# Predicting Aircraft Maneuvers to Evaluate the Reliability of Human Observers

2d Lt David Crow, USAF

May 28, 2019

An aircraft's maneuvers in flight are often directly indicated by the plane's roll, pitch, and yaw. Takeoff, for example, can be defined by a high, positive pitch and low, well-correlated values for roll and yaw. Humans, who are naturally error-prone, might misidentify an aircraft's movement when given only a visual observation of the plane. The Air Force is often concerned with the movement of its various aircraft, but, in situations without access to the aircraft's flight data, the Air Force must rely on human observers. Presently, the Air Force does not know much about the reliability of non-expert human observers in identifying aircraft maneuvers.

In this research, multiple machine learning models are fit to a set of aircraft characteristics generated by a flight simulator. A non-expert human observer (that's me) labels each of the simulated observations with the maneuver of the aircraft at that point in time. I evaluate each model's performance and then identify which model best predicts the aircraft's maneuver. An expert system (as defined by a few Air Force pilots) also labels the observations, and these labels are then compared to those predicted by the best model. In doing so, I can determine whether the human observer provides reliable testimony.

The Air Force Research Lab maintains a flight simulator, the Avionics Vulnerability Assessment System, which generates the dataset used in this research. At any given moment, the system computes a multitude of metrics about the aircraft; this reseach concerns the airplane's roll, pitch, yaw, altitude, airspeed, vertical velocity, and acceleration. The simulator is able to display all values as they change over time, and Lt. Marvin and I modified the AVAS source code to enable parameter filtering and file output.

As the non-expert, I flew the simulator and noted time periods where I was taking off, turning, or cruising. By repeating this process, I generated about 5,000 observations evenly distributed over the three classes.

After generating the dataset, I proportionally-sampled 90% of each class to build a training set. I converted all orientation measurements from radians to degrees and scaled the training and testing sets using the training set's mean and standard deviation. In my exploration phase, I found that the simulator's yaw data is just another measure of the airplane's heading, so I dropped the yaw column to remove noise.

I then utilized several different machine learning approaches, all with a fair amount of parameter tuning, to identify the best possible model. I validated every version of every model with 10-fold cross-validation and, to ensure reproducibility, I used the same random seed for the entire process.

To evaluate model performance, I computed Cohen's kappa coefficient between the model's predicted labels and the truth data. Like accuracy, the coefficient is a measure of agreement between two categorical sets, but, unlike accuracy, it considers the possibility of chance agreement between the sets. For two sets of labels $a$ and $b$, the coefficient is defined by

$$\kappa = \frac{p_o - p_c}{1 - p_c} = 1 - \frac{1 - p_o}{1 - p_e},$$

where $p_o$ is the observed agreement between $a$ and $b$, and $p_c$ is the chance agreement between $a$ and $b$.

Results are promising. This table shows Cohen's kappa coefficient between the layman's labels and the labels predicted by the best version of each of the six models. Similarly, it shows the coefficient between the expert's labels and the labels predicted by each model. Clearly, a machine learning model can predict the non-expert's labels very well, but the same model fails to predict the expert's labels at the same level of performance.

The expert system applies labels using just roll, pitch, and yaw data. Out of curiosity, I again computed the performance of the best models, but this time I removed roll, pitch, and yaw data from the dataset. As expected, the models perform worse, but, aside from QDA, the kappa coefficients are relatively close to those computed with the full dataset.

The results indicate that the layman's label for a given observation is often *not* representative of the true aircraft maneuver for that observation. Thus, we can conclude that a human observer needs expert knowledge to accurately classify aircraft maneuvers. This conclusion likely applies to more than just aircraft maneuvers (for example, in evaluating the validity of crime-scene witness testimony), but further research is required to validate this claim.

Although the results show that a model fit to the layman's labels does not adequately predict the expert's labels, they also show that the model fits the layman's labels exceptionally well. The best model's kappa coefficient is 0.993946 (where 1 is the maximum possible value), so it is clear that the model is representative of the non-expert. It is assumed that an

expert applies labels in a more consistent manner than does a non-expert, so it's likely that a machine learning model can fit an expert at least as well as it fits the layman. Thus, one could fit a model to an expert and then use such a model to classify large amounts of aircraft flight data. In doing so, one would reduce manpower requirements – at least for its expert maneuver-labelers.

The AVAS, while useful for this research, is limited in scope. The control scheme is low-quality, the visual representation is sub-par, and the code is far too convoluted to allow for significant modifications. Future researchers who wish to affirm my conclusions should generate data with a higher-quality, professional simulator, or they should work to obtain real data from real aircraft.

Additionally, future research should analyze whether the conclusions hold when the number of classes is significantly large. A powered aircraft's movements can be classified by so much more than just *takeoff*, *cruise*, and *turn*, and future researchers would do well to apply more labels to their observations.

| Model |
|---|
| Logistic Regression |
| Quadratic Discriminant Analysis |
| Ridge Classification |
| Classification Tree |
| $k$-Nearest Neighbors |
| Support Vector Classification |

| Model | Parameter | Values |
|---|---|---|
| Ridge | Regularization Strength $\alpha$ | Twenty logarithmically-spaced values in $[0.01, 1000]$ |
| Classification Tree | Tree Depth | Every integer in $[1, 20]$ |
| | Number of Features | Every integer in $[1, 3]$ |
| $k$-Nearest Neighbors | Number of Neighbors $n$ | Every integer in $[1, 50]$ |
| Support Vector Classifier | Kernel Type | {linear, polynomial, rbf, sigmoid} |
| | Budget $C$ | Ten logarithmically-spaced values in $[1, 1000]$ |
| | Polynomial Degree | Every integer in $[1, 7]$ |

| Model | Layman | Expert |
|---|---|---|
| Logistic Regression | 0.696015 | 0.677175 |
| Quadratic Discriminant Analysis | 0.957674 | 0.927529 |
| Ridge Classification | 0.657370 | 0.631847 |
| Classification Tree | 0.996973 | 0.900345 |
| $k$-Nearest Neighbors | 0.993946 | 0.897330 |
| Support Vector Classification | 0.993946 | 0.897330 |

| Model | Layman | Expert |
|---|---|---|
| Logistic Regression | 0.656764 | 0.637979 |
| Quadratic Discriminant Analysis | 0.739439 | 0.719577 |
| Ridge Classification | 0.587684 | 0.564931 |
| Classification Tree | 0.993946 | 0.897330 |
| $k$-Nearest Neighbors | 0.984867 | 0.894308 |
| Support Vector Classification | 0.993946 | 0.897330 |

| Model | Layman | Expert |
|---|---|---|
| Logistic Regression | 0.696015 | 0.677175 |
| Quadratic Discriminant Analysis | 0.957674 | 0.927529 |
| Ridge Classification | 0.657370 | 0.631847 |
| Classification Tree | 0.996973 | 0.900345 |
| $k$-Nearest Neighbors | 0.993946 | 0.897330 |
| Support Vector Classification | 0.993946 | 0.897330 |