International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 17 (2015) pp 38628-38637

Improvement of K-Mean Clustering Algorithm Using Support Vector Machine on Reusable Software Components

Amarjeet Kaur
PG Scholar, Department of Computer Science and Engineering, Chandigarh University, acronyms acceptable Chandigarh University, India kauramarjeet392@gmail.com

Iqbaldeep Kaur,
Associate Professor, Department of Computer Science and Engineering Chandigarh University, acronyms acceptable Chandigarh University, India iqbaldeepkaur.cu@gmail.com

Abstract
Data mining is important and challenging task in the field of Machine learning. Data mining refers to the process of deriving high quality, extracting relevant information from huge set of data. There are various functionalities of data mining, clustering is one of them. Clustering is an effort to group similar data into single cluster. Clustering to reduce the search space, reuse test cases of grouping similar entities according to requirements ensuring reduced time complexity as it reduces the search time for retrieval the test cases. So in this paper k-mean algorithm is used to reduce the search time for information retrieval. The feature extraction and classification of such text documents needs an efficient machine learning algorithm which performs automatic text classification. In the proposed work, to automate the process using support vector machine is applied on the data set which gave the classified data The work also explains the performance of the proposed approach for efficient text classification.

Keywords: Data mining, Clustering, K-Mean, Support vector machine.

Introduction

Data mining is used to extract the useful information from the huge data set. A data-warehouse is a place where data is stored. The type of information or data stored depends on the type of the company and industry. For effective data mining four things are required and these things are an adequate sample size, the "right" data, high-quality data, and the right tool. There are many tools available for data mining. These include decision trees, various types of regression, neural networks and support vector machine. Data mining has various techniques like classification, prediction and clustering. Most of the domains use clustering method for the reuse of software components, text documents and patterns. In software engineering, the need of clustering arises due to software component classification, component clustering, and performing software component search and for the software component retrieval from the software repository.

Clustering is the task to examine the structure and the groups of the data, which have similarity. In the cluster, data items are mainly grouped according to the relationship between them. There are various clustering techniques are available like, hierarchical methods, portioning methods, density-based methods, model based clustering method, grid based etc. clustering may be supervised and unsupervised. In unsupervised clustering algorithms, the partitions can be viewed as the unlabelled patterns for example k-mean, k-nearest neighbor etc. Supervised clustering algorithms, label the patterns which is further used to classify or distinguish the software components for decision-making process for example support vector machine, neural network etc. Hence the partitions which can be obtained through clustering techniques may be labeled, or it is unlabeled. So the inspiration behind the design of an algorithm automation of component clustering. These clusters thus help in choosing the required component with high cohesion and coupling quickly and efficiently.

Related Work

This section discusses the related work done by researcher in the field of information retrieval and clustering. In today's world there is a never ending increase in the documents that are available in the repositories of corporate and same is the case with the Internet. As a result the focus of document clustering has shifted towards finding more productive way of going through large databases of documents and to get organized search results for display in a structured, preferably hierarchical manner.

Mark Sanderson, et.al [5] described the history of "information retrieval systems". "Information retrieval systems" was beginning with the development of the electro-mechanical searching devices that was used to find out the relevant items according to user's query. In 1950s, the information was retrieve with the help of ranked retrieval method. In 1960s, at this time the similarity measure function was used to find out the similarity between the two documents. In 1970, the vector space model was introduced. Nowadays, the information is retrieves by different methods like clustering, classification etc.

Mingyu Yao et.al [8] proposed an algorithm which is used to cluster the Chinese text documents. This algorithm is based on k-mean algorithm. In this paper firstly documents are pre-processed. After pre-processing, transformation of the text using term frequency (TF) and inverse document frequency (IDF) has been done and then apply the k-mean algorithm. After all this process the improving in k –mean algorithm is applied. In experimental results, this algorithm is not satisfactory.

Richa Loohach et.al [9] discussed the k-means clustering algorithm and various distance functions used in k-means clustering algorithm such as Euclidean distance function and Manhattan distance function. Experimental results are shown to observe the effect of Manhattan distance function and Euclidean distance function on k-means clustering algorithm. These results also show that distance functions furthermore affect the size of clusters formed by the k-means clustering algorithm.

Atreya Basu et.al on [10] compared two supervised learning algorithms that are SVM and NN. These algorithms are used for the classification of the documents and also use to decrease the feature set that provides better results."Reuters-21578" data set is used; it is a collocation of 21,578 documents which is in SGML format. This data set contains 118 predefined categories. In experimental results, the precision and recall of support vector machine and neural network are significantly different for the "Reuters-21578" data set. Support vector machine gives better results as compare to neural network because support vector machine is less complex.

Zhexue Huang [11] discussed k-mode algorithm. This algorithm is used to overcome the drawbacks of k-mean algorithm. So k-mode algorithm extends the k-means paradigms to categorical domains. And new dissimilarity measure is discussed which is used to deal with categorical objects. K-mode used modes instead of means. K-mode clustering technique is also used to reduce CCF (clustering cost function) by using frequency based method to update modes in the clustering process.

Thorsten Joachims et.al [14] proposed an algorithm for text categorization that is support vector machine. This technique is suitable for text categorization. In experimental results, they conclude that support vector machine gives better performance on text categorization process as compare to other methods. The main advantage of support vector machine is provides robustness as compare to conventional methods. It does not need any parameter tuning.

Proposed System

The proposed work Support Vector Machine as a linear classification, non linear classification and Multi-class classification for the test case documents.

The entire system is organized into four major modules namely, Preprocessing, Learning, Classification and Evaluation. The preprocessing stage involves the techniques and processes which completes task of text mining. The support vector machine is formulated by the training and testing modules which indeed represents the learning and classification tasks. Finally the evaluation phase measures the accuracy, precision and recall of the system.

In our experiment, NLTK toolkit is used for tokenization and stop word removing processing. The tokenization process helps to split the sentence into the tokens or words. The stop word remover help to remove the extra common words from the data set and help to reduce the size of the data set and it will help us easy to identify the key words in the data set and frequency distribution of concept words in overall concepts.

The workflow of the proposed system is represented as follows.

