© 2017 IJSRST | Volume 3 | Issue 1 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
Themed Section: Science and Technology

A Novel Fuzzy-Bayesian Classification Method for Automatic Text Categorization

Swathi V[1], Swetha S Kumar[2], Dr. P. Perumal[3]
[1,2]Student, Bachelor of Computer Science and Engineering,
[3]Professor, Department of Computer Science & Engineering,
[1,2,3]Department of Computer Science and Engineering, Sri Ramakrishna Engineering College, Coimbatore, Tamil Nadu, India

ABSTRACT

Text categorization is mostly required to label the documents automatically with the predefined set of topics. It has been achieved by the large number of advanced machine learning algorithms. In the proposed system, fuzzy rule along with Bayesian classification method is proposed for automatic text categorization using the class-specific features. The proposed method selects the particular feature subset for each class. Then, these class features are applied for the classification. To achieve this, Baggenstoss's PDF Projection Theorem is followed to reconstruct PDF in raw data space from the class-specific PDF in low-dimensional feature space and build the fuzzy based Bayes classification rule. The noticeable significance of this method is that most feature selection criteria such as information gain and maximum discrimination which can be easily incorporated into the proposed method. The proposed classification performance is evaluated on different datasets and compared with the different feature selection methods. The experimental results illustrate that the effectiveness of the proposed method and further indicates its wide applications in text categorization.

Keywords : Text Mining, Categorization, Machine Learning, Discrimination, Feature Selection.

I. INTRODUCTION

Generally, Data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Although data mining is a relatively new term, the technology is not. Companies have used powerful computers to sift through volumes of supermarket scanner data and analyze market research reports for years. However, continuous innovations in computer processing power, disk storage, and statistical software are dramatically increasing the accuracy of analysis while driving down the cost.

As the volume of information available on the Internet and corporate increases, there is growing interest in developing tools to help people better find, filter, and manage these electronic resources. Text categorization [9][10][11] – the assignment of natural language texts to one or more predefined categories based on their content – is an important component in many information organization and management tasks. Machine learning methods, including Support Vector Machines (SVMs) [5], have tremendous potential for helping people more effectively organize electronic resources.

Human categorization is very time-consuming and costly, thus limiting its applicability especially for large or rapidly changing collections. Additional concerns such as the lack of consistency in category assignment and the need to adapt to changing category structures further limit the applicability of purely human systems. Consequently there is growing interest in developing technologies for semi automatic text categorization. Rule-based approaches similar to those used in expert systems are popular (e.g., Hayes and Weinstein's

Construe system for classifying Reuters news stories, 1990), but they generally require manual construction of the rules, make rigid binary decisions about category membership, and are typically difficult to modify. Another strategy is to use inductive learning techniques to automatically construct classifiers using labeled training data. The resulting classifiers have many advantages: they are easy to construct and update, they depend only on information that is easy for people to provide (i.e., examples of items that are in/out of categories), they can be customized for individual users, and they allow users to smoothly tradeoff precision and recall depending on their task.

A growing number of statistical classification and machine learning techniques have been applied to text categorization, including multivariate regression, nearest neighbor classifiers [3], probabilistic Bayesian models, decision trees, neural networks, symbolic rule learning, and multiplicative update algorithms. Good overviews of this text classification work can be found in Lewis and Hayes (1994) and Yang (1998). More recently, Joachims (1998) and Dumais et al. (1998) have explored the use of Support Vector Machines (SVMs) for text categorization with promising results.

II. METHODS AND MATERIAL

A. Literature Survey

Paul M. Baggenstoss [1] proposed a PDF Projection Theorem and class-specific method for optimal classification. The proposed system utilized class-dependent and data-dependent reference classes and showed the relatedness to asymptotic maximum likelihood theory. Data dependent reference classes used maximum likelihood (ML) and central limit theorem (CLT) for analyzing the feature sets. It also used class-specific approach for enabling the feature selection model. However, likelihood ratio of both raw data and feature domains was not justified by the author.

