RESEARCH ARTICLE OPEN ACCESS

Dynamic & Attribute Weighted KNN for Document Classification Using Bootstrap Sampling

Dharmendra S Panwar, Kshitij Pathak
Mahakal Institute of Technology Ujjain

Abstract—
Although publicly accessible databases containing speech documents. It requires a great deal of time and effort required to keep them up to date is often burdensome. In an effort to help identify speaker of speech if text is available, text-mining tools, from the machine learning discipline, it can be applied to help in this process also. Here, we describe and evaluate document classification algorithms i.e. a combo pack of text mining and classification. This task asked participants to design classifiers for identifying documents containing speech related information in the main literature, and evaluated them against one another. Expected systems utilizes a novel approach of k -nearest neighbour classification and compare its performance by taking different values of k.

Keywords— Data Mining, Text Mining, Classification, K-nearest neighbour, KNN

I. INTRODUCTION

Text mining is a prosper new field that attempts to extract meaningful information from natural language text. It can be characterized as the process of analysing unstructured text to extract or retrieve information that is useful for particular purposes. Compared with other type of data stored in databases, algorithmically dealing with text is very difficult because text is irregular and unstructured. In modern or recent culture, text is the most and common way for the formal exchange of information. Emphasis in text mining is to find hidden patterns or opinions and extraction of information related to communication from an individual or parties. Mining the text is fruitful even if it provides partial success.

II. TEXT MINING

Recently text mining has become an important research area. Text can be placed in organization files, mails, online chats, product reviews, newspaper articles, journal and SMS. Text mining can also be called as text data mining, intelligent text analysis or knowledge discovery in text (KDT) refers to extracting the useful information from the natural language text. Text mining extracts the new pieces of knowledge from textual data.

Fundamentally text mining is used to joint countless pages of plain-language digitized text to calculate useful information that has been hiding in plain sight (Gary Belsky: 2012).

Approximately 80% of the world's data is in not structured format. Data are stored in most of the government sectors, organization, industries and institutes in electronic form. These all data are stored in text database format. Text database is semi structured format which contains many structured fields and few unstructured fields. For example students name, roll no, class, semester are the structured fields and Address, remarks are unstructured fields in an educational institution. Text mining is very essential for an organization or institution because most of the information in the organizations is in text format.

Text Mining [1] is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources.

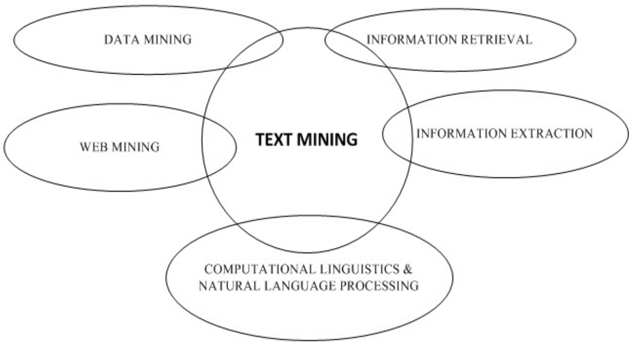


Figure 2.1 Text Mining Basic

A main element is linking to together to form new facts or new conclusion to be research further by conventional means of experimentation. Text mining is different from the web search. In web search, the user is looking for something that is already known and which has been written by someone. The problem is pushing aside all the material that currently is not relevant to your needs in order to find the relevant information. In text mining, the aim is to discover unknown information from the datasets that no one yet knows and so could not have yet written down.

Text mining is a part of data mining [2] that tries to find interesting patterns from large databases and data warehouse. Text mining can also be known as Knowledge-Discovery in Text (KDT), Text Data Mining or Intelligent Text Analysis is the process of extracting interesting and valuable information and knowledge from unstructured text. Text mining is a new and young interdisciplinary field which draws on data mining, information retrieval, machine learning, computational and statistics linguistics. As per the study, 80% information is stored as text form, text mining is considered to have a big commercial potential value. Knowledge may be discovered from many or multiple sources of information but unstructured text remains the largest readily available source of knowledge.

