Multinomial Naive-Bayes for text classification

Nikhil V

First Year Student
M.Sc. Data Scince
Chennai Mathematical Institute,
Siruseri, Chennai-603103

March 16, 2019

1 Introduction

The problem is of multi-label text classification. The dataset contains data in form of SGML files. Each file has 1000 articles with SGML tags.

2 HTML Parser

I Extracted relevant data from SGML file using HTML parser and brought it in form of list of tuples where the first entry is list of topics and second entry is the text. This can be done easily using beautiful soup also.

In parser I've kept track of when the tags are entered and exited using in_body, in_topics, in_topic_d which is reset on every encounter of reuters end tag. When the start tag is observed corresponding switch gets activated and corresponding content is parsed. To save on memory I fed the data in form of chunks.

3 cleaning

After getting the data in format such that labels and text can be easily accessed, I filtered the labels of each text and removed the erroneous labels. Raw text contains lot of things like punctuation which can act as noise while training our model and therefore reduce the accuracy. Therefore one must clean the data before training the model. I've removed the punctuations and short forms like can't, I'm, I've.

4 Vectorization

Then I vectorised data using tf-idf, tf-idf stands for Term frequency- Inverse document frequency. Term Frequency is defined as how frequently the word appear in the document or corpus. Inverse Document Frequency is based on concept that less frequent words are more informative.

Features in machine learning is basically numerical attributes from which anyone can perform some mathematical operation such as matrix factorisation, dot product etc. But there are various scenario when dataset does not contain numerical attribute, like in this case.

When it came to labels of the data, Although a list of sets or tuples is very intuitive format for multi-label data but it is hard to process. Therefore I converted this data into a supported multi-label format using MultiLabelBinarizer. Which is nothing but a (samples x classes) binary matrix indicating the presence of a class label.(form on which we can apply mathematical operations) This function has a inverse method through which after predicting we can invert the matrix and get labels in form of text.

5 Test Train Split

For different fractions of test train split I was getting different accuracies. When the fraction of training data was more accuracy was more. for example 70-30 split gave 65.64% accuracy, 90-10 gave 68.50% accuracy

Any train-test split is likely to give better accuracy on test data if we put more fraction in training set. But we are interested in improving the accuracy on real data and not on the test data. which is estimated by test set. Therefore what test-train split to choose is always subjective.
So, depending on the following factors we can decide what split we should use.

1. If we want good estimate of our real error then we should increase the size of test set.

2. If its hard getting more training data and more data in training set improves the model then we can reduce the size of test set.

6 Training model

I used Label Powerset function from sklearn. Label Powerset is a problem transformation approach to multi-label classification that transforms a multi-label problem to a multi-class problem with 1 multi-class classifier trained on all unique label combinations found in the training data. The method maps each combination to a unique combination id number, and performs multi-class classification using the classifier as multi-class classifier and combination ids as classes. Lable powerset function take base estimator as input. The base estimator used was ComplementNB from sklearn. Which trained faster than GaussianNB and gave more accuracy than GaussianNB.

7 Accuracy

Accuracy on training set was depending following factors- train-test split, cleaning data, base estimator, transformation method for transforming multi-label to multiclass problem. By cleaning the text by removing punctuation and digits accuracy increased a little bit.