Assignment 04 - Bayesian Networks

Peter Nadel

July 21, 2025

1 Introduction

Classification is a common task, especially when working with data taken from sensors or other instrumentation. Often these devices will observe different distributions of the value that they measure and either rules-based on statistical models can be trained to tease out and isolate these distributions so that a new observation can be classified easily.

For this assignment, we were given several sequences of different velocity time series, a set representing bird speeds and another airplane speeds. Additionally, we were supplied with likelihood values for each of these and a testing set. For a random given sequence, to discriminate between birds and airplanes, we can develop a Naïve Bayesian Network, which will use conditional probabilities to iteratively predict the class to which that random sequence belongs.

The velocity of an object, however, is not enough for a Naïve Bayesian Network to consistently classify these two classes correctly. As a result, we also look at the rate of change of the velocity, the acceleration, to classify the test sequences.

2 Methodology

Naïve Bayesian Classification uses Bayes' theorem (below) to calculate the probability of a given sequence of velocities belonging to either the bird or airplane class by comparing prior information and current evidence.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

This approach proceeds through the following method:

- First, we find our likelihoods for a given velocity value, that is, the probability of observing a velocity given that it is a bird or that it is an airplane.
- Next, we initialize our prior probabilities. These values will change as data propagates through the network. To start, though, we set these values to .5, as we could never guess the correct class before any training or fitting. Choosing .5 has an initial value signals that we are starting at random. At the beginning, a velocity has as much chance to be from a bird as it does from an airplane.
- We update and normalize our prior probabilities using Bayes' theorem and set these posterior updates as our new prior values.
- We follow these steps for the whole dataset, by the end fitting our priors to the given data. When we do our posterior update (the above step) in subsequent steps, we must also multiple the priors with transition probabilities. These values reflect the likelihood of staying the same class rather than transitioning to another class. Because our network is conditioned solely off of how the past timestep conditions the current one, it can become unstable very quickly. These transition probabilities allow for the model to learn at more regular rate.

To then classify a given sequence after fitting, we follow these steps:

- We determine the likelihood values as we did above.

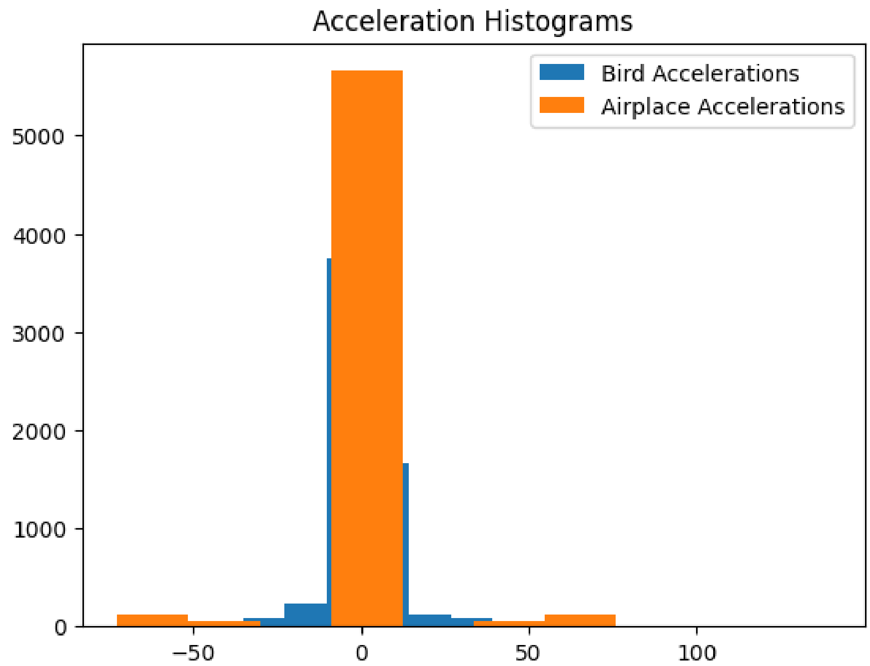

Figure 1: Distribution of acceleration data

- Retrieve our stored priors from the fitting process.
- Continue the update process as we did above. Instead of updating the fit priors though, we assign a class to the likelihood value by choosing the larger posterior update.

Applying this method to the data, we are able to output not just the classification at each likelihood for a given velocity sequence, but also the confidences that the model has in its classification.

3 Additional features

As mentioned in the Introduction, though velocity is a driving factor in classification, the addition of other features may help the classification process. In particular, the acceleration of the object may be a helpful way of discriminating birds from airplanes. Thankfully, this value is very simple to derive from the velocity values as the acceleration is its first derivative. Multiplying the acceleration likelihood with the velocity likelihood during the posterior update will account for the differences in the acceleration between birds and airplanes.

That said, while the likelihoods for velocity were given, the likelihoods for acceleration would have to be determined. There are two main ways of doing so: non-parametric and parametric.

