International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 1 Issue 7, September - 2012

An Efficient Self Constructing Algorithm for Text Categorization

B.RAMA KRISHNA[1], J.RAMESH[2]

[1] *M.tech Student, Dept of Computer Science and Engineering,*

QIS COLLEGE OF ENGG. & TECH.,ONGOLE,Andhra Pradesh, India.

[2] *Asst.Professor, Dept. of Computer Science and Engineering*

QIS COLLEGE OF ENGG. & TECH., ONGOLE, Andhra Pradesh, India

***Abstract*— Text categorization is a predefined category to the natural language text. One of the major characteristic of text document classification problem is extremely reduce high dimensionality of text data in to low dimensionality. In this paper we introduce a naive algorithm for feature/word selection for the purpose of text classification. We use sequential forward selection methods based on improved mutual information criterion functions. The performance of the proposed evaluation functions compared to the information gain which evaluate features individually is discussed. We present experimental results using naive Baye's classifier based on multinomial model, linear support vector machine and k-nearest neighbour classifiers on the Reuters, Webkb's, 20 news subgroups data sets .**

Feature clustering is a powerful method to reduce the dimensionality of feature vectors for text classification. In this paper, we propose a fuzzy similarity-based self-constructing algorithm for feature clustering. The words in the feature vector of a document set are grouped into clusters, based on similarity test. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature, corresponding to a cluster, is a weighted combination of the words contained in the cluster. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided.

***Keywords*— Feature/word selection, Support vector machine, feature reduction, feature clustering, feature extraction, Tfidf and weight.**

I. INTRODUCTION

The goal of text document classification is to assign automatically a new document into one or more predefined classes based on its contents. An increasing number of statistical classification methods and machine learning algorithms have been explored to build automatically a classifier by learning from previously labelled documents including naive Bayes, k-nearest neighbour, support vector machines, neural network, decision trees and logistic regression. The overview of discusses the main approaches to text classification.

We propose a fuzzy similarity-based self-constructing feature clustering algorithm, which is an incremental feature clustering approach to reduce the number of features for the text classification task. The words in the feature vector of a document set are represented as distributions and processed one after another. Words that are similar to each other are grouped into the same cluster. Each cluster is characterized by a membership function with statistical mean and deviation. If a word is not similar to any existing cluster, a new cluster is created for this word. Similarity between a word and a cluster is defined by considering both the mean and the variance of the cluster. When all the words have been fed in, a desired number of clusters are formed automatically. We then have one extracted feature for each cluster. The extracted feature corresponding to a cluster is a weighted combination of the words contained in the cluster. Three ways of weighting, hard, soft, and mixed, are introduced. By this algorithm, the derived membership functions match closely with and describe properly the real distribution of the training data. Besides, the user need not specify the number of extracted features in advance, and trial-and-error for determining the appropriate number of extracted features can then be avoided. Experiments on real-world data sets show that our method can run faster and obtain better extracted features than other methods.

II. DEFINITIONS AND NOTATIONS

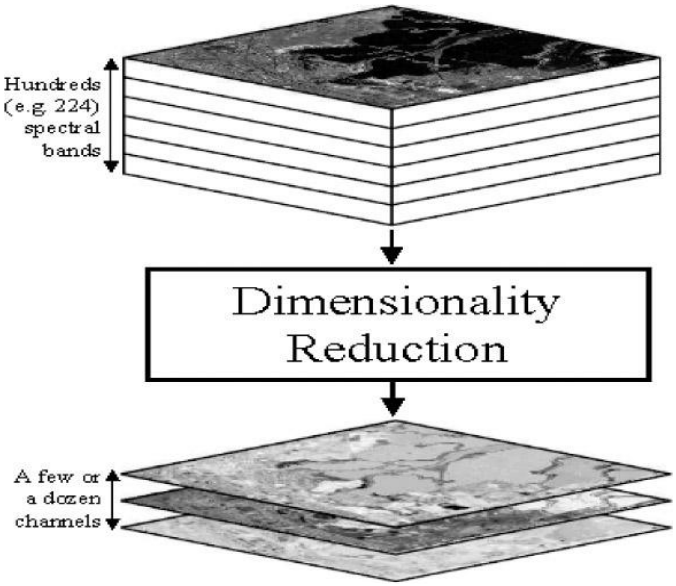
Let $C = \{c_1, \ldots, c_{|C|}\}$ be the set of $|C|$ predefined classes and let $D = \{d_1, \ldots, d_{|D|}\}$ be the finite training document set. Let $V = \{w_1, \ldots, w_{|V|}\}$ be the vocabulary set containing $|V|$ distinct words occurred in training documents. Given a set of document vectors $\{d_1, \ldots, d_{|D|}\}$ and their associated class labels $c(d_j) = \{c_1, \ldots, c_{|C|}\}$, text classification is the problem of estimating the true class label of a new document. Text documents cannot be directly interpreted by a classifier. According to the bag-of-words representation, the document d can be represented by a feature vector consisting of one feature variable for each word in the given vocabulary set V. A common characteristic of text data is its extremely high