The problem of Knowledge Discovery from Text (KDT) [3] is to extract explicit and implicit concepts and semantic relations between concepts using Natural Language Processing techniques (NLP). Its purpose to get insights into large quantities of text data. Knowledge Discovery from Text, while deeply rooted in Natural Language Processing, draws on methods from statistics, information extraction, machine learning, reasoning, knowledge management, and others for its discovery process. KDT plays an important and increasingly significant role in emerging applications for example Text Understanding.

Text mining [1] is just like of data mining, except that data mining tools [2] are designed to work with structured data from databases, but text mining can apply with unstructured, semi-structured and structured data sets such as Internet files (HTML), emails and full text files etc. As a result, text mining is a better solution for companies. To date, however, most research and development efforts have centered on data mining efforts using structured data. The problem introduced by text mining is obvious: natural language was developed for humans to communicate with one another and to record and save information, and computers are a large way from comprehending natural language.

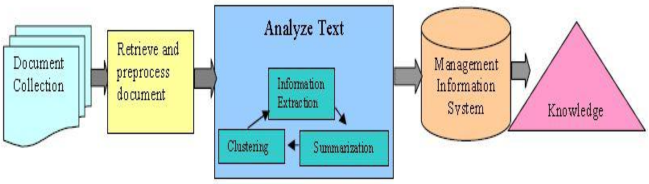


Figure 2.2 An Example of Text Mining

Humans have the ability to distinguish and apply linguistic patterns to text and humans can easily overcome obstacles that computers cannot easily handle such as contextual meaning slang and spelling variations. However, although our language capabilities allow us to comprehend unstructured data and lacking of the computer's ability to process unstructured text in large volumes or at high speeds. Figure 2.2 shows a generic process model [4] for a text mining application.

Starting with a collection of documents and a text mining tool would be retrieve a specific document and pre-process it by checking character sets and format. Then it would be go through by a text analysis phase and repeating techniques of mining until information is extracted. The final resulting information would be placed in a management information system and that helps to yielding an abundant amount of knowledge that is very useful for the user of that system.

III. CLASSIFICATION

All Classification, which is the task of assigning objects to one of several predefined categories or class, is a pervasive problem that encloses many diverse applications. Examples that include classifying galaxies based upon their shapes, categorizing cells as malignant or benign based upon the results of MRI scans and detecting spam email messages based upon the message content and header.

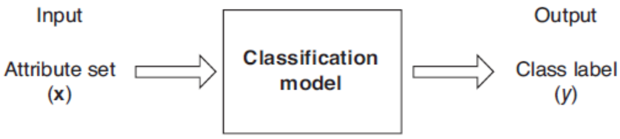


Figure 3.1 Classification as the task of mapping an input attribute set x into its class label y

The input data for a classification task is a set of records. Every record also know as instance or can characterized by a tuble(x,y),where x is attribute set and y is a unique attribute as class label also know as target or category attribute. Mainly Classification is the task of learning a target function f that Assign each attribute of set x to one of the predefined class labels y.

The target function is also known informally as a classification model. A classification model is useful for the following purposes:-

Descriptive Modeling: It is a classification model that serves as an informing tool to distinguish between objects of different classes.

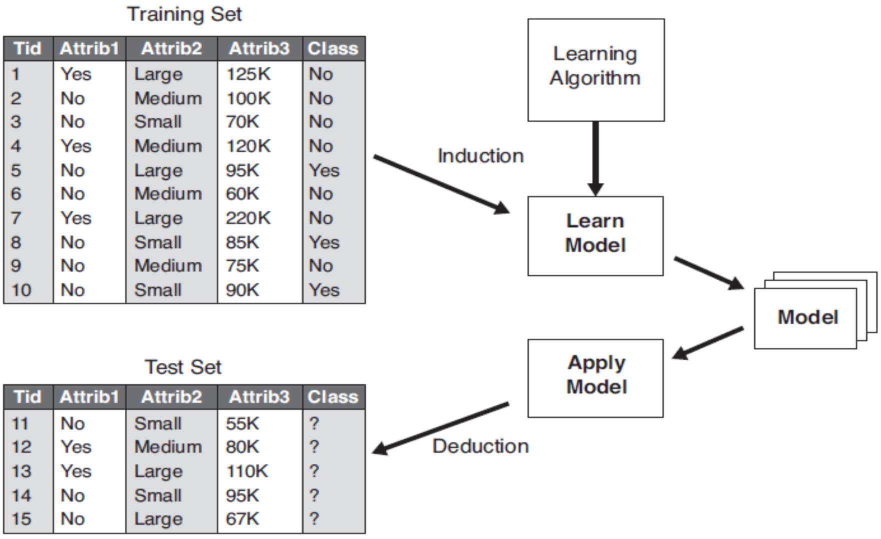
Predictive Modeling: It is a classification model that also be used to predict the class label of unknown records.

