Naive Bayes Classification Using the Iris Data Set

Dalton Rothenberger

1. Introduction

Naive Bayes is a method of using features to contribute independently to the probability that some set of features belongs under a specific classification. The individual probabilities of the features given a specific class are multiplied together to produce the overall probability that the set of features falls under that class. This calculation is then repeated for all the other classes and whichever class produces the highest probability is the class that is selected for classification.

2. Setup and Methodology

For this Naive Bayes Classifier, the Iris Data Set was used which is available here. The data contains three different plants inside it which will be the classifications for our Naive Bayes Classifier. The three classifications are Iris-setosa, Iris-versicolor, and Iris-virginica. Also, each entry in the data set has four columns associated with it and the four columns are sepal length, sepal width, petal length, and petal width. This data was parsed into a Python program and loaded into a 3-D array. The rows of this 3-D array were individual entries, the columns were the different feature types, and the layers were the different classes. Since the data was continuous it had to be made into discrete points for Naive Bayes. To accomplish this, binning was used. The binning was done based off of the mean for each feature. All the entries for a given feature were counted up and divided by the total to calculate the mean. Then break points for four bins were created from the mean. For bin 1, it was from 0 to $\frac{mean}{2}$. For bin 2, it was from $\frac{mean}{2}$ to $mean$. For bin 3, it was from $mean$ to $\frac{3*mean}{2}$. For bin 4, it was anything greater than $\frac{3*mean}{2}$. This binning was applied to both the training data and test data. The means calculated while binning the training data were used in the binning of the test data. The probabilities for each bin was calculated for each plant for every given feature and stored into another 3-D array. This was done so that the calculation for these probabilities was only done once for a given set of test data. These probabilities were then used to classify every entry in the given test data using Naive Bayes with log probability to prevent underflow. The probabilities for the 3 classes of plants were compared and whichever had the greatest probability was selected as the classification.

3. Results and Analysis

Using the Iris data set as both the training data and test data at the same time resulted in 145 correct classifications out of 150 total. This means an accuracy of 96.67%. At first, this surprised me since it was using the same data to learn as it was to test but after looking at what the entries I understood what was happening. All five cases occurred between Iris-versicolor and Iris-virginica. Iris-versicolor and Iris-virginica had values that were relatively close together. The five cases were extremes where the features for a given Iris-versicolor were larger than most other Iris-versicolor or the features for a given Iris-virginica were smaller than most other Iris-virginica. This resulted in the entries of the given plant being binned into a different category than most others for that plant. Thus, when calculating the probabilities this resulted in the wrong plant having a higher probability than the correct classification resulting in a wrong classification.

When splitting the data into a set of training data and a set of test data that was representative of the training set, the classifier had 100% accuracy with 0 wrong classifications. The data was split so that 120 entries, 40 entries from each plant type, were used as training data and 30 entries, 10 entries from each plant type, were used as test data. Also, running Vowpal Wabbit against this setup of the data resulted in 30 correct predictions out of 30 examples, thus a 100% accuracy for its predictions.

4. Conclusion

The Naive Bayes classifier worked well with the Iris Data set. It was able to properly classify the plants for average cases but not at the extremes of each plant type. The Naive Bayes classifier leaves room for improvement when it comes to classification. If the data points were closer together between all the plants this could have caused even more wrong classifications. The binning technique as well could have been improved to create more diverse categories to separate the data even more. Overall, the Naive Bayes classifier works well in a simple scenario like this but the ”naive” part of its name is there for a reason.