dimensionality. The number of potential features (several tens of thousands) exceeds the number of training documents.

To process documents, the bag-of-words model is commonly used. Let **D** = { d_1; d_2; . . . ; d_n } be a document set of *n* documents, where d_1, d_2; . . . ; d_n are individual documents, and each document belongs to one of the classes in the set {c_1; c_2; . . . . . ; c_p}. If a document belongs to two or more classes, then two or more copies of the document with different classes are included in **D**. Let the word set W = {w_1; w_2; . . . ; w_m} be the feature vector of the document set. Each document d_i, $1 \le i \le n$, is represented as d_i = <d_{i1}; d_{i2}; . . . ; d_{im}>, where each d_{ij} denotes the number of occurrences of w_j in the *i*th document. The feature reduction task is to find a new word set W' = {w'_1; w'_2; . . . ; w'_k}, $k << m$, such that **W** and **W**' work equally well for all the desired properties with **D**. After feature reduction, each document d_i is converted into a new representation d'_i = <d'_{i1}; d'_{i2}; . . . ; d'_{ik}> and the converted document set is D' = {d'_1; d'_2; . . . ; d'_n}. If *k* is much smaller than *m*, computation cost with subsequent operations on D' can be drastically reduced.

A. Dimensionality Reduction

Dimensionality reduction is a very important step in text classification because irrelevant and redundant features often degrade the performance of classification algorithms both in speed and classification accuracy. The number of features can be dramatically reduced by the domain dependent methods which include the elimination of stop words, stripping of special characters as well as stemming algorithms or morphological analysis. For a further dimensionality the domain independent methods can be used.



B. Feature Subset Selection

In text classification the dominant approach to dimensionality reduction is feature selection. Given a predefined integer | V' |, methods for word selection attempt to select from the original set V, the set V' of words with | V' | $\ll$ |V| that, when used for document representation, yields the highest effectiveness. Different methods for feature subset selection have been developed in pattern recognition and machine learning using different evaluation functions and search procedures.

C. Feature Reduction

In general, there are two ways of doing feature reduction, feature selection, and feature extraction. By feature selection approaches, a new feature set W' = {w'_1, w'_2, . . . , w'_k} is obtained, which is a subset of the original feature set **W**. Then W' is used as inputs for classification tasks. Information Gain (IG) is frequently employed in the feature selection approach [10]. It measures the reduced uncertainty by an information-theoretic measure and gives each word a weight. The weight of a word w_j is calculated as follows:

$$IG(w_j) = -\sum_{l=1}^{p} P(c_l)\, log\, P(c_l) + P(w_j)\sum_{l=1}^{p} P(c_l|w_j)\, log\, P(c_l|w_j) + P(\overline{w}_j)\sum_{l=1}^{p} P(c_l|\overline{w}_j)\, log\, P(c_l|\overline{w}_j) \quad (1)$$

where P (c_l) denotes the prior probability for class c_l, P(w_j) denotes the prior probability for feature w_j, P ($\overline{w}_j$) is identical to 1-*P*(w_j), and $P(c_l|w_j)$ and P (c_l |$\overline{w}_j$) denote the probability for class c_l with the presence and absence, respectively, of w_j. The words of top k weights in W are selected as the features in W'.