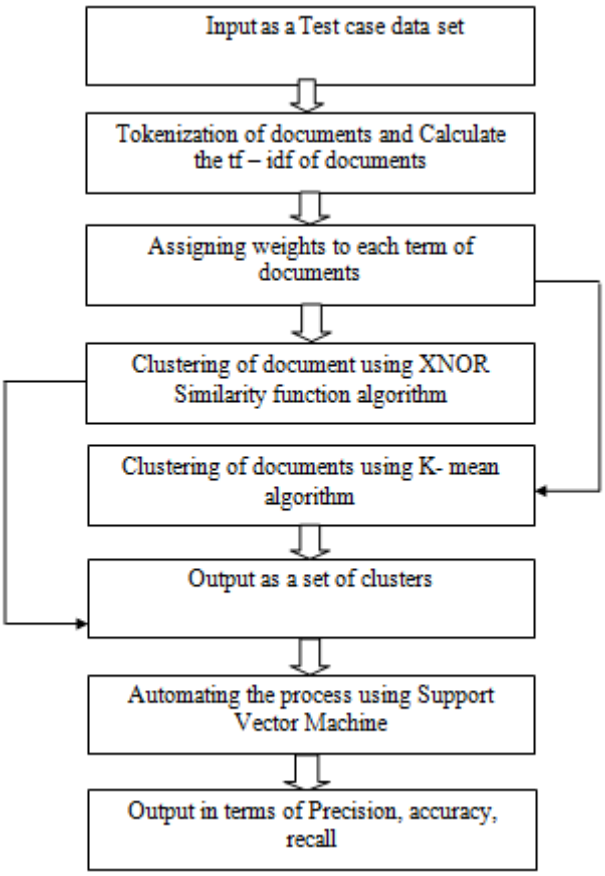

Fig.1. Methodology

A. *Preprocessing of Documents*

Preprocessing of documents helped to improve the data and get appropriate information. Different preprocessing techniques and tools are used to normalize the data. These techniques are used to remove the noisy data, null values, missing values and incorporate field or column from the dataset. The objective of pre-processing of documents is to represent the documents in such a way that their retrieval from the software repository and storage in the software repository are very efficient.

i. Tokenization of Documents

The first step while creating cluster of document is to convert the whole document into single statement. Tokenization of software documents is the process of breaking up sentence into words is called tokens.

For example:

Input: Support vector machine is an algorithm

Output:"Support", "vector", "machine","is","an", "algorithm"

ii. Removal of stop words

Stop word removing is one of the pre-processing stage of natural language processing. It is the method of removing the common stop words in English like 'is', 'was', 'where', 'the',

'a', 'for', 'of', 'in' exe. The stop words in corpus make little difficult for coups processing and feature extraction. To avoid this issues we are choose stop word remover.

B. *Transformation of Documents*

i Term Frequency

TF (Term Frequency) measures how commonly a term appears in a document. Since each document have its respective length then it may be probable that frequent entity will occur higher times in longer documents than in shorter ones. Therefore, the term frequency is usually divided by the length of the document as a method of normalization.

TF (t) = (k / (f))

Where k is Number of documents with term t in it and f is Total number of terms in the document.

ii Inverse Document Frequency

IDF (Inverse Data Frequency) measures how relevant a term is while calculating TF, all entities are judged equally relevant. "is" "of" and "that" are entities that may appear a lot of times but have little importance. So, we are required to reduce down the frequent entities while prioritize the rare ones, by calculating the following:

$IDF = log\ (t_n) / (F)$

Where t_n is Total number of documents and F is Number of documents with term t in it.

iii Weighted Text

Weighted text is the product of term frequency and inverse document frequency to each term in each document.

$$w = tf * idf$$

Where w is weighted, text tf is term frequency and idf is inverse document frequency.

C. *K-Mean Clustering Algorithm*

After preprocessing and transformation of documents, the process of k-mean algorithm is carried out.

K-mean is an unsupervised learning algorithm for solving clustering problem in any field. It follows simple procedure by fixing the number of cluster priory to make it easy to classify given data set. Mainly for each cluster there should centers defined denoted by k. The centers (k) should be positioned in a manipulative way so that they produce same result as different position of centers (k) causes different outcomes. So, best way to position these centers is far away from each other. After that, each point which belongs to its respective data set and relate it to its closest center. First step will complete when there is no point pending. Now re-compute (k) new center as bar center of the clusters resulting from the previous step. After this, the same data set points and the nearest new center should be binded together. The value of k keeps on changing until convergence in results occurs.

This algorithm partitions the data into K clusters (Cluster1, Cluster2, Cluster3, ……, ClusterN) represented by their centers or means. The center of each cluster is calculated as the mean of all points belonging to that cluster. Initial cluster center is randomly selected. Then findout out the distance between the two points by using distance function. Distance function in k-mean clustering algorithm plays an important role. There are Different types of distance functions which is used to measure the distance between the two points. Here Euclidean distance is used to find out the distance between the two points. The Euclidean distance equation is:-

$$dist\ ((a,b),(c,d)) = \sqrt{(a-c)^2 + (b-d)^2}$$

Notation:

Let the set of data points (or instances) D be {x1, x2, …, xn}, where xi = (xi1, xi2, …, xir) is a vector in a real-valued space X Í Rr, and r is the number of attributes (dimensions) in the data.

Input:-Number of attributes

Output:-Set of clusters

k:-Specified by user

Pseudo code of K-Mean Algorithm

```
k-mean(k,D)
Choose k data points as the initial means (cluster centers)

Repeat
    for each point x € D do
        compute the distance from x to each mean;
        assign x to the closest mean
endfor
re-compute the mean using the current cluster memberships

until the stopping criterion is met.
```

Fig.2. Pusedo code of support vector machine

D. *Support Vector Machine*

After getting the number of clusters from the k-mean clustering algorithm then divides randomly whole data set into two parts. First part of data set is 80 percent that will undergo training and second part of data set is 20percent will undergo testing. Training of data set is using support vector machine.