Bo Tang [2] presented efficient feature selection framework based on the Information Theory. The proposed framework ranked the features with their discriminative capacity for classification. In addition to this framework, Jeffreys-Multi-Hypothesis (JMH) divergence was presented to evaluate multi-distribution divergence and also developed feature selection methods such as maximum discrimination (MD) and MD-x^2 methods for text categorization. However dependency of features was not considered in this method.

Bo Tang and Haibo [3] proposed an Extended Nearest Neighbor (ENN) Method for predicting the input pattern in two-way communication style. ENN predicts the pattern by considering nearest neighbors of the test sample and also considers test sample as their nearest neighbors. This method enhances the classification result. This method shows computational complexity.

Xiao-Bing Xue and Zhi-Hua Zhou [4] considered the Distributional features for categorizing text. These Distributional features were used to enhance the performance of text categorization. Distributional features encode a word's distribution from various aspects. The ensemble learning techniques were further used for classification which utilizes the constructed features for text categorization. However this method was suitable when the document was long.

G. Forman [5] presented an empirical comparison of twelve feature selection methods (e.g. Information Gain) evaluated on a benchmark of 229 text classification problem instances that were gathered from Reuters, TREC, OHSUMED, etc. The results are analyzed from multiple goal perspectives—accuracy, F-measure, precision, and recall—since each is appropriate in different situations. The results reveal that a new feature selection metric we call 'Bi-Normal Separation' (BNS) outperformed the others by a substantial margin in most situations. This margin widened in tasks with high class skew, which is rampant in text classification problems and is particularly challenging for induction algorithms. A new evaluation methodology is offered that focuses on the needs of the data mining practitioner faced with a single dataset who seeks to choose one metrics that are most likely to yield the best performance.

J. Sreemathy & P. S. Balamurugan [6] proposed an efficient classification method for text classification. In this approach, for text classification, Naive Bayesian and K-Nearest Neighbor classification methods were used [3]. The classification algorithm used in this approach measures the attribute importance and use them to evaluate similarity measure. These two methods provide

better classification result in terms of effectiveness and efficiency of classification. But naïve Bayesian is strong feature independence.

B. Research Methodology

In proposed system, the automatic text categorization is performed by using fuzzy rule along with the Bayesian approach. The text categorization [8][9][10] is performed based on the fuzzy rule with Bayesian classification approach using class-specific features. The class-specific features are classified by combining fuzzy rule with Bayesian classification which facilitates improve in the probabilities of each of the classes by considering many of the features that are left un-classified or incorrectly classified by applying the Bayesian classification alone. Therefore, the classification accuracy is increased and efficiency of the classification approach is also enhanced.

Each data sample is represented as feature vector and assigned class for each feature vector. Bayesian classifier is provided for predicting the class label for labeled data sample based on the highest posterior probability conditioned on the class-specific features. The performance of Bayesian classifier is improved by the fuzzy logic approach. Fuzzy rules are simple conditional "If Then" rules which is represented as,

"If x is A Then y is B"

where, x and y are linguistic variables and A and B are linguistic values. In fuzzy logic certain missing values are considered in between the precise rules being defined by formulating the fuzzy rules accordingly. The combination of fuzzy and Bayesian classification method provides increase in probabilities of each classes. This approach is also used for improving the classification accuracy and building the classifier at a faster rate.

Thus, the figure 1 below shows the architecture diagram of the proposed system.

