News Classification System Using Naïve Bayes

A thesis submitted in partial fulfillment of the requirement for the

Degree of Master of Computer Application

of

Jadavpur University

By

SOHINI ACHARYA

Registration Number: **133669** of 2015-2016

Examination Roll Number: MCA186007

Under the Guidance of

Dr. Kamal Sarkar

Professor

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

May, 2018

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE OF RECOMMENDATION

This is to certify that the thesis entitled “NEWS CLASSIFICATION SYSTEM USING NAÏVE BAYES” has been satisfactorily completed by Sohini Acharya (University Registration No.: 133669 of 2015-16, Examination Roll No.: MCA186007).It is a bonafide piece of work carried out under my guidance and supervision and be accepted in partial fulfillment of the requirement for the Degree of Master of Computer Application , Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, Kolkata.

____

Dr. Kamal Sarkar (Thesis Supervisor)
Professor
Department of Computer Science and Engineering
Jadavpur University, Kolkata-700032

Countersigned

____

Prof Ujjwal Maulik
Head, Department of Computer Science and Engineering,
Jadavpur University, Kolkata-700032.

__________________________________[1]

Prof. Chiranjib Bhattacharjee
Dean, Faculty of Engineering and Technology,
Jadavpur University, Kolkata-700032.

FACULTY OF ENGINEERING AND TECHNOLOGY
JADAVPUR UNIVERSITY

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled "NEWS CLASSIFICATION SYSTEM USING NAIVE BAYES" is a bonafide record of work carried out by Sohini Acharya in partial fulfilment of the requirements for the award of the degree of Master of Computer Application in the Department of Computer Science and Engineering, Jadavpur University during the period of February 2018 to May 2018. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn there in but approve the thesis only for the purpose for which it has been submitted.

____

Signature of Examiner

Date:

____

Signature of Supervisor

Date:

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that this thesis entitled "NEWS CLASSIFICATION SYSTEM USING NAIVE BAYES" contains literature survey and original research work by the undersigned candidate, as part of his Degree of Master of Computer Application.

All information in this document has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Sohini Acharya
University Registration No. : 133669 of 2015-16
Examination Roll No. : MCA186007

Thesis Title: News Classification System Using Naive Bayes

____

Signature

Date:

ACKNOWLEDGEMENT

First and foremost, I would like to start by thanking God Almighty for showering me with the strength, knowledge and potential to embark on this wonderful journey and to persevere and complete the embodied research work satisfactorily.

I am pleased to express my deepest gratitude to my thesis guide, **Dr. Kamal Sarkar**, Department of Computer Science and Engineering, Jadavpur University, Kolkata for his invaluable guidance, constant encouragement and inspiration during the period of my dissertation.

I am highly indebted to **Jadavpur University** for providing me the opportunity and the required infrastructure to carry on my thesis.

I am thankful to all the teaching and non-teaching staff whose helping hands have smoothed my journey through the period of my research.

Last but not the least; I would like to thank my family members, classmates, seniors and friends for giving me constant encouragement and mental support throughout my work.

____

Sohini Acharya
University Registration No. : 133669 of 2015-16
Examination Roll No. : MCA186007
Master of Computer Application
Department of Computer Science and Engineering
Jadavpur University

ABSTRACT

Document classification has become an emerging technique in the field of research due to the abundance of documents available in digital form. Document classification can be used to organize data into smaller and meaningful classes. Correctly identifying a document into a particular class is still a huge challenge particularly in e-news industry as very few work has been done in this field . In this paper we have done document classification using Naïve Bayes classifier. For the last few years, text mining has been gaining significant importance. Since Knowledge is now available to users through variety of sources i.e. electronic media, digital media, print media, and many more. Due to huge availability of text in numerous forms, a lot of unstructured data has been recorded by research experts and have found numerous ways in literature to convert this scattered text into defined structured volume, commonly known as text classification. Focus on full text classification i.e. full news, huge documents, long length texts etc. is more prominent as compared to the short length text. In regards to the various classifying approaches, Naïve Bayes is potentially good at serving as a document classification model due to its simplicity. I present a system for automatic categorization of news items into a standard set of categories. The system has been built specifically for news stories written in English language taken from news articles of respected Dailies. The aim of this paper is to highlight the performance of employing Naïve Bayes in document classification. In this paper the document is classified into one of the five classes i.e. cricket, football politics , entertainments and business. To build and evaluate the classification model, a total 125 documents is split into two datasets, namely training set and testing set, in which 80% of the documents is used as training set whereas the remaining 20% is used as the testing set. Results show that Naïve Bayes is a good classifiers.

Keywords — Document classification, Naive Bayes.

Contents

Introduction

Text analysis, as a whole, is an emerging field of study. Fields such as Marketing, Product Management, Academia, and Governance are already leveraging the process of analysing and extracting information from textual data. In a previous post, we discussed the technology behind Text Classification, one of the essential parts of Text Analysis. Text classification or Text Categorization is the activity of labelling natural language texts with relevant categories from a predefined set. In laymen terms, text classification is a process of extracting generic tags from unstructured text. These generic tags come from a set of pre-defined categories. Classifying your content and products into categories help users to easily search and navigate within application. A series of challenges have recently emerged in the field of document classification due to the advancement of web and social network technology .Text categorization is the procedure of labelling a textual document with one or more predefined categories. Due to proliferation of easily available textual data within the past decade or so, the interest in automated text categorization has steadily increased. The applications range from automatic document indexing for information retrieval systems, document organization, text filtering, word sense disambiguation, categorization of web pages and, most recently, spam filters. One particularly interesting application area is the news industry. In the news industry metadata is a very important part of a news item. Fast spreading of Internet decreased complexity of news exchange, which resulted in dramatic increase in the number of available news sources and the volume of news items an average recipient received every day. As a paradox, that led to over flooding of consumers with the information and actually decreased its usability. On the other hand, speed has always been very important factor in the news industry. Due to the inability to process all the content they receive fast enough, news recipients have to rely on metadata to find out the content they are interested in, which means that it is very important for metadata to be consistent, accurate and comprehensive. It could be even said that news story with inaccurate or insufficient metadata does not exist, because it will rarely reach the consumers no matter how important its content might be. Apart from manual classification and hand-crafted rules, there is a third approach to text classification, namely, machine learning-based text classification. In machine learning, the set of rules or, more generally, the decision criterion of the text classifier, is learned automatically from training data. This approach is also called statistical text classification if the learning method is statistical. In statistical text classification, we require a number of good example documents (or training documents) for each class. The need for manual classification is not eliminated because the training documents are labelled under the supervision of a supervisor i.e. the person who defines the classes and labels each document with its class. Hence this type of learning is called supervised learning. Document classification can be done using any statistical approach. Naive Bayes is the de-facto standard text classifier. It is commonly used in practice and is a focus of research in text classification. In this paper I have done document classification using Naïve Bayes approach. The Naïve Bayes approach is particularly appealing because of its simplicity, elegance , robustness as well as the speed with which it can be applied to do the classification task. It is one of the oldest formal classification algorithms and yet even in its simplest form it is surprisingly effective.