In feature extraction approaches, extracted features are obtained by a projecting process through algebraic transformations. An incremental orthogonal centroid (IOC) algorithm was proposed in [14]. Let a corpus of documents be represented as an $m \times n$ matrix $X \in R^{m \times n}$, where *m* is the number of features in the feature set and *n* is the number of documents in the document set. IOC tries to find an optimal transformation matrix $F^* \in R^{m \times k}$, where *k* is the desired number of extracted features, according to the following criterion:

$$F^* = arg\ max\ trace(F^T S_b F), \quad (2)$$

where $F^* \in R^{m \times k}$ and $F^T F = I$, and

$$S_b = \sum_{q=1}^{p} P(c_q)(M_q - M_{all})(M_q - M_{all})^T \quad (3)$$

with P (c_q) being the prior probability for a pattern belonging to class c_q, M_q being the mean vector of class c_q, and M_{all} being the mean vector of all patterns.

D. Feature Extraction

Feature Extraction is a method of retrieving information from reduced data. That is which data you would like to retrieve then automatically extract from that area. Extracted data in the sense already pre processed data. This is also one of the method for text classification.

E. Text Classification

Text classification [10] (also known as text categorization or topic spotting) is the task of automatically sorting a set of

documents into categories from a predefined set. This task has several applications, including automated indexing of scientific articles according to predefined thesauri of technical terms, filing patents into patent directories, selective dissemination of information to information consumers, automated population of hierarchical catalogues of Web resources, spam filtering, identification of document genre, authorship attribution, survey coding, and even automated essay grading. Automated text classification is attractive because it frees organizations from the need of manually organizing document bases, which can be too expensive, or simply not feasible given the time constraints of the application or the number of documents involved. The accuracy of modern text classification systems rivals that of trained human professionals, thanks to a combination of information retrieval (IR) technology and machine learning (ML) technology.

F. Feature Clustering

Feature clustering is an efficient approach for feature reduction, which groups all features into some clusters, where features in a cluster are similar to each other. The feature clustering methods [are "hard" clustering methods, where each word of the original features belongs to exactly one word cluster. Therefore each word contributes to the synthesis of only one new feature. Each new feature is obtained by summing up the words belonging to one cluster. Let D be the matrix consisting of all the original documents with m features and D' be the matrix consisting of the converted documents with new *k* features. The new feature set $W' = \{w'_1; w'_2; \ldots; w'_k\}$, corresponds to a partition $(W_1, W_2, \ldots, W_k)$ of the original feature set W, i.e., $W_t \cap W_q = \emptyset$ where $1 \leq q$; $t \leq k$ the partition. Then, the *t*th feature value of the converted document d'_i is calculated as follows:

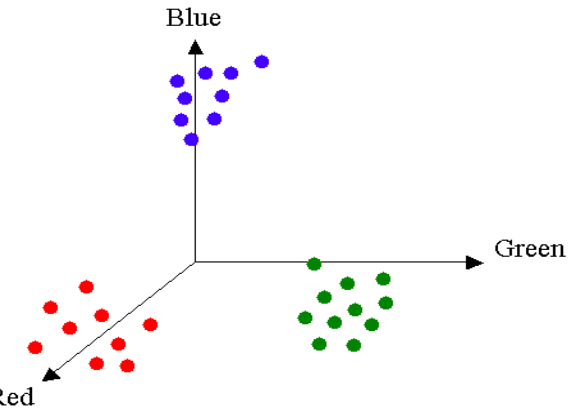
$$d'_{it} = \sum_{w_j \in w_t} d_{ij}, \qquad (4)$$

which is a linear sum of the feature values in Wt. The divisive information-theoretic feature clustering (DC) algorithm, calculates the distributions of words over classes, $P(C_j|w_j)$, $1 \leq j \leq m$,

where $C = \{c_1; c_2; \ldots; c_p\}$, and uses Kullback-Leibler divergence to measure the dissimilarity between two distributions. The distribution of a cluster W_t is calculated as follows:

$$P(C|W_t) = \sum_{w_j \in w_t} \frac{P(w_j)}{\sum_{w_j \in w_t} P(w_j)} P(C|w_j). \qquad (5)$$

The goal of DC is to minimize the following objective function:



$$\sum_{t=1}^{k} \sum_{w_j \in w_t} P(w_j) KL((P)\, P(C|w_j)\, P(C|W_t)) \qquad (6)$$

which takes the sum over all the *k* clusters, where *k* is specified by the user in an Pixel Feature Clustering

G. Term Frequency and inverse document frequency:

Tfidf is a form of assigning frequent words in the documents. Basically, the term frequency describes how frequently the word appears in the document. At the same time the inverse document frequency describes how the term or word frequently appears in the remaining documents. We can add another feature for this classification, i.e, compactness. This compactness describes position of the word in particular document. Totally these features represents what is the first appearance of the word, what is the last appearance of the word. It is also describes middle of the word. Tfidf is most powerful method for classification.