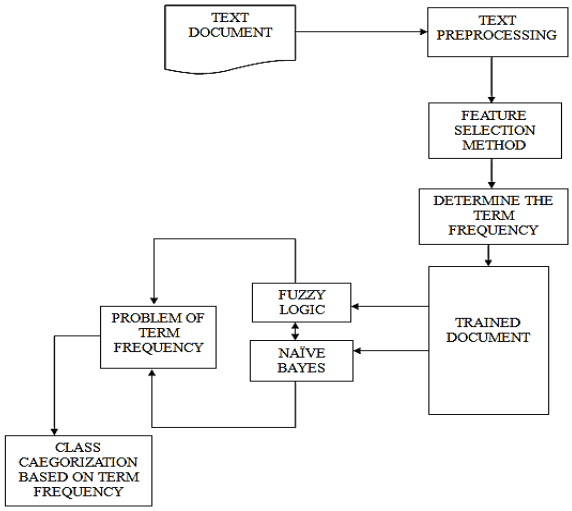

Figure 1. System Architecture Diagram

C. Pre-Processing

The preparation of input texts for elaboration is also known in the machine learning context as "data cleaning". One often-applied transformation to the input text is the substitution of characters outside of the usual 26-letter English alphabet with a single space. Multiple spaces are then reduced to one, and upper case letters are folded into lower case. These mappings will make any punctuation indistinguishable from white space, which is accepted as an un-influential loss of information. This simple transformation is only appropriate to the English language. For foreign languages; more complex transformation rules are needed. For European languages, e.g. accented characters may be replaced with their unaccented equivalents. More in detail, non-English transformations will depend on the encoding of the specific text, language, and several other factors (known as the locale).

Thus, the figure 2 below shows the pre processing of stopword /stemmer-process.

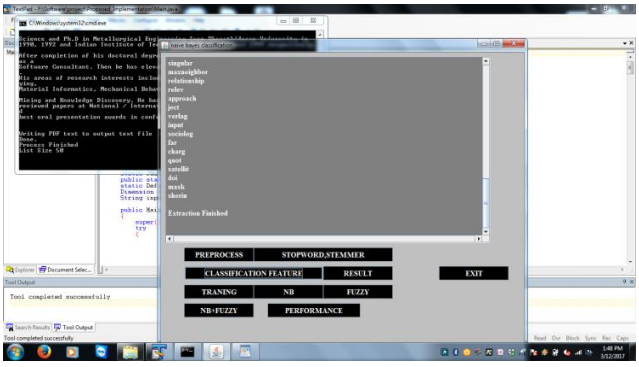

Figure 2. Pre-processing (Stopword/Stemmer–Process)

D. Feature Selection

Before any classification task, one of the most fundamental tasks that need to be accomplished is that of document representation and feature selection. While feature selection is also desirable in other classification tasks, it is especially important in text classification due to the high dimensionality of text features and the existence of irrelevant (noisy) features. In general, text can be represented in two separate ways. The first is as a bag of words, in which a document is represented as a set of words, together with their associated frequency in the document. Such a representation is essentially independent of the sequence of words in the collection. The second method is to represent text directly as strings, in which each document is a sequence of words. Most text classification methods use the bag-of-words representation because of its simplicity for classification purposes.

Thus, the figure 3 below shows the Feature Extraction from the stopword/stemmer-process.

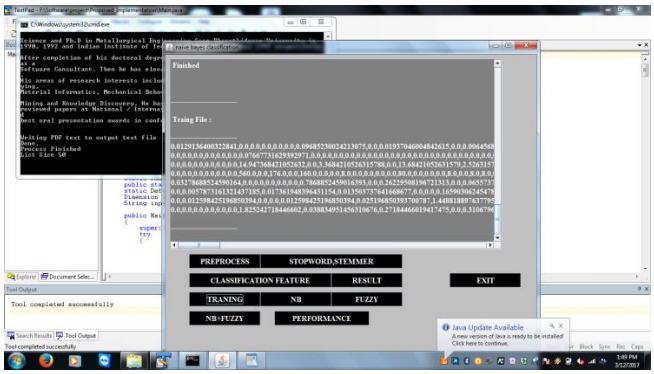

Figure 3. Feature Extraction

Considering a Text Classification (TC) problem with N predefined topics, let c_i be the class label taking value i $\in \{1, 2,\ldots,N\}$. For a given data set, we form a dictionary D with M terms. According to the concept of "bag of words", a document can be represented by a feature vector $x = [x1, x2,\ldots, x_m]^T$, where the m-th element x_m in x corresponds to the m-th term in D. In TC, both Binary and Real-valued feature models have been widely used. In Binary-valued feature model, the feature value is either 1 or 0 indicating whether or not a particular term occurs in the document. In Real valued feature model, the feature usually refers to the term frequency (TF) which is defined as the number times that a particular term appears in the document.