SVM stands for "support vector machine. It is supervised learning algorithm used in classification. Classification can be viewed as the separating of two or more than two classes in features space. "Support vector machine" is simple, scalable and easy way to classify the documents. Basically, SVM is used for boundary data analysis which is not classified by the clustering techniques. This algorithm is used to get the high accuracy in the "information retrieval process". The main objective of SVM technique is to find out the optimum hyperplane that separates clusters of vector in such a way that

cases with one class of the target variable are on one side of the hyperplane and cases with the other class are on the other side of the hyperplane. There are number of vectors on the feature space. But the vectors that are close to the hyperplane are the support vectors. Support vector machine consider two approaches

i. *When the data is linearly separable*
ii. *When the data is not linearly separable*

i. Linear Support Vector Machine

This technique is used to classify the two categories when the data is linearly separable. In this algorithm, we considered one class is negative class and other is positive class and these two classes are separated by the hyperplane.

Let D be the dataset with n points in d dimensional space which is to be classified. D = {(xi, yi)}, with i = 1, 2,..., n and let there be only two class labels such that yi is either +1 or-1. if the dataset is linearly separable, a separating hyperplane can be found such that for all points with label-1, h(x) < 0 and for all points labeled +1, h(x) > 0. In this case, h(x) serves as a linear classifier or linear discriminant that predicts the class for any point.

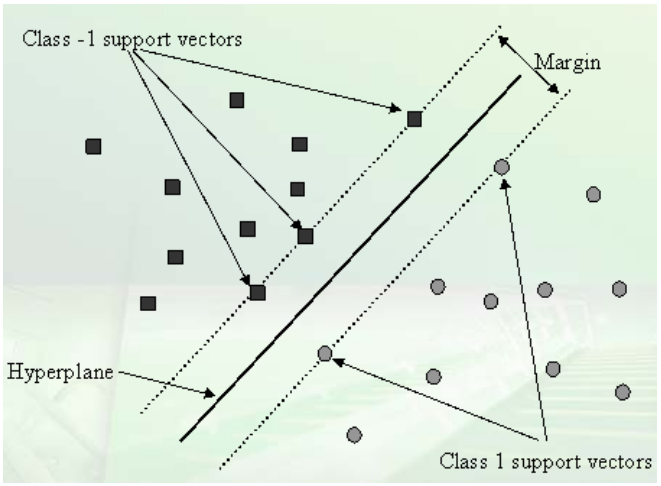

Fig.3. Linear Classification [8]

In optimal hyperplane, the distance to the closest negative point should be equal to the distance to the closest positive point.For calculating the LSVM; the objective is to accurately classify all the data points. The mathematical calculations of LSVM are:-

- For all points with label +1

$$h(x) = w^T x + b \geq 1$$

- For all points with label-1

$$h(x) = w^T x + b \leq 1$$

where w is a d-dimensional weight vector, x is vector points and b is a scalar bias. So to separate the data should always be greater than zero.

- Maximum Margin

$$M = 2 / ||w|| = 2 \sqrt{w^T.w}$$

- Then decision function f(x) is:-

$$f(x) = w^T x + b$$

Here x is the test data and w is weight and b is bias.

ii Non Linear Support Vector Machine

This technique is used to classify the two categories when the data is not linearly separable. Nonlinear classification uses kernel method for classification. The network consists of the following components

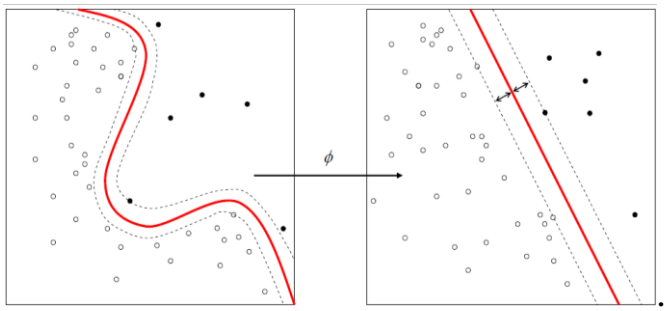

Fig.4. Non Linear Classification [8]

a) Kernel functions

Kernel function is use to map all points with a mapping function Φ(x) to a space of sufficiently high dimension so that they will be separable by a hyperplane.

- Input space: the space where the points xi are located
- Feature space: the space of Φ(xi) after transformation

There are three types of kernel functions and these kernel functions are;-

- Quadratic kernel function
- Polynomial kernel function

$K(x_i x_j) = (x_i x_j + 1)q$,where q is the degree of polynomial

- Radial basis kernel function

$$\phi_j(\vec{x}) = \exp(-\gamma \|\vec{x} - \vec{x}_j\|^2)$$

iii. Multiclass classification

Multiclass classification is used when there are three and more than three categories. In this method "one versus all" is used for classification. According to this method, each class or category is split out and all the other class or categories are combined and to choose the class which classifies the test data (new input) with greatest margin. It divides an m class problem into m binary problem. The equation of multilayer classification method is

$$W^{(y_i)}.x_j + b^{(y_i)} \geq W^{(y')}.x_j + b^{(y')} + 1, \forall y' \neq y_i,$$

Where w is weight, b is bias, yi represents the class where i=1, 2, 3, 4…..n and y' represents the other class.

Multiclass SVM aims to assign labels to instances by using support vector machines, where the labels are drawn from a finite set of several elements.

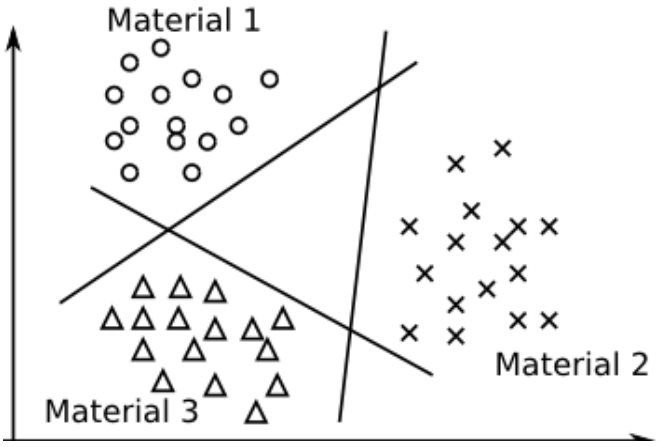

Fig.5.Multi Class Classification

Notation:
Input: Requirement (R) and test(T) cases document
OutPut: Cluster of document according to document similarity

