Genetic Algorithm and Confusion Matrix for Document Clustering

A. K. Santra, C. Josephine Christy,
[1] Dean, CARE School of Computer Applications , Trichy – 620 009, India.

[2] Research Scholar, Bharathiar University, Coimbatore – 638401,India.

Abstract

Text mining is one of the most important tools in Information Retrieval. Text clustering is the process of classifying documents into predefined categories according to their content. Existing supervised learning algorithms to automatically classify text requires sufficient documentation to learn exactly. In this paper, Niching memetic algorithm and Genetic algorithm (GA) is presented in which feature selection an integral part of the global clustering search procedure that attempts to overcome the problem of finding optimal solutions at the local less promising in both clustering and feature selection. The concept of confusion matrix is then used for derivative works, and finally, hybrid GA is included for the final classification. Experimental results show benefits by using the proposed method which evaluates F-measure, purity and results better performance in terms of False positive, False negative, True positive and True negative.

Keywords: Text mining, GA, Confusion matrix, F-measure

1. Introduction

In Text data mining, Text classification has become one of the most important techniques. The task is to automatically classify documents into predefined classes based on their content. Many algorithms have been developed to deal with document clustering. With the existing algorithms, a number of newly established processes are involving in the automation of Document clustering. It has been observed that for the purpose of Document clustering the concept of association rule is very well known. Association rule mining finds interesting association or correlation relationships among a large set of data items. The discovery of these relationships among huge amounts of transaction records can help in many decision making process. On the other hand, the confusion matrices use the maximum a posteriori estimation for learning a classifier. It assumes that the occurrence of each word in a document is conditionally independent of all other words in that document given its class.

The confusion matrix is more commonly named contingency table in which the matrix could be arbitrarily large. The number of correctly classified instances is the sum of diagonals in the matrix; all others are incorrectly classified accurately. Improved Genetic algorithm starts with an initial population which is created consisting of randomly generated rules. Each rule can be represented by a string of bits. Based on the notion of survival of the fittest, a new population is formed to consist of the fittest rules in the current population, as well as offspring of these rules. Typically, the fitness of a rule is assessed by its classification accuracy on a set of training examples.

This paper presents an improved genetic algorithm which is used to evaluate the weights of the metrics such as F-measure, purity and accuracy. We apply improved genetic algorithm to find out and identify the potential informative features combinations for classification and then use the F-Measure to determine the fitness in genetic algorithm. The improved GA is general purpose search algorithm which provides rules inspired by natural genetic populations to evaluate solutions to problems. In our method, not as usual, an individual is joined together of the real-coded metrics' weight, and it's more natural to indicate the optimization problem in the continuous domain.

2. Literature Review

A. K. Santra, C. Josephine Christy and B. Nagarajan [1] have proposed that cluster based niche memetic and genetic algorithm have been designed & implemented by optimizing feature selection of text in the document repository. The contribution of genetic algorithm works with an evaluation of fitness function. Accuracy can be calculated through the document clustering. S. Areibi and Z. Yang [2] have proposed several local search operations to effectively design an MA for simultaneous clustering and feature selection. which incorporate local searches with traditional GAs,

have been proposed and applied successfully to solve a wide variety of optimization problems. These studies show that pure GAs are not well suited to fine tuning structures in complex search spaces and that hybridization with other techniques can greatly improve their efficiency. S. Wu *et al.*[3] have proposed about data clustering is a common technique for statistical data analysis and has been used in a variety of engineering and scientific disciplines such as biology (genome data). Y. Zhao and G. Karypis [5] have proposed the purity of a cluster represents the fraction of the cluster corresponding to the largest class of documents assigned to that cluster; thus, the purity of the cluster.

One way of approaching this challenge is to use stochastic optimization schemes, prominent among which is an approach based on genetic algorithms (GAs). The GA is biologically inspired and embodies many mechanisms mimicking natural evolution. It has a great deal of potential in scientific and engineering optimization or search problems. Recently, hybrid methods [8], which incorporate local searches with traditional GAs, have been proposed and applied successfully to solve a wide variety of optimization problems. These studies show that pure Gas [16] are not well suited to finetuning structures in complex search spaces and that hybridization with other techniques can greatly improve their efficiency. GAs that have been hybridized with local searches are also known as memetic algorithms (MAs) [7].