Classification techniques are most suited for predicting or describing or defining data sets with nominal or binary categories. They are less effective for ordinal categories (e.g., to classify a person in terms of their income group like high, medium or low income group) because they do not consider the implicit order among the categories. Other forms of relationships, such as the subclass–super class relationships among categories (e.g., apes and humans are primates which is a subclass of mammals) are also ignored.

Following are the general Approach to Solving a Classification Problem

A classification technique (or classifier) is a systematic approach to building classification models from an input data set. Examples include decision neural networks, naıve Bayes classifiers, rule-based classifiers, tree classifiers, support vector machines. Each technique uses learning algorithm to identify a model that best fits the relationship between class label and the attributes set of the input datasets. The algorithm generated model should both fit the input data and correctly predict the class labels of records it hasn't seen before. That's why a main objective of the learning algorithm is to build models with good generalization capability; that is to say, models that accurately predict the class labels of previously unknown records.

Figure 3.2 shows a general approach for solving classification problems. First, a training set containing the records whose class labels are known must be given.

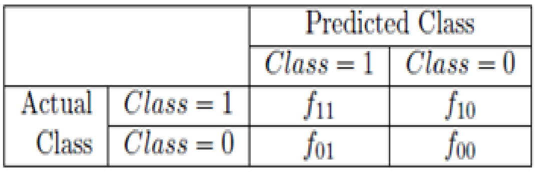

Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

Test Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Figure 3.2 General approach for building a classification model.



		Predicted Class	
		$Class = 1$	$Class = 0$
Actual Class	$Class = 1$	f_{11}	f_{10}
	$Class = 0$	f_{01}	f_{00}

Table 3.1 Confusion matrix for a 2-class problem

The training set is used to generate a classification model that is applied to the test dataset, which consist unknown class labels records. The performance and evaluation of the model is based on correctly or incorrectly predict the counts of the test records. These counts are tabulated in a table known as a confusion matrix. Table 3.1 depicts the confusion matrix for a binary classification problem. Each entry f_{ij} in this table shows the number of records from class i predicted to be of class j. Here f_{01} is the number of records from class 0 incorrectly predicted as class 1. On the basis of the entries in the confusion matrix and the total number of correct predictions made by the model is (f_{11}+f_{00}) and the total number of incorrect predictions is (f_{10}+f_{01}).

Confusion matrix provides the useful information that determine how well model perform or work and its summarized this information with single number that it more convenient to compare with different models performance.

This can also be done using a performance metric like that accuracy, which is defined as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} = \frac{f_{11} + f_{00}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Likewise, the performance of a model can be calculated in terms of its error rate, using the following equation:

$$\text{Error rate} = \frac{\text{Number of wrong predictions}}{\text{Total number of predictions}} = \frac{f_{10} + f_{01}}{f_{11} + f_{10} + f_{01} + f_{00}}.$$

Most of the classification algorithms seek models that attain the highest accuracy, or equivalently, the lowest error rate when applied to the test set.