Support Vector Machine Algorithm

```
Input :Requirement (R) and test(T) cases document
OutPut : Cluster of document according to document similarity
For I=0 to Len(R and T)… do
        Tokenize input text of R and T
        Remove Stop word
        Make a Vector space model (TF-IDF) Log N/D

For I=0 to Len(R and T)… do
        Vectorize all Documents (t1, t2, t3.....tn) for test cases
        Input these vectorization on K-mean
        Output is K number of cluster
        Make K number of classes
        Put features of these documents and make training set

For I=0 to Len (Doc.training.Features)… do
        Input on Support Vector Machine
        Support Vector Machine Model

For I=0 to Len (Doc.test.Features)… do
        Test on Support Vector Machine, linear, non linear, multi
classification modeluse following cluster classes

        C2 # when data is linearly classified, when data is not linearly
classified#
        C3,C4,C5,C6,C7 # three and more than three classes #
Analysis the cluster on

    • Accuracy = TP+TN/P+N
    • Precision= TP/TP+FP
    • Recall =TP/TP+FN
# after getting the output we will check the precision and recall of previous
defined algorithms with svm #
```

Fig.6. Pusedo code of support vector machine

Simulation
In the field of machine learning, a confusion matrix, also known as a contingency table or an error matrix, is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning one (in unsupervised learning it is usually called a matching matrix). Each column of the matrix represents the instances in a predicted class, while each row represents the instances in an actual class.
TABLE.1. 2 class confusion matrix

Confusion Matrix		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.
- D is the number of correct predictions that an instance is positive.

a) Accuracy
Accuracy is the measure to determine the accuracy of a classifier. This measure indicates that what percentage of the total test set records correctly classified.

$$A = S_{CC} / S_N$$

Where SCC is Sum of correct classifications and SN is Sum of all number of classifications.

b) Precision
Precision for a class is the number of true positives (TP) (i.e. the number of items correctly labeled as belonging to the positive class) divided by the total number of elements labeled as belonging to the positive class.

TP/(TP+FP)

c) Recall
Recall is defined as the number of true positives divided by the total number of elements that actually belong to the positive class (i.e. the sum of true positives and false negatives (FN), which are items which were not labeled as belonging to the positive class but should have been).

TP/(TP+FP)