Traditional GAs and MAs are generally suitable for locating the optimal solution of an optimization problem with a small number of local optima. Complex problems such as clustering, however, often involve a significant number of locally optimal solutions. In such cases, traditional GAs and MAs cannot maintain controlled competitions among the individual solutions and can cause the population to converge prematurely [3]. To improve the situation, various methods [7], (usually called niche methods) have been proposed. The research reported shows that one of the key elements in finding the optimal solution to a difficult problem with a GA approach is to preserve the population diversity during the search, since this permits the GA to investigate many peaks in parallel and helps in preventing it from being trapped in local optima. GAs are naturally applicable to problems with exponential search spaces and have consequently been a significant source of interest for clustering [6, 10]. For example, in [4] proposed the use of traditional GAs for partitioned clustering. These methods can be very expensive and susceptible to becoming trapped in locally optimal solutions for clustering large data sets.

In [8] introduced hybrid GAs by incorporating clustering-specified local searches into traditional GAs. In contrast to the methods proposed in [11] and [12], clustering based on hybrid GAs can be more efficient, but these techniques can still, however, suffer from premature convergence. Furthermore, all of the above methods may exhibit limited performance, since they perform clustering on all features without selection. GAs have also been proposed for feature selection [7]. However, they are usually developed in the supervised learning context, where class labels of the data are available, and the main purpose is to reduce the number of features used in classification while maintaining acceptable classification accuracies. The second (and related) theme is feature selection for clustering, and feature selection research has a long history, as reported in the literature.

Feature selection in the context of supervised learning, adopts methods that are usually divided into two classes filters and wrappers based on whether or not feature selection is implemented independently of the learning algorithm. To maintain the filter/wrapper distinction used in supervised feature selection, we also classify feature selection methods for clustering into these two categories based on whether or not the process is carried out independently of the clustering algorithm [13, 14, 15]. The filters in clustering basically preselect the features and then apply a clustering algorithm to the selected feature subset. The principle is that any feature carrying little or no additional information beyond that subsumed by the remaining features is redundant and should be eliminated.

3. Document Clustering

While document clustering can be valuable for categorizing documents into meaningful groups, the usefulness of categorization cannot be fully appreciated without labeling those clusters with the relevant keywords or key phrases that describe the various topics associated with them. A highly accurate key phrase extraction algorithm, called Core Phrase is proposed for this particular purpose.

Core Phrase works by building a complete list of phrases shared by at least two documents in a cluster. Phrases are assigned scores according to a set of features calculated from the matching process. The candidate phrases are then ranked in descending order and the top L phrases are output as a label for the cluster. While this algorithm on its own is useful for labeling document clusters, it is used to produce cluster summaries for the collaborative clustering algorithm.

Document clustering is used to organize a large document collection into distinct groups of similar documents. It discerns general themes hidden within the corpus. Applications of document clustering go beyond organizing document collections into knowledge maps. This can facilitate subsequent knowledge retrievals and accesses. Document clustering, for example, has been applied to improve the efficiency of text categorization

and discover event episodes in temporally ordered documents. In addition, instead of presenting search results as one long list, some prior studies and emerging search engines employ a document clustering approach to automatically organize search results into meaningful categories and thereby support cluster-based browsing.

