Implement a Mining Web Document through New Data Clustering Algorithm

[1] Najim Sheikh, [2] Kasim Ali Saiyed, [3] Ajeet Malviya, [4] Swapnil Sharma Dikshit

[1] M-tech Scholar, RTMNU Nagpur.

[2] M-tech Scholar, RGPV Bhopal.

[3] M-tech Scholar, RGPV Bhopal.

[4] M-tech Scholar, JNTU Hydrabaad.

Abstract - This paper formulates, simulates and assess an new data clustering algorithm for mining web documents with a view to preserving their conceptual similarities and eliminating the problem of speed while increasing accuracy. The improved data clustering algorithm was formulated using the concept of K-means algorithm. Real and artificial datasets were used to test the proposed and existing algorithm. The simulated results were compared with the existing data clustering algorithm using accuracy, response time, and entropy as performance parameters. The results show an improved data clustering algorithm with a new initialization method based on finding a set of medians extracted from a dimension with maximum variances. The results of the simulation showed that the accuracy is at its peak when the number of clusters is 3 and reduces as the number of clusters increases. When compared with existing algorithm, the proposed clustering algorithm showed an accuracy of 89.3% while the existing had an accuracy of 88.9%. The entropy was stable for both algorithms with a value of 0.2485 at k = 3. This also decreases as the number of clusters increase until when the number of clusters reached eight where it increased slightly. In addition, the response time decreased from 0.0451 seconds to 0.0439 seconds when the number of clusters was three. This showed that the proposed data clustering algorithm decreased by 2.7% in response time as compared to the K-means data clustering. Finally this clustering algorithm helps to improve search result.

***Keywords* - Web Documents, Mining, Data Clustering, RealDataset, Artificial Datasets.**

1. Introduction

The World Wide Web is a vast resource of information and services that continues to grow rapidly. Powerful search engines have been developed to aid in locating unfamiliar documents by category, contents, or subjects. However, queries often return inconsistent results, with document referrals that meet the search criteria but are of no interest to the user [5].

While it may be currently feasible to extract in full the meaning of an HTML document, intelligent software agents have been developed which extract features from the words or structures of an HTML document and employ them to classify and categorize the documents [5]. Under classification, the researcher attempts to assign a data item to a predefined category based on a method that is created from pre-classified training data (supervised learning). Clustering's goal is to separate a given group of data items (the data set) into groups called clusters such that items in the same cluster are similar to each other and dissimilar to items in other clusters or to identify distinct groups in a dataset [3]. The results of clustering could then be used to automatically formulate queries and search for other similar documents on the web, or to organize bookmark files, or to construct a user profile. In contrast to the highly structured tabular data upon which most machine learning methods are expected to operate, web and text documents are semi structured.

Web documents have well defined structures such as letters, words, sentences, paragraphs, sections, punctuation marks, HTML tags and so forth. Hence, developing improved methods of performing machine learning techniques in this vast amount of non tabular, semi structured web data is highly desirable. In this work solutions to problems such as high dimensionality and scalability associated with existing techniques of mining web documents on the web were provided by proposing an improved data clustering algorithm.

2. Related Work

Document clustering is widely applicable in areas such as search engines, web mining, information retrieval and topological analysis. Most document clustering methods perform several pre-processing steps including stop words removal and stemming on the document set. Each document is represented by a vector of frequencies of remaining terms within the document. Some document clustering algorithms employ an extra pre-processing step that divides the actual term frequency by the overall frequency of the term in the entire document set.

A. Data Clustering: A Review [1]

The problem of clustering has been addressed in many contexts and by researchers in many disciplines; this reflects its broad appeal and usefulness as one of the steps in exploratory data analysis. It has great potentials in applications like object recognition, image segmentation and information filtering and retrieval.

Most of the clustering techniques fall into two major categories, and these are the hierarchical clustering and the partitioned clustering. Hierarchical clustering can further be divided into agglomerative and divisive, depending on the direction of building the hierarchy. Hierarchical techniques produce a nested sequence of partitions, with a single all inclusive cluster at the top and singleton clusters of individual objects at the bottom.