III. PROPOSED PROCEDURE

There are some issues pertinent to most of the existing feature clustering methods. First, the parameter *k*, indicating the desired number of extracted features, has to be specified in advance. This gives a burden to the user, since trial-and-error has to be done until the appropriate number of extracted features is found. Second, when calculating similarities, the variance of the underlying cluster is not considered. Intuitively, the distribution of the data in a cluster is an important factor in the calculation of similarity. Third, all words in a cluster have the same degree of contribution to the resulting extracted feature. Sometimes, it may be better if more similar words are allowed to have bigger degrees of contribution. Our feature clustering algorithm is proposed to deal with these issues.

Suppose, we are given a document set D of *n* documents $d_1, d_2, \ldots, d_n$, together with the feature vector W of *m* words $w_1, w_2, \ldots w_m$ and *p* classes $c_1, c_2, \ldots, c_p$, as specified in Section 2. We construct one word pattern for each word in W. For word w_i, its word pattern x_i is defined, similarly as in [21], by

$$x_i = \langle x_{i1}, x_{i2}, \ldots, x_{ip} \rangle,$$
$$= \langle P(c_1|w_i), P(c_2|w_i), \ldots, P(c_p|w_i) \rangle$$

Where

$$P(c_i|w_j) = \frac{\sum_{q=1}^{n} d_{qi} \times \delta_{qj}}{\sum_{q=1}^{n} d_{qi}} \qquad (8)$$

for $1 \leq j \leq p$. Note that d_{qi} indicates the number of occurrences of w_i in document d_q, as described in Section 2. Also, δ_{qj} is defined as

$$\delta_{qj} = \begin{cases} 1; & \text{if document } d_q \text{ belongs to class } c_j; \\ 0; & \text{otherwise:} \end{cases} \qquad (9)$$

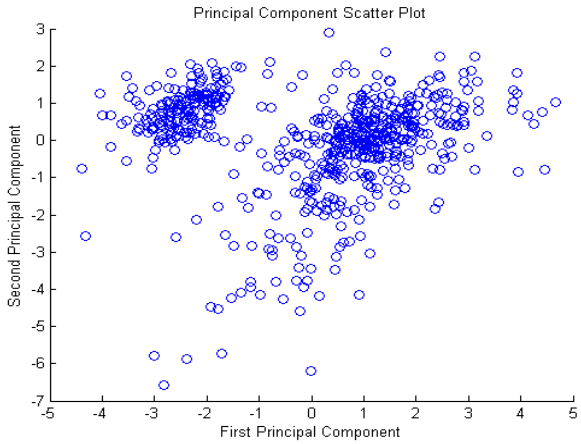


Fig. 2 principal components can be now be **clustered** for **Self**-Organizing

Therefore, we have m word patterns in total. For example, suppose we have four documents d_1, d_2, d_3, and d_4 belonging to c_1, c_1, c_2, and c_n, respectively. Let the occurrences of w_1 in these documents be 1, 2, 3, and 4, respectively. Then, the word pattern x_1 of w_1 is:

$$P(c_1|w_1) = \frac{1\times1+2\times1+3\times0+4\times0}{1+2+3+4} = 0.3$$

$$P(c_2|w_1) = \frac{1\times0+2\times0+3\times1+4\times1}{1+2+3+4} = 0.7 \qquad (10)$$

X_1 = <0:3; 0:7>:

We consider the global filtering approach to feature selection in text document task. In this section novel algorithms for feature selection using mutual information are presented.

1. *Feature Selection using Mutual Information*

Our feature subset selection problem is formulated as follows: Given an initial set V with $|V|$ features, find the subset $S_s \subset V$ with $|S|$ features that maximizes the mutual information for text defined as mutual information for a set of words averaged over all classes given by the following formula:

$$MI(S) = \sum_{k=1}^{|c|} P(c_k) I(c_k, S) \qquad (1)$$

The mutual information for a feature/word w (word w occurred) averaged over all classes is defined as:

$$MI(w) = \sum_{k=1}^{|c|} P(c_k) \frac{P(w|c_k)}{P(w)} = \sum_{k=1}^{|c|} P(c_k) I(c_k, w) \qquad (2)$$

Here P(w) is the probability, that the word w occurred, $P(c_k)$ is the probability of the class c_k, $P(w|c_k)$ is the conditional probability of the word w given the class ck, $I(c_k,w)$ is the mutual information between class ck and word w. We can consider three strategies for solving our feature selection problem.