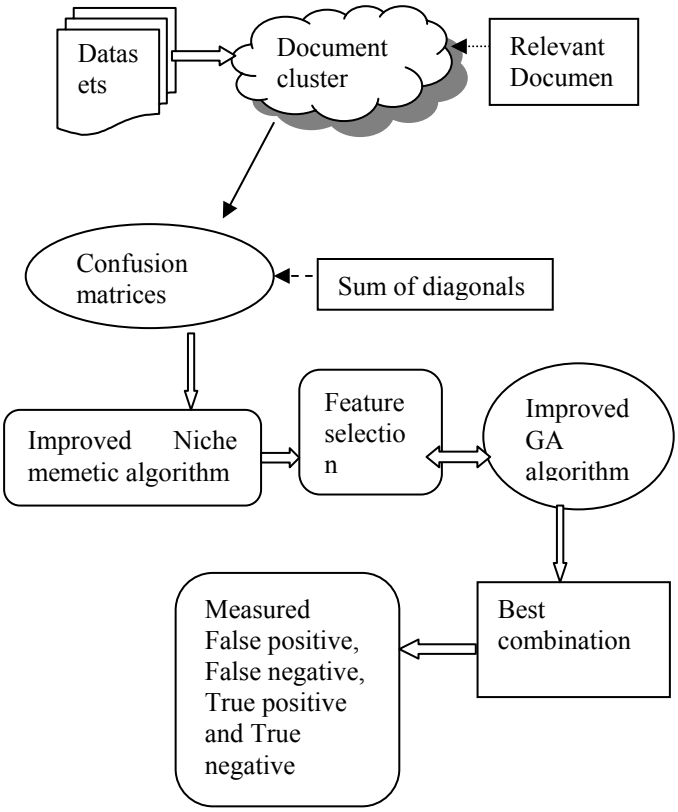

Fig 1 : Document Clustering using confusion matrices on Improved GA algorithm

4. Feature Selection

Feature selection is important for clustering efficiency and effectiveness because it not only condenses the size of the extracted feature set but also reduces any potential biases embedded in the original (i.e., non-trimmed) feature set . Previous research commonly has employed feature selection metrics such as TF (term frequency), TF×IDF (term frequency × inverse document frequency), and their hybrids.

Unlike the non-LSI-based document clustering approach, which typically involves a feature selection phase, the LSI-based approach to clustering monolingual documents employs LSI to reduce the dimensions and thereby improve both clustering effectiveness and efficiency. Its process generally commences with feature extraction, followed by document representation.

4.1 Confusion Matrix

A confusion matrix contains information about actual and predicted classifications done by a classification system. Performance of such systems is commonly evaluated using the data in the matrix. The following table shows the confusion matrix for a two class classifier.

		Predicted	
		Negative	Positive
Actual	Negative	a	b
	Positive	c	d

The entries in the confusion matrix have the following meaning in the context of our study:

- a is the number of correct predictions that an instance is negative,
- b is the number of incorrect predictions that an instance is positive,
- c is the number of incorrect of predictions that an instance negative, and
- d is the number of correct predictions that an instance is positive.

Several standard terms have been defined for the 2 class matrix:

- The accuracy (AC) is the proportion of the total number of predictions that were correct. It is determined using the equation:

$$AC = \frac{a + d}{a + b + c + d} \quad \text{----------> (1)}$$

The recall or true positive rate (TP) is the proportion of positive cases that were correctly identified, as calculated using the equation:

$$TP = \frac{d}{c + d} \quad \text{-----> (2)}$$

- The false positive rate (FP) is the proportion of negatives cases that were incorrectly classified as positive, as calculated using the equation:

$$FP = \frac{b}{a + b} \quad \text{------> (3)}$$

- The true negative rate (TN) is defined as the proportion of negatives cases that were classified correctly, as calculated using the equation:

$$TN = \frac{a}{a + b} \quad \text{-------> (4)}$$

- The false negative rate (FN) is the proportion of positives cases that were incorrectly classified as negative, as calculated using the equation:

$$FN = \frac{c}{c + d} \quad \text{-------> (5)}$$

- Finally, precision (P) is the proportion of the predicted positive cases that were correct, as calculated using the equation:

$$P = \frac{d}{b + d} \quad \text{-------> (6)}$$

The accuracy determined using equation 1 may not be an adequate performance measure when the number of negative cases is much greater than the number of positive cases. Suppose there are 1000 cases, 995 of which are negative cases and 5 of which are positive cases. If the system classifies them all as negative, the accuracy would be 99.5%, even though the classifier missed all positive cases. Other performance measures account for this by including TP in a product: for example, geometric mean (g-mean), as defined in equations 7 and 8, and F-Measure (Lewis and Gale, 1994), as defined in equation 9.