1. Form a reference class c_0 which consists of all documents;
2. Calculate the score of each feature based on a specific criteria, and rank the feature with the score in a descending order;
3. Choose the first K features z_i, the index of which is denoted by I_i
4. Estimate the parameters ($\theta_{(i|0)}$) under the reference class c_0 and the parameters (θ_i) under the class c_i;

Naïve Bayesian Classification

Considering a N-class classification problem, suppose that for each class c_i, where i =1, 2, …,N, we select a class-specific feature subset $z_i=f_i(x)$, where $f_i(x)$ could be a linear or nonlinear function such that the dimension of z_i is much smaller than x. Notice that we cannot apply these class-specific features z_i, where i=1,2,…,N, to the Bayes classification rule, because it is invalid to compare the discriminative information on different feature spaces. Here, we follow Baggenstoss's PDF Projection Theorem to build a classification rule using these class-specific features. The idea is to reconstruct the PDF $p(x|c_i)$ in the original feature space from the PDF $p(z_i|c_i)$ in the class-specific feature space, if we know both PDFs $p(x|c_0)$ and $p(z_i|c_0)$ under a reference hypothesis (class) c0. The reconstructed PDF can be written as

$$p(x|c_i) = \frac{p(x|c_0)}{p(z_i|c_0)} p(z_i|c_i) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (1)$$

as the PDF Projection Theorem since it projects the PDF from a low-dimensional feature space into a high-dimensional feature space. Note that in this equation, one can use class-specific reference hypotheses $c_{0,i}$ for the PDF construction of each class in theory, but we choose to use a common one c_0 in this system. By incorporating the above reconstructed PDF into the Bayes classification rule as

$$c^* \propto \underset{i\epsilon 1,2,..,N}{argmax}\, log\, \frac{p(x|c_0)}{p(z_i|c_0)} + logp(z_i|c_i) + logp(c_i) \propto \underset{i\epsilon 1,2,..,N}{argmax}\, log\, \frac{p(z_i|c_i)}{p(z_i|c_0)} + logp(c_i) \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (2)$$

The key challenge of using this equation for classification is to find a reference class c_0 in which both $p(x|c_0)$ and $p(z_i|c_0)$ can be estimated or derived. A good choice of reference class c_0 is the combination of all classes.

Thus, the figure 4 shows the classification of Naïve Bayesian algorithm.

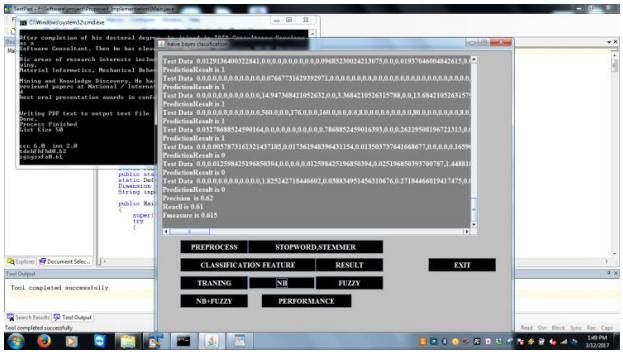

Figure 4. Naïve Bayesian Classification

E. Fuzzy Naïve Bayesian Classification

Each data sample is represented as feature vector and assigned class for each feature vector. Bayesian classifier is provided for predicting the class label for labeled data sample based on the highest posterior probability conditioned on the class-specific features. The performance of Bayesian classifier is improved by the fuzzy logic approach. Fuzzy rules are simple conditional "If Then" rules which is represented as,

"If x is A Then y is B"

Where x and y are linguistic variables and A and B are linguistic values. In fuzzy logic certain missing values are considered in between the precise rules being defined by formulating the fuzzy rules accordingly. The combination of fuzzy and Bayesian classification method provides increase in probabilities of each classes. This approach is also used for improving the classification accuracy and building the classifier at a faster rate.

Thus, the figure 5 shows the classification of fuzzy and Naïve Bayesian algorithm.