The optimal strategy generates all the word subsets S and compares their *MI(S)*. It is almost impossible for too many combinations. In the backward elimination strategy we remove the worst word from the complete set V one by one till the required number of words remain. This procedure has a lot of difficulties in computing $I(c_k, S)$.

IV. Clustering Algorithm

The whole clustering algorithm can be summarized below.

Initialization
  #of original word patterns: m
  #of classes: p
  Threshold: Initial
  deviation: ρ
  Initial deviation: σ_0
  Initial of clusters: $k = 0$
Input:
  $x_i = < x_{i1}; x_{i2}; \ldots ; x_{ip}>$, $1 \leq i \leq m$
Output:
  Clusters $G_1, G_2; \ldots ; G_k$
procedure Self-Constructing-Clustering-Algorithm
  for each word pattern x_i, $1 \leq i \leq m$
    *temp*_W = $G_j | \mu_{Gj}(x_i)$; $1 \leq j \leq k$};
    if(*temp*_W==∅)
      A new cluster G_h, $h = k+1$
  else let $G_t \in$ *temp*_W be the cluster to which x_i is
      closest by (19);
Incorporate x_i into G_t by (20)-(24);
      endif;
endfor;
return with the created k clusters;
endprocedure

Note that the word patterns in a cluster have a high degree of similarity to each other. Besides, when new training patterns are considered, the existing clusters can be adjusted or new clusters can be created, without the necessity of generating the whole set of clusters from the scratch. The order in which the word patterns are fed in influences the clusters obtained. We apply a heuristic to determine the order. We sort all the patterns, in decreasing order, by their largest components. Then the word patterns are fed in this order. In this way, more significant patterns will be fed in first and likely become the core of the underlying cluster. For example, let x_1 = <0:1; 0:3;

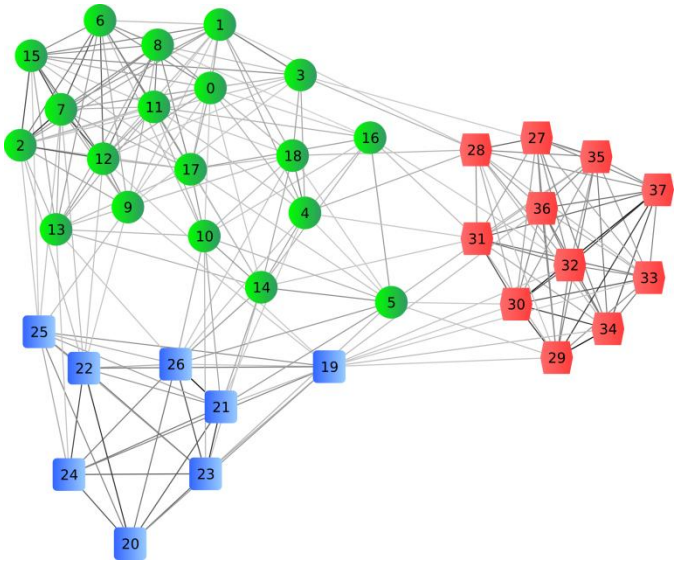


Fig. 3 Feature Clustering Diagram

The following are the results for text classification. This classification basically depend on clustering after that each word in the input file must be compared to trained data set. Based on weight this algorithm classifies input data and represent individual classification results.

TABLE I
COMPARE OF RESULTS AMONG THREE ALGORITHMS IN IRIS DATASET

	Accuracy Rate of classification			Running time		
	Best	Worst	Avg.	Best	Worst	Avg.
SVM	1	0.833	0.9033	0.1406	0.2813	0.19064
GA-SVM	1	1	1	6.9063	7.8906	7.3172
EGA-SVM	1	1	1	6.6094	7.3594	6.93439

TABLE 2
CLASSIFICATION RESULT

List of Input Text Documents	Classification(Based on all algorithms)
Crude	C1
Wheat	C1
Reuters	C3
Webkb	C5

V. CONCLUSIONS

This paper mainly focused on the problem of feature selection for document classification. In this paper we presented methods based on novel improved mutual information measures. The proposed algorithms are new in the text classification field. It uses good optimization performance of support vector machines to improve classification performance of genetic algorithm with elite strategy. Iris dataset and a text dataset are chosen to validity performance of the combing algorithm. It's obviously that the hybrid algorithm and feature Clustering can be applied to classify literatures in the field of electrical engineering. Future study direction will focus on the effect to performance when related parameters, such as crossing-over rate, mutation rate, size of population, etc., have different values, and improve computational efficiency of the new algorithm further.

