Fuzziness in Text Classification Using Different Similarity Metrics

M.A. Wajeed[1] and T. Adilakshmi[2]

[1] SCSI, Sreenidhi Institute of Science & Technology,
Ghatkesar, Hyderabad, AP, India
wajeed.mtech@gmail.com
[2] Dept., of CSE, Vasavi College of Engineering,
Ibrahimbagh, Hyderabad, AP, India
t_adilakshmi@gmail.com

Abstract. We are living in the information era where vast amount of data is generated at the end of the day, which can also be in textual form. To cater the further needs and to make decisions effective we need to classify the generated data and store it in the classified repository, so that later it can efficiently be retrieved with minimum effort. The paper attempts to mix the concepts of supervised learning and unsupervised learning techniques, by forming clusters which could act as features so that feature reduction can be made possible. Clusters are formed based on the word patterns, soft, hard and mixed clustering is also considered in the processes of text classification. We employee different similarity measures like Euclidean, square Euclidean, Manhattan, chebyshev, bray-Curtis etc., in the processing of finding the category of the document. The results obtained were encouraging.

Keywords: Text classification, data Clusters, soft-hard-mixed clusters, eucledean, chebyshev, manhattan,bray-curtis, canberra similarity measures.

1 Introduction

To lead a comfortable life man today has stepped in to the era of information technology, where on daily basis huge amount of data is generated which needs to be stored in a proper fashion so as to make it useful in future. The generated data can also be in textual format, so the need of classification of documents based on its contents become inevitable. In this process we have supervised classification and unsupervised classification. In case of supervised classification based on the training data which has a set of independent attribute, with its corresponding decision attribute, a function which can learn how the decision attribute is dependent on the independent attributes is made to learn. But in case of unsupervised learning as we don't have the training data, the given data is split into clusters such that the inter-similarity among the clusters is maximized and intra-similarity among the clusters is reduced. The paper explores the construction of the clusters in case of supervised learning to reduce the no: of features involved in the learning process of the classification function, as in case of text classification large no of features are involved. Different similarity measures are explored in the process of learning.

N. Meghanathan et al. (Eds.): CCSIT 2012, Part III, LNICST 86, pp. 249–260, 2012.

The paper is organized in the following manner, in section 2 basic concepts are furnished in section 3 proposed method is explored. In section 4 diferent similarity measures are discussed and in section 5 implementation details and results are furnished and finally in section 6 conclusion and future enhancements are given.

2 Related Work

Without feature selection text categorization (TC) is almost not possible as in TC naturally large no: of features exists. Need for feature reduction becomes inevitable step. Feature reduction generally can be done either by feature selection or feature reduction. In Feature selection we attempt to find a subset of the original features that can be obtained by the process of either filtering technique or by using wrapper technique. The other possibility to reduce features is to transform the relevant features from a high-dimension space to a space of fewer dimensions such a transformation may be either linear as in case of principal component analysis (PCA), or can be through other nonlinear technique.

In supervised classification Information Gain, Gain Ratio, Odds Ratio, Gini-index, Chi-Square etc techniques are very popular as feature selection measures. On the other hand document frequency, term frequency, and inverse-document frequency are considered as feature selection in unsupervised learning. A brief discussion of few supervised feature selection techniques are given below.

Given a set of categories C_m, where m is the no: of classes the information gain of term t is given by

$$\begin{aligned} IG(t) = -\sum_{i=1}^{m} P(c_i)\log P(c_i) + P(t)\sum_{i=1}^{m} P(c_i \mid t) \\ \log P(c_i \mid t) + P(\bar{t})\sum_{i=1}^{m} P(c_i \mid \bar{t})\log P(c_i \mid \bar{t}) \end{aligned} \tag{1}$$

Feature reduction can be made possible using the values obtained by the equation (1). The features whose IG is less compared to the other features are discarded.