$$\text{g-mean}_1 = \sqrt{TP * P} \quad \text{--------> (7)}$$

$$\text{g-mean}_2 = \sqrt{TP * P} \quad \text{--------> (8)}$$

$$F = \frac{(\beta 2 + 1) * P * TP}{B2 * P + TP} \quad \text{--------> (9)}$$

In equation 9, b has a value from 0 to infinity and is used to control the weight assigned to TP and P. Any classifier evaluated using equations 7, 8 or 9 will have a measure value of 0, if all positive cases are classified incorrectly.

4.2. Niching Memetic Algorithm

One of the key elements in overcoming less promising locally optimal solutions of a difficult optimization problem with a GA approach is to preserve the population diversity during the search. In this section, we introduce a modification of the niching method and integrate it into our GA to preserve the population diversity during the simultaneous search for clustering and feature selection.

The niching method presented was designed for clustering where no feature selection is required and the number of clusters is known beforehand. In this method, a niching selection with a restricted competition replacement was developed to encourage mating among similar solutions while allowing for some competitions among dissimilar solutions. The flow of the algorithm is given as follows:

Step 1: Initialize the population_size p
Step 2: For each p in initial population, p = local search (p)
Step 3: Calculate unified criterion for each of the offspring. If the fitness of the offspring is better than its paired solution, then the latter is replaced.
Step 4: Provide the feature subset and cluster centers of the solution from the terminal population with the best fitness.

4.3. GA Algorithm

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution. This heuristic is routinely used to generate useful solutions to optimization and search problems. Genetic algorithms belong to the larger class of evolutionary algorithms (EA), which generate solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection, and crossover. The flow of the algorithm is given as follows:

Input: document set DS, number of generations n
Output: best classifier over DS

Step 1: Evaluate the sets of candidate positive and negative terms
Step 2: Create the population oldPop and initialize each chromosome
Step 3: Evaluate the fitness of each chromosome in oldPop
Step 4: Copy in NewPop the best r chromosomes of oldPop

Step 5: While size(newPop) < size(oldPop)
- select parent1 and parent2 in oldPop
- generate kid1, kid2 through crossover(parent1, parent2)
- apply mutation, i.e., kid1 = mut(kid1) and kid2 = mut(kid2)
- apply the repair operator ρ to both kid1 and kid2
- add kid1 and kid2 to newPop

step 6: oldPop = newPop
- Select the best chromosome K in oldPop;
- Eliminate redundancies from K;

step 7: classifier associated with K.

5. Performance Evalution

The performance of improved GA on Documents is evaluated in this section. Let us suppose that we have obtained a clustering solution with feature selection. Since the quality of clusters depends on the particular application, there is no standard criterion for evaluating

clustering solutions. We compute classification errors, since we know the "true" clusters of the synthetic data and the class labels of the real data. This is done by first running the algorithm to be tested on each data set. Next, each cluster of the clustering results is assigned to a class based on examining the class labels of the data objects in that cluster and choosing the majority class. After that, the classification errors are computed by counting the number of misclassified data objects. For the identification of correct clusters, initially we report the number of clusters found. We stress that the class labels are not used during the generation of the clustering results, and they are intended only to provide independent verification of the clusters.

The feature recall and precision are reported on synthetic data, since the relevant features are known a priori. Recall and precision are concepts from text retrieval. Feature recall is the number of relevant features in the selected subset divided by the total number of relevant features. Feature precision is the number of relevant features in the selected subset divided by the total number of features selected. These indices give us an indication of the quality of the features selected. High values of feature recall and precision are desired. Note that, with respect to the real data, we report only the number of feature selected, since the relevant features are unknown.

