Machine Problem 1:

Naive Bayes Classifier for Athletes Dataset

CS165A
2/17/2022

Teagan Connon
teagan@ucsb.edu
7851405

Architecture:

The main class of my project is the NaiveBayesClassifier, which itself contains a two element vector of a struct called meanVariance. meanVariance is a 2D vector which contains the mean and variance for each feature given a certain value of the class, as well as the probability of that class taking that value. These meanVariance elements are constructed using the fit() method which takes in a training dataset as a parameter. To make predictions, a testing dataset is passed into the predict method which will return a vector containing the predictions for each element of the testing set. To accomplish all the calculations, a separate file called probability.h was used to perform operations such as acquiring the mean, standard deviation, or variance of an input vector.

Preprocessing:

My preprocessing was fairly minimal, mostly consisting of taking in a CSV file, and converting it into a 2D vector with 12 vector<vector<double>> representing each feature plus the class. To accomplish this, first I had to replace the gender column with either a 1 or a 0 to represent female or male, and then I simply iterated over the whole CSV taking turns pushing into each feature vector. As well, there is a function called splitDataByClass which can take a 2D vector input, and a class label, and extract only the members who belong to whatever class was specified.

Model Building:

Using the 2D vectors split by class, I was able to calculate the mean and variance for each feature for each class. In this case, that means only doing it for C=1 and C=0, but theoretically my code could be modified slightly to accommodate a class with any number of possible values. Once the means and variances are calculated, they are stored into their corresponding feature vector in their corresponding meanVariance 2D vector. To get the conditional probability for a

given feature value, the value, mean, and variance are input into the gaussian conditional probability formula.

Results:

In my testing, my classifier had a 77.8% accuracy on the testing data, and was able to finish running with output completely in ~0.076s.

Challenges:

One challenge I ran into was handling probability distributions for boolean variables since I assumed all the distributions were gaussian which doesn't work very well for boolean variables. I was able to work around this by setting the probability of a positive result equal to the mean, and the negative to be one minus the mean. Class was the only boolean variable that I ended up running into the problem for, since the only other boolean variable was gender which the raw probability never ended up being necessary to determine.

Weaknesses:

The Naive Bayes model makes a key assumption that all features are equally important as well as conditionally independent which for this dataset isn't necessarily a good assumption to make. For instance, height and weight are probably fairly correlated, and roughly represent the same thing so the classifier is giving unfair weight to these two features. As well, the gender feature probably has no bearing on how high class an athlete is since I would expect the proportion of high performance male athletes to be roughly the same as the proportion of high performance female athletes, since they have separate pools. Also, as mentioned above, I assumed that all the features had a gaussian probability distribution which isn't necessarily a good assumption. Were I to do it again I'd probably analyze the dataset a bit more rigorously to determine if perhaps a kernel function was needed to really get good probability values.