Literature Survey

This section, describe the related work of document classification using different statistical approach. Automatic document classification studies are gaining more interests in text mining research recently. Consequently, an increasing number of approaches have been developed for accomplishing such purpose, including k-nearest-neighbour (KNN) classification, Naïve Bayes classification, support vector machines (SVM), decision tree (DT), neural network (NN), and maximum entropy. Among these approaches, the Naïve Bayes text classifier has been widely used because of its simplicity in both the training and classifying stage. Although it is less accurate than other discriminative methods (such as SVM), numerous researchers proved that it is effective enough to classify the text in many domains. It has been shown that Naive Bayes Classifier can be used effectively for text classification in Indian languages. Naïve Bayes classification has been used to classify text in different languages before with good accuracy.

Some of the related works are:

- Seongwook Youn et al. (2007) proposed a comparative study for email classification. Neural Network, SVM, Naive Bayesian and J48 classifiers are used to filter spam from the datasets of emails. Neural network consists of data pre-processing, data training and testing. Feature selection extracts more informative and removing irrelevant and redundant features. They feeded pre-processed features to the NN and email classifier got generated through the NN. In the third step of testing the email classifier is used to verify the efficiency of NN. An error back propagation algorithm is used for the experiment. Through SVM successful implementation of learning and generalization tasks can be achieved. SVMs learn by examples and each example consists of a number of data points followed by a label, which is in the two class classification. They are +1represents one state and -1 represents another state. The optimum hyper plane separates the two classes. Support Vector minimizes the distance between the closest +1 and -1points. It divides two separate classes which are generated from training examples. SVM introduced a separate hyper plane which maximizes the margin between two classes for obtaining a well generalized test data. Naive Bayesian classifier is an effective classifier based on Bayesian theorem and theorem of total probability. J48 is a decision tree creates a binary tree used for classification of legitimate and spam. They evaluated the four classifiers on different datasets and different features.

$$\text{Accuracy(\%)} \quad = \quad \frac{\text{Correctly\_ Classified\_ Emails}}{\text{Total\_ Emails}} * 100$$

For evaluating the performance precision and recall were used as metrics for email classification. They suggested J48 and NB classifiers obtained a better result and accuracy than SVM and NN classifiers.

- Ali Ciltik et al. (2008) proposed a method of spam email filtering methods with high accuracies and low time complexities. They took Turkish mails for their research. They used PC-KIMMO system, a morphological analyser to extract root forms of words as input and produce parse of words as output. This method is based on the n-gram approach and a heuristic. They developed two models, a class general model and an e-mail specific model. The general model classifies the mail as spam or legitimate by using Bayes rule. The determination is done in the second model where the correct class of a message gets compared it with the similar previous message for matching. The third model is a combined perception refined model. It is achieved by combination of above two models. Free word order is used for ordering the word in fixed order for n gram model. This spam filtering method is based on classifying text contents and raw contents of emails obtaining results from the categorization of data sets. They faced the increase of time complexity problem when handling the larger number of words. Adaboost ensemble algorithm is used to compare with its previous work. They performed extensive tests on various number datasets sizes and initial words. They have obtained a result of high success rates in both Turkish language and English.

- Han, Weili et al. (2008) proposed this method of automated individual white list approach is a tool used to build white list and automatically maintained by the naïve Bayesian classifier protects user’s web digital identities and also recognize the successful login process. AIWL is an efficient automated tool specializing in detecting phishing and pharming. AIWL checks the LUI information and recognize the phishing and pharming. If AIWL is installed in a machine it is difficult to fight against the Trojan horse and viruses. Although it was found that it had a synchronization problem when the user has many machines.

- Liu Pei-yu et al. (2009) suggested the method of improved Bayesian algorithm for filtering spam. KNN algorithm, SVM, decision tree, and improved Bayesian algorithm are used for classifying texts. KNN algorithm is a simple and accurate method for spam filtering by using the k nearest neighbour. SVM is also used for filtering spam and finds hyper plane to classify the legitimate and spam mails. It works with smaller training set. Decision tree is used for faster and simple classification which gives higher accuracy of judgment. Bayesian algorithm is a base and simple classification method classifies the mail as C legal and spam C rubbish. In the Bayesian method one feature is treated as independent of other. Improved naive Bayesian algorithm is a combination of Bayesian algorithm with boosting method, developed to reduce the rate of misjudgement and improve the accuracy of classification. Boosting is a universal learning algorithm. They treated the naive Bayesian algorithm as weaker learning algorithm and made it stronger by boosting it with boosting algorithm. And doing so they obtained better result by applying this boosted naive Bayesian algorithm for filtering spam.

- Chiristina.V et al. (2010) proposed a study on email spam filtering methods. They discussed about various spam identification methods and spam filtering techniques. In spam identification methods Whitelist/Blacklist, Naïve Bayesian analysis, Mail header analysis, and Keyword checking are discussed. In spam filtering techniques they discussed about the naïve Bayesian classifier, SVM, rule based, content based filters, K nearest neighbours, distributed adaptive blacklists, the multilayer networks, technique of search engines and technique of artificial immune system. They concluded that there is a need to develop a method to provide an ideal solution with 0% false positive and 0% false negative.

- Mehdi Samiei yeganeh et al. (2012) developed a model for fuzzy logic based machine learning approach for filtering spam. They discussed the methods of automatic spam filters like naive Bayes classifier, artificial immune classifier and fuzzy logic. They built a classification model from a set of pre-classified email instances by using fuzzy similarity approach. They used three stages for filtering spam. In the pre-processing stage the html tags are stripped off and stop word are removed from the mail. In the second stage of training they built a model based on the characteristics of each category in a pre-classified set of email messages. Each Sample message is labelled with a specific category and the message is then classified by comparing its fuzzy similarity measures. The functionality of the model got enhanced and also the feature identification of emails and deletion of spam mails on its own. They suggested that the fuzzy logic is adaptable for spammer tactics.

- Sivakumar (2012) proposed a paper on A Fuzzy Similarity Approach for Automated Spam Filtering and Naïve Bayes Classifier is a near-duplicate phenomenon of spams. SAG is focused on email layout and used html content in email. Structure abstraction generation process composed of three types. They are tag extraction, tag reordering and appending types. SAG captures the near-duplicate phenomenon of spams. They used SpTable and SpTrees to store large amounts of the email abstractions in reported spams. The values assigned to tags by Bayes theorem helps to find the tag as spam or not.

- Mehdi Samiei yeganeh et al. (2012) this paper discussed about the machine learning methods they are Naïve Bayes, Artificial Neural Networks, Artificial Immune System Classifier methods, and fuzzy logic method to filter the spam mails. They built a classification model from a set of pre-classified e-mail instances by using fuzzy similarity approach. Their method consists of preprocessing, training and classification stages. They concluded that their model is an enhanced fuzzy model for feature identification and deletion of spam mails.