Chi-Square another supervised feature selection technique for feature reduction which is given by

$$\chi^2 = N.\frac{P(t_k, c_i)P(\bar{t}_k, \bar{c}_i) - P(t_k, \bar{c}_i)P(\bar{t}_k, c_i)}{P(t_k)P(\bar{t}_k) - P(c_i)P(\bar{c}_i)} \tag{2}$$

Odds Ratio yet another supervised feature selection method which is given by

$$\text{OR}= \frac{P(t_k \mid c_i).P(t_k \mid \bar{c}_k)}{(1 - P(t_k \mid c_i)).P(t_k \mid \bar{c}_k)} \tag{3}$$

Based on the values obtained in the above equations, which determines the relationship between the terms and the class label, feature selection can be made.

Definitions: - We are provided with the training document set T_s whose class label is given along with the documents. We are also given a set of document set T_t termed as test data set whose class label is to be determined by the classifier.

In the process of obtaining the class label of the test document lexicon set **£,** which is a set which contains all words, that have appeared in the training document set T_s is built.

We obtain the word-patterns for each word which appears in the documents, which is the conditional probability of the class given the terms appearance in the document. We find the probability of the terms appearance for all the classes and for all the terms, we denote such a set as $W_{p.}$ Once we obtain the word patterns then we find the self constructive clusters based on the word patterns using the Gaussian functions.

3 Fuzzy Classifier

[1] gives the detailed steps of pre-processing the training set. Preprocessing is needed so as to remove the noise in the training data. We obtain the lexicon set which is a set of all words which appeared in the training documents. [6] gives the procedure for generating word pattern, we obtain the clusters once word-pattern for all the words in the lexicon set. In the process of generating the cluster we proceed with the given '*n*' no: of documents in the training data, that are spread across '*m*' no: of classes. Using the lexicon we generate the word patterns for each member of the lexicon set $X_i =< x_{i1}, x_{i2}, x_{i3}, ...x_{in} >$ the elements of the set are defined as

$$P(c_m \mid t_i) = \frac{\sum_{r=1}^{n} d_{ri} * \varepsilon_{rm}}{\sum_{r=1}^{n} d_{ri}} \tag{4}$$

where d_{ri} is the no: of times the term t_i occurs in the document d_r, $\boldsymbol{\varepsilon_{rm}}$ is 1 if the document d_r belongs to the class c_m, otherwise it is 0.

Clusters have the property that the inter-similarity among clusters is minimized and intra-similarity is maximized. In order to achieve optimal clusters, they are characterized by the product of m – one dimensional Gaussian function. Let ζ be a cluster containing q word patterns $x_1, x_2, ... x_q$. Let $x_j = < x_{j1}, x_{j2}, ... x_{jm} >$ $i \le j \le q$ the mean $\bar{x} = \; < \bar{x}_1, \bar{x}_2, ... \bar{x}_m >$, the mean is defined as

$$\overline{x_i} = 1/|\zeta| \sum_{i=1}^{n} x_i \tag{5}$$

where $|\zeta|$ gives the no: of elements in the i^{th} clusters. The deviation $\sigma = < \sigma_1, \sigma_2, ... \sigma_m >$ of ζ are given by

$$\sigma_i = \sqrt{\frac{1}{|\zeta|} \sum_{i=1}^{n} (x_{ji} - \bar{x}_i) * (x_{ji} - \bar{x}_i)} \tag{6}$$

The fuzzy similarity of a word pattern to cluster ζ is defined by gaussian membership function

$$\zeta_j(x) = \prod_{i=1}^{m} \exp\left[\frac{-(x_i - \bar{x}_i)^2}{\sigma_i}\right] \tag{7}$$

The values of the $\zeta_j(x)$ are bounded in the interval [0, 1], where $1 \leq j \leq k$. A word pattern close to the mean of a cluster is regarded to be very similar to the cluster i.e. $\zeta(x) \approx 1$, on the other hand a word pattern far distant from a cluster is hardly similar to the cluster so $\zeta(x) \approx 0$. On the basis of $\zeta(x)$ and on the threshold value ς, which is provided by the user one can control the formation of no: of clusters. If we wish to have many clusters then smaller value of the threshold ς is considered otherwise larger value of the threshold is taken.