6. Experimental Result and Discussion

The proposed method was tested with a file of 100 historical documents. The datasets were taken as related topic of Data mining, Image processing and Networking. For each dataset, 30% of the documents are randomly selected as test documents, and the rest are used to create training sets as follows: γ percent of the documents from the positive class is first selected as the positive set P. The rest of the positive documents and negative documents are used as unlabeled set U. We range γ percent from 10%- 50% to create a wide range of scenarios.

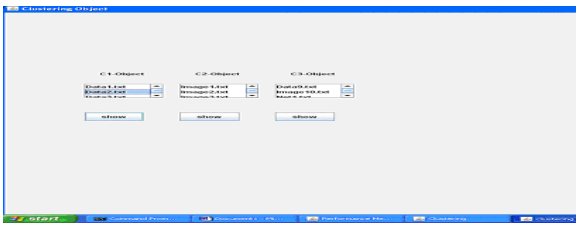

Fig 2 Clustered DataSet

Preliminarily, documents were subjected to the following pre-processing steps: (1) First, we removed all words occurring in a list of common stopwords, as well as punctuation marks and numbers; (2) then, we extracted all n-grams, defined as sequences of maximum three words consecutively occurring within a document (after stopword removal)5; (3) at this point we have randomly split the set of seen data into a training set (70%), on which to run the GA, and a validation set (30%), on which tuning the model parameters. We performed the split in such a way that each category was proportionally represented in both sets (stratified holdout). Based on the term frequency and inverse document frequency, the term weight will be calculated.

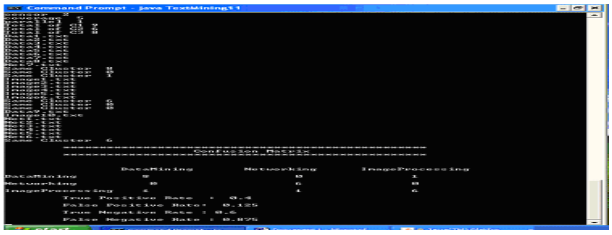

Fig 3: Confusion matrix on text documents

The performance results are measured in terms of F-measure, purity and false positive rate according to the Number of documents and Cluster object.

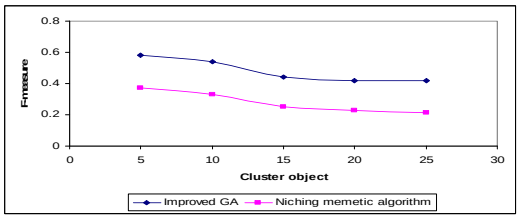

Fig 4: Cluster object vs F-measure

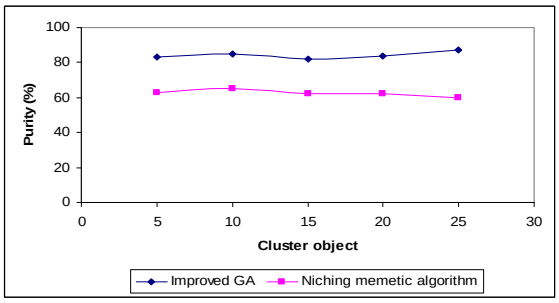

Fig 5: Cluster vs purity

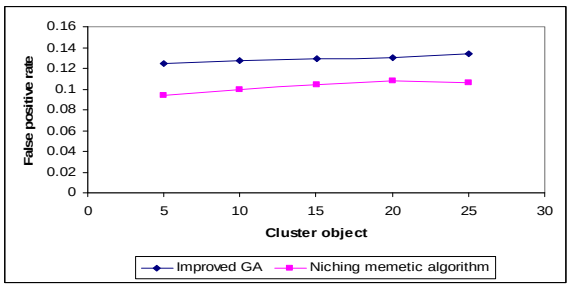

Fig 6: Cluster Object vs False positive rate

Figure (4), (5) and (6) shows the result of F-measure, purity and false positive rate with respect to the cluster object. The proposed algorithm improved GA gives the better result compared with existing method, Niching memetic algorithm.

