Performance of Naïve Bayes Classifier – Multinomial Model on Different Categories of Documents

Hetal Doshi
M. E. EXTC Semester III
K. J. Somaiya COE,
Vidyavihar, Mumbai

Maruti Zalte
Assistant Professor
K. J. Somaiya COE,
Vidyavihar, Mumbai

ABSTRACT
Automatic sorting of documents is progressively becoming vital because manual handling and organization of documents is not a feasible solution as it can be very time consuming given the number of documents. One of the machine learning applications – text classification which is employed for document classification is explored in this paper. Generative learning algorithm – Naïve Bayes classifier is discussed in this paper. Documents from the 20 Newsgroups dataset are distributed in two groups. Group 1 consists of relatively unrelated two categories of documents and group 2 consists of relatively similar two categories of documents. Naïve Bayes classifier - Multinomial model is implemented to perform classification on both groups and it is observed that Accuracy can be improved with increasing the training set size for both the groups and Classification accuracy is higher for category of documents with lower similarity.

1. INTRODUCTION
Machine learning is a field of Artificial Intelligence (AI) that deals with the conceptualization, design and development of techniques or algorithms that will let computers to understand behavior based on given data. The learning algorithm takes advantage of existing examples to capture characteristics/features of interest of their unknown underlying probability distribution and thus generalize efficiently from the training set to produce a useful output for every new input condition.

Modern applications of Machine learning are learning from biological sequences, learning from text and learning in complex environments such as web [8]. In this paper, learning from text is used for document classification. Text classification also known as text categorization deals with the assignment of text to a particular category from a predefined set of categories based on the words of the text. Text classification finds immense applications in information management tasks. Some of the applications are like document classification based on defined vocabulary, sorting emails as spam or non spam, or sorting emails into various folders, documents filtering, topic identification etc. In this paper, document classification is implemented. Generally Machine learning tasks can be classified into supervised learning and unsupervised learning.

Supervised learning is also known as learning from examples. Here the algorithm has to build a function that is actually the description of a model. The system is provided with a set of examples. The output of the built function for each of the training examples is also available. The algorithm has to discover the behavior of the model based on the output of the function. A model is built using a subset of the data (training set) and evaluation of this model is done on the remaining data which is called as test set [8].

Unsupervised learning is also known as learning from observation. In unsupervised learning the system has to discover any patterns (or clusters) based only on the common properties of the training examples without knowing how many or even if there are any patterns [8].

Text classification can be achieved using various classifiers like Adaboost, Support Vector Machines [6]. In this paper we will explore document classification using Naïve Bayes classifier which adopts supervised learning scheme.

This paper is organized as follows. In the next section, types of classifier are discussed followed by description of Naïve Bayes Classifier. Section III describes the Multinomial model of Naïve Bayes Classifiers. Multinomial model implementation and results are given in the section IV followed by related work and conclusion in section V and VI respectively.

2. CLASSIFIER
A classifier is a function that maps input feature vectors x Є X to output class labels y = (1,…C), where X is the whole feature space. Aim is to learn and understand the function f from available labeled training set of N i/p – o/p pairs (xn, yn), n = 1…N. This is called as supervised learning as opposed to unsupervised learning which doesn't comprise of labeled training set. To achieve this, we use fundamentals of probability. The ways of implementing classifier is as follows [2]

1. Discriminating model: The aim is to learn function that computes the class posterior p(y/x). This is named as discriminative model as it discriminates between different classes given the input.[2]
2. Generative model: The aim is to learn the class conditional density p(x/y) for each value of y and also learn class priors p(y) and then by applying Bayes rule, [2]

$$p(y/x) = \frac{p(x/y).p(y)}{p(x)} \qquad (1)$$

This is known as generative model as it specifies a way to generate the feature vector x for each possible class y.

2.1 Naïve Bayes Classifier
The probabilistic technique used in basic Bayesian classifiers assumes the way data is generated and offers a probability model that symbolizes these assumptions made. The parameters of the generative model are estimated using a set of labeled training examples and every new examples is classified using Bayes rule by selecting the class with the highest probability [3].