If $\zeta_j(x_i) \geq \varsigma$, then the word pattern x_i can be added to the cluster ζ_i, and the no: of elements in the cluster ζ_i is increased by 1, corresponding values of the cluster also are updated, i,e the mean $\overline{x_i}$ of the cluster and deviation of the cluster σ_i. In case if $\zeta_j(x_i) < \varsigma$, then a new cluster is created with its mean as $\overline{x} = x_i$, the deviation of the newly formed cluster as $\sigma = 1$ and the no: of clusters so far formed is also incremented. Once all the word patterns are constructed we have 'k' no: of clusters with updated mean values of each of the cluster in the form of the vector $\overline{x} =< \overline{x}_1, \overline{x}_2$, ..., $\overline{x}_k$> and updated deviation values of the cluster in the form of the vector $\sigma = < \sigma_1, \sigma_2, ... \sigma_k >$.

We deviate from [6] in the process of text classification based on fuzzy measures. Once we obtain all the updated values of cluster deviation, mean and no: of clusters, we proceed in the following manner.

As explained in section 3 word pattern for all the words which are members of the lexicon are obtained, the no: of clusters, each of the cluster size is also obtained. From the training documents words frequency is obtained as described in [7]. We then find for each word, the membership of it in the clusters. We create soft, hard and mixed mapping based on the membership of the word in the clusters. We create 14,822 vectors with 'k' no: of elements in each, where 'k' is the no: of clusters formed. In case of hard classification based on the equation 9 a word is allowed in a single cluster. But in case of soft classification single word can be considered in more than a single cluster, based on the equation 7. We generate the mixed classification based on the equation 10.

The same steps are repeated for the test documents too. In the case of test data we have 2189, so we get 2189 vectors with k elements in each. Now for each of the test document we find the euclidean distance similarity measure, which is given by the sum of square of the difference of the individual elements [8] of test and train data.

4 Similarity Measures

The sum of the squares of the difference of the individual elements gives the Euclidean similarity [7]. The same can be expresses mathematically as

$$Dist(X,Y) = \sqrt{(x_2 - x_1)*(x_2 - x_1) + (y_2 - y_1)*(y_2 - y_1) + ...} \tag{8}$$

Figure-1 shows the results obtained for different k values varying from 1 to 10, and figure-2 gives the results for different values of k varying through 10 to 100.

- Squared Euclidean Distance is similar to the Euclidean distance, but does not have the square-root over the summation. Figure-3 shows the results obtained for k values varying from 1 to 10, figure-4 gives the results for different values of k varying through 10 to 100. Mathematically square Euclidean distance is expressed as

$$Dist(X,Y)=(x_2-x_1)*(x_2-x_1)+(y_2-y_1)*(y_2-y_1)+... \tag{9}$$

- Manhattan Distance is a simple similarity measure when compared to the Euclidean and square-Euclidean distance measure; it takes the summation of the absolute difference among the individual elements of the vector. Figure-5 shows the results obtained for k value varying from 1 to 10, and figure-6 gives the results for different values of k varying through 10 to 100. Mathematical expression of Manhattan distance is expressed as

$$Dist(X,Y) =| x_2 - x_1 | + | y_2 - y_1 | +... \tag{10}$$

- Chessboard distance is also called as Chebyshev distance, Tchebychev distance), Maximum metric, it is a metric defined on a vector space where the distance between two vectors the greatest of their differences along any coordinate dimension is. It is named after Pafnuty. Figure-7 shows the results obtained for k value varying from 1 to 10, and figure-8 gives the results for different values of k varying through 10 to 100. Mathematically the same can be expressed as

$$Dist(X,Y) = Max(| x_1 - y_1 |,| x_2 - y_2 |,...) \tag{11}$$

- Bray Curtis Distance is also called as Sorenson. It is defined as the fraction of absolute difference in the individual elements of the vector to the sum of the individual elements of the two vectors. The same can be expressed mathematically as

$$Dist(X,Y) = \frac{(| x_1 - y_1 | + | x_2 - y_2 |,...)}{(| x_1 + y_1 | + | x_2 + y_2 |,....)} \tag{12}$$