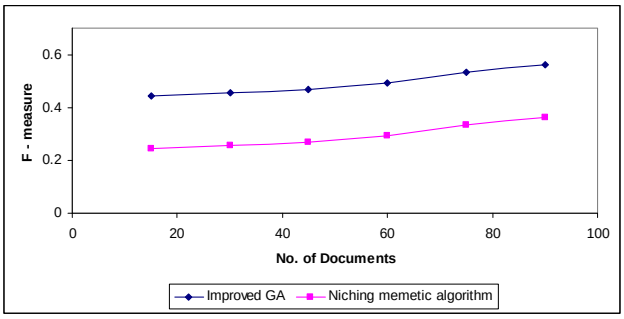

Fig 7: No. of Documents vs F-measure

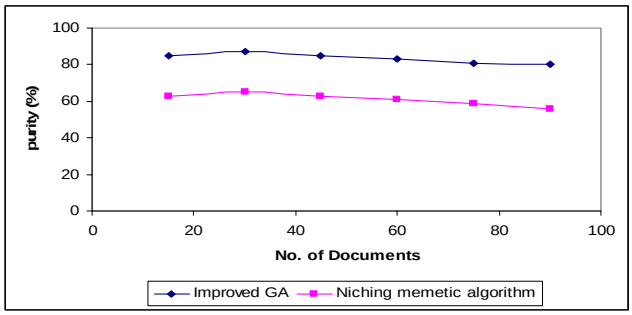

Fig 8: No. of Documents vs Purity

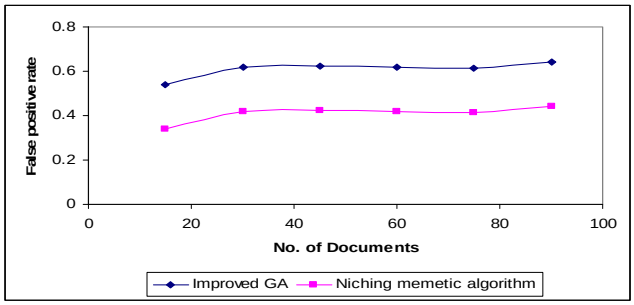

Fig 9: No. of documents vs false positive rate

Figure (7), (8) and (9) depicts the performance result of F-measure, purity and false positive rate according to the Number of documents. It is observed that improved GA performs the well. By comparing niching memetic algorithm with improved GA, proposed improved GA can efficiently recover solutions with low classification errors

6. CONCLUSION

The improved Niche memetic algorithm and improved genetic algorithm have been designed and implemented by using confusion matrices. Our proposed method is applied to real data sets with an abundance of irrelevant or redundant features. Improved GA relies on confusion matrices and uses the F-measure as the fitness function. In this case, identifying a relevant subset that adequately captures the underlying structure in the data can be particularly useful. Additionally, as a general optimization framework, the proposed algorithm can be applied for text mining. In such a case, an unbiased clustering criterion in some sense is produced by computing the mutual information between clusters, thus enabling a better verification of the properties of the proposed optimization scheme. We conclude by remarking that we consider the experimental results can further be improved through a fine-tuning of the GA parameters.

References

[1] A.K. Santra, C. Josephine Christy and B.Nagarajan, "Cluster Based Hybrid Niche Memetic and Genetic Algorithm for Text Document Categorization", IJCSI, vol.8, Issue 5, no. 2,pp. 450-456, Sep 2011.

[2] S. Areibi and Z. Yang, "Effective Memetic Algorithms for VLSI Design Automation = Genetic Algorithms + Local Search + MultiLevel Clustering," Evolutionary Computation, vol. 12, no. 3, pp. 327- 353, 2004.

[3] S. Wu, A.W.C. Liew, H. Yan, and M. Yang, "Cluster Analysis of Gene Expression Database on Self-Splitting and Merging Competitive Learning," IEEE Trans. Information Technology in Biomedicine, vol. 8, no. 1, 2004.

[4] H.K. Tsai, J.M. Yang, Y.F. Tsai, and C.Y. Kao, "An Evolutionary Approach for Gene Expression Patterns," IEEE Trans. Information Technology in Biomedicine, vol. 8, no. 2, pp. 69-78, 2004.