K-Nearest Neighbors (KNN)

One of the various classifier, 'KNN classifier' is a case based learning algorithm which is based on a distance or similarity function for various pairs of observation such as the Euclidean distance function. It is tried for many applications because of its effectiveness, non-parametric & easy to implementation properties. However, under this method, the classification time is very long & it is difficult to find optimal value of K. Generally, the best alternative of k to be chosen depends on the data. Also, the effect of noise on the classification is reduced by the larger values of k but make boundaries between classes less distinct. By using various heuristic techniques, a good 'k' can be selected. In order to overcome the above said drawback, modify traditional KNN with different K values for different classes rather than fixed value for all classes.

How KNN Algorithm Works

KNN algorithm is used to classify instances based on nearest training examples in the frame space. KNN algorithm is known as lazy learning algorithm in which function is approximated locally & computations are delayed until classification. A majority of instances is used for classification process. Object is classified into the particular class which has maximum number of nearest instances.

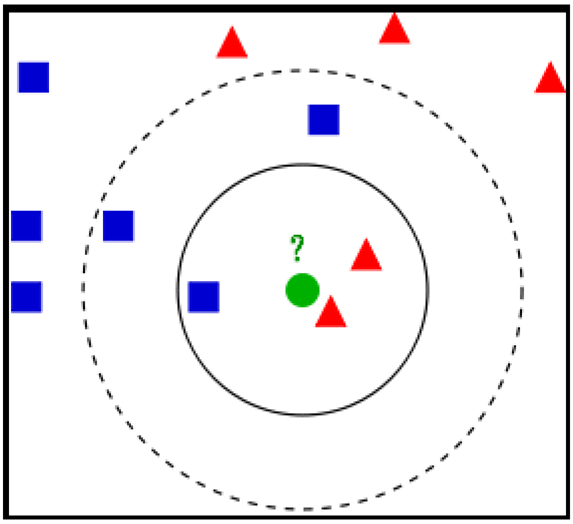


Fig 3.3 Example of k- nearest neighbor [11]

In above figure the test instance (green circle) should be classified either into blue square class or into red triangle class. If k = 3 (solid line circle) test object (green circle) is classified into red triangle class because there are 2 triangle instances and only 1 square instance in the inner circle.

If k = 5 (dashed line circle) test object (green circle) is classified into blue square class because there are 3 blue square instances and only 2 red triangle instances in the inner circle.

SCOPE OF KNN

Improvement of KNN can be done on following parameters. [12]

A. ***Distance or similarity Function:*** The distance or similarity function is used for measuring the difference or similarity between two instances is the standard Euclidean distance.
B. ***Selection of K Value:*** It represents neighbourhood size, which is artificially assigned as an input parameter.
C. ***Calculating Class Probability***: The class probability assumption, based on a simple voting.

IV. BOOTSTRAP SAMPLING

Bootstrap methods are more than 30 years old [8], but they are computer-intensive and have only recently become widely available in statistical programs. Powerful and widely available computing capability is transforming statistics. Originally, computers just speeded up data handling and statistical computations. Now they enable exploratory analysis of gigantic data sets (data mining), creating neural network algorithms, analysing complex genomic data records, and supplementing traditional statistical methods using large numbers of data simulations.

As per statistics, bootstrapping is a method for assigning measures of accuracy to sample estimates. This technique also allows estimation of the sampling distribution of almost any statistic using only very simple methods.[9,10] Generally, it falls in the broader class of resampling methods.

Bootstrapping is the process of estimating properties of an estimator (like variance) by measuring those properties when sampling from an approximating distribution.

To understand bootstrap, suppose it were possible to draw repeated samples (of the same size) from the population of interest, a large and more number of times. Then one would obtain a fairly good idea about the sampling distribution of a particular statistic from the collection of its values arising from these repeated samples. But, that does not mean that it would be too expensive and defeat the purpose of a sample study. The aim of a sample study is to collect information cheaply in a timely fashion. The main idea behind bootstrap sampling is to use the data of a sample study at hand as a “surrogate population”, for the aim of approximating the sampling distribution of a statistic; i.e. to resample (with replacement) from the sample data at hand and create a large number of “phantom samples” known as bootstrap samples. The summary of the sample is then computed on each of the bootstrap samples (usually a few or more thousand). A histogram of the set of these computed values is referred to as the bootstrap distribution of the statistic.