The Naïve Bayes classifier is the simplest of these Bayesians models. This model assumes that all the attributes of the training examples are independent of each other given the

context of the class. But this assumption obviously doesn't hold true in natural language text. There are various types of dependencies observed between words which are often induced by semantics, pragmatic, syntactic and conversational structure inherent in the given text example [1]. Also the assumptions about the distribution of words in documents are violated in real world examples. Still it is observed that Naïve Bayes performs well and is often compared with other sophisticated classification algorithms. This contradiction is well explained by the fact that classification estimation is only a function of sign of function estimation. The function approximation can be affordably poor as the classification accuracy remains high. [3]

The parameters for each attribute can be learned separately, courtesy independent assumption and this significantly simplifies learning especially when the number of attributes is large. In application of document classification, the attributes of the training/testing examples to be classified are words and number of different words can be quite large [3].

Defining the aim of document classifier [4]: If document D is to be classified, the learning algorithm should be able to classify it in required category Ck

$$p\left(\frac{Ck}{D}\right) = \frac{p\left(\frac{D}{Ck}\right).p(Ck)}{p(D)}$$

$$\alpha\, p(D/Ck).p(Ck) \qquad (2)$$

Hence it is required to have a procedure to represent Document D and estimating the probabilities $p(D/Ck).p(Ck)$. There are two models for representing documents and calculating probabilities.

2.2 models of Naïve Bayes Classifier

Multivariate Bernoulli model: A document is represented by a binary feature vector, whose elements (1/0) indicate presence or absence of a particular word in a given document. In this case the document is considered to be the event and the presence and absence of words are considered as attributes of the event. This approach is more traditional in the domain of Bayesian networks particularly implemented for tasks with fixed number of attributes [3].

Multinomial model: A document is represented by an integer feature vector, whose individual elements indicate frequency of corresponding word in the given document. Thus the individual word occurrence is considered to be events and document is considered to be collection of word events [3].

The Bernoulli model is build on only presence and absence of words in the given document and the frequency of words in a document is not captured. This is one of the most important distinguishing factors of Multivariate Bernoulli and Multinomial model. In Multinomial model, document feature vector captures word frequency information and not just its presence or absence. Ref. [3] suggest that with a large vocabulary, multinomial model is more accurate than the multivariate Bernoulli model for many classification tasks.

3. MULTINOMIAL MODEL

In Multinomial generative model, a biased V sided dice is considered and each side of the dice represents the word Wt with probability $p(Wt/Ck)$. Thus at each position in the document, a dice is rolled and a word is inserted. Thus a document is generated as bag of words which includes which words are present in the document and their frequency of occurrence.

Mathematically this can be achieved by defining Mi as multinomial model feature vector for the i[th] document Di. Mit is the frequency with which word Wt occurs in document Di and $ni = \sum_t Mit$ is the total number of words in Di. Using word frequency information from the multinomial model feature vectors can be used for estimating $p(Wt/Ck)$.

Naive Bayes approximation: Generation of documents is modeled by multinomial distribution [4]

$$p(Mi/Ck) = ni! * \prod_{t=1}^{V} p(Wt/Ck)^{Mit} / \prod_{t=1}^{V} Mit! \qquad (3)$$

If likelihoods of the same document for different classes are compared, then

$$p(Mi/Ck) \alpha \prod_{t=1}^{V} p(Wt/Ck)^{Mit} \qquad (4)$$

As $X^0 = 1$, the above product is affected by words that are present in the Di. If Di is sequence of l words, w1, w2, w3…wl [4]

$$p(Mi/Ck) \alpha \prod_{h=1}^{l} p(wh/Ck) \qquad (5)$$

A: Defining the parameters of the Multinomial model

$Zik = 1$ When Di has class / category Ck and $Zik = 0$, otherwise. Let N be the total number of documents then, [4]

$$p(Wt/Ck) \qquad (6)$$

$$= \sum_{i=1}^{N} Mik * Zik \,/\, \sum_{s=1}^{V} \sum_{i=1}^{N} Mis\, Zik$$

The relative frequency of Wt in documents of class Ck w. r. t. the total number of words in documents of that class is estimated as $p(Wt/Ck)$ and priors are estimated as

$$p(Ck) = \frac{Nk}{N} \qquad (7)$$

B: Steps in building a Multinomial model

1. Defining the vocabulary V, the number of words which provides the dimension of the feature vector
2. Scan the training set to obtain following counts
   N: Number of documents
   Nk: Number of documents of class Ck, for all classes

Mit: The frequency of word Wt in document Di for all words in V and all documents.

