Bayesian learning for classifying Netnews text articles

Saichand bandarupalli
Machine learning class project
Department of Computer Science
Colorado School of Mines, Golden , CO,80401
November 28, 2017

Methodology

The implementation for this project is carried out using the python language. The naive bayes classifier is trained on the news data provided. The data is divided into 80 is to 20 ratio for training and testing of the bayes classifier. The bayes classifier is tested and the results for the pair wise class comparison and the three class comparison are documented in this report. Further, The results for many vs one classification is also documented in the report

Running

For running the code open the python file from the home directory in a python work environment like IDE. The python file which consists of the code is "**Naive_bayes.py**". For pair wise, triple and many to one comparisons bring the classes from the sorted data folder to the **20-newsgroups"** folder.

Creation of the Testing and training datasets:

Here as discussed earlier the data is split into two groups of 80 percent data and 20 percent data which are the training and the testing data respectively.

So essentially,

1. Training instances: 800
2. Testing instances: 200

The Naive bayes classifier:

In this project a probabilistic approach for classing the data, the Naive-Bayes has been used. Given a problem instance to be classified, represented by vectors x = [x1; : : : ; xN], representing N training instances.

In the code node.prob acts as a bag of words approach in keeping track of how many time the word is repeated and creates a data structure based on that. The different approaches used are Class-vs-Class, Tri-class and Total class classification. In Class-vs-Class we take different pairs of classes, train and test them to get the accuracy of classification. In tri-class we take three different class and process the same steps as above. The results are provided in the upcoming section. At last we see how classification would work by taking all the twenty classes at once.

Results

Pair wise comparison:

The comp.os.ms-windows.misc class vs comp.sys.ibm.pc.hardware class have low accuracy. This is because both the classes have many words in common which make them closely related. The other two classes have close to 100 percent separation given the reason the two fields have distinct terminologies.

1.	Comp.os.ms-windows.misc vs comp.sys.ibm.pc.hardware	87.5
2.	Alt.atheism vs comp.graphics	99.25
3.	Comp.sys.mac.hardware vs comp.windows.x	98.5

Tri class classification

1.	Alt.atheism vs comp.graphics vs comp.os.ms-windows.misc	94.83
2.	Comp.sys.ibm.pc.hardware vs comp.sys.mac.hardware vs comp.windows.x	95
3.	Misc.forsale vs rec.autos vs rec.motorcycles	95.29
4.	Rec.sport.baseball vs rec.sport.hockey vs sci.crypt	98.34
5.	Sci.electronics vs sci.med vs sci.space	97.35

As described above the difference in accuracies among the various classes is due to the distinctness of the keywords in each news class. From this we can see that all classes in the comparison have more distinct terminologies in the news text with Rec.sport.baseball vs rec.sport.hockey vs sci.crypt having the highest accuracy given the same reason

Total classification:

The code was run on a 2.8 Ghz processor. It took a minute to train and another couple minutes to test the model and come up with the accuracy results. The results are presented in the upcoming subsections.

1.	No break words	86.05
2.	25 break words	87.00