- The ratio of the sum of the absolute difference in the individual elements to the sum of the absolute values of the individual elements in the two vectors gives the canberra similarity measure. Figure-9 shows the results obtained for k value varying from 1 to 10, and figure-10 gives the results for different values of k varying through 10 to 100. This is mathematically expressed as

$$Dist(X,Y) = \sum_{k=1}^{n} \frac{| x_{ik} - x_{jk} |}{| x_{ik} | + | x_{kj} |} \tag{13}$$

5 Implementation

[5] has the text categorization corpus, which has 5485 documents across 8 classes. [1] gives the details of the lexicon construction phase; we had 14833 words in the lexicon. We assume that the corpus has no noise, though we found noise in the corpus, but no effort was put to remove the noise. We had vector, with 8 columns representing 8 different classes and 14833 rows for the lexicon entries; the elements in the vector are the word patterns. Using these word patterns, we build the clusters, and the goodness of the clustering algorithm is that we don't need to provide to the algorithm the input stating, the no: of clusters to be build, but no: of clusters are defined on the basis of input value of the similarity threshold ς. By varying the threshold value the no: of clusters obtained can be controlled, for more clusters the threshold value has to be smaller, and for less no: of clusters the value of the threshold has to be larger.

Table 1. showing different threshold values and clusters obtained

Threshold value (ς)	**Clusters**
0.5	14
0.6	12
0.7	10

We tried with 3 different values of the threshold, table 1 gives the details of the threshold value and the clusters obtained. Once the no: of clusters are formed, we implement the KNN algorithm for the text categorization.

All documents belonging to the training data are processed first. We perform stemming on the training data, and the words obtained after stemming with respect to word patterns are grouped to form clusters. The no: of elements for each of the clusters for each of the document was obtained, thus the training documents were mapped to the numeric values. As the no: of clusters obtained for different values of the threshold is different so the experiment was repeated 3 times and we obtained different clusters.

We take 3 types of clusters into consideration, a word is considered to belong to a single cluster only we refer such a clusters as hard cluster, using the given below equation we obtain the membership of the hard clusters.

$$t_{ij} = \left\{ \begin{matrix} 1 \text{ if j= arg } \max\limits_{1<=\alpha<=k} (\zeta(x_i)) \\ 0 \text{ otherwise} \end{matrix} \right\} \quad .(14)$$

But in case of soft-weighting approach we allow the word pattern to belong to more than a single cluster so rather than considering the maximum value of the function ($\mu G_\alpha(x_i)$) we take its direct value for all the clusters. In case of the soft, hard mixed-weighting cluster we employ the below equation where γ is the constant which dictates domination of the cluster type.

$$\mathrm{t_{ij}} = (\gamma) * t_{ij}{}^{H} + (1-\gamma) * t_{ij}{}^{S} \tag{15}$$

where $\mathrm{t_{ij}}^{\mathrm{H}}$ is hard-weighting clustering approach and $\mathrm{t_{ij}}^{\mathrm{S}}$ is the soft-weighting clustering membership function. The value of γ can be between 0 and 1. If it is very near to 0 then the hard and mixed weighting equation coincides with the soft-clustering approach and if its value is 1 then the hard and mixed weighting coincides with hard-clustering approach. Taking γ value as 0.1 we have performed the experiment.

In figures 1 through 10 soft1, hard1 and mixed1 refers to the threshold(ς) value as 0.5 which formed 14 clusters, soft2, hard2 and mixed2 refers to the threshold(ς) value as 0.6 which formed 12 clusters, and soft3, hard3 and mixed3 refers to the threshold(ς) value as 0.7 which formed 10 clusters. For 5485 documents in the training data we have 5485 vectors with the no: of elements in the vector equals to the no: of clusters. We have 2189 vectors for the same no: of elements in the vector as equal to the no: of clusters as part of the test data.

The procedure for test data is repeated as that of training, so all the test documents are too converted into vectors. Once we obtain the vectors of the training and test data we apply the KNN algorithm for the vectors. We use the Euclidean measure and obtain the similarity between the training data and the test data. For different values of K we obtain the confusion matrix, in table 2 we give the confusion matrix for k=1 for Euclidean similarity measure.