3. Estimate likelihoods $p(Wt/Ck)$ and priors $p(Ck)$

Once the training is performed and parameters are ready, for every new unlabelled document, Dj, the posterior probability for each class is estimated as [4]

$$p(Ck/Dj) = p(Ck/Mj) \quad (8)$$

$$\alpha\ p(Mj/Ck) * p(Ck)$$

$$\alpha\ P(Ck) * \prod_{t=1}^{V} p(Wt/Ck)^{Mit}$$

$$\alpha\ p(Ck) * \prod_{h=1}^{len(Dj)} p(wh/Ck)$$

C: Laplace smoothing

If a particular word doesn't appear in the class Ck, then the probability calculated by equation (6) will become zero. But this doesn't mean it cannot occur for any documents of that class. Stating the problem more broadly, it is statistically not appropriate to estimate the probability of some event to be zero just because it hasn't been seen before in the finite training set available [7].

To avoid this problem, Laplace smoothing is applied. In this one is added to the count of each word type and denominator is modified to compensate for additional count of one for each word. Thus Laplace smoothing is incorporated in the implementation of the multinomial model described in the next section and it modifies the equation 6 as follows,

$$p(Wt/Ck) \quad (9)$$

$$= 1 + \sum_{i=1}^{N} Mik * Zik/|V| + \sum_{s=1}^{V}\sum_{i=1}^{N} Mis\ Zik$$

4. EXPERIMENTAL RESULTS

To evaluate the performance of Multinomial Model Naïve Bayes Classifier, the 20 Newsgroup Dataset is used. The 20 Newsgroup Dataset is a common standard used for testing text classification algorithm.

The dataset, introduced in (Lang 1995), contains approximately 20,000 newsgroup posts which are divided across 20 different newsgroup.

There are varied types of newsgroups used in the given dataset. Some of these newsgroups are closely related and some are highly unrelated. In this paper, the Multinomial model Naïve Bayes Classifier is implemented on two groups of dataset obtained from the 20 Newsgroups Dataset. First group consists of 960 documents from category alt.athesim (480 documents) and category computer graphics (480 documents). Second group consists of 960 documents from category computer graphics (480 documents) and category computer OS MS-Windows misc (480 documents). First group categories are much unrelated while second group categories are related. These two groups are used as training set. For testing the performance of the classifier on both the groups, 636 unlabelled documents for group 1 and group 2 are used including equal number of documents from both categories for both the groups.

The training algorithm implemented in MATLAB is used to calculate $p(Wt/Ck)$ and priors $p(Ck)$ for both the categories of documents in both groups i.e. $p(Wt/C1)$ & $p(Wt/C2)$ and $p(C1)$ &$p(C2)$. Probability $p(C1)$ & $p(C2)$ is 0.5 as equal number of documents are used for both the categories. Dictionary size |V| for the two categories of documents in group 1 is 54708. Dictionary size |V| for the two categories of documents in group 2 is 55493

The testing algorithm implemented in MATLAB is used to classify 636 documents in one of the two categories. Classification results obtained are compared with the correct labeling provided in the dataset from the 20 Newsgroups to find out how many documents are correctly classified.

The experiment is further extended to perform classification accuracy check on the same testing examples using a smaller training set, once with using 480 documents and then using 240 documents for both the groups. In both cases equal number of documents from both the categories is used. The results obtained for group 1are tabulated below in table 1 and the results obtained for group 2are tabulated below in table 2

TABLE I
Accuracy evaluation for different training set sizes for group 1

Sr. No	Training set	Testing set	Number of documents classified correctly	Accuracy %
1	960	636	620	97.48
2	480	636	607	95.44
3	240	636	497	78.14

It is observed in table I that for training examples with 960 documents, classification accuracy obtained is 97.48% and the accuracy drops to 95.44% for training examples with 480 documents and further to 78.14% for training examples with 240 documents.

TABLE II
Accuracy evaluation for different training set sizes for group 2

Sr. No	Training set	Testing set	Number of documents classified correctly	Accuracy %
1	960	636	549	86.32
2	480	636	535	84.11
3	240	636	492	77.35

It is observed in table II that for training examples with 960 documents, classification accuracy obtained is 86.32% and the accuracy drops to 84.11% for training examples with 480 documents and further to 77.35% for training examples with 240 documents.