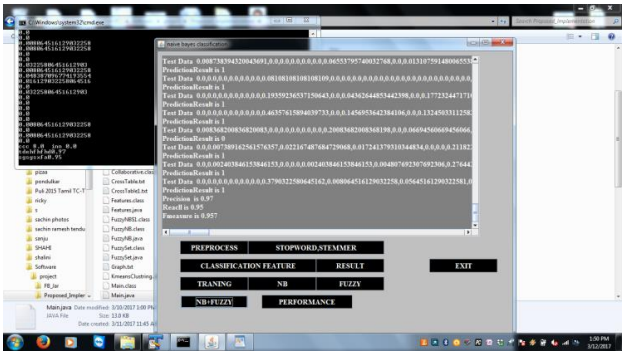

Figure 5. NB + Fuzzy Classification

F. Performance Evaluation

The performance evaluation of this work is done to prove the performance improvement over the proposed methodology than the existing system in terms of accuracy, precision, recall G-mean and F-measure.

III. RESULTS AND DISCUSSION

We further apply the performance metrics of Accuracy, Precision, Recall, F-Measure and G-Mean to measure the classification performance, which are defined as follows:

Accuracy, is the proximity of measurement results to the true value

$$\frac{TP + TN}{D}$$

Recall, ability to find positive document $\frac{TP}{P^*}$

Precision, accuracy on positive documents $\frac{TP}{P}$

F-measure, a harmonic mean of precision and recall

$$\frac{2 \times Precision \times Recall}{Precision + Recall}$$

G-mean, is the geometric mean

$$\sqrt{precsion \times recall}$$

where,

TP-True Positive, P*-Predicted Positive, P-Total Positive, D-Total Documents

Thus, the table below compares the classification metrics of Naïve Bayesian and Fuzzy-Naïve Bayesian algorithm.

TABLE.1 Benchmark of Time taken to process

Classification		Accuracy	F-measure	Precision	Recall	G-mean
Naïve Bayesian	DM	84.0	0.579	0.584	0.574	0.714
	NW	83.0	0.436	0.441	0.431	0.571
	SE	86.0	0.7221	0.727	0.717	0.654
	NG	85.89	0.769	0.657	0.678	0.693
Fuzzy-Naïve Bayesian	DM	92.81	0.9671	0.684	0.664	0.812
	NW	91.4	0.815	0.827	0.807	0.857
	SE	94.68	0.95	0.97	0.957	0.929
	NG	93.81	0.935	0.97	0.978	0.980

Thus, the figure 6 shows the performance metrics of Accuracy.

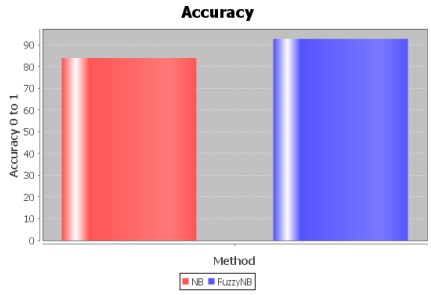

Figure 6. Performance metrics of Accuracy

Thus, the figure 7 shows the performance metrics of Precision.

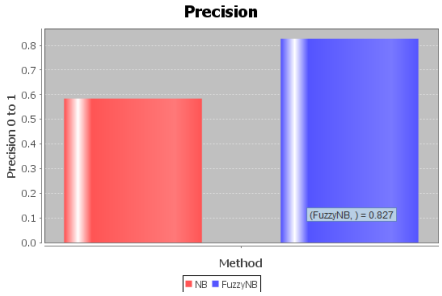

Figure 7. Performance metrics of Precision

Thus, the figure 8 shows the performance metrics of F-Measure.

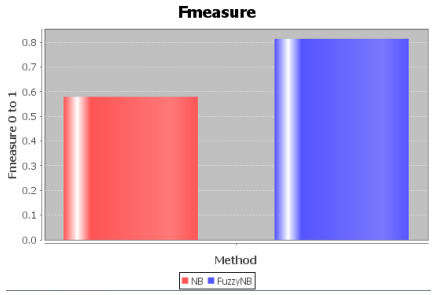

Figure 8. Performance metrics of F-measure

Thus, the figure 9 shows the performance metrics of G-Mean.

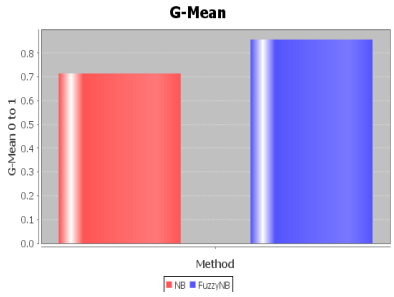

Figure 9. Performance metrics of G-mean

Thus, the figure 10 shows the performance metrics of Recall.