K-Nearest Neighbour algorithm is also called as instance based learning algorithm. It is a classifier which is based on learning by analogy. It computes the similarity between a given test tuple with training tuples that are similar to it [10]. The training tuples are described by ‘n’ attributes, in our case attributes are words which are in the lexicon set. Each tuple represents a point in an n-dimension pattern space. When given an unseen tuple, a k-nearest neighbor classifier searches the pattern space for different values of k (which can take any value 1 through some arbitrary number) the training tuples that are closest to the unseen tuple. Depending on the value of k, k training tuples are used which are near to the unseen tuple. Different similarity measures would be applied between two points or tuples say X1 and X2 which have ‘n’ component elements. It is used to find the similarity (closeness) between the tuples.

For different k tuples, the majority class label is taken, and the unseen tuple class label is declared to be the same as the majority class labels. In case of a tie, arbitrary the tie is resolved. In other words, the test data consists of 2189 documents, the distance between the training and a particular test documents is measured, the class with the nearest training data is taken as the class of the test data, as here K value in K-NN is 1. In case of k value 2 we take two smallest distances, and if both belong to same class than the test tuple also belongs to the same class as it is the nearest distance of the training data class, in case of tie an arbitrary consensus is used to resolve the conflict. Based on the similarity between the training and test tuples we obtain confusion matrix which is a good tool for analyzing, how well the classifier can classify the tuples of different classes. A confusion matrix can be treated as a tool which could be helpful in determining the performance of a classifier in supervised learning. It is a matrix plot of the classifier predicted versus the actual classes of the tuples.

For a dataset that has m different classes, a confusion matrix is a table which has m rows with m columns in each. An entry *CMi,j* in the first m rows and m columns indicates the number of tuples of class *i* that are labeled by the classifier as class *j*. For a classifier to have good accuracy, i,e to be an ideal classifier tuples along the diagonal of the confusion matrix must have non-zero values and rest of the elements, very close to zero. Table 1 gives the confusion matrix obtained, in the experiment for different values of k. The table has an additional row and columns to provide totals and recognition rates per class

In figure 1 we draw a graph showing the accuracy of the classifier, on X-axis we take different value of K, in K-NN algorithm, and on Y-axis we take the accuracy of the classifier, for the 3 types of clusters soft, hard and mixed results for ς = 0.5 are shown. Similarly in figure 2 classifier accuracy for k values varying from 10 to 100 are provided for ς =0.5

In figure 3 for ς=0.6 for k values 1 to 10 are provided and in figure 4 for ς=0.6 for k values 10 to 100 are provided. In figure 5 for ς=0.7 for k values 1 to 10 are provided and in figure 6 for ς=0.7 for k values 10 to 100 are provided.