B. Clustering Web Documents [6]

These algorithms start with the set of objects as individual clusters, then, at each step merges the two most similar clusters. This process is repeated until a minimal number of clusters have been reached, or if a complete hierarchy is required then the process continues until only one cluster is left. These algorithms are slow when applied to large document collections; single link and group-average can be implemented in O+ (n2) time (where n is the number of items), while complete link requires O (n3) time and therefore tends to be too slow to meet the speed requirements when clustering several items. In terms of quality, complete links tend to produce “tight” clusters, in which all documents are similar to one another, while single link have the tendency to create elongated clusters which is a disadvantage in noisy domains (such as the web), because it results in one or two large clusters, and many extremely small ones. This method is simple but needs to specify how to compute the distance between two clusters. The three commonly used methods for computing distance are the single linkage, complete linkage and the average linkage method respectively.

Divisive hierarchical clustering methods work from top to bottom, starting with the whole data set as one cluster, and at each step split a cluster until only singleton clusters of individual objects remain. They basically differ in two things, (i) Which cluster to split next (ii) How to perform the split. A divisive method begins with all patterns in a single cluster and performs the split until a stopping criterion is met.

C. A New Algorithm for Cluster Initialization [2]

CLUSTERING techniques have received attention in many areas including engineering, medicine, biology and data mining. The purpose of clustering is to group together data points, which are close to one another. The k-means algorithm [1] is one of the most widely used techniques for clustering.

The k-means algorithm starts by initializing the K cluster centers. The input vectors (data points) are then allocated (assigned) to one of the existing clusters according to the square of the Euclidean distance from the clusters, choosing the closest. The mean (centroid) of each cluster is then computed so as to update the cluster center. This update occurs as a result of the change in the membership of each cluster. The processes of re-assigning the input vectors and the update of the cluster centers is repeated until no more change in the value of any of the cluster centers.

The steps of the k-means algorithm are written below.

1. Initialization: choose K input vectors (data points) to initialize the clusters.
2. Nearest-neighbor search: for each input vector, find the cluster center that is closest, and assign that input vector to the corresponding cluster.
3. Mean update: update the cluster centers in each cluster using the mean (centroid) of the input vectors assigned to that cluster.
4. Stopping rule: repeat steps 2 and 3 until no more change in the value of the means.

However, it has been reported that solutions obtained from the k-means are dependent on the initialization of cluster centers.

Two simple approaches to cluster center initialization are either to select the initial values randomly, or to choose the first K samples of the data points. As an alternative, different sets of initial values are chosen (out of the data points) and the set, which is closest to optimal, is chosen. However, testing different initial sets is considered impracticable criteria, especially for large number of clusters.

D. Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values [13]:

Several variants of K-mean algorithm have been reported in the literature, such as the K-median. The K-mode algorithm is a recent partitioning algorithm that uses the simple matching coefficient measure to deal with categorical attributes. The K-prototype algorithm integrated the K-means and the K-modes algorithm to allow for clustering instance described by mixed attributes. Some of them attempt to select a good initial partition so that the algorithm is more likely to find the global minimum value. Another variation is to permit splitting and merging of the resulting clusters, i.e. a cluster is split when its variance is above a specified threshold, and the two clusters are merged when the distance between their centroids is below another pre-specified threshold. Using this variant, it is possible to obtain the optimal partition starting from any arbitrary initial partition, provided proper threshold values are specified. Another variation of the K-means algorithm involves selecting a different criterion function altogether.

E. Web Document Clustering [8]

Suffix Tree Clustering (STC) is a linear time clustering algorithm that is based on identifying the phrases that are common to groups of documents. A phrase in this context is an ordered sequence of one or more words and a base cluster to be a set of documents that share a common phrase. Suffix tree, as defined by a concept representation of a tie (retrieval) corresponding to the suffixes of a given string where all the nodes with one „child" are merged with their „parents". It is a divisive method which begins with the dataset as a whole and divides it into progressively smaller clusters, each composed of a node with suffixes branching like leaves.

STC has three logical steps:

(1) Document “cleaning”,

(2) Identifying base clusters using a suffix tree, and

(3) Combining these base clusters into clusters.

Step 1 - Document "Cleaning"

In this step, the string of text representing each document is transformed using a light stemming algorithm (deleting word prefixes and suffixes and reducing plural to singular). Sentence boundaries (identified via punctuation and HTML tags) are marked and non-word tokens (such as numbers, HTML tags and most punctuation) are stripped. The original document strings are kept, as well as pointers from the beginning of each word in the transformed string to its position in the original string. This enables us, once we identify key phrases in the transformed string, to display the original text for enhanced user readability.

