

# Author Identification Using Naïve Bayes Classification

James Conigliaro

Marquette University

Department of Electrical and Computer Engineering

jconigliaro@ieee.org

## Abstract

Bayesian learning techniques are becoming widely applied to the categorization of short texts, such as email messages, however, the problem of classifying long texts (i.e. novels) is typically approached from a linguistic stand point, using both word choice and grammatical structures as a basis for manual analysis [9]. In this study, the Naïve Bayes classification technique is applied to the problem of identifying whether or not a specific individual authored a document. A minor modification to the Naïve Bayes algorithm is applied to make the algorithm better suited for application to large text documents. This study does show that the technique is effective for determining whether or not a document has been authored by a specific individual, though the work presented serves only to prove the concept, opening the problem for further study.

## Introduction

There are situations in which an anonymous text is believed to be written by a specific author but there is a lack of evidence to prove this. In such cases, experts painstakingly analyze known texts by the author and performed detailed, manual comparisons to determine if a text is in-fact authored by a given individual. While linguistic computer applications, known as concordance software, are now being used to perform such tasks, the process is still largely manual [9].

With the advent of widely available electronic texts, research into the automated classification of such documents has steadily increased over the past decade [3,4]. A number of methods for categorizing texts have been proposed ranging from extraction of keywords to the application of artificial neural networks [1,3,5,8]. Of these

techniques, the Bayesian techniques for categorization have gained the widest popularity[3], particularly as a method for classifying short texts such as email messages and web pages.

In this paper, a method is presented for training a computer application to recognize texts authored by a given individual for the purposes of classifying an anonymous text. The methods presented utilize databases of texts published by known author to train an application to determine whether or not a text was written by a particular author, specifically Jane Austen. A number of texts written by Jane Austen along with a number of other texts written by randomly chosen authors were used to train to a Naïve Bayes classifier. Additional texts, some written by Jane Austin and others written by randomly chosen authors were then used to test the results of the training process. Experimental results indicate that the Naïve Bayes classifier may be used to accurately indicate whether a particular individual authored a certain text

This paper begins with a detailed description of the algorithms utilized to characterize a text by its author. The algorithms are first described mathematically; the details of the implementation are then presented. Once the algorithms are presented, the experimental procedures and data sets are described, followed by a summary of the experimental results.

## Algorithm Description

In this experiment the Naïve Bayes classifier is used with one minor modification which will be described henceforth. The Naïve Bayes classifier is a statistical classifier that proposes a hypotheses based upon the maximum likelihood of the data supporting a specific hypothesis.

## Bayes Optimal

The Naïve Bayes classifier is based upon the Bayes Optimal classifier [1, 2, 3] which is represented with the following equation:

$$\hat{c} = \arg \max_{c_j \in \mathbf{C}} \left[ \sum_{h_i \in \mathbf{H}} P(c_j | h_i) P(h_i | D) \right] \quad 1.$$

where  $\hat{c}$  is the estimated value,  $\mathbf{C}$  is the set of all values,  $c_j$  is the  $j$ th value,  $\mathbf{H}$  is the set of all hypotheses,  $h_i$  is the  $i$ th hypothesis, and  $D$  is the training data. Expressed simply, the Bayes Optimal classifier states that the value  $v$  that has the highest probability given some hypothesis  $h$  and training data  $D$  is the most likely classifier of the data. Here it is seen that the calculation involves computing the probability of the hypothesis given the training data and the probability of the each possible value given a hypothesis the space of all possible hypotheses. Given this task, the implementation of equation 1 is not feasible [1].