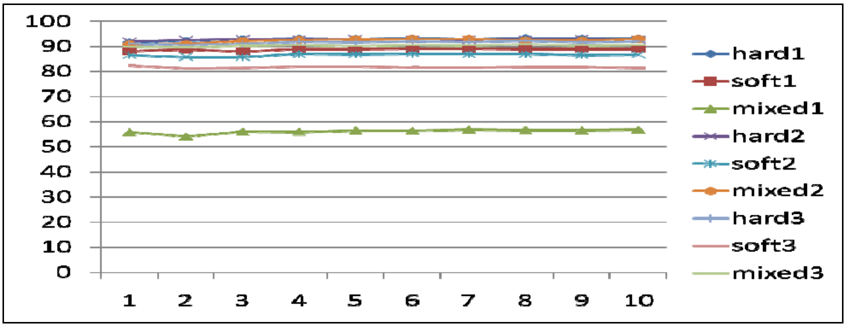

Fig. 1. Graph showing accuracy for Euclidean k values 1 to 10

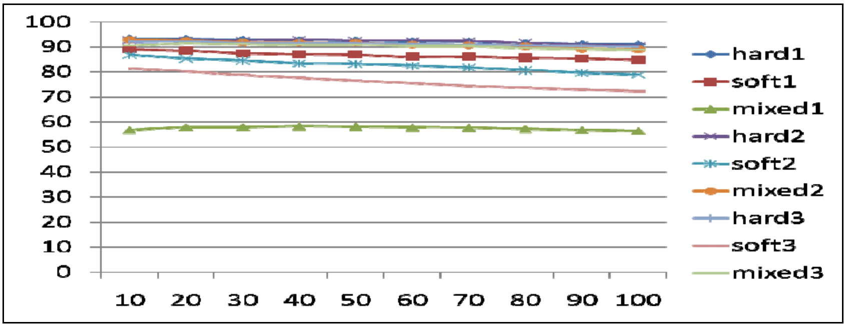

Fig. 2. Graph showing accuracy for Euclidean k values 10 to 100

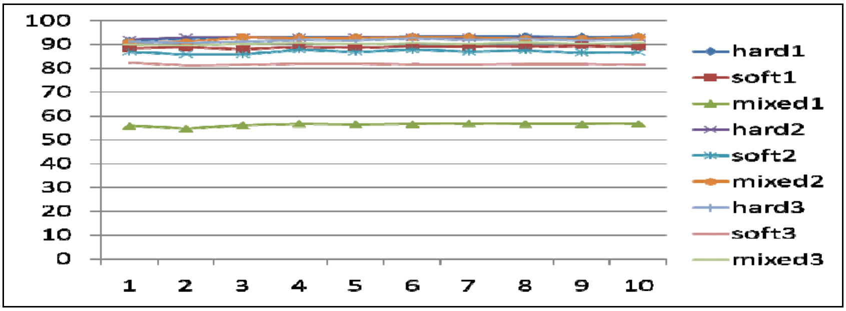

Fig. 3. Graph showing accuracy for Squared Euclidean k values 1 to 10

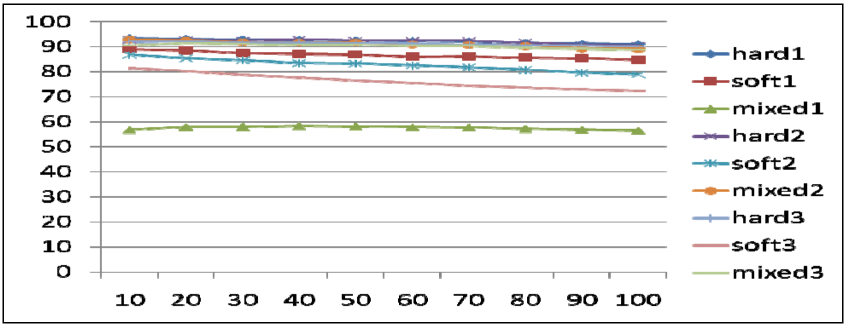

Fig. 4. Graph showing accuracy for Squared Euclidean k values 10 to 100

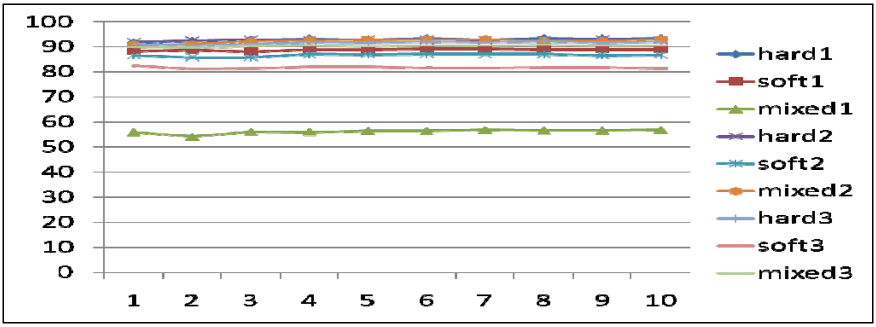

Fig. 5. Graph showing accuracy for Manhattan k values 1 to 10

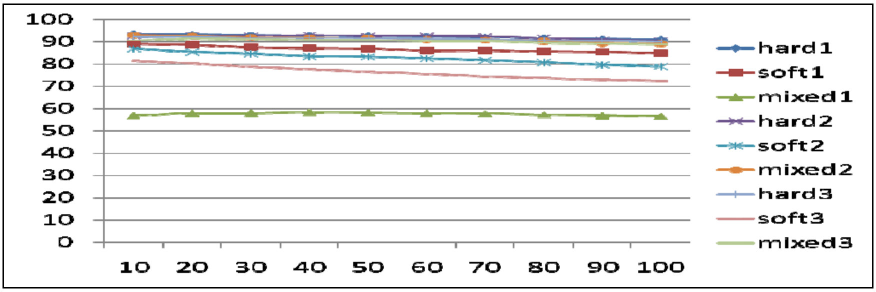

Fig. 6. Graph showing accuracy for Manhattan k values 10 to 100

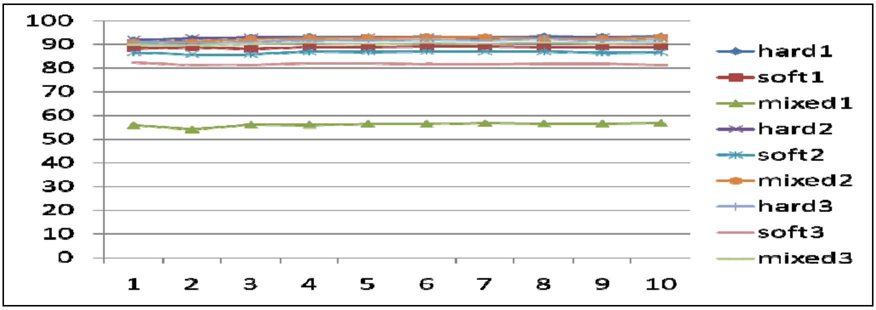

Fig. 7. Graph showing accuracy for Chebyshev k values 1 to 10

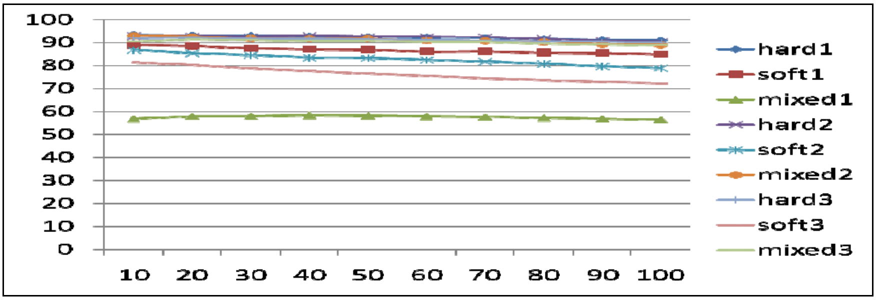

Fig. 8. Graph showing accuracy for chebyshev k values 10 to 100

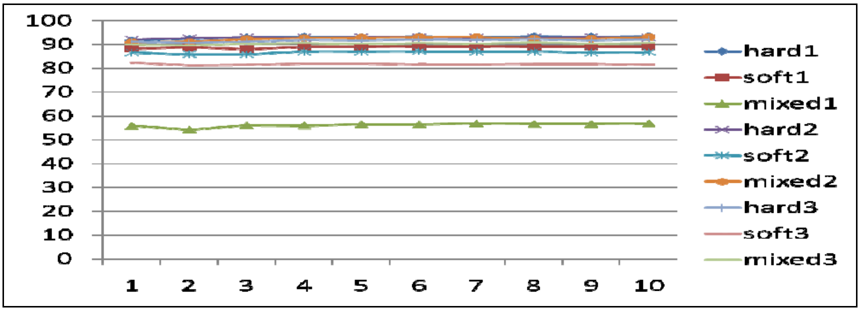

Fig. 9. Graph showing accuracy for Canberra k values 1 to 10

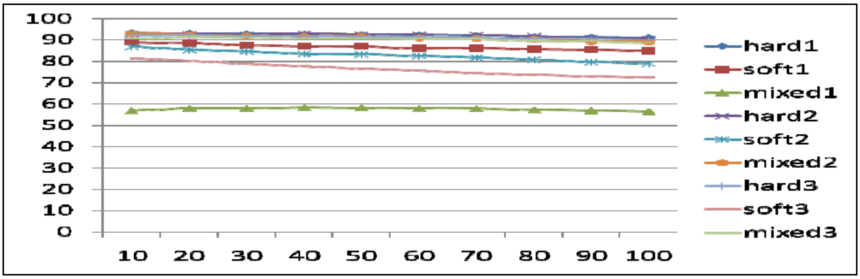

Fig. 10. Graph showing accuracy for Canberra k values 10 to 100

6 Conclusions

Presently bulk data is available which needs to be analyzed, utilizing the data generated decisions making can become effective. The data generated can also be textual form. The paper attempts in achieving learning by building a classifier which has fuzzy capabilities. In the process 3 types of clusters were explored based on the words in the documents occurrence, soft cluster where a word can belong to more than a single cluster at the same time, hard cluster where a word can belong to a single cluster at a time, mixed cluster which is devised using both the soft and hard clusters.