- In existing classification algorithm such as Naïve Bayes, Centroid Based techniques are used for Punjabi Text Classification. And one new approach is proposed for the Punjabi Text Documents which is the combination Naïve Bayes (to extract the relevant features so as to reduce the dimensionality) and Ontology Based Classification (that act as text classifier that used extracted features). From these three algorithms, Hybrid Approach has better performance than others two, as features extracted with Naïve Bayes are less in count than others two which results in less computations and less time consuming.

- In Naïve Bayes classifier has been performed over Telugu News articles in four major classes: Politics, Sports, Business and Cinema; to about 800 documents. In this, normalized TFXIDF is used to extract the features. Without any stop word removal and morphological analysis, at the threshold of 0.03, the classifier gives 93% precision.

- In text classification is done using Vector Space Model and Artificial Neural network on morphological rich Dravidian classical language Tamil. The experimental results that was obtained show that Artificial Neural network model achieves 93.33% which is better than the performance of Vector Space Model which yields 90.33% on Tamil document classification.

- In work on automatic text categorization in Indian languages is presented. They have used purely corpus based machine learning techniques. The presented were completely language independent - no language specific knowledge is used. Experiments is performed on ten of the major Indian languages including Assamese, Bengali (Bangla), Gujarati, Hindi, Kannada, Malayalam, Oriya, Punjabi, Tamil and Telugu. Several machine learning techniques have been used including naive Bayes classifier, k-Nearest Neighbour classifier and SVMs to test the results although Naive Bayes yielded satisfactory results.

Methodology

In recent years, there has been growing interest in probabilistic methods for induction. Such techniques have a number of clear attractions: they accommodate the flexible nature of many natural concepts; they have inherent resilience to noise; and they have a solid grounding in the theory of probability. Moreover, experimental studies of probabilistic methods have revealed behaviours that are often competitive with the best inductive learning schemes. Although much of the recent work on probabilistic induction (e.g., Anderson & Matessa, 1992; Cheeseman et al., 1988; Fisher, 1987; Hadzikadic & Yun, 1989; McKusick & Langley, 1991) has focused on unsupervised learning, the same basic approach applies equally well to supervised learning tasks. Supervised Bayesian methods have long been used within the field of pattern recognition (Duda & Hart, 1973), but only in the past few years have they received attention within the machine learning community (e.g., Clark & Niblett, 1989; Kononenko, 1990, 1991; Langley, Iba, & Thompson, 1992).

Naïve Bayes Methodology in layman's terms:

Starting more than half a century ago, scientists became very serious about addressing the question: "Can we build a model that learns from available data and automatically makes the right decisions and predictions?" Looking back, this sounds almost like a rhetoric question, and the answer can be found in numerous applications that are emerging from the fields of pattern classification, machine learning, and artificial intelligence.

Data from various censoring devices combined with powerful learning algorithms and domain knowledge led to many great inventions that we now take for granted in our everyday life: Internet queries via search engines like Google, text recognition at the post office, barcode scanners at the supermarket, the diagnosis of diseases, speech recognition by Siri or Google Now on our mobile phone, just to name a few.
One of the sub-fields of predictive modelling is supervised pattern classification; supervised pattern classification is the task of training a model based on labelled training data which then can be used to assign a pre-defined class label to new objects. One example that we will explore throughout this article is news filtering via naive Bayes classifiers in order to predict whether a new news article can be categorized in one of the five classes that we have considered. Naive Bayes classifiers, a family of classifiers that are based on the popular Bayes' probability theorem, are known for creating simple yet well performing models, especially in the fields of document classification and disease prediction.

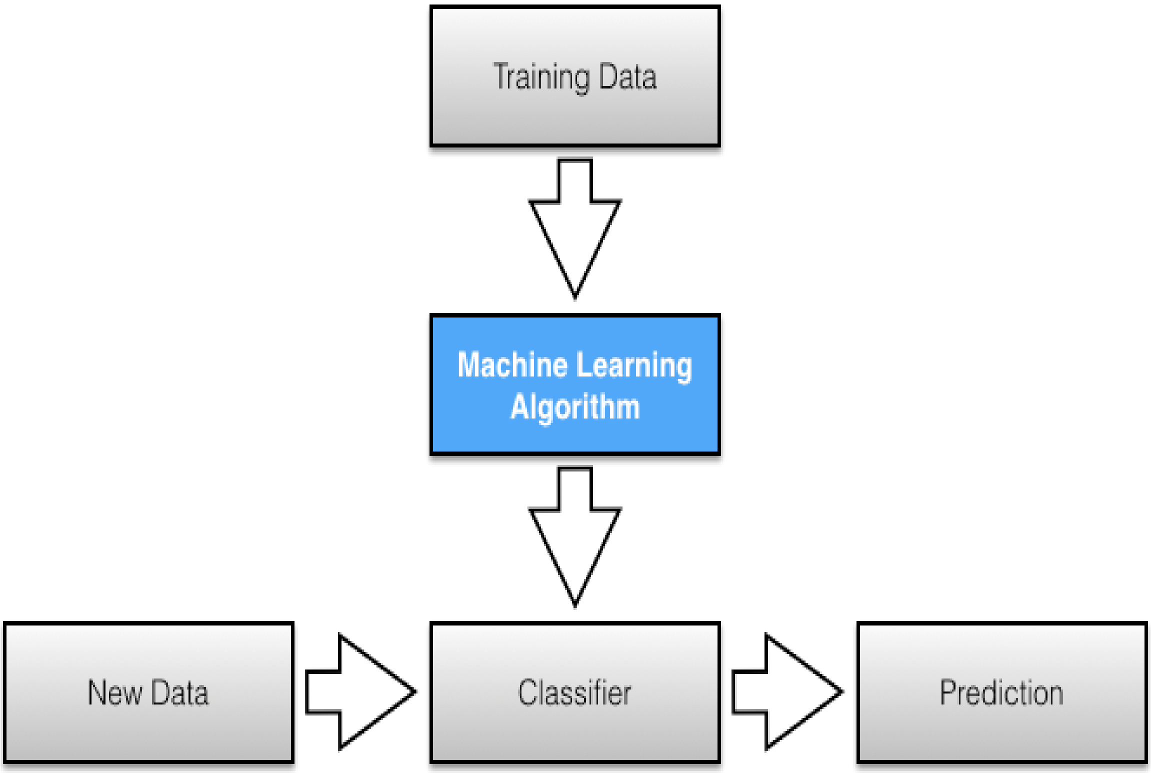

Figure 1. Naive Bayes Model

Let us take a closer look at the probability model of the naive Bayes classifier and apply the concept to a simple toy problem though a diagrammatic representation:

