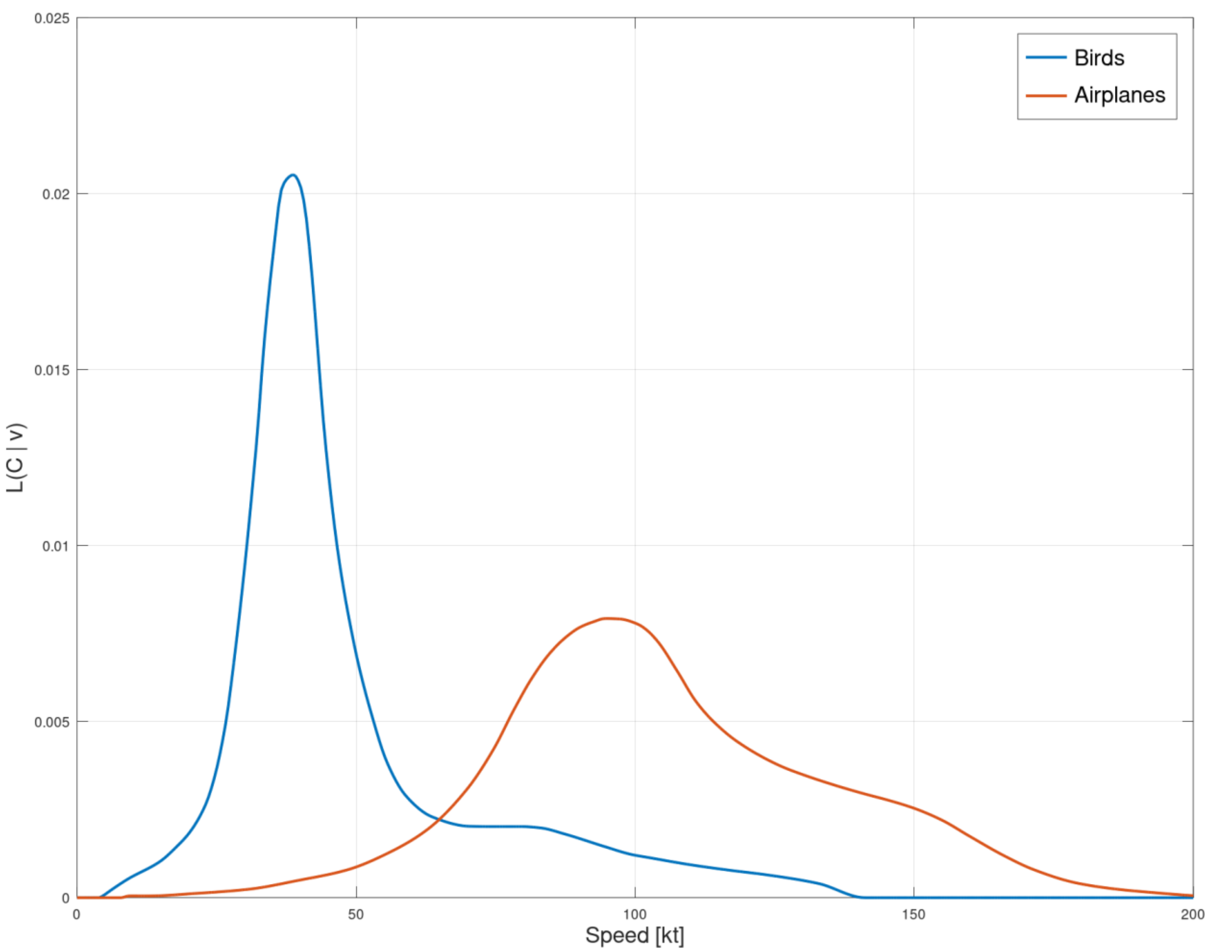
Naïve Bayesian Classification

A RADAR TRACE CLASSIFIER (100 Points)

A frequent problem at airports is the collision between airplanes and birds. You are to solve this problem by classifying radar tracks into two classes: birds and airplanes. Using a Naïve Recursive Bayesian classifier, your job is to calculate and report the probability that the object belongs to one of the two classes for each data point provided.