Step 2 - Identifying Base Clusters

The identification of base clusters can be viewed as the creation of an inverted index of phrases for our document collection. This is done efficiently using a data structure called a *suffix tree*. This structure can be constructed in time linear with the size of the collection, and can be constructed incrementally as the documents are being read. The idea of using a suffix tree for document clustering was first introduced in. Here we present an improved clustering algorithm, which introduces the merger of base clusters (step three of the STC algorithm), and compare it using standard IR methodology to classical clustering methods in the Web domain. A suffix tree of a string *S* is a *compact trie* containing all the suffixes of *S*. We treat documents as strings of words, not characters, thus suffixes contain one or more whole words. In more precise terms:

1. A suffix tree is a rooted, directed tree.
2. Each internal node has at least 2 children.
3. Each edge is labeled with a non-empty sub-string of *S* (hence it is a *trie*). The label of a node in definedto be the concatenation of the edge-labels on the path from the root to that node.
4. No two edges out of the same node can have edge-labels that begin with the same word (hence it is *compact*).
5. For each suffix *s* of *S*, there exists a *suffix-node* whose label equals *s*. The suffix tree of a collection of strings is a compact trie containing all the suffixes of all the strings in the collection. Each suffix-node is marked to designate from which string (or strings) it originated from (*i.e.*, the label of that suffix node is a suffix of that string).

Step 3 - Combining Base Clusters

Documents may share more than one phrase. As a result, the document sets of distinct base clusters may overlap and may even be identical. To avoid the proliferation of nearly identical clusters, the third step of the algorithm merges base clusters with a high overlap in their document sets (phrases are not considered in this step).

3. Proposed Model

The proposed model for the clustering algorithm consists firstly of introducing a new initialization algorithm into the K-means data clustering algorithm. The new initialization method was taken from the method proposed by [2], i.e. a new algorithm for cluster initialization which was based on finding a set of medians extracted from a dimension with maximum variance.

The algorithm can be described as follows:

Step1. For a data set with dimensionality d, compute the variance of data in each dimension (column).

Step2. Find the column with maximum variance, call it cvmax and sort it in descending order.

Step3. Divide the data points of cvmax into k subsets, where k is the desired number of clusters.

Step4. Find the median of each subset.

Step5. Use the corresponding data points (vectors) for each median to initialize the cluster centers.

This new K-means algorithm performs proper clustering without pre-determining the exact cluster number and it is proven to be efficient and accurate[2].

The flowchart of the cluster center initialization algorithm embedded in the K-means routine is depicted in “Fig.1”.

Figure

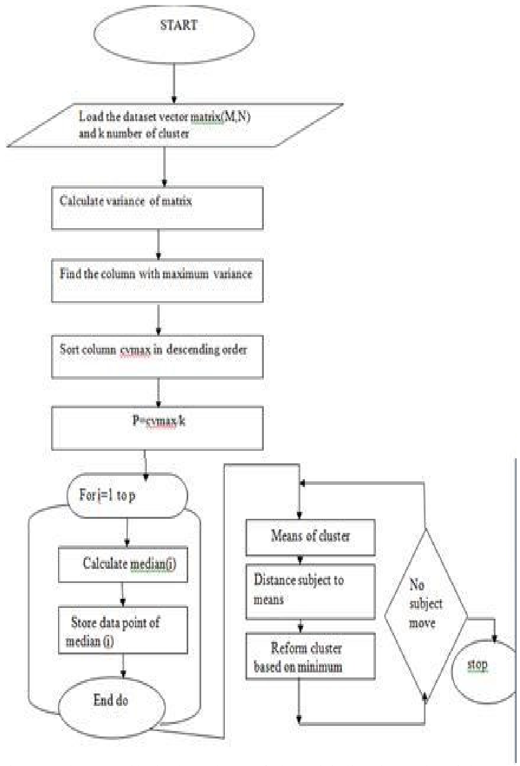
START
Load the dataset vector matrix(M,N) and k number of cluster
Calculate variance of matrix
Find the column with maximum variance
Sort column cvmax in descending order
P=cvmax/k
For i=1 to p
Calculate median(i)
Store data point of median (i)
End do
Means of cluster
Distance subject to means
Reform cluster based on minimum
No subject move
stop

Fig. 1: The Detailed Flowchart of the Initialization Algorithm

The reason for replacing the random initialization method is as a result of the disadvantage of the random initialization of the traditional K-means, which causes solutions to converge to local optimal, which are inferior to global optimum solutions. Local optimal is a selection from a given domain which yields either the highest value or lowest value (depending on the objective) when a given function is applied. If for instance, f(x)=-x+2 defined on real numbers, then the global optimum occurs at x=0 where f(x)=2 for all values of x f(x) is smaller. Hence, this new initialization method solves this problem and therefore improves the performance of the K-means algorithm.