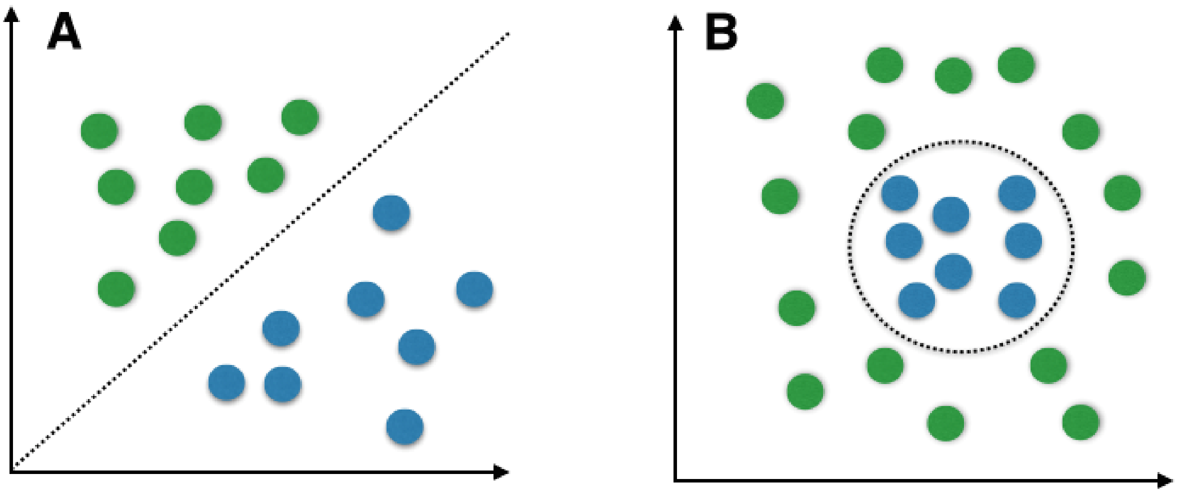

Figure 2. *Linear (A) vs. non-linear problems (B). Random samples for two different classes are shown as coloured spheres, and the dotted lines indicate the class boundaries that classifiers try to approximate by computing the decision boundaries. A non-linear problem (B) would be a case where linear classifiers, such as naive Bayes, would not be suitable since the classes are not linearly separable. In such a scenario, non-linear classifiers (e.g. Instance-based nearest neighbour classifiers) should be preferred.*

Modes of Naïve Bayes Classifier

1) **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.

2) **Multinomial model:**

A document is represented by an integer feature vector, whose 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. Multinomial model is more accurate than the multivariate Bernoulli model for many classification tasks because it considers the frequency of the words too.

3) **Probabilistic Model:**

Consider D be the set of documents and C be the set of classes. The probability of assigning a document d to a class c is given by:

$$C_{NB}=\text{argmax}_{cj€C}\ P(c|d)= \text{argmax}_{cj€C}\frac{P(c)P(d|c)}{P(d)}$$

As P(d) is independent of the class, it can be ignored.

$$C_{NB}=\text{argmax}_{cj€C}\ P(c)P(d|c)$$

According to Naïve Bayes assumption,

$$P(d|c) = P(w_1|c)\ P(w_2|c)..\ P(w_d|c) =\prod_{1<k<d}P(w_k|c)$$

Therefore,

$$C_{NB}=\text{argmax}_{cj€C}\ P(c) *\prod_{1<k<d}P(w_k|c)$$

Where P(c) is the prior probability of the class c_j, which is calculated as ,$\frac{N}{n}$ where N is the total number of training documents in class c, n is the total number of training documents. P(c|d) is the posterior probability.

$$P(w_k|c) = \frac{T}{\sum_{t£V} T}$$

Where T is the number of occurrences of w in d from class c, is the total number of words in d from class c.

General approach to Naïve Classification:

The most straightforward and widely tested method for probabilistic induction is known as the naive Bayesian classifier .1 This scheme represents each class with a single probabilistic summary. In particular, each description has an associated class probability or base rate, $p(C_k)$, which specifies the prior probability that one will observe a member of class C_k. Each description also has an associated set of conditional probabilities, specifying a probability distribution for each attribute. In nominal domains, one typically stores a discrete distribution for each attribute in a description. Each $p(V_j \mid C_k)$ term specifies the probability of value Vj, given an instance of class ck.

In numeric domains, one must represent a continuous probability distribution for each attribute. This requires that one assume some general form or model, with a common choice being the normal distribution, which can be conveniently represented entirely in terms of its mean and variance. To classify a new instance I, a naive Bayesian classifier applies Bayes' theorem to determine the probability of each description given the instance,

$$P(C_i|I)=A/B;$$

where

$$A=P(C_i)P(I|C_i)$$

$$B=P(I)$$

However, since I is a conjunction of j values, one can expand this expression to

$$P(C_i|\Lambda v_j) =\frac{C}{D}$$

$C= P(C_i)P(\Lambda v_j|C_i)$

$$D=\sum_{k} P(\Lambda v_j|C_k)P(C_k)$$

where the denominator sums over all classes and where $P(\Lambda v_j|C_i)$ is the probability of the instance I given the class C_i. After calculating these quantities for each description, the algorithm assigns the instance to the class with the highest probability. In order to make the above expression operational one must still specify how to compute the term $P(\Lambda v_j|C_k)$. The naive Bayesian classifier assumes independence of attributes within each class which lets it use the equality

$$P(\Lambda v_j|C_k) =\prod_{J}(P(v_j|C_k)$$

where the values $p(v_j |C_k)$ represent the conditional probabilities stored with each class. This approach greatly simplifies the computation of class probabilities for a given observation. The Bayesian framework also lets one specify prior probabilities for both the class and the conditional terms. In the absence of domain-specific knowledge, a common scheme makes use of 'uninformed priors', wh1ch assign equal probabilities to each class and to the values of each attribute. However, one must also specify how much weight to give these priors relative to the training data. For example, Anderson and Matessa (1992) use a Dirichlet distribution to initialize probabilities and give these priors the same influence as a sin? le training instance. Clark and Niblett (1989) describe another approach that does not use explicit priors, but instead estimates $P(C_k)$ and $p(v_j | C_k)$ directly from their proportions in the training data. When no instances of a value have been observed, they replace the zero probability with $p(C_i)/N$, where N is the number of training cases.

Learning in the naive Bayesian classifier is an almost trivial matter. The simplest implementation increments a count each time it encounters a new instance along with a separate count for a class each time it observes an instance of that class. These counts let the classifier estimate $p(C_k)$ for each class C_k. For each nominal value, the algorithm updates a count for that class-value pair. Together with the second count lets the classifier estimate p(v_i| C_k). For each numeric attribute, the method retains and revises two quantities, the sum and the sum of squares which let it compute the mean and variance for a normal curve that it uses to find p(v_j |C_k)- In domains that can have missing attributes, it must include a fourth count for each class-attribute pair. In contrast to many induction methods, the naive Bayesian classifier does not carry out an extensive search through a space of possible descriptions. The basic algorithm makes no choices about how to partition the data, which direction to move in a weight space, or the like, and the resulting probabilistic summary is completely determined by the training data and the prior probabilities. Nor does the order of the training _instances have any effect on the output; the bas1c process produces the same description whether it operates incrementally or non-incrementally. These features make the learning algorithm both simple to understand and quite efficient.