Similarity-based clustering is one of the techniques we have developed in our machine learning research. In this paper, we apply this clustering technique to text categorization problems. We are also applying it to other problems, such as image segmentation, data sampling, fuzzy modelling, web mining, etc. We found that when a document set is transformed to a collection of word patterns, the relevance among word patterns can be measured, and the word patterns can be grouped by applying our similarity-based clustering algorithm. Our method is good for text categorization problems due to the suitability of the distributional word clustering concept.

REFERENCES

[1] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using em," Machine Learning, vol. 39, pp. 103–134, 2000.
[2] A. McCallum and K. Nigam, "A comparison of event models for naive
[3] Bayes text classification," in Proceedings of the AAAI-98 Workshop on Learning for Text Categorization, 1998.
[4] S. M. Metev and V. P. Veiko, *Laser Assisted Microtechnology*, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.
[5] J. Breckling, Ed., *The Analysis of Directional Time Series: Applications to Wind Speed and Direction*, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.
[6] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," *IEEE Electron Device Lett.*, vol. 20, pp. 569–571, Nov. 1999.
[7] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, "High resolution fiber distributed measurements with coherent OFDR," in *Proc. ECOC'00*, 2000, paper 11.3.4, p. 109.
[8] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, "High-speed digital-to-RF converter," U.S. Patent 5 668 842, Sept. 16, 1997.
[9] (2002) The IEEE website. [Online]. Available: http://www.ieee.org/
[10] M. Shell. (2002) IEEEtran homepage on CTAN. [Online]. Available: http://www.ctan.org/texarchive/macros/latex/contrib/supported/IEEEtran/*FLEXChip Signal Processor (MC68175/D)*, Motorola, 1996.
[11] Support vector machine. http://en.wikipedia.org/wiki/Support_vector_machine [2009-10-6]
[12] Genetic algorithm. http://en.wikipedia.org/wiki/Genetic_algorithm. [2009-10-6]
[13] Text classification. http://en.wikipedia.org/wiki/Text_classification[2009-10-6]
[14] Vector space model. http://en.wikipedia.org/wiki/Vector_space_model [2009-10-6]
[15] "PDCA12-70 data sheet," Opto Speed SA, Mezzovico, Switzerland.
[16] A. Karnik, "Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP," M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.
[17] J. Padhye, V. Firoiu, and D. Towsley, "A stochastic model of TCP Reno congestion avoidance and control," Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.
[18] *Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification*, IEEE Std. 802.11, 1997.
[19] Li, S.T., Wu, X.X., and Hu, X.Y.: 'Gene selection using genetic algorithm and support vectors machines', Soft Computing, 2008, 12, (7), pp. 693-698
[20] FENG He-long and XIA Sheng-ping. Web Page Classification Method Based On RSOM-Bayes. Computer Engineering, 2008,34(13),pp.61-63.

[21] Liu Li-zhen, HE Hai-jun, Lu Yu-chang and Song Han-tao. Application Research of Support Vector Machine in Web Information Classification. MINI-MICRO SYSTEM, 2007,28(2), pp.337-340.
[22] Distributional features for text categorization, IEEE march 2009, Xiao-Bing Xue and Zhi-Hua Zhou, Senior Member, IEEE

AUTHORS PROFILE

Mr..Ramesh currently working as an Asst.Professor in Dept of CSE. He received his B.Tech from Chundi Ranganayakulu Engg. college, JNTU Hyderabad and M.Tech from QISCET,JNTU Hyderabad. He has 5 years of teaching experience and published many papers at various international journals and conferences. His Research areas includes Data Mining and Data Warehousing .

B.Rama Krishna currently pursuing M.Tech (CSE) from QIS College of Engineering and Technology, Ongole, Affiliated to JNTU Kakinada. His Interested areas includes Data Mining and Data Warehousing.