A. Existing pseudo code for data clustering algorithm

The pseudo code of the existing data clustering algorithm is listed below

Input : $\alpha_{n\times m}$: dataset matrix, k: number of clusters

Output: c^k : the centroides

Procedure begin

```
1.  Input k: // Integer number
2.  Memory ⟵ $\alpha_{n\times m}$
3.  // Partition dataset into k clusters
    $r^1_{(1, nm/k)}$ ⟵ $\alpha_{[(1,\ldots\ldots n/k),m1}$
    $r^2_{(1, nm/k)}$ ⟵ $\alpha_{[(n/k+1,\ldots\ldots 2n/k),m1}$
    $r^k_{(1,\ldots\ldots nm/k)}$ ⟵ $\alpha_{[(n(k-1)/k+1,\ldots\ldots n),m1}$
4.  //select a center at random
    $C_1$ ⟵ randomize
    $(r^1)$ $C_2$ ⟵
    randomize $(r^1)$ $C_3$ ⟵
    randomize $(r^1)$
5.  While(C==Q)d
    o For y=1 to k
    do
            For i=1 to nm/k do
        $P_{y=}\sum (r^1-C_y)^2$
```

Equation (1) is a general mathematical model for the performance/objective function of the proposed data clustering algorithm.

$$KM(X, C) = \sum_{i=1\ j\varepsilon\{1\ldots k\}}^{n} \min\|x - c\|^2 + \text{Initialization}, \quad (1)$$

$$\text{Where initialization} = \sum □[\max [[\sum_{j=1}^{k} (x_i - c_j)^2]/n-1]/k]$$

$$\text{and } □= \text{median}_i$$

```
End for
$B_y$=under root $(P_y)$; //Vector for Euclidian distance
6.  G=$\min_{(1\ldots k)}$ (ceil (($\beta_y$)));
7.  $r_{(1\ldots,nm/k)}$ ⟵ cluster _reform($r^{(1\ldots k)}$ ,G)
8.  Q ⟵ C //Move old centers into Q
9.  $C^1$ ⟵ mean $(r^1)$; $C^2$
    ⟵ mean $(r^2)$; $C^k$
    ⟵ mean $(r^k)$; kk
    } end while
18. Procedure stop
```

B. Proposed pseudo code for data clustering algorithm

```
    Replace step 4 with steps 11 -17
11: //Calculate the variance form columns of $\propto_{n\times m}$
    $D_1 \leftarrow$ Variance ($\propto_{[(1,\ldots.,n),1]}$)
    $D_2 \leftarrow$ Variance ($\propto_{[(1,\ldots., n),2]}$)
    $D_m \leftarrow$ Variance ($\propto_{[(1,\ldots.,n),m]}$)
12: E $\leftarrow$ ceil (max (D));//find the max of column variance.
13: F $\leftarrow$ Sortpro ($\propto_{[(1,\ldots\ n), E]}$); //Sort the column variance
14: // Partition matrix $\propto_{[(1,\ldots.,n),E]}$ into k subset and store in
    //vector form
    $G^1_{(1,\ldots..,n/k)} \leftarrow \propto_{[(1,\ldots.,n),E]}$
    $G^2_{(n/k+1,\ldots.,2n/k)} \leftarrow \propto_{[(n/k+1,\ldots.,2n/k),E]}$
    $G^1_{((k-1)\ n/k+1,\ldots.n)} \leftarrow \propto_{[((k-1)n/k+1,\ldots.,n),E]}$
15: For q=1 to k do
        $C_q \leftarrow$ median ($C_1^{\,q}$);
16: //find the index for the median data point in vector G
    end for
17: //return to step 5 in the existing algorithm
```

IV. Simulation And Result Analysis

Step 1: k-means

Step 2: Proposed Clustering Algorithm

The simulation model was implemented using the Matlab language. To first implement the proposed data clustering algorithm, the K-means m file that comes along the statistical toolbox was modified and the command to run the algorithm was called with the necessary parameters. The input to the program was the simulated multivariate normal distributed dataset and the iris dataset, while the output of the result was again called by the confusion matrix m file to derive the confusion matrix for each consecutive run. After generating the matrix, (2) given by [11] was used to calculate the accuracy for each run. This procedure was repeated several times and sometimes produced irregular results. However, the best run was chosen at the end.