V. TEXT CLASSIFICATION

The Text classification is a supervised learning algorithm whose approach is to classify a text documents into pre-defined categories [5]. Classification has basic four types as per applications – Binary Hard, Multiclass single label, Multi-class Multi label hard and multi-class multi label soft classification [6].

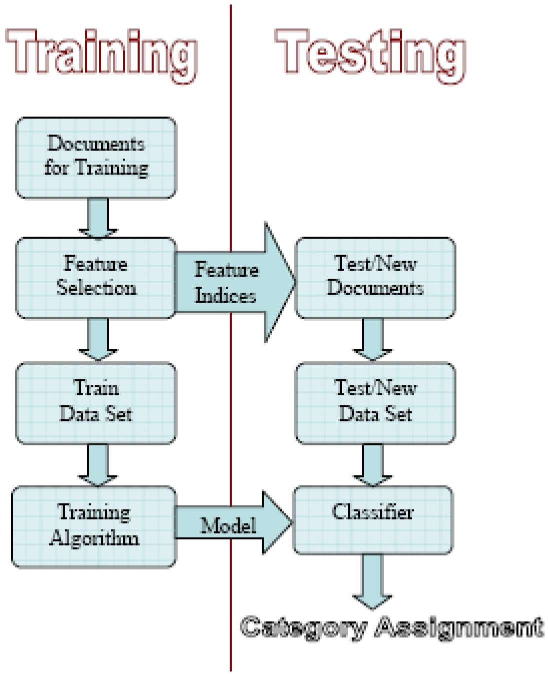


Figure -5.1 General model of text classification

Multi-class means multi categories are allowed like sport, jokes, and business in news data. Multi –label means document belong to more than one category like any documents belong to business and sport categories. Hard classification means document has Boolean membership into classes and soft means fuzzy membership into all classes. In this paper, our approach is regarding multi-class single label classification.

For classification, general model is represented in figure -5.1. There are two phases - training and testing phase. In training phase, training documents that are pre-labeled are used to build a model using its feature vectors which are generated by pre-processing. Actual class assignment process performed in test phase. Whenever a new test document come for classification, then feature vector of test document is applied in model and class label assigned. In our literature research we found that most methods and developments can be divided into following distinct steps.

The steps are:

1. Text gathering,
2. Text pre-processing,
    i. Stopword
    ii. Stemming
    iii. Pruning
3. Document Indexing,
4. Pattern Mining and Result Evaluation
    i. Categorization
        a. Sentiment Classification
        b. Authorship identification
        c. Web classification
    ii. Information Retrieval

VI. PROPOSED WORK

The simplicity and robustness of k-NN has been evident, and has been discussed already. This work target on one of the most vital factors of k-NN, namely, the selection of the value of k. Various experiments using different adaptations of the traditional k-NN to both improve its accuracy, and address its shortcomings have been researched in [7].

With the increasing variety of document, and the complex nature displayed by these document pages, a dynamic value of k is of ever increasing importance. In this work value of k is set dynamically and classify the document using dynamic weighted K-nearest neighbor classifier using bootstrap sampling.

In the proposed work, first set the value of k which is dynamically and classify the document using dynamic weighted K-nearest neighbour classifier using bootstrap sampling.

Algorithm 6.1

Dynamic & Attribute Weighted KNN for Document Classification Using Bootstrap Sampling (DAWk-nn)

```
Phase-I
  1. Clean Corpus by   // Corpus: Collection of documents.
       i.   Removing Punctuation
       ii.  Removing White space
       iii. Converting text to either upper or lower case
       iv.  Removing Stopwords
  2. Make Term document matrix.
  3. Remove sparse terms.
  4. Let r=maximum expected value of K
     For i=1 to r
       i.   Take a bootstrap sample
       ii.  Evaluate the performance by simple distance based KNN
       iii. Stored in performance matrix (performance[i])
  5. Select the best value from performance array & store it in K.
Phase-2
  1. Again apply bootstrap sampling on original dataset, divide dataset into training & testing module.
  2. Clean Corpus by
       i.   Punctuation
       ii.  White space
       iii. Either select upper or lower case
       iv.  Stopwords
  3. Make document matrix
  4. Remove sparse terms
  5. Assign the Weight to training dataset by using following equation:-
         $W_r = 1/ [d (x_i, x_q)^2 + 0.001]$
  6. Calculate Distance using K-NN algorithm with the help of K Picked in Step 5.
```

