Bayesian Learning for Classifying Netnews Text Articles

Mayur Muralidhar
1001434196

October 29 2017

1 Problem

The net-news text articles consist of 10,000 articles that are spread over 20 categories. Using this data set, a Naive Bayes classifier was trained, so that a text article presented to it can be easily categorized.

When solving this problem, two different approaches were used. In one approach, only the top 500 features were taken, and a lapace smoothing value of 100 was used. In the other approach, a shuffled training set was used downloaded from the scikitlearn python machine learning repository, and the top 9500 features were used. The classifier was written in the python programming language.

2 Implementation

The Naive Bayes classifier uses the fundamentals of Bayes Theorem to achieve classification of data. Data can be classified into various classes, and probabilities of data being in a class are calculated to classify data into a class.

To train the classifier, given a data sample x with n features $x_1, x_2, ..., x_n(x)$ represents a feature vector and $x = (x1, x2, ..., xn))$, the goal of naive Bayes is to determine the probabilities that this sample belongs to each of K possible classes $y_1, y_2, ..., y_K$, that is

$$P(y_k|x) \tag{1}$$

or

$$P(y_k|x_1.x_2, ..., x_n) \tag{2}$$

where k = 1, 2, ..., K.

To compute $P(y_k|x)$, we apply Bayes Theorem as follows :

$$P(y_k|x) = \frac{P(x|y_k)P(y_k)}{P(x)} \tag{3}$$

$P(y_k)$ is known as the prior, $P(x|y_k)$ is known as the posterior and $P(y_k|x_1.x_2, ..., x_n)$ is known as the likelihood in Bayesian terminology.

For testing, the following equation is applied to the entire document:

$$h_{NB}(x) = \underset{y}{\operatorname{argmax}}\, P(y) \prod_{i=1}^{LengthDoc} P(x_i|y) \tag{4}$$

2.1 Implementation 1

In this implementation, 9,999 samples were taken for training and 9,998 samples were taken for testing. Data was cleaned by filtering out names of people, and by removing all punctuation marks. The people's names were obtained from a dataset provided by the nltk text processing python library. Stemming was also used while data processing.

Here, while calculating the posterior, to prevent overflow due to the multiplication of hundreds of small conditional values, the summation of the natural logarithms of the probabilities were taken and then converted back by taking the exponential.

2.2 Implementation 2

Here, the dataset was taken from the scikit-learn machine learning repository, which has 11,314 training samples and 7532 testing shuffled samples. The dataset also had the headers, footers and quotation blocks of the messages removed, to allow for better training of the classifier.

3 Result

With implementation 1, an accuracy of 10.321% was achieved when tested against testing data, whereas with implementation 2, an accuracy of 45.193% was achieved.

References

[1] Yuxi Liu. *Python Machine Learning By Example.* Packt Publishing.

[2] Tom Mitchell. *Machine Learning.* McGraw-Hill Science/Engineering/Math, 1997.

[3] Dajiang Zhu. Lecture 8 probability distribution, naïve bayes. Class Presentation Slides.