It is also seen that accuracy results are comparatively higher in group 1 than group 2. With the above results, it can be analyzed that

- Accuracy can be improved with increasing the training set size for both the groups.
- Classification accuracy is higher for category of documents with lower similarity.

To improve accuracy further, every new document that is correctly classified can be added to the training set. Also a user can manually move the incorrectly classified document into the appropriate folder which can be added to the training document. This will increase the size of training examples on the continual basis which can improve accuracy as suggested in the table above.

5. RELATED WORK

Amount of information: In Ref. [5], both the models are run on the test data which is divided into three categories: proverb; proverb + meaning and proverb + meaning + example and the results presented in Ref. [5] suggests that multinomial model provides better accuracy and also suggests that when more information is provided, classification accuracy improves as it was exhibited by the results of the proverb + meaning + example data set. Thus better results can be obtained if more test and training data is used.

Accommodation of variance: In Ref. [3], it is suggested that multinomial model should be more accurate classification model for data sets that possess large variance in document length as it handles documents of varying length by incorporating the evidence of every occurrence of each appearing word. For such situations Multivariate Bernoulli model proves to be a poor fit for data with varying lengths as it is more likely for a word to occur in a long document irrespective of the class.

Presence of non text features: Non text features like number of recipients of the email (considering applications related to email classification or sorting) can be included exactly as word features in case of multivariate Bernoulli model. But in case of multinomial model this is not as simple. This is due to the fact that if non text features are added to vocabulary, then event spaces for different features would compete for the same probability mass even though they are mutually exclusive [3].

Repeating word probability: Although multinomial model treats each occurrence of words in a given document independent to any other occurrence of the same word in the given document. In real world this is not true. Repeated occurrence of the same word in the given document is dependent. When the word occurs for the first time, it is likely to occur again i.e. the probability of the second occurrence is much higher than that of the first occurrence. This fact is overlooked in multinomial model thus underestimating the probability of documents with multiple occurrences of the same word [1].

6. CONCLUSION

With the growing usage of information, automatic document classification can assist and speed up the process of information handling and management. There are various efficient and sophisticated algorithms for implementing classification task but Naïve Bayes classifier is very popular due to its simplicity and effectiveness. From the results it is observed that the performance of the classifier is dependent on the training set size and larger training set size can significantly increase the accuracy of classification task. Also degree of similarity between the two categories of documents influences the accuracy results. To further improve the accuracy, other aspects like amount of information, variance in document length, presence of non text features and repeating word probability can be considered and modify the classifier accordingly.

7. REFERENCES

[1] Karl - Michael "Techniques for Improving the performance of Naïve Bayes for text Classification" University of Passau, department of general Linguistics Innstr. 40, 94032 Passau, Germany

[2] Kevin P. Murphy, " Naïve Bayes classifier", Department of Computer Science, University of British Columbia

[3] Andrew McCallum and Kamal Nigam, "A Comparison of Event Models for Naïve Bayes Text Classification ", In: Learning for Text Categorization: Papers from the AAAI workshop, AAAI pressc(1998) 41 – 48 Technical report Ws – 98 - 05

[4] "Text Classification using Naïve Bayes", Steve Renals, Learning and Data lecture 7, Informatics 2B, http://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b11-learnlec07-nup.pdf

[5] S. A. Noah and F. Ismail, "Automatic Classification of Malay Proverbs using Naïve Bayesian Algorithm", in Information Technology Journal 7 (7): 1016-1022, 2002 ISSN 1812-5638

[6]Carl Liu, "Experiments on Spam Detection with Boosting, SVM and Naïve Bayes", CMPS 242, Final project, Winter 2008, UCSC

[7] "Generative learning algorithm", lecture notes2 for CS229, Department of Computer Science, University of Stanford. Available online: http://www.stanford.edu/class/cs229/notes/cs229-notes2.pdf

[8] George Tzanis, Ioannis Katakis, Ioannis Partalas, Ioannis Vlahavas, " Modern Applications of Machine Learning", in Proceedings of the 1st Annual SEERC Doctoral Student Conference – DSC 2006