[5] Y. Zhao and G. Karypis, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering," Machine Learning, vol. 55, no. 3, pp. 311-331, 2004.

[6] J. Kogan, C. Nicholas, and V. Volkovich, "Text Mining with Information-Theoretic Clustering," IEEE Computational Science and Eng., pp. 52-59, 2003.

[7] W. Sheng, A. Tucker, and X. Liu, "Clustering with Niching Genetic K-Means Algorithm," Proc. Genetic and Evolutionary Computation Conf. (GECCO '04), pp. 162-173, 2004.

[8] K. Deep and K. N. Das. Quadratic approximation based Hybrid Genetic Algorithm for Function Optimization. AMC, Elsevier, Vol. 203: 86-98, 2008.

[9] C. Wei, C.S. Yang, H.W. Hsiao, T.H. Cheng, Combining preference- and content-based approaches for improving document clustering effectiveness, Information Processing & Management 42 (2) (2006) 350–372.

[10] Renchu Guan, Xiaohu Shi, Maurizio Marchese, Chen Yang, and Yanchun Liang, "Text Clustering with Seeds Affinity Propagation" IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 4, APRIL 2011

[11] Y.J. Li, C. Luo, and S.M. Chung, "Text Clustering with Feature Selection by Using Statistical Data," IEEE Trans. Knowledge and Data Eng., vol. 20, no. 5, pp. 641-652, May 2008.

[12] B.J. Frey and D. Dueck, "Non-Metric Affinity Propagation for Un- Supervised Image Categorization," Proc. 11th IEEE Int'l Conf. Computer Vision (ICCV '07), pp. 1-8, Oct. 2007.

[13] L.P. Jing, M.K. Ng, and J.Z. Huang, "An Entropy Weighting KMeans Algorithm for Subspace Clustering of High-Dimensional Sparse Data," IEEE Trans. Knowledge and Data Eng., vol. 19, no. 8, pp. 1026-1041, Aug. 2007.

[14] Z.H. Zhou and M. Li, "Distributional Features for Text Categorization," IEEE Trans. Knowledge and Data Eng., vol. 21, no. 3, pp. 428-442, Mar. 2009.

[15] F. Pan, X. Zhang, and W. Wang, "Crd: Fast Co-Clustering on Large Data Sets Utilizing Sampling-Based Matrix Decomposition," Proc. ACM SIGMOD, 2008.

[16] Jung-Yi Jiang, Ren-Jia Liou, and Shie-Jue Lee, "A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification", IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 3, MARCH 2011

A. K. Santra received the P. G. degree and Doctorate degree from I.I.T., Kharagpur in the year 1975 and 1981 respectively. He has got 20 years of Teaching Experience and 19 years of Industrial (Research) Experience. His area of interest includes Artificial Intelligence, Neural Networks, Process Modeling, Optimization and Control. He has got to his credit (i) 35 Technical Research Papers which are published in National / International Journals and Seminars of repute, (ii) 20 Research Projects have been completed in varied application areas, (iii) 2 Copy Rights for Software Development have been obtained in the area of Artificial Neural Networks (ANN) and (iv) he is the contributor of the book entitled **"Mathematics and its Applications in Industry and Business"**, Narosa Publishing House, **New Delhi.** He is the recognized Supervisor for guiding Ph. D. / M. S. (By Research) Scholars of Anna University-Chennai, Anna University-Coimbatore, Bharathiyar University, Coimbatore and Mother Teresa University, Kodaikanal. Currently he is guiding 12 Ph. D. Research Scholars in the Department. He is a Life member of CSI and a Life member of ISTE.

C. Josephine Christy received her M.Sc., M.Phil., M.B.A., from Bharathiar University, Coimbatore. Currently she is working as Asst.Professor in Bannari Amman Institute of Technology, Sathyamangalam. Her area of interest includes Text Mining, Web Mining. She presented a paper in International Journal, 2 papers international conferences and 6 papers in national Conferences. She is a Life member of Computer Society of India and a Life member of Indian Society for Technical Education.