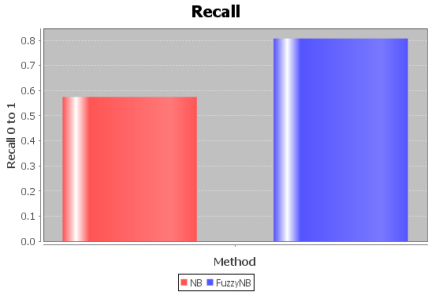

Figure 10. Performance metrics of Recall

Thus, the Performance Evaluation of the above figures concludes that fuzzy gives better performance when compared to Naïve Bayesian algorithm.

IV. CONCLUSION

In the proposed system, an efficient method called as fuzzy naïve bayes classifier is proposed for automatic text categorization. Here the most important features are selected for each classes those features are termed as class-specific features. The class specific features reduced the dimensionality of features and it speed up the process of fuzzy naïve bayes classifier. The texts are categorized by combining the fuzzy rule with Bayesian classification which facilitates improvement in the probabilities of each of the classes by considering many of the features that are left un-classified or incorrectly classified by applying the Bayesian classification alone. The experimental results proved that the proposed fuzzy naïve bayes classifier has high accuracy, high F-measure and high G-mean than the existing method.

Furthermore, in future the proposed system is to categorize large datasets that are stored and analyzed directly in the cloud.

V. REFERENCES

[1] Bo Tang, Haibo He, Paul M. Baggenstoss and Steven Kay, "A Bayesian Classification Approach Using Class-Specific Features for Text Categorization", IEEE Transactions on Knowledge and Data Engineering Vol. 28, Issue. 6, pp. 1602 – 1606, 2016.

[2] Bo Tang, Steven Kay and Haibo He, "Toward Optimal Feature Selection in Naïve Bayes for Text Categorization", IEEE Transactions on

Knowledge and Data Engineering, Vol.28, Issue. 9, pp. 2508 – 2521, 2016.

[3] Bo Tang and Haibo He, "ENN: Extended Nearest Neighbor Method for Pattern Recognition", IEEE Computational Intelligence Magazine, Vol. 10, Issue. 3, pp. 52 – 60, 2015.

[4] Xiao-Bing Xue and Zhi-Hua Zhou, "Distributional Features for Text Categorization", IEEE Transactions on Knowledge and Data Engineering, Vol. 21, Issue. 3, pp. 428 – 442, 2009.

[5] G. Forman, "An extensive empirical study of feature selection metrics for text classification," The Journal of machine learning research, vol. 3, pp. 1289–1305, 2003.

[6] J.Sreemathy and P. S. Balamurugan, "An Efficient Text Classification using KNN and Naive Bayesian", International Journal on Computer Science and Engineering (IJCSE), Vol. 4, No. 03, pp. 392 – 396, 2012.

[7] J Upendra Singh and Saqib Hasan, "Survey Paper on Document Classification and Classifiers", International Journal of Computer Science Trends and Technology (IJCST) – Vol. 3, Issue. 2, pp. 83 – 87, 2015.

[8] Mital Vala and Jay Gandhi, "Survey of Text Classification Technique and Compare Classifier", International Journal of Innovative Research in Computer and Communication Engineering, Vol. 3, Issue 11, pp. 10809 - 10813, 2015.

[9] Rajni Jindal, Ruchika Malhotra and Abha Jain, "Techniques for text classification: Literature review and current trends", Webology, Vol. 12, No. 2, pp. 1 - 28, 2015.

[10] Anuradha Purohit, Deepika Atre, Payal Jaswani and Priyanshi Asawara, "Text Classification in Data Mining", International Journal of Scientific and Research Publications, Vol. 5, Issue. 6, pp. 1 – 7, 2015.

[11] Kujguhj Bhumika, Sukhjit Singh Sehra, and Anand Nayyar, "A Review Paper on Algorithms used for Text Classification", International Journal of Application or Innovation in Engineering & Management (IJAIEM), Vol. 2, Issue. 3, pp. 90 – 99, 2013.