B. Results of Support Vector Machine

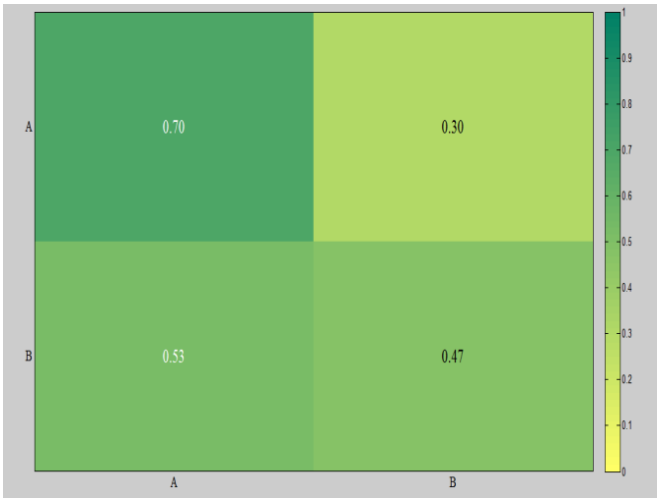

Fig.7 Linear Support Vector Machine

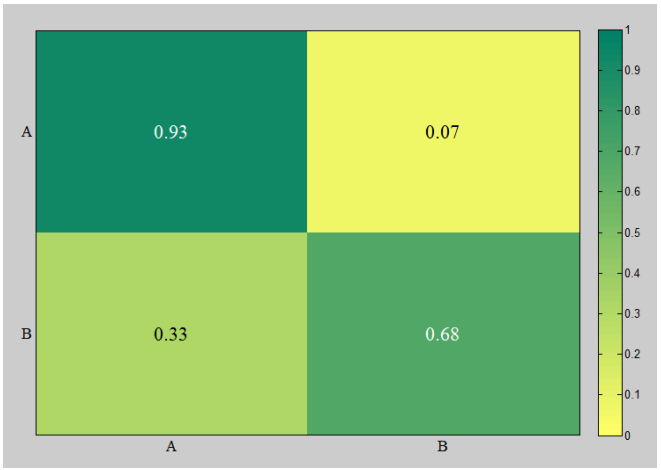

Fig.8 Polynomial kernel function

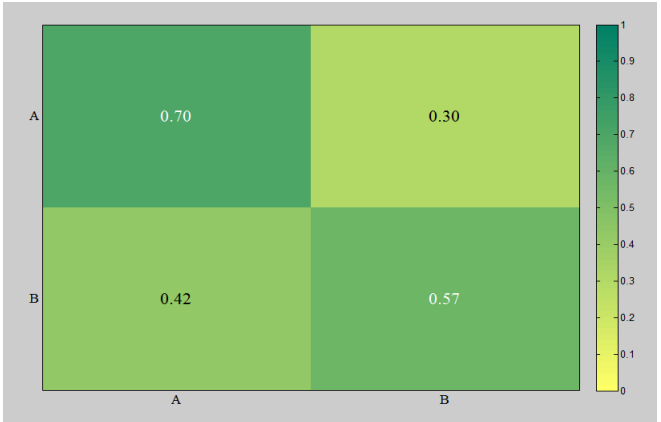

Fig.9 Quadratic kernel function

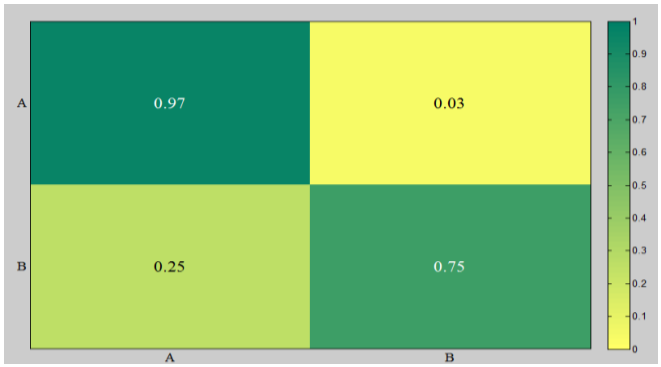

Fig. 10 rbf kernel function

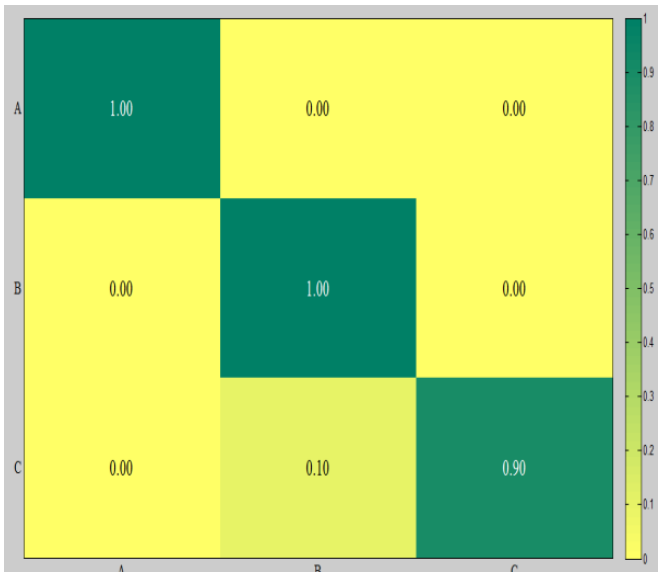

Fig. 11 Three class classification

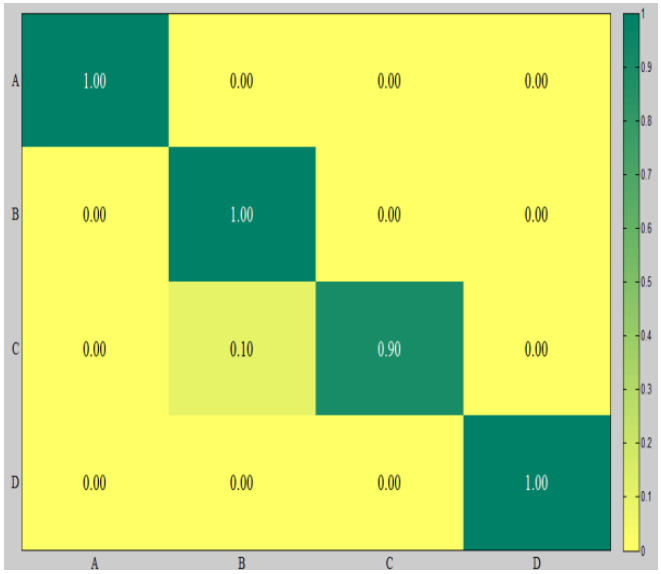

Fig. 12 Four class classification

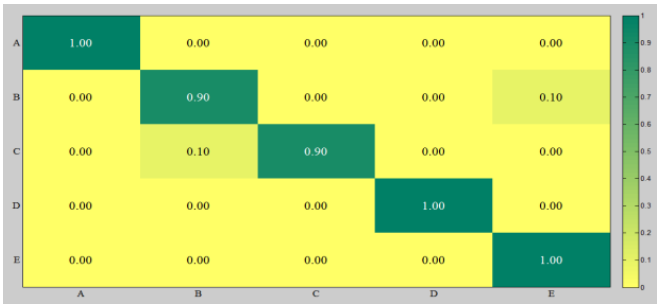

Fig.13 Five class classification

Fig. 14 Six class classification

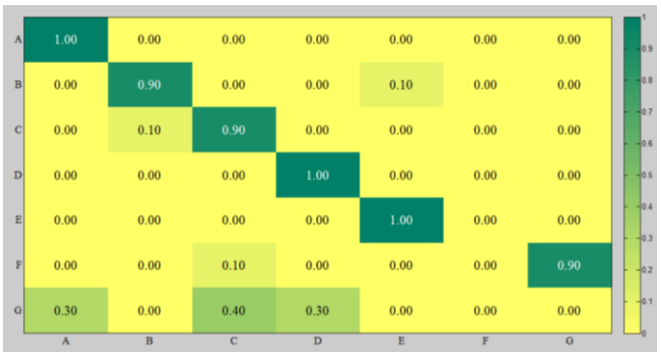

Fig. 15 Seven class classification

C. Comparison of K-mean and Support Vector Machine

Number of clusters	Support Vector Machine		
	Accuracy	Precision	Recall
Cluster 2	When the data is linearly separable (LSVM)		
	58.5%	61.5%	47%
	When the data is not linearly separable		
	Polynomial = 80.0%	80.1%	81.8%
	Quadratic = 63.81%	63.5%	64.0%
	Rbf = 83.5%	86%	87.82%
Cluster 3	Multi Class Classification		
	96.6%	96.66%	96.97%
Cluster 4	97.5%	97.5%	97.72%
Cluster 5	96.0%	96.0%	96.18%
Cluster 6	80.0%	80.0%	69.91%
Cluster 7	68.57%	68.5%	56.57%

D. Comparison of K-mean and Support Vector Machine in form of Graphs

Number of clusters	k-Mean Algorithm		
	Accuracy	Precision	Recall
Cluster 2	81.6%	58.3%	31.7%
Cluster 3	70.5%	59%	44.0%
Cluster 4	74.3%	57.14%	54.9%
Cluster 5	81.3%	52.8%	62.9%
Cluster 6	54.3%	61.90%	60.8%
Cluster 7	50.5%	66.66%	73.9%