## Naïve Bayes

The Naïve Bayes classifier attempts to simplify the problem by computing the probability of some value given the set of all attributes. While the evaluation of this value is also computationally prohibitive [3], if the assumption is made that the attributes are conditionally independent, the computation becomes much less complex. While this assumption is likely to be false, the error that it induces in the predictive results has been shown to be minimal [1]. The Naïve Bayes classifier may be summarized with the following equation:

$$\hat{c} = \arg \max_{c_j \in \mathbf{C}} \left[ P(c_j) \prod_{i=0}^n P(a_i | c_j) \right] \quad 2.$$

where  $\hat{c}$  is the estimated classification,  $\mathbf{C}$  is the set of all possible categories,  $c_j$  is the  $j$ th category,  $a_i$  is the  $i$ th attribute, and  $n$  is the number of attributes. Expressed simply, the Naïve Bayes classifier estimates a target value by assuming that the data to be evaluated belongs to a category,  $c$ , in  $\mathbf{C}$ , and then estimating the probability of the given

attributes being present. The value with the highest probability is chosen as the resulting estimate.

## Modification to the Naïve Bayes

The Naïve Bayes classifier has been used to extensively in text categorization. A number of modifications to the algorithm have been proposed [1,2,3]. One of the main attractions to this algorithm is the relative simplicity of the computation, even as the number of attributes increases. However, when dealing with very large sets of attributes, problems arise relating to the limits of precision in computers.

By the very nature of probability, it can be shown that:

$$P(a_i | c_j) < 1 \quad 3.$$

From Equation 3 it follows that

$$\prod_{i=0}^n P(a_i | c_j) = 0 \quad 4.$$

$\lim_{n \rightarrow \infty}$

While in practice,  $n$  does not approach infinity, it grows large enough for the value to exceed the limits of double precision floating point numbers in modern computers. The end result is that under certain circumstances, the probability for each value is computed as zero, resulting in an inconclusive result.

To address this problem, the logarithm of the conditional probabilities was added to the Naïve Bayes algorithm as follows:

$$\hat{c} = \arg \max_{c_j \in \mathbf{C}} \left[ -\text{Log}_{10} \left[ P(c_j) \prod_{i=0}^n P(a_i | c_j) \right] \right] \quad 5.$$

The logarithm may be moved inside the product operator as follows:

$$\hat{c} = \arg \max_{c_j \in \mathbf{C}} \left[ -\text{Log}_{10} P(v_j) + \sum_{i=0}^n -\text{Log}_{10} [P(a_i | c_j)] \right] \quad 6.$$

Note that when moving the logarithm operator inside a product term, the product term becomes a summation. The repetitive

summation is easily within the computational limits of a modern computer.

For example, assume:

$$P(a_i|c_j) = \{.0001,.0002,.0003,.0001\}$$

for i=0 to 3. With the standard Naïve Bayes algorithm, the product of these terms is:

$$0.000000000000000006.$$

One can imagine what will happen with n in the hundreds or thousands. To see the affect of the modification perform the sum of the logarithms as follows:

$$-\text{Log}_{10}(.0001) = 4.$$

$$4 - \text{Log}_{10} (.0002) = 4 + 3.699 = 7.699$$

$$7.699 - \text{Log}_{10} (.0003) = 7.699+3.523 =11.222$$

$$11.222 - \text{Log}_{10} (.0001) = 11.222 + 4 = 15.222$$

While the introduction of the logarithm, has little bearing on an example in which there are only four attributes, when one is dealing with hundreds or thousands of attributes, this modification becomes an important player in the computability of the Naive Bayes classifier.

### Implementation

The Modified Naive Bayes algorithm was implemented in a manner similar to that described in [1]. The detailed of which are described herein.

The attributes in equation 5 are defined as the unique words present in all of the text presented for training. This choice of attributes is especially desirable in that it requires no underlying knowledge of grammatical constructs or historical contexts, resulting in an algorithm that is genuinely ignorant of linguistics. As such, the probabilities we are concerned with are the probabilities of the author using each specific word. Other techniques have been proposed that attempt to reduce the size of the attribute space [3, 6, 7].