7. Return the class label.

In this algorithm, in Phase-1, first step we have to clean the raw data or unstructured text that is called clean corpus by removing punctuation, removing white space, converting text to either upper or lower case, removing stopwords. In the next step make the document matrix and remove the sparse terms. In the fourth step calculate the value of K for proposed work using bootstrap sampling and performance calculation and after the select the best K in performance matrix. When get the value of K from Phase-1, use it in Phase-2 for calculating beast nearest neighbor. Now Phas-2, again we apply the bootstrap sampling but in this time work on original dataset and clean that original data sets using clean corpus, make document matrix, remove sparse terms. In fifth step Find the best K nearest neighbor to meet training dataset by using following equation:-

$$\mathbf{W_r = 1/\, [d\,(x_i,\, x_q)^2 + 0.001]}$$

After calculate the weight return the class label.

VII. CONCLUSION

In this proposed work, we investigated the factors that reduced the time complexity of document classification. The proposed approach overcomes the limitation of traditional K-nearest neighbor by incorporating neighbor weighted and dynamically picking K to provide best result, this proposed work also use the bootstrap sampling technique which is the method for assigning measure of accuracy to sample estimate. We have measured accuracy of the proposed approach with the standard K-nearest neighbor on 20,000 newsgroup documents dataset downloaded from [13] and accuracy of the classifier increases by 11.32 % (on average) in comparison to the earlier approach.

REFERENCES

[1] Berry Michael W., (2004), “Automatic Discovery of Similar Words”, in “Survey of Text Mining: Clustering, Classification and Retrieval”, Springer Verlag, New York, LLC, 24-43.

[2] Navathe, Shamkant B., and Elmasri Ramez, (2000), “Data Warehousing and Data Mining”, in “Fundamentals of Database Systems”, Pearson Education pvt Inc, Singapore, 841-872.

[3] Haralampos Karanikas and Babis Theodoulidis Manchester, (2001), “Knowledge Discovery in Text and Text Mining Software”, Centre for Research in Information Management, UK Tavel, P. 2007 Modeling and Simulation Design. AK Peters Ltd.

[4] Weiguo Fan, Linda Wallace, Stephanie Rich, and Zhongju Zhang, (2005), “Tapping into the Power of Text Mining”, Journal of ACM, Blacksburg.

[5] Mita K. Dalal, Mukesh A. Zaveri (2011), “Automatic Text Classification: A Technical Review”, International Journalof Computer Applications (0975 – 8887), Volume 28– No.2.

[6] Mohammed Abdul Wajeed (PhD), Dr.T.Adilakshmi, “Text Classification Using Machine Learning”, Journal of Theoretical and Applied Information Technology, 2005 - 2009 JATIT.

[7] Jiang L., Cai Z., Wang D., and Jiang S (2007), “Survey of Improving K-Nearest-Neighbor for Classification”. In Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD ’07), vol. 1, pp. 679-683.

[8] Efron B. Bootstrap methods: another look at the jackknife. Ann Statist 1979;7:1–26.

[9] Varian, H.(2005). "Bootstrap Tutorial". *Mathematica Journal*, 9, 768-775.

[10] Weisstein, Eric W. "Bootstrap Methods." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/BootstrapMethods.html

[11] K-nearest neighbor algorithm http://en.wikipedia.org/wiki/K-earest_neighbor_algorithm.

[12] Jia Wu, Zhihua Cai, Zhechao, Gao (2010), “Dynamic K-Nearest-Neighbor with Distance and Attribute Weighted for Classification,” International Conference on Electronics and Information Engineering (ICEIE), 978-1-4244-7681 IEEE 2010.

[13] http://web.ist.utl.pt/acardoso/datasets/ (Visited before 2 Month before)