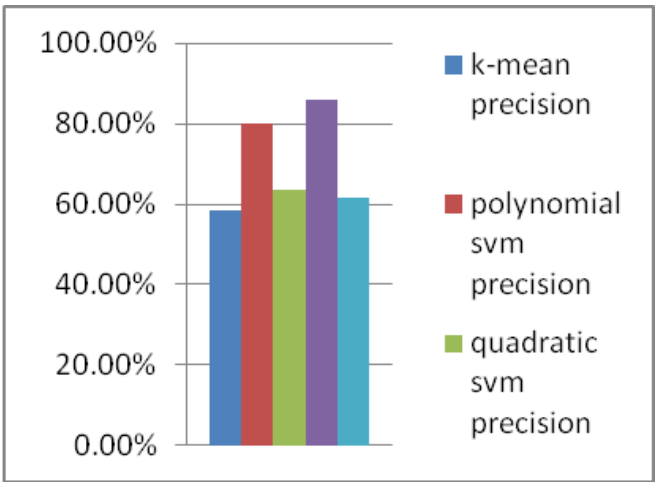

Figure 16 Precison of k-mean and Two class classification

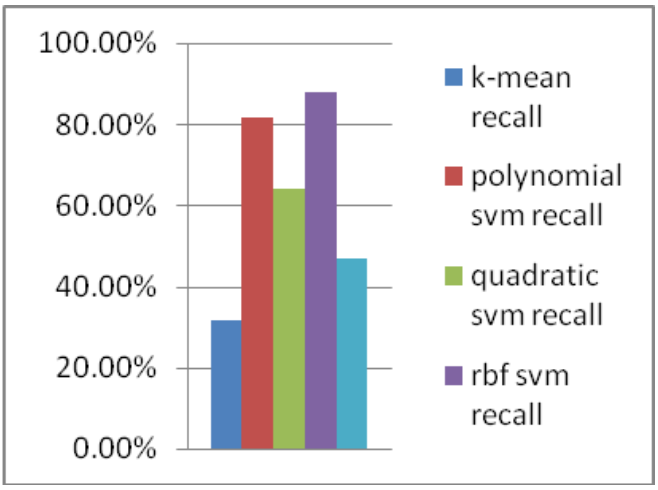

Figure 17 Recall of k-mean and Two class classification

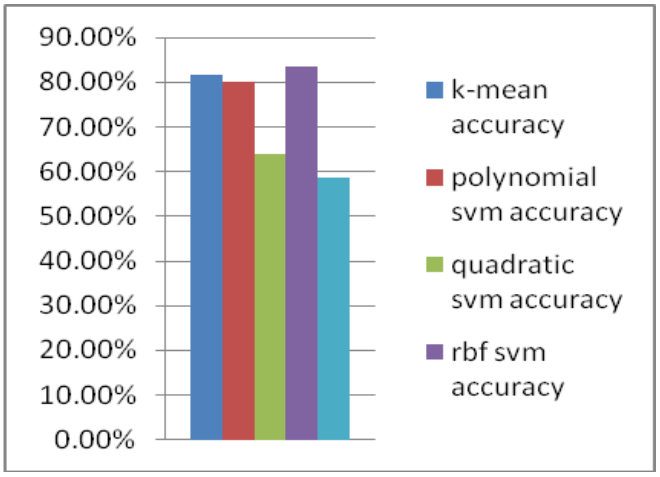

Figure 18 Accuracy of k-mean and multiclass class classification

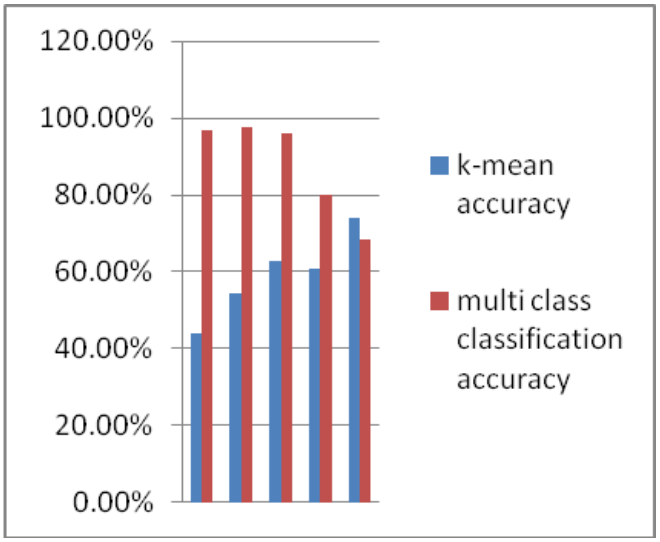

Figure 19 Accuracy of k-mean and multiclass class classification

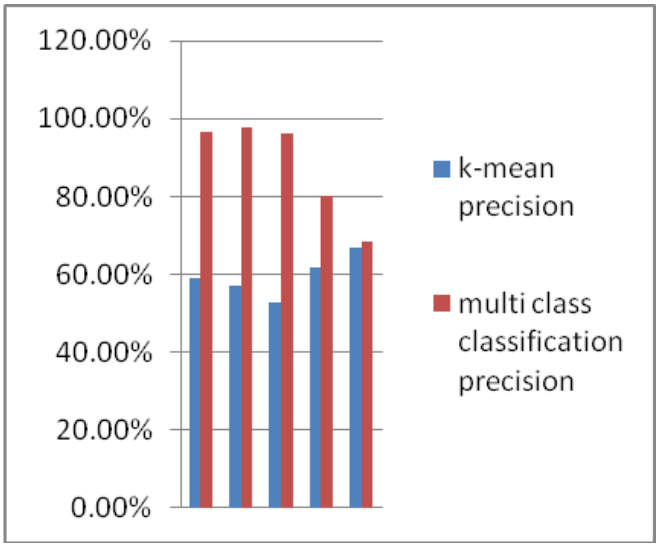

Figure 20 Precison of k-mean and multi class classification

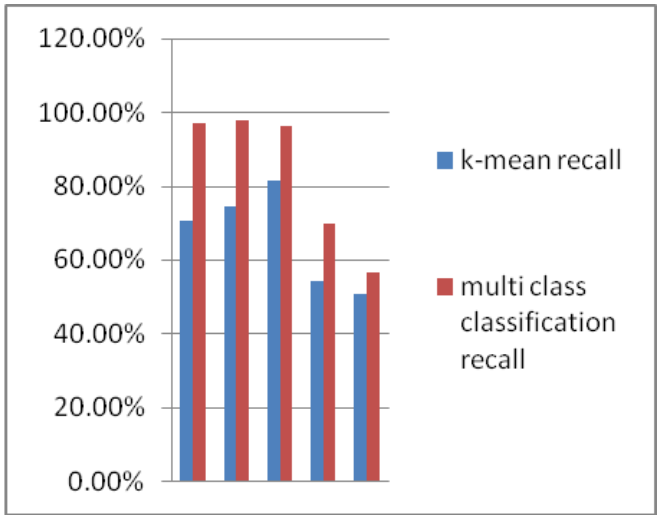

Figure 21 Recall of k-mean and multi class classification

Conclusion

We formulated Support Vector Machine learning paradigm for the classification of documents from the test case data set. In this research we have represented that how supervised and unsupervised learning plays important role in component based test case documents. Unsupervised clustering methods are use to cluster the similar type of documents, reduce the search space and provide higher efficient and supervised learning is use to automate the process. The available data may contain anomalies, redundant values. All these are removed in first phase i.e preprocessing. In order to retrieve documents efficiently k-mean clustering algorithm and Support Vector Machine has been implemented. K-mean clustering algorithm is implemented using Euclidian distance function. But this algorithm doesn't automate the process. In the proposed work, to automate the process using support vector machine is applied on the data set which gave the classified data and then parameter evaluation is done. We have implemented, when the data is linearly separable, when the data is not linear separable data and when there are three and more than three categories. The problem is represented as a multi-class multi-label problem and addressed by SVM matlab Implementation. In future support vector machine can be used with ranking support vector machine learning

References

[1] Mili, Ali, Rym Mili, and Roland T. Mittermeir, "A survey of software reuses libraries." Annals of Software Engineering 5, pp.349-414, Year 1998.

[2] Lu, Hongjun, Rudy Setiono, and Huan Liu, "Effective data mining using neural networks." Knowledge and Data Engineering, IEEE Transactions on 8, vol.8, no. 6, pp.957-96, Year 1996.

[3] Tracz, Will. "Software reuse myths." ACM SIGSOFT Software Engineering Notes 13, no., pp.17-21, Year 1988.

[4] Mili, A., Chmiel, S. F., Gottumukkala, R., and Zhang, L, "An integrated cost model for software reuses", In Proceedings of the 22nd international conference on Software engineering, ACM, pp.157-166, Year 2000.

[5] Sanderson, Mark, and W. Bruce Croft, "The history of information retrieval research." Proceedings of the IEEE 100, no, Special Centennial Issue, 1444-1451, Year 2012.

[6] Pressman, Roger S, "Software engineering: a practitioner's approach", Palgrave Macmillan, Year 2005.

[7] Sharma, Manish, and Rahul Patel, "A Survey on Information Retrieval Models, Techniques And Applications", International Journal of Emerging Technology and Advanced Engineering, ISSN, pp.2250-2459, Year 2013.