In order to estimate these probabilities, the m-estimate is used. The m estimate is defined as

$$\frac{n_c + m \cdot p}{n + m} \tag{7.}$$

where  $n_c$  is the number of occurrences of a specific value, n is the number of values m is the equivalent sample size, and p is the prior estimate of the probability. In the absence of any statistics pertaining to distributions of word use, a uniform distribution of the words is assumed. Thus, m will be the number of words in the vocabulary of all documents in the training set and p will be 1/n [1]. Therefore, the estimate of the probability for a given word  $n_i$  for corpus j is defined as:

$$\frac{n_i + 1}{n_j + m} \tag{8.}$$

where  $n_i$  is the number of occurrences of the ith word in corpus j,  $n_j$  is the number of words in corpus j, and n is the number of words in the vocabulary.

The training documents used to create to corpora, one consisting of documents penned by the target author and the other consisting of documents written by all other authors. The vocabulary is built by extracting all unique tokens (words) from both corpora. The probabilities for each word occurring in each corpora are then computed. Once the probabilities are computed, the test documents are then applied.

Figures 1 and 2 represent the pseudo-code for these algorithms. The actual program was implemented in the java programming language.

```

For each document
  If document.author = target
    add document to corpus A
  Else
    add document to corpus B
  End If
End For

Set Vocabulary = {}
For each Word in corpus A and B
  If Word is in Vocabulary
    Add word to Vocabulary
  End If
End For

Set N = number of words in Vocabulary
Set Nd = Number of Documents
Using Corpus A
  Set nd = number of documents in corpus
  Set nw = number of distinct words in corpus
  Set pc = nd/Nd
  For each distinct Word in corpus
    Set ni = number of times Word appears in corpus
    Set  $pi = \eta (ni+1)/(nw+n)$ 
  End For
End Using
Repeat for corpus B

```

**Figure 1: Pseudo-Code for Training**

```

Set T = Test Document
Using Corpus A
  Set P = pc
  For each word in Test-Documents that is in Vocabulary
    P = P*pc
  End For
End Using
Repeat for Corpus B
If A > B choose A else choose B

```

**Figure 2: Pseudo-code for evaluation**

## Experimental Description

### Data

Data was provided by the Oxford Text Archive, <http://ota.ahds.ac.uk/>. The following texts were utilized to train and test the algorithm:

1. Jane Austen - Emma
2. Jane Austen - Love and Friendship
3. Jane Austen - Mansfield Park
4. Jane Austen - Northanger Abbey
5. Jane Austen - Persuasion
6. Jane Austen - Pride and prejudice
7. Jane Austen - Sandition
8. Jane Austen - Sense and sensibility
9. Jane Austen - The Watson
10. Jane Austen - Miscellaneous Correspondences
11. Charlotte Bronte - Jane Eyre
12. Robert Browning - Dramatic lyrics
13. Robert Browning - A blot in the 'scutcheon
14. Robert Browning - Men and women
15. Robert Browning - Essay on Chatterton
16. Charles Darwin - The voyage of the Beagle
17. George Eliot - Silas Marner
18. George Eliot - Janet's repentance
19. George Eliot - Mr. Gilfil's love story
20. George Eliot - The sad fortunes of the Reverend Amos Barton
21. Nathaniel Hawthorn - The house of the seven gables
22. Nathaniel Hawthorn - The scarlet letter
23. Nathaniel Hawthorn - The Blithedale romance
24. Henry James - The ambassadors
25. Henry James - The turn of the screw
26. Henry James - Daisy Miller
27. Herman Melville - Billy Buddy
28. Herman Melville – Bartleby
29. Karl Mark - Manifesto of the Communist Party
30. Mary Shelly -Frankenstein
31. Bram Stoker – Dracula
32. Bram Stoker - Dracula's guest
33. Leo Tolstoy - Anna Karenina
34. Leo Tolstoy - War and peace
35. Mark Twain – The Adventures of Huckleberry Fin
36. Mark Twain - The tragedy of Pudd'nhead Wilson
37. Mark Twain - The adventures of Tom Sawyer
38. Mark Twain - Roughing it
39. Mark Twain -Tom Sawyer abroad
40. Mark Twain - What is man? and other essays of Mark Twain
41. Irving Washington - The legend of Sleepy Hollow

The texts were provided in ASCII text format, tagged with SGML tags to provide information about the text. The header contains information about the author, publisher, publication date, encoding mechanisms, and notes from the corpus publishers. The header provides valuable information used to determine in which training corpus the document belongs, however, provides little other information and is stripped from the document prior to training.

