A Report on
Classifying Text Documents using Naive Bayes Classifier

by

Kshitij Sharma - 2012A7PS009H
Rohit Sharma - 2012A7PS050H
Abhishek Kaushik - 2012A7PS056H
Prakhar Gupta - 2012A7PS059H

BITS F464 - Machine Learning

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE PILANI (RAJASTHAN)
HYDERABAD CAMPUS

(12th November 2014)

Contents

Description

A **Naive Bayes classifier** is a simple probabilistic classifier based on applying Bayes' Theorem(from Bayesian statistics) with strong independence assumptions. A more descriptive term for the underlying probability model would be “independent feature model".

In simple terms, a naive Bayes classifier assumes that the presence (or absence) of a particular feature of a class is unrelated to the presence (or absence) of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 4" in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier considers all of these properties to independently contribute to the probability that this fruit is an apple.

Step by Step Algorithm

Learn_Naive_Bayes_Text(*Examples*, *V*)

Examples is a set of text documents along with their target values. *V* is the set of all possible target values. This function learns the probability terms $P(w_k|v_j)$, describing the probability that a randomly drawn word from a document in class v_j will be the English word w_k. It also learns the class prior probabilities $P(v_j)$.

1. Collect all words that occur in examples and assign them to a set called *Vocabulary*.

2. calculate the required $P(v_j)$ and $P(w_k|v_j)$ terms.

For each target value v_j in *V* do,

- $docs_j \leftarrow$ the subset of documents from examples for which the target value is v_j
- $P(v_j) \leftarrow |docs_j| / |Examples|$
- $Text_j \leftarrow$ a document created by concatenating all members of $docs_j$
- $n \leftarrow$ total number of distinct word position in $Text_j$
- for each word w_k in *Vocabulary* ,

$n_k \leftarrow$ number of times w_k occurs in $Text_j$

$P(w_k|v_j) \leftarrow (n_k + 1)/(n + |Vocabulary|)$

Classify_Naive_Bayes_Text(*Doc*)

Return the estimated target values for the document *Doc*. A_i denotes the word found in the ith position in the *Doc*.

- positions ← all word positions in *Doc* that contains tokens found in *Vocabulary*
- Return V_{NB} , where V_{nb} = argmax $P(v_j)$ x (product of($P(a_i|v_j)$))

Experiment

1. Five files have been used to train the classifier out of which three have been classified as dislike and two as like prior to experiment. Three documents of like category contain data related to mathematics and two documents of dislike category contain data about computational biology.

   Testing data contains : machine learning description in test1, data related to biology in test2, data related to education in test3
   Files used for training : set1, set2, set3, set4, set5
   Files used for testing : test1, test2, test3

   All files are in .txt format.

2. Probabilities of distinct words from testing data have been mapped to their respective probabilities according to classes like and dislike.

3. Special characters have not been taken into consideration.

Results:

For the three files used to testing data, output is as shown below:

1. Test File 1: Dislike
2. Test File 2: Like
3. Test File 3: Like