Bayesian classifiers would appear to have advantages over many induction algorithms. For example, their collection of class and conditional probabilities should make them inherently robust with respect to noise. Their statistical basis should also let them scale well to domains that involve many irrelevant attributes. Langley, Iba, and Thompson (1992) present an average case analysis of these factors' effect on the algorithm's behaviour for a specific class of target concepts.

The experimental literature is consistent with these expectations, with researchers reporting that the naive Bayesian classifier gives remarkably high accuracies in many natural domains. For example Cestnik Kononenko, and Bratko (1987) included this ,method as a straw man in their experiments on decision-tree induction, but found that it fared as well as the more sophisticated techniques. Clark and Niblett (1989) reported similar results, finding that the naive Bayesian classifier learned as well as both rule-induction and decision-tree methods on medical domains. And Langley et al. (1992) obtained even stronger results, in which the simple probabilistic method outperformed a decision-tree algorithm on four out of five natural domains.

However, the naive Bayesian classifier relies on two important assumptions. First, this simple scheme posits that the instances in each class can be summarized by a single probabilistic description, and that these are sufficient to distinguish the classes from one other. If we represent each attribute value as a feature that may be present or absent, this is closely related to the assumption of linear separability in early work on neural networks. Other encodings lead to a more complex story, but the effect is nearly the same. Nevertheless, like perceptrons, Bayesian classifiers are typically limited to learning classes that can be separated by a single decision boundary.3 Although we have addressed this limitation in other work (Langley, 1 993), we will not focus on it here.

Another important assumption that the naive Bayesian classifier makes is that, within each class, the probability distributions for attributes are independent of each other. One can model attribute dependence within the Bayesian framework (Pearl, 1988), but determining such dependencies and estimating them from limited training data is much more difficult. Thus, the independence assumption has clear attractions. Unfortunately, it is unrealistic to expect this assumption to hold in the natural world. Correlations among attributes in a given domain are common. For example, in the domain of medical diagnosis, certain symptoms are more common among older patients than younger ones, regardless of whether they are ill. Such correlations introduce dependencies into the probabilistic summaries that can degrade a naive Bayesian classifier's accuracy.

To illustrate this difficulty, consider the extreme case of redundant attributes. For a domain with three features, the numerator we saw earlier becomes

$$P(C_i)P(v_1|C_i)P(v_2|C_i)P(v_3|C_i)$$

If we include a fourth feature that is perfectly correlated (redundant) with the first of these features, we obtain

$$P(C_i)P(v_1|C_i)^2P(v_2|C_i)P(v_3|C_i)$$

in which v_1 has twice as much influence as the other values. The emphasis given to the redundant information reduces the influence of other features, which can produce a biased prediction. For example, consider a linearly separable target concept that predicts class A is any two of three features are present and that predicts class B otherwise. A naive classifier can easily master this concept, but given a single redundant feature, it will consistently misclassify one of the eight possible instances no matter how many training cases it encounters.

Surprisingly, many of the domains in which the naive Bayesian classifier performs well appear to contain significant dependencies. This evidence comes in part from Holte's (1993) studies, which show that one-level decision trees do nearly as well as full decision trees on many of these domains. In addition, Langley and Sage (1994) found that the behaviour of a simple nearest neighbour algorithm does not suffer in these domains, as one would expect if there were many irrelevant attributes. Since one attribute is sufficient for high accuracy and the remaining ones do not degrade a nearest neighbor method, then many of the attributes would appear to be highly correlated.

The strong performance of the naive Bayesian method despite violation of the independence assumption is intriguing. It suggests that a revised method which circumvents dependencies should outperform the naive algorithm in domains where dependencies occur, while performing equally well in cases where they do not. In the following section, we discuss a variant Bayesian algorithm that selects and uses a subset of the known features in an attempt to exclude highly correlated attributes. This should let one continue to make the convenient assumption of independence while minimizing its detrimental effects on classification accuracy.

Naïve Classification adopted in this project:

Naive Bayes Classifier (NBC) is a simple probabilistic classifier based on applying Bayes theorem with strong (naive) independence assumptions. Naive Bayes classifier assumes each term independent to each other. Depending on the precise nature of the probabilistic model, naive Bayes classifiers are trained efficiently in a supervised learning approach. For news articles Text Classification, Multinomial Model is used as it performs better than Multi-variate Bernoulli model. The multinomial model specifies that a document be represented as " bag of words". An advantage of the naive Bayes classifier is that it only requires a small amount of training data to estimate the parameters necessary for classification. For news articles text classification we have taken the following steps using Naive Bayes Classifier.

A. Bag of words representation :

The objective of feature selection is to find a subset attributes that best describes a set of documents with respect to the classification task. This definition suggests that we try all subsets and pick the one that maximizes accuracy. Therefore we make the assumption that attributes are independent. Different methods exist to construct the features of each document. Each method has its own strengths and weakness and must be evaluated for each given document domain.

One common approach when constructing features is to treat the document as a " bag of words". This approach takes all the words in a document and places them in an arraylist that represents the feature set of the document. An example of the bag of words" approach can be seen. The example uses a simple boolean weight approach with 1 representing the feature as present and 0 as not present i.e. each word that occur in the training set is associated with a count of how it occurs .

The biggest advantage to the “ bag of words " approach is its simplicity in representation. No special knowledge of the language in question is needed in order to design and implement a classifier . Simply tokenization of the given words is necessary to construct the document vectors.

For example:

Let D1 and D2 be two documents in a training dataset.

D1: ” Ram is a good boy.”

D2: ” Shyam is a bad boy.”

Based on these two documents the bag of words would be:

B = { Ram : 1, is : 2, a : 2, good : 1, boy : 2, bad : 1 }

The vocabulary can then be used to construct the d-dimensional feature vectors for the individual documents where the dimensionality is equal to the number of different words in the vocabulary (d=|B|) .

B. Training set :

(1) Prepared training set for Naïve Bayes Classifier. The training set is a set of documents, each labelled with its class called labelled documents. A class is represented by a collection of words and their frequencies. The frequency is the number of times that each word has been seen in the documents used to train the classifier. The documents in the training set are tokenized and pre-processed. Stopwords, punctuations , special symbols , name entities are extracted from the documents as they are irrelevant .

2) Total number of documents in each class in the training set is calculated

3) Total number of documents in the training set is calculated

My training set consist of 100 documents : 20 of Cricket, 20 of Football, 20 of Politics, 20 of Entertainments and 20 of Business in that order. Thus each class has 20 documents and the whole training set has 100 documents in all.