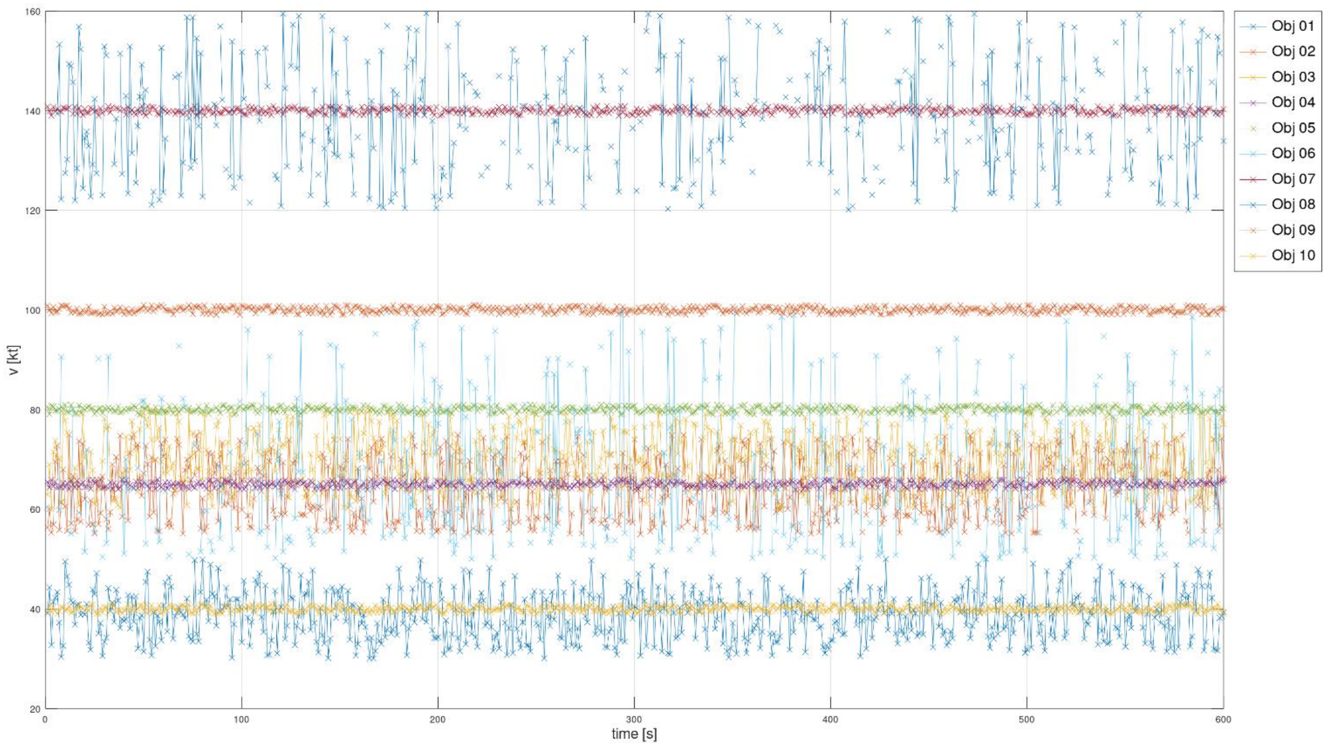
For your classification, you are given the following data:

a) The likelihood of birds and airplanes for a specified speed is represented here:



b) Twenty (20) tracks representing the velocity of the birds and airplanes (10 birds followed by 10 airplanes) measured by a military-grade radar (1s sampling frequency for a total length of 600s). If the radar could not acquire the target and perform the measurement, the corresponding data point would be a **NaN** value. These tracks are curated to have a maximum sample drop rate of 5% of the total number of samples per track. This data can be used if you want to improve the naïve classification with additional signal features.

c) Ten (10) tracks representing the velocity of the unidentified flying object measured by a military-grade radar (1s sampling frequency for a total length of 600s). If the radar could not acquire the target and perform the measurement, the corresponding data point would be a **NaN** value. These tracks are raw data.



For each test track, your solution will have to return a classification ("a" for airplane, "b" for bird) for each data sample and a final classification that summarizes the object class for the entire track.

While testing your application, you must consider that, in certain cases, the object's speed alone is insufficient to make a reasonable determination as there is insufficient data or it is too noisy. Also, assume that the classifier is conservative when transitioning between classes of objects. A probability of transition $P(C_{t+1} = bird \mid C_t = bird) = 0.9$ and $P(C_{t+1} = airplane \mid C_t = airplane) = 0.9$ should be sufficient. However, feel free to change these values as appropriate.

As initial probabilities for the classes, it is normal practice to start the classification from equally distributed values (for two classes, it would be 0.5 for each class). Expect these values to change as the classifier acquires more signal information.

Could you extract an additional feature from the training data to improve the classification? If yes, can you modify your original solution to include this feature in the classifier? You can use the traces in the training data file to extract additional information. Make sure to explain your rationale in the README file.

SUBMISSION

Python or C++ are the preferred implementation languages. If you are writing in C++, please include a Makefile and other compilation instructions. For Python, provide a plain PY file (do not use Jupyter notebooks).

Your solution may use any numerical libraries for pre-processing, fundamental calculations (i.e., linear algebra), and visualization. However, the core portion of your solution must be implemented from scratch.

Submit your solution via Canvas and include a README file explaining its assumptions.

SOLUTION

$O_1 = \boldsymbol{bird}, O_2 = \boldsymbol{bird}, O_3 = \boldsymbol{bird}, O_4 = \boldsymbol{airplane}, O_5 = \boldsymbol{airplane}, O_6 = \boldsymbol{bird},$
$O_7 = \boldsymbol{airplane}, O_8 = \boldsymbol{airplane}, O_9 = \boldsymbol{airplane}, O_{10} = \boldsymbol{bird}$