The training documents are used to generate two corpora, one containing only documents written by Jane Austen and the other containing documents written by all other authors in the training set. These two corpora are then used to compute the probability of each word occurring in each of the corpora. Once the probabilities are computed, the test documents are evaluated using these probabilities.

## Experimental Results

The texts utilized in this experiment were divided into training and testing sets. The training sets consisted of all but two of the documents authored by Jane Austen five documents authored by the remaining authors with three of the authors providing other texts used in the training set and two of the authors being new to the program. Once the Naive Bayes algorithms were trained, the classifier was testing using randomly selected excerpts from the testing texts.

		Actual	
		True	False
Predicted	True	99	6
	False	1	144

Figure 3: Confusion Matrix

The total number of words in the training set was just over 2.3 million. The total number of words in the vocabulary was 56,000. When the experiment was run, the application classified the documents with 97% accuracy. The confusion matrix shows a total of 250 training documents, 100 written by Jane Austen and 150 written by other authors. From this test set, there was only one false positive and one false negative.

## Conclusion

The research indicates that the use of the Naïve Bayes classifier is well suited for the identification of authors of long texts. This particular study resulted in 97% accuracy in classifying documents as being written by a particular author. It should be noted that this is only a preliminary study and serves merely to prove the feasibility of work in the area. Additional studies are recommended in both the machine learning and linguistic fields. Additional tests should be run against a much larger document base, varying the author to be identified. Furthermore, it is recommended that the study be expanded to the linguistics field to better identify additional attributes that are likely to improve

classification accuracy. In addition, a detailed analysis of the results as compared to a comprehensive linguistic study would be helpful for justifying the conclusions drawn by the Bayesian classifier.

## References

- [1] Mitchel, Tom; Machine Learning, McGraw-Hill Companies, Inc. Boston, 1997.
- [2] Chai, K.; H. T. Hn, H. L. Chieu; "Bayesian Online Classifiers for Text Classification and Filtering", Proceedings of the 25th annual international ACM SIGIR conference on Research and Development in Information Retrieval, August 2002, pp 97-104
- [3] Sebastiani, Fabrizio "Machine Learning in Automated Text Classification", ACM Computing Surveys, Vo.l 34, No. 1, 2002. pp 1-47
- [4] Harold Borko , Myrna Bernick, "Automatic Document Classification", Journal of the ACM (JACM), v.10 n.2, p.151-162, April 1963
- [5] William W. Cohen , Yoram Singer, "Context-Sensitive Learning Methods for Text Categorization", ACM Transactions on Information Systems (TOIS), v.17 n.2, p.141-173, April 1999
- [6] Dumais, Susan; John Platt; David Heckerman; Mehran Sahami; "Inductive Learning Algorithms and Representations for Text Categorization", Proceedings of the seventh international conference on Information and knowledge management, p.148-155, November 02-07, 1998, Bethesda, Maryland, United States
- [7] Lanquillion, Carsten; "Little Words Can Make a Big Difference for Text Classification", Proceedings of the Eighth International Conference on Information and Knowledge Management November 1999, pp 538-544.
- [8] Basili, Roberto and Alessandro Moschitti "A Robust Model for Intelligent Text Classification", 13th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'01), 2001, pp 265-272.

[9] McCombe, Niamh Methods of Author Identification, Thesis – Trinity College, Dublin Ireland, online  
<http://www.cs.tcd.ie/courses/csll/mccombe0102.pdf>