Prior Probability of each class P(Ci) is calculated

$P(C_i) = N_c / N$ ---------------------(3)

Where N_c is the number of documents in class C and N is the total number of documents in training set. Since in my training set each class have equal number of documents and there are a total of five documents in the training set each class has probability

P(Cricket) = P(Football) = P(Politics) = P(Entertainments) = P(Business) = $\frac{20}{100} = \frac{1}{5} =$ 0.20

$P(C_i)$ is the prior probability of a given category i.e. probability of a document being in class C without considering its content.

C.Test set :

During test phase when a new document is given to the trained classification model, it should predict the correct class of the document. There are a total of 25 test documents which are categorized in to 5 classes. First five belonging to the Cricket, five belonging to Football, five belonging to Politics, five belonging to Entertainments and last five belonging to Business in that order. All the pre-processing techniques should be applied to the test documents such as feature selection, tokenization, stop words removal and stemming.

After pre-processing and feature extraction steps, each unlabelled document are represented as bag of words i.e. w_1, w_2 …. w_n, where w_n is the nth word of the document. We have considered only five classes : Cricket, Football, Politics, Entertainments and Business. Determining which class a document D is most associated with means calculating the probability that document D is in class C_i.

Algorithm

In order to understand how I have used naive Bayes classifiers in this paper, we have to briefly recapitulate the concept of Bayes' rule. The probability model that was formulated by Thomas Bayes (1701-1761) is quite simple yet powerful. It can be written down in simple words as follows:

$$\text{posterior probability} = \frac{conditional\ probability * prior\ probability}{\text{evidence}}$$

Bayes' theorem forms the core of the whole concept of naive Bayes classification. The *posterior probability*, in the context of a classification problem, can be interpreted as: "What is the probability that a particular object belongs to class *i* given its observed feature values?" A more concrete example would be: "What is the probability that a person has diabetes given a certain value for a pre-breakfast blood glucose measurement and a certain value for a post-breakfast blood glucose measurement?"

Let

- x_i be the feature vector of sample i, i∈{1,2,...,n}
- ω_j be the notation of class j, j∈{1,2,...,m}
- and $P(x_i|\omega_j)P(x_i|\omega_j)$ be the probability of observing sample x_i given that is belongs to class ω_j.
- The general notation of the posterior probability can be written as

$P(\omega_j|x_i)$=A/B where,
A=$P(x_i|\omega_j)P(\omega_{j)}$
B=$P(x_i)$

The objective function in the naive Bayes probability is to maximize the posterior probability given the training data in order to formulate the decision rule.

To continue with our example above, we can formulate the decision rule based on the posterior probabilities as follows:

person has diabetes if

$P(\text{diabetes} \mid x_i) \geq P(\text{not-diabetes} \mid x_i)$

else classify person as healthy.

Bayes Logic applied to my program is also the same except that the formula to

calculate the probability of the document to fall in a particular class or category is by using the equation:

$$\mathbf{P(C_i|D)=P(D|C_i)*P(C_i)}$$

Where C_i is a particular class and D is the query document.

D is split into set of words w_1, w_2 …. w_n. Hence $P(D|C_i)$ is calculated as $P(D|C_i) = \log(P(w_1 \mid C_i) * P(w_2 \mid C_i)*\ldots* P(w_n \mid C_i))$

But since I have shown earlier $P(C_{i)=}$ N_c/N where Nc is the number of documents in the class i and N is the total number of documents in the training set. Since each category has 20 document each P(Cricket)=P(Football)=P(Politics)=P(Entertainments)=P(Business)=20/100=0.20

Since each class has the same probability according to our considered dataset we eliminate this term. Thus our final equation is:

$$\mathbf{P(C_i|D)=P(D|C_i)}$$

Now as stated earlier we evaluate $P(D|C_i)$ by calculating the probability of each word in the document to be in class C_i and multiplying them into a product and taking the logarithmic value of this product. We know Log is an increasing function so the greater the value the greater is the probability of the word or the document to fall in a certain class.

Now to calculate the probability of a single word in a class we need to find the frequency of the word in the class, the size of the vocabulary or the corpus that consists of all the unique words of the training dataset and the size of the query document D. The equation used to do that is:

$$\mathbf{P(w_i|C_i)=(freq(w_i)+1)/Vs+Qs)}$$

Where freq(w_i) is the frequency of the word I the class Vs is the vocabulary size and Qs is the query document size.

We add 1 to the numerator to avoid the zero problem i.e. if a word is not present in a class in the training set but is present in he test query document the frequency of that word would be 0 and since we are calculating the product of each word probability and then taking its log results gets undefined. Thus to eliminate the zero problem I add 1 to the numerator although there are many other techniques to eliminate he zero problem I adopt the easiest one.

Since we know probabilities of each word runs from $0<p<1$ and since we are taking log of each of these probabilities, log of these values gives a negative number and since we are adding each of these log values (log (A*B)=log A+ log B) the probability of a query document to belong to a particular class is a big negative number. As stated earlier log is an increasing function so we take the class where the result is the smallest negative number (bigger number on the whole (-1>-2)) and infer that the query document belongs to that particular category or class.

Experimental Results and Observations

Our dataset consists of documents from most major e-newspapers likes Times, The Telegraph, The Hindu, The Bloomberg etc for the recent year of 2018. A dataset with 125 documents classified in four different classes is used for evaluation. The selected dataset consists of five classes of document: Cricket, Football, Politics , Entertainment and Business each containing 25 articles each. All the five categories are easily differentiated. 80% data i.e. 100 documents are extracted randomly to build the training dataset . The other 20% data i.e 25 documents are used as the testing dataset

In this paper , the training set is used to train the classification model and the test set collection is then applied for the classification of documents in to respective classes. Training set with 100 documents and test set with 825 documents is divided into 5 classes as shown in the table 1

Table 1 Training and test document collection of five classes

Class	***Training Set***	***Testing Set***
1. Cricket	20 articles	5 articles
2. Football	20 articles	5 articles
3. Politics	20 articles	5 articles
4. Entertainments	20 articles	5 articles
5. Business	20 articles	5 articles

Now we calculate the accuracy of our algorithm by calculating the accuracy matrix.The accuracy matrix compares the classification a done by a human to that done by the program and tries to draw an inference based on the result.

Accuracy matrix:

Document no.	**Human**	**Code Prediction**
D1	Cricket	Cricket
D2	Cricket	Cricket
D3	Cricket	Cricket
D4	Cricket	Cricket
D5	Cricket	Cricket
D6	Football	Football
D7	Football	Football
D8	Football	Football
D9	Football	Football
D10	Football	Football
D11	Politics	Politics
D12	Politics	Politics
D13	Politics	Politics
D14	Politics	Politics
D15	Politics	Entertainments
D16	Entertainments	Entertainments

D17	Entertainments	Entertainments
D18	Entertainments	Entertainments
D19	Entertainments	Entertainments
D20	Entertainments	Entertainments
D21	Business	Business
D22	Business	Business
D23	Business	Entertainments
D24	Business	Business
D25	Business	Football

We see that there are three instances where human prediction and code prediction has disagreed upon namely in documents D15, D23 and D25 according to the accuracy matrix. We calculate the appropriate accuracy by which this code works:

$$\textbf{Accuracy=} \frac{\textbf{No.of matched documents}}{\boldsymbol{Total\ nof\ documents\ in\ Training\ set}} \ \textbf{* 100}$$

$$=\frac{22}{25} * 100$$

$$= 88\%$$

Thus we see that the Naïve Bayes Algorithm adopted by us in this classification has borne 88% accuracy with the considered dataset.