[8] Burges, Christopher JC, "A tutorial on support vector machines for pattern recognition", Data mining and knowledge discovery 2, pp.121-167, Year 1998.

[9] Yao, Mingyu, Dechang Pi, and Xiangxiang Cong, "Chinese Text Clustering Algorithm Based k-means." Physics Procedia 33, pp.301-307, year 2012.

[10] Loohach, Richa, and Kanwal Garg, "Effect of Distance Functions on K-Means Clustering Algorithm", International Journal of Computer Applications, pp.231-236, year 2012.

[11] Basu, Atreya, C. Walters, and M. Shepherd, "Support vector machines for text categorization", In System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on, pp. 7-pp. IEEE, Year 2003.

[12] Huang, Zhexue, "A Fast Clustering Algorithm to Cluster Very Large Categorical Data Sets in Data Mining", DMKD, p. 0, Year 1997.

[13] Hwee Tou, Wei Boon Goh, and Kok Leong Low, "Feature selection, perceptron learning, and a usability case study for text categorization", In ACM SIGIR Forum, vol. 31, no. SI, pp. 67-73, ACM, Year 1997.

[14] Joachims, Thorsten, "Text categorization with support vector machines: Learning with many relevant features", Springer Berlin Heidelberg, pp.281-287, Year 1998.

[15] Veropoulos, K., N. Cristianini, and C. Campbell, "The application of support vector machines to medical decision support: a case study." Advanced Course in Artificial Intelligence, pp.1-6, Year 1999.

[16] Hidber, Christian, "Online association rule mining", Vol. 28, ACM, Year 1999.

[17] Steinbach, Michael, George Karypis, and Vipin Kumar,"A comparison of document clustering techniques", KDD workshop on text mining, vol. 400, no. 1, pp. 525-526, Year 2000.

[18] Pathak, Praveen, Michael Gordon, and Weiguo Fan. "Effective information retrieval using genetic algorithms based matching functions adaptation." InSystem Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on, pp. 8-pp. IEEE, Year 2000.

[19] Han, Jiawei, Jian Pei, and Yiwen Yin, "Mining frequent patterns without candidate generation", In ACM SIGMOD Record, vol. 29, no. 2, pp. 1-12, ACM, Year2000.

[20] Nikravesh, Masoud, Roy D. Adams, and Raymond A. Levey, "Soft computing: tools for intelligent reservoir characterization (IRESC) and optimum well placement (OWP)", Journal of Petroleum Science and Engineering 29, pp.239-262, Year 2001.

[21] Zhong, Shi, and Joydeep Ghosh, "A comparative study of generative models for document clustering", In Proceedings of the workshop on Clustering High Dimensional Data and Its Applications in SIAM Data Mining Conference, Year 2003.

[22] Maqbool, Onaiza, and Haroon Atique Babri, "The weighted combined algorithm: A linkage algorithm for software clustering", In Software Maintenance and Reengineering, 2004. CSMR 2004. Proceedings, Eighth European Conference on, pp. 15-24. IEEE, Year 2004.

[23] Hung, Ming-Chuan, Jungpin Wu, Jin-Hua Chang, and Don-Lin Yang, "An Efficient k-Means Clustering Algorithm Using Simple Partitioning", journal of information science and engineering 21, pp.1157-1177, Year 2005.

[24] Finley, Thomas, and Thorsten Joachims, "Supervised clustering with support vector machine", In Proceedings of the 22nd international conference on Machine learning, pp. 217-224. ACM, Year 2005.