3.1 Non-parametric likelihoods

To obtain likelihoods non-parametrically, we are able to use the histogram of the accelerations to determine how likely it is that if were to randomly choose an acceleration value it would be that given acceleration value. Thankfully, there are several built-in methods in Python packages like numpy and matplotlib that allow us to easily do this. In Figure 1, we see the histogram of all of the acceleration data.

3.2 Parametric likelihoods

On the other hand, we can also obtain likelihood data parametrically, that is, by fitting a Gaussian distribution to our data, based on the parameters: mean and standard deviation. This distribution will be able to give us the likelihoods that we want but not based solely on the data itself. By

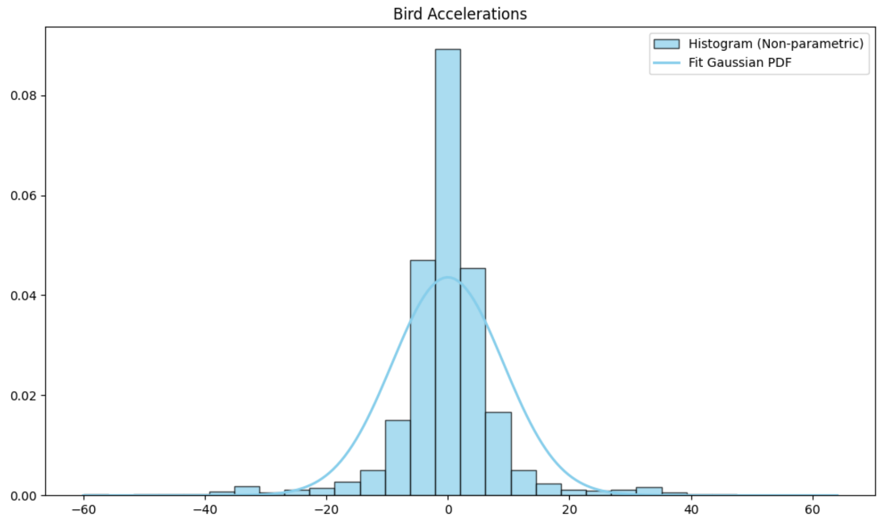

Figure 2: Bird acceleration data with fit distribution

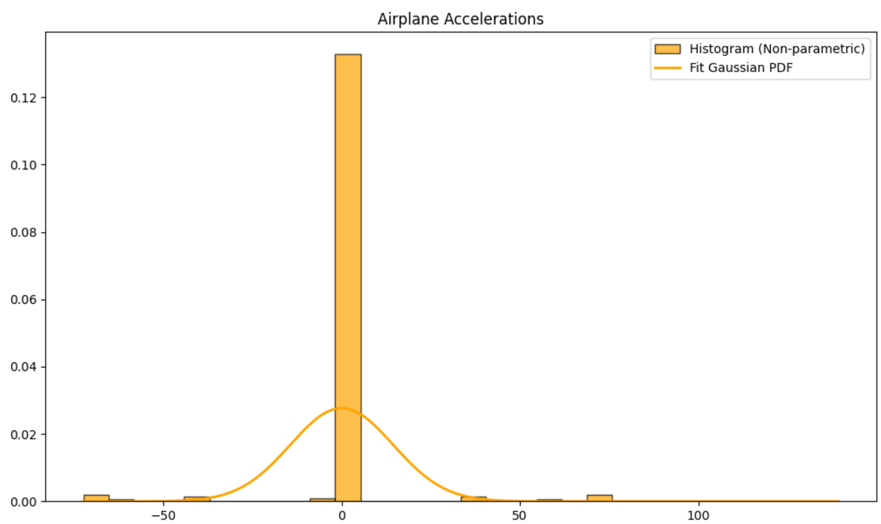

Figure 3: Airplane acceleration data with fit distribution

employing this method, we make the following hypothesis: the underlying distribution of accelerations is approximated by the Gaussian distribution. The performance and results of the classifier trained on these likelihoods will give us some sense as to how true or false this assertion may be. In Figures 2 and 3, we see the histogram data plotted with the probability distribution function of the fit Gaussian.

4 Results

Evaluation on the test set told an interesting story about the kinds of interventions we made on the data and how we modeled this classifier.

Model name	Accuracy
Just velocity	.8
Non-parametric Acceleration	.9
Parametric Acceleration	.7

Based on these results, our hypothesis about non-parametric and parametric acceleration likelihoods is likely wrong, meaning that the underlying distribution of the acceleration is not normally distributed.

Looking at the plots above may give us some indication of this. Especially in the case of the airplane data, the Gaussian distribution fails to capture the complexity of the acceleration data. High peaks and long tails tend to disrupt these fit Gaussian distribution and it seems that the acceleration data contains both of these. In the future, we may be able to use other parameterizations of the acceleration data to fit different distributions which better fit the underlying distribution.