Next we calculate the confusion matrix for the wrongly classified documents. A confusion matrix is a technique for summarizing the performance of a classification algorithm.

Classification accuracy alone can be misleading if you have an unequal number of observations in each class or if you have more than two classes in your dataset. Calculating a confusion matrix can give you a better idea of what your classification model is getting right and what types of errors it is making.

Confusion Matrix:

Prediction Human	Cricket	Football	Politics	Entertainments	Business
Cricket	5	0	0	0	0
Football	0	5	0	0	0
Politics	0	0	4	1	0
Entertainments	0	0	0	5	0
Business	0	1	0	1	3

A. Observations on Naïve Bayes Classification

1) Based on this problem , the performance on the Naïve Bayes classification model has worked with 88% accuracy.

2) The Naïve Bayes method assumes that the component features are independent within each class.

3) Naïve Bayes model is often surprisingly effective.

4) Naïve Bayes classifies an unknown instance by computing the class which maximizes the posterior.

5) In naive Bayes classifiers, every feature contributes towards determining the class of the document by calculating the prior probability of each document , which is determined by checking frequency of each document in the training set.

6) Naïve Bayes classification is flexible and robust to errors. The prior and the likelihood can be updated dramatically with each training example.

7) Naïve Bayes is very efficient and linearly proportional to the time needed just to read in all the data.

8) It is easy to implement and compute when compared with other algorithms.

9) Time Complexity:

Training Time :

$O(|D|L_d + |C||V|)$

where Ld is the average length of a document in D. | D | is the number of documents. The complexity of computing the parameters is O (|C||V|) because the set of parameters consists of |C||V| conditional probabilities and |C| priors. The preprocessing computations on the parameters can be done in one pass through the training data. The time complexity of this component is therefore $O(|D|\ L_d)$.

Testing Time: O(|C| L_t)

- Where Lt is the average length of a test document.
- Very efficient overall, linearly proportional to the time needed to read in all the data to have optimal time complexity.
- Plus, robust in practice.

Disadvantages of Naive Bayes Classifier

- One of the problems of Naïve Bayes is known as the "Zero Conditional Probability Problem." This problem wipes out all the information in other probabilities too. There are several sample correction techniques to fix this problem such as "Laplacian Correction."

- Another disadvantage is the very strong assumption of independence class features that it makes. It is near to impossible to find such data sets in real life.

Uses of Text Classification in other industries:

Let's talk about the current and emerging applications of text classification in industry other than e-news. We have been using text classification to simplify things for us for a long time now. Classification of books in libraries and segmentation of articles in news are essentially examples of text classification. Adding AI tech to it, the process becomes automated and simpler with minimum manual work. The concept of using AI to classify text has been around for a fair amount of time.

It can essentially be used whenever there are certain tags to map to a large amount of textual data — especially in marketing, as it has moved from search engines to social media platforms where real communication between brands and users take place. As marketing is becoming more targeted, marketers are using personalization to drive better engagements. Thus, listening to user conversations and analyzing them becomes a must-do task for marketers.

Classification can be done on any set of data. The ability of text classification to work on a tagged dataset (in the case of a CRM automation) or without it (reading social sentiments online) just opens up the spaces where this technology can be implemented. A lot of these classifications can be done or rather has been done by the Naïve Bayes Classification algorithm and has been successfully implemented

- **Tagging content or products using categories as a way to improve browsing or to identify related content on your website.** Platforms such as e-commerce, news agencies, content curators, blogs, and directories can use automated technologies to classify and tag content and products.
- **Text classification can also be used to automate CRM tasks**. The text classifier is highly customizable and can be trained accordingly. The CRM tasks can

directly be assigned and analyzed based on importance and relevance. This reduces manual work and thus is highly time-efficient.

- **Text classification of content on the website using tags helps Google crawl your website easily, which ultimately helps in SEO**. Additionally, automating the content tags on your website and app can make the user experience better and help standardize it. Another use case for marketers would be to research and analyze tags and keywords used by competitors. Text classification can be used to automate and speed up this process.
- **A faster emergency response system can be made by classifying panic conversation on social media**. Authorities can monitor and classify emergency situations to make a quick response if any such situation arises. This is a case of very selective classification. You can check out this study to read an elaborate post on one such emergency response system.
- **As marketing is becoming more targeted every day, automated classification of users into cohorts can make the marketer's life simple**. Marketers can monitor and classify users based on how they talk about a product or brand online. The classifier can be trained to identify promoters or detractors, thus helping brands serve cohorts better.
- **Academia, law practitioners, social researchers, government, and non-profit organizations can also make use of text classification technology**. As these organizations deal with a lot of unstructured text, handling the data would be much easier if it were standardized by categories/tags.
- **Spam filtration.** It is an example of text classification. This has become a popular mechanism to distinguish spam email from legitimate email. Several modern email services implement Bayesian spam filtering. Many server-side email filters, such as DSPAM, Spam Bayes, Spam Assassin, Bogofilter, and ASSP, use this technique.
- **Sentiment Analysis**. It can be used to analyze the tone of tweets, comments, and reviews—whether they are negative, positive or neutral.

- **Recommendation System.** The Naive Bayes algorithm in combination with collaborative filtering is used to build hybrid recommendation systems which help in predicting if a user would like a given resource or not.

Conclusion

In this paper we have done document classification of news text using Naïve Bayes model. We have also observed the accuracy of Naïve Bayes classifier on huge amount of documents. As such very few work has been done on document classification of news using any statistical approach, this method shows promising result. Prospect of doing document classification of Indian language is very high. Deeper analysis can be done using different statistical approach. In the future we propose to perform document classification using different statistical approach such as centroid based, k-nn and SVM. Also this study can be further extended by implementing Naïve Bayes classification on larger datasets. we have presented initial results of an implementation of an automated news categorization system. The obtained results are encouraging and practically useful. Text classification has become a major issue, now a days and one reason of it is the lack of single technique, which is able to produce good classification for different data sets. There are various classification methods such as Decision Trees, Neural Networks, Naïve Bayes and Centroid Based, but Naïve Bayes performs better for different data collections and is easy and computationally cheap. Along with its simplicity, Naïve Bayes also suffers from the some issues like unseen words. We can use various smoothing techniques like JK method, Absolute Discounting method, Dirichlet Smoothing and Two-stage Smoothing to enhance the performance and accuracy of Naïve Bayes. Further work will continue on extending the system to cover not only the main IPTC categories, but also the sub-categories in a hierarchical categorization scheme. Further experiments are also needed in order to optimize the size and structure of the training set in order to strike a balance between precision, for which a larger training set is beneficial, and performance, which is dragged down by increasing the training set. An indexing method putting more weight on news item title and news lead is also expected to yield better results. With these optimizations and improvements, we expect to achieve even higher accuracy and faster processing.