[25] Cao, Yunbo, Jun Xu, Tie-Yan Liu, Hang Li, Yalou Huang, and Hsiao-Wuen Hon. "Adapting ranking SVM to document retrieval", In Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 186-193. ACM, Year 2006.

[26] Khan, Shehroz S., and Shri Kant, "Computation of Initial Modes for K-modes Clustering Algorithm Using Evidence Accumulation", In IJCAI, pp. 2784-2789, Year 2007.

[27] Abbas, Osama Abu, "Comparisons between Data Clustering Algorithms." Int. Arab J. Inf. Technol. 5, pp.320-325, Year 2008.

[28] Hammouda, Khaled M., and Mohamed S. Kamel, "Hierarchically distributed peer-to-peer document clustering and cluster summarization", Knowledge and Data Engineering, IEEE Transactions on 21, pp.681-698, Year 2009.

[29] Luo, Congnan, Yanjun Li, and Soon M. Chung, "Text document clustering based on neighbors." Data & Knowledge Engineering 68, pp.1271-1288, Year 2009.

[30] Chen, Xiuguo, Wensheng Yin, Pinghui Tu, and Hengxi Zhang, "Weighted k-means algorithm based text clustering", In Information Engineering and Electronic Commerce, 2009. IEEC'09, International Symposium on, pp. 51-55. IEEE, Year 2009.

[31] Naldi, Murilo Coelho, Ricardo JGB Campello, Eduardo R. Hruschka, and A. C. P. L. F. Carvalho, "Efficiency issues of evolutionary k-means." Applied Soft Computing 11, pp.1938-1952, Year 2011,

[32] Singh, Shalini S., and N. C. Chauhan, "K-means v/s K-medoids: A Comparative Study",In National Conference on Recent Trends in Engineering & Technology, vol. 13, Year 2011.

[33] Peleja, Filipa, Gabriel Pereira Lopes, and Joaquim Silva, "Text Categorization: A comparison of classifiers, feature selection metrics and document representation", In Proceedings of the 15th Portuguese Conference in Artificial Intelligence, pp. 660-674., Year 2011.

[34] Sudhakaran, Jasmine Kalathipparambil, and Ramaswamy Vasantha, "A mixed method approach for efficient component retrieval from a component repository", Journal of Software Engineering and Applications 4, pp. 442-448, Year 2011.

[35] Sembiring, Sajadin, M. Zarlis, Dedy Hartama, S. Ramliana, and Elvi Wani. "Prediction of student academic performance by an application of data mining techniques", International Proceedings of Economics Development & Research 6 Year 2011.

[36] Nithya, K., M. Saranya, and C. R. Dhivyaa, "Concept Based Labeling of Text Documents Using Support Vector Machine", pp-456-460, Year 2012.

[37] Menaka, S., and N. Radha. "Text Classification using Keyword Extraction Technique." International Journal of Advanced Research in Computer Science and Software Engineering 3, no. 12, Year 2013.

[38] KS, Manjula, Sarvar Begum, and D. Venkata Swetha Ramana. "Extracting Summary from Documents Using K-Mean Clustering Algorithm." measurement 2, no. 8, Year 2013.

[39] Radhakrishna, Vangipuram, Chintakindi Srinivas, and CV Guru Rao. "Document Clustering Using Hybrid XOR Similarity Function for Efficient Software Component Reuse." Procedia Computer Science 17, pp.121-128, Year 2013.

[40] Yadav, Shweta, Kaur, Kamaljeet, "Design of Rank Based Reusable Component Retrieval Algorithm", International Journal of Advanced Research in Computer Science and Software Engineering, vol.3, issue 11, pp. 850-857, Year 2013.

[41] Kashyap, Preeti, Shailendra Kumar Shrivastava, and Babita Ujjainiya. "A weighted seeds affinity propagation clustering for efficient document mining." In2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT), pp. 1-7. IEEE, Year 2013.

[42] Kaur, Manjot, and Navjot Kaur. "Web Document Clustering Approaches Using K-Means Algorithm." International Journal of Advanced Research in Computer Science and Software Engineering 3, pp.861-864, Year 2013.

[43] Lin, Yung-Shen, Jung-Yi Jiang, and Shie-Jue Lee. "A similarity measure for text classification and clustering." IEEE Transactions on Knowledge and Data Engineering, pp.1-15, year 2013.

[44] Kumar, Amit, "Software Reuse Libraries Based Proposed Classification for Efficient Retrieval of Components", International Journal of Advanced Research in Computer Science and Software Engineering, pp. 884-89, Year 2013.

[45] Radhakrishna, Vangipuram, Chintakindi Srinivas, and CV Guru Rao, "Document Clustering Using Hybrid XNOR Similarity Function for Efficient Software Component Reuse", Procedia Computer Science 17, pp.121-128, Year 2013.

[46] Sood, Tamanna, "Optimal Component Software Development based on Meta Data Repositories", International Journal of Advanced Research in Computer Science and Software Engineerin, pp.74-79, Year 2013.

[47] Srinivas, Chintakindi, Vangipuram Radhakrishna, and CV Guru Rao", Clustering and Classification of Software Component for Efficient Component Retrieval and Building Component Reuse Libraries", Procedia Computer Science 31, pp.1044-1050, Year 2014.

[48] Radhakrishna, Vangipuram, Chintakindi Srinivas, and C. V. GuruRao, "A modified Gaussian similarity measure for clustering software components and documents", In Proceedings of the International Conference on Information Systems and Design of Communication, pp. 99-104, ACM, Year 2014.

[49] Renukadevi, D., and S. Sumathi, "Term based similarity measure for text classification and clustering using fuzzy C-means algorithm", Int. J. Sci., Eng. Technol. Res, (IJSETR) 3, pp.1093-1097,Year 2014.

[50] Zhou, Xiaofei, Yue Hu, and Li Guo, "Text Categorization based on Clustering Feature Selection." Procedia Computer Science 3, pp.398-405, Year 2014.

[51] Syal, Rishi, and V. Vijaya Kumar, "Innovative Modified K-Mode Clustering Algorithm", pp.390-398, Year 2014.