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," in *Neural Networks: tricks of the trade*, G. Orr and K. M uller, Eds. Springer, 1998.

[13] L. Bottou, "Learning with large datasets," `http://leon.bottou.org/slides/largescale/lstut.pdf`.

[14] "N90 flows visualizer," `http://nyartcc.org/N90Flows/NY_AirSpace.swf`, January 2012.

[15] A. M. Dwyer, L. Epstein, A. Futer, M. Hogan, K. Howard, R. Oiesen, and B. Sharick, "Interactions of multiple traffic management initiatives: An initial analysis," Volpe National Transportation Systems Center, Cambridge, MA, Technical Report VNTSC-TFM-11-11, October 2011.

[16] K. Bilimoria, B. Sridhar, G. Chatterji, K. Sheth, and S. Grabbe, "FACET: Future ATM Concepts Evaluation Tool," *Air Traffic Control Quarterly*, vol. 9, no. 1, pp. 1–20, 2001.

[17] "Orange," `http://orange.biolab.si/`, December 2011.