References

1. Sebastiani, F. 2002. Machine Learning in Automated Text Categorization. ACM Computing Surveys, Vol. 34, No. 1, March 2002, pp- 1-47

2. Yiming Yang and Xin Liu, 1999. A re-examination of text categorization methods. Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99, pp 42--49)

3. Sebastiani, F. 2000. A Tutorial On Automated Text Categorization «Istituto di Elaborazione dell' Informazione, Consiglio Nazionale delle Ricerche,», pp. 11-22

4. Rafael A. Calvo, Jae-Moon Lee, Xiaobo Li, 2004. Managing Content with Automatic Document Classification. J. Digit. Inf. 5(2)

5. Bharati, Akshar, Kiran Varanasi, Chaitanya Kamisetty, Rajeev Sangal, S M Bendre, 2002. A Document Space Model for Automated Text Classification based on Frequency Distribution across categories, Published in the proceedings of ICON2002, Mumbai, 18-21 Dec 2002

6. Yihua Liao and V. Rao Vemuri, "Use of K-Nearest Neighbor Classifier for Intrusion Detection", Computers & Security, Volume 21, Issue 5, October 2002, Pages 439-448

7. Eui-Hong (Sam) Han, George Karypis, Vipin Kumar. 1999. “Text Categorization Using Weight Adjusted k-Nearest Neighbor Classification”, Department of Computer Science and Engineering, Army HPC Research Center, University of Minnesota

8. Thomas Hofmann, 2003. Introduction to Machine Learning, Draft Version 1.1.5, November 10, 2003

9. Li Baoli, Yu Shiwen, and Lu Qin, 2003. “An Improved k-Nearest Neighbor Algorithm for Text Categorization”, Proceedings of the 20th International Conference on Computer Processing of Oriental Languages, Shenyang, China, 2003

10. http://www.iptc.org

11. http://www.iptc.org/NewsCodes

12. http://www.iptc.org/NewsCodes/nc_ts-table01.php?TsByName=iptc-subjectcode

13. http://www.nitf.org/

14. E.H. Han, G. Karypis, and V. Kumar, Text categorization using weight adjusted k-nearest neighbour classification, Department of Computer Science and Engineering, Army HPC Research Center, University of Minnesota, 1999.

15. A. McCallum, and K. Nigam, "A comparison of event models for naïve Bayes text classification", Journal of Machine Learning Research, Vol. 3, 2003, pp. 1265–1287.

16. S. Chakrabarti, S. Roy, and M.V. Soundalgekar, "Fast and accurate text classification via multiple linear discriminant projection", The VLDB Journal The International Journal on Very Large Data Bases, 2003, pp. 170–185. International Journal of Software Engineering and Its Applications Vol. 5, No. 3, July, 201146

17. J.R. Quinlan, C4.5: programs for machine learning, Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993.

18. S. Wermter, "Neural network agents for learning semantic text classification", Information Retrieval, Vol. 3, No. 2, 2004, pp. 87-103.

19. K. Nigam, J. Lafferty, and A. McCallum, "Using maximum entropy for text classification", In Proceedings: IJCAI-99 Workshop on Machine Learning for Information Filtering, pp. 61–67, 1999.

20. T. Joachims, "Text categorization with support vector machines: Learning with many relevant features", In Proceedings: Machine Learning: ECML-98, 10th European Conference on Machine Learning, pp. 137–142, 1998.

21. Nidhi, Vishal Gupta, 2012. "Domain Based Classification of Punjabi Text Documents using Ontology and Hybrid Based Approach" Proceedings of the 3rd

Workshop on South and Southeast Asian Natural Language Processing (SANLP), COLING

22. G Siva Charan, Kavi Narayana Murthy, and S Durga Bhavani. “Text categorization in indian languages”. In R M K Sinha and V N Shukla, editors, Proceedings of ICSLT-OCOCOSDA – I STRANS 2004 International Conference - Vol 1, pages 56-61. Tata McGraw-Hill Publishing Company Ltd, 2004.

23. Murthy, Kavi Narayana. “Automatic Categorization of Telugu News Articles”. In: Department of Computer and Information Sciences, University of Hyderabad, Hyderabad, DOI= 202.41.85.68. (2003)

24. Yang, Y., Chute, C.G.: “An example-based mapping method for text categorizationand retrieval”. In: ACM Transaction on Information Systems: 253-277(1994)

25. R ajan, K., Ramalingam, V.,Ganesan, M., Palanivel, S. and Palaniappan, B “Automatic Classification of Tamil documents using Vector Space Model and Artificial Neural network”. In: Expert Systems with Applications, Elsevier, Volume 36 Issue 8, DOI= 10.1016/j.eswa.2009.02.010. . (2009).

26. K Raghuveer and Kavi Narayana Murthy, “Text Categorization in Indian Languages using Machine Learning Approaches”

27. Naïve Bayes Text Classification. http://nlp.stanford.edu/IR/book/html/htmledition/naivebayes-text-classification1 .html

28. Anderson, J. R., & Matessa, M. (1992). Explorations of an incremental, Bayesian algorithm for categorization. Machine Learning, 9, 275-308.

29. Brodley, C. E., & Utgoff, P. E. (1992). Multivariate versus univariate decision trees (Coins Technical Report 92-8) . Amherst: University of Massachusetts, Department of Computer and Information Science.

30. Bun tine, W. (1990). A theory of learning classification rules. Dissertation, Department of Computer Science, University of Technology, Sydney.

31. Caruana, R. A., & Freitag, D. (in press). Greedy attribute selection. Proceedings of the Eleventh International Conference on Machine Learning. New Brunswick, NJ.

32. Cestnik, G. (1990). Estimating probabilities: A crucial task for machine learning. Proceedings of the Ninth European Conference on Artificial Intelligence (pp. 147-149). Stockholm, Sweden.

33. Cestnik, G., Kononenko , I, & Bratko, I. (1987). AssiSTANT-86: A knowledge-elicitation tool for sophisticated users. In I. Bratko & N. Lavrac (Eds .) Progress in machine learning. Sigma Press.

34. Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., & Freeman, D. (1988). AUTOCLASS: A Bayesian classification system. Proceedings of the Fifth International Conference on Machine Learning (pp. 54-64). Ann Arbor: Morgan Kaufmann.

35. Clark, P., & Niblett , T. (1989). T he CN2 induction algorithm. Machine Learning, 3, 261-284.

36. Connolly, D. {1993). Constructing hidden variables in Bayesian networks via conceptual clustering. Proceedings of the Tenth International Conference on Machine Learning (pp. 65-72). Amherst, MA: Morgan Kaufmann.