Programming Assignment 1

Abstract

The purpose of this write-up is to examine the accuracy rate of different classifiers (Naive-Bayes and Vowpal Wabbit) on the Iris dataset. https://archive.ics.uci.edu/ml/datasets/Iris

1. Introduction

I set out to explore the effectiveness of two different classifiers on the Iris dataset, a popular dataset for teaching in machine learning. The data is composed of different measurements of different class of the iris flower. These include: sepal Length in cm, sepal width in cm, petal length in cm, and petal width in cm. Using these different data points and by training a model using both a custom naive bayes program and the Vowpal Wabbit machine learning software, i attempted to classify unseen Iris examples into three catagories: Iris Setosa, Iris Versicolour, and Iris Virginica. The goal was to use the model generated from both the custom program and the Vowpal Wabbit software to produce a high accuracy prediction rate.

2. Running the Program

The classifier works out of the box. Simply type:

python3 NaiveBayes.py

Into the terminal and it will go through 100 iterations of random train/test splits with the Iris dataset.

Similarly for the Vowpal Wabbit software:

sh test.sh

Will train and test the data with a report on the accuracy.

3. Challenges

This section will address two challenges that were faced when attempting to build the Naive Bayes model for the classification of the Iris dataset.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

3.1. Continuous Values

Normally Naive Bayes lends itself to a binary feature set. That is to say, the features of a given class are either on or off. (0—1) The default Naive Bayes formula:

$p(x|c) = \prod_{i=1}^{D} p(x_i|C)$

Is only really useful if the nature of x_i is binary. The nature of the Iris data is not binary, but rather continuous. You do not have petal length that is either 1cm, or 0cm, but rather 1 to n cm of possibilities. Because of the continuous nature of the data, means and standard deviation of a given attribute in a given class were used to build the model on a gaussian distribution. That is to say the mean value of an attribute was calculated by with:

mean = $\frac{\sum(x)}{numberofx}$

and standard deviation with:

stdeviation = $\frac{\sum(x-mean)^2}{numberofx}$

Using these two statistics we can then summarize the data to see what the average value for sepal length, sepal width, petal length, and petal width were for the Iris Setosa, Iris Versicolour, and Iris Virginica respectively. These were then used to calculate probabilities based on Gaussian probability density function (normal distribution):

$\frac{1}{\sqrt{2\pi}*\rho} exp[-\frac{(x-\mu)^2}{2\rho^2}]$

This method makes sense for continuous data. If all of the same kind of flower have similar petal length, then their average petal length should be different than the average petal length for another type of Iris. By using averages and the above formula we can classify unseen data.

`See http://users.isr.ist.utl.pt/~mir/pub/`

`probability.pdf for more on Normal Distribution`

3.2. Handling Underflow

Because the nature of multiplying this many probabilities by one another could easily lead to underflow in a computer, I took advantage of the fact that log(p_1) + log(p_2) = log($p_1 p_2$) as described in the writeup.

This actually encountered some underflow errors by itself, but the decimal python library and the natural log function were able to solve that.

Overall the performance of the classifier took a hit using logs, it is significantly slower than it was when logs where not being used. This is to be expected with having to bring in and use another entire python package. The tradeoff between time and no underflow errors is worth it.

4. Accuracy

This section will cover the accuracy of both the Naive Bayes model and the abnormal accuracy of the Vowpal Wabbit model. As well as how those abnormalities were handled and corrected.

4.1. Training and Test Data Set Methodology

The Iris data set is composed of 150 samples. In order to train and test the model the data was split into a 2/3, 1/3 ratio for training and test data respectively. This was done completely at random for both the Naive Bayes model and the Vowpal Wabbit model.

4.2. Naive Bayes Accuracy

Using the methodology described above, and a few hours of time, one million iterations of the naive bayes model was trained, and tested against. The highest accuracy achieved was 98% and the lowest 92%.

On average the model, over one million iterations, succeeded with 95.355308% accuracy.

This is not entirely surprising. While the data set is continuous and we had to handle that, it is a very simple set with very few features. I would expect the model to do much worse on a set that had hundreds of features, but because we are only dealing with 4 features and 3 classes here, it is not surprising that Naive Bayes was capable of an accuracy in the high 90's.

Looking at the actual data, with the method I used taking the average and standard deviation of each of the features, you can see how the model would be accurate. The Iris flowers are pretty distinct in terms of each of their features. For example, the Iris Versicolour's Petal Width is almost entirely inside of the 1.0 - 2.0 range. It would be easy for the model to assume off of averages that any new flower with petal width around that length would fit into this category.

We can account for the variance in between runs simply because the data is being shuffled before training and testing the model each time. Overall we achieved a spread of only 6% for all tests.

4.3. Vowpal Wabbit Accuracy

Vowpal Wabbit proved to be quite effective with classifying unseen sets of iris flowers. An interesting anomaly appeared in my testing of the dataset with Vowpal Wabbit. I was seeing a correct classification of the data 100% of the time.

In order to make sure this was not just an anomaly, a script was created to randomize the training and test data similar to the method used with Naive Bayed. Once again in this case we used 2/3 of the 150 samples as training data and 1/3 as test data. I started with 7 passes to generate the model and recieved a 100% accuracy rating. After 100's of tests the lowest achieved rating was 100%.

After tweaking the passes down to 3 per model train, I managed to get a 98& accuracy rate to show up in a single pass. Going lower than 3 passes resulted in dramatically reduced the accuracy to the 60% - 70% range. Which makes sense. With the amount of data we have here reducing the total passes down below three would not allow the logistic model Vowpal Wabbit uses enough information to train the β 's on.

5. Conclusion

It would appear that while the Naive Bayes model for the classification of Iris Flowers is rather accurate at a high 90's% rate, the Vowpal Wabbit logistic regression style of classification is much more effective. I had to go through drastic measures to reduce the accuracy of the Vowpal Wabbit model. The algorithm behind Vowpal Wabbit was much more effective at classifying the data than the Naive Bayes implementation. This was suprising to me because I thought with how small the dataset was, Naive Bayes would have an easier time classifying such a small amount of data. I did not believe that there was enough information for a logistic regression style classifier a'la Vowpal Wabbit to train and classify effectively on. I was wrong, the constant iteration on the β's, even when it was as small as 3 runs, produced results on par or exceeding the Naive Bayes implementation.