The performance measures, such as accuracy, adjusted rand index, entropy and speed, were used to show the improvement of the proposed data clustering algorithm over the existing algorithm. The results are reported on a PC with the following configuration: Intel (R) core (TM) 2 Duo, 1.83GHz, 2038MB, 160GB HD and WLAN and Blue tooth. Where N is the number of samples in the dataset, a_i is the number of data samples occurring in both cluster i and its corresponding class, which have the maximal value.

$$r = \sum_{i=1}^{k} (a_i/N) \qquad (2)$$

A. Artificial datasets

These datasets were generated from a multivariate normal distribution, whose mean vector and variance of each variable (is assumed to be equal; and hence covariance is zero). Also, in order to compare the performance when some outliers are present among objects, outliers were added to the generated datasets. These outliers were generated from a multivariate normal distribution.

B. Fisher's Iris datasets

The Iris flower data set or Fisher's Iris data set is a multivariate data set introduced by Sir Ronald Aylmer Fisher in 1936 as an example of discriminate analysis. It is sometimes called Anderson's Iris dataset, because Edgar Anderson collected the data to quantify the geographic variation of *Iris* flowers in the Gaspe Peninsula (Wikipedia Free Encyclopedia). R.A. Fisher's Iris dataset is often referenced in the field of pattern recognition. It consists of 3 groups (classes) of 50 patterns each. One group corresponds to one species of Iris flower: Iris Setosa (class *C*1), Iris Versicolor (class *C*2), and Iris Verginica (class *C*3). Every pattern has 4features (attributes), representing petal width, petal length, sepal width, and sepal length (expressed in centimeters).

Tables I and II show the summary of results on accuracy of the K-means and proposed clustering algorithm. The number of clusters was varied from 3 to 10 for a fixed number of iterations 10 and the best results were used at the end of the iterations. The results show that the accuracy is at its peak when the number clusters is 3 and reduces as the number of clusters increases. In comparing the standard results in Table I with the simulated results in Table II, it is shown that the accuracy of the proposed method is higher at K=3 and also reduces as the number of clusters increases.

The accuracy of 88.9% and 89.3% were obtained at the same number of clusters K=3.Hence, the proposed algorithm was able to achieve an accuracy of 89.3%, as against 88.9% of the existing method, thereby improving it by 1.12%.

TABLE I : SUMMARY OF RESULTS ON ACCURACY FOR EXISTING DATA CLUSTERING ALGORITHM

NO OF CLUSTERS (K)	3	4	5	6	7	8	9	10
ACCURACY	0.889	0.693	0.666	0.666	0.66	0.66	0.6	0.6

TABLE II : SUMMARY OF RESULTS ON ACCURACY FOR PROPOSED DATA CLUSTERING ALGORITHM

NO OF CLUSTERS (K)	3	4	5	6	7	8	9	10
ACCURACY	0.893	0.693	0.666	0.666	0.66	0.66	0.6	0.6

C. Adjusted rand index versus number of clusters (Iris datasets) using Euclidean distances

Fig 2 shows the adjusted rand Index against number of clusters when number of clusters was varied from n=1 to n=0 for both the existing and proposed method. The adjusted rand index value varies from 0 to 1 and is best at 1. The graph shows that at n=2 to 5. The adjusted rand indices are equal and do not vary, but at K=6 clusters the existing algorithm decreases and shows the characteristics of local optimum, while the proposed algorithm is stable and decreases but at a lower rate than the existing method. Also, at K=10 there is a small difference in the adjusted rand index method with the existing method. The existing method achieves an adjusted rand index of 53%, as compared to the proposed which achieves an adjusted rand index of 63.7%.On the average the proposed algorithm performed better than the existing method.