For k values varying from 1 to 10 we find that soft1 is better than soft2, soft2 is better than soft2. As k value increases accuracy also increases. Again decreases in soft2 but in soft3 we get mixed results. Hard is better than soft cluster. Hard also increases with k values, both hard1 and hard2 are the same we find mixed results in hard3. Mixed is the least in case of mixed1 and good in mixed2, mixed3 again decreases.

For k value 10 to 100 soft1, soft2 and soft3 decreases as k value increases. Mixed cluster approach also gives most decreasing accuracy result. Mixed2 is better when compared to mixed1 and mixed3. Hard1 is better than hard2, hard3 and decreases as k value increases. We also find that almost all the similarity measures gave same results giving the choice for the user to choose the similarity measure of his choice.

In future we wish to decrease the size of the lexicon and see how best the classifier can learn from the training data to classify the textual data

References

[1] Wajeed, M.A., Adilakshmi, T.: Text Classification Using Machine Learning. Journal of Theoretical and Applied Information Technology 7(2), 119–123 (2009)

[2] Yen, J., Langari, R.: Fuzzy Logic-Intelligence, Control, and Information. Prentice-Hall (1999)

[3] Wang, J.S., Lee, C.S.G.: Self-Adaptive Neurofuzzy Inference Systems for Classification Applications. IEEE Trans. Fuzzy Systems 10(6), 790–802 (2002)

[4] Correa, R.F., Ludermir, T.B.: Automatic Text Categorization: Case Study. In: Proceedings of the VII Brazilian Symposium on Neural Networks, Pernambuc, Brazil (November 2002)

[5] http://www.daviddlewis.com/resources/testcollections/reuters21578/

[6] Jiang, J.-Y., Liou, R.-J., Lee, S.-J.: Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification. IEEE Transaction on Knowledge & Data Engineering 23(3) (March 2011)

[7] Wajeed, M.A., Adilakshmi, T.: Different Similarity Measures for Text Classification Using KNN. In: To Be Presented in International Conference on Computer Communication Technology at NIT Allahabad (2011)

[8] Sebastiani, F.: Text classification, automatic. In: Brown, K. (ed.) The Encyclopedia of Language and Linguistics, 2nd edn., vol. 14, Elsevier Science, Amsterdam (2004)

[9] http://tartarus.org/~martin/PorterStemmer

[10] Yeung, C.-M.A., Gibbins, N., Shadbolt, N.: A k-Nearest-Neighbour Method for Classifying Web Search Results with Data in Folksonomies. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, WI-IAT, vol. 1, pp. 70–76 (2008)

[11] Capdevila Dalmau, M., Márquez Flórez, O.W.: Experimental Results of the Signal Processing Approach to Distributional Clustering of Terms on Shape Reuters-21578 Collection. In: Amati, G., Carpineto, C., Romano, G. (eds.) ECiR 2007. LNCS, vol. 4425, pp. 678–681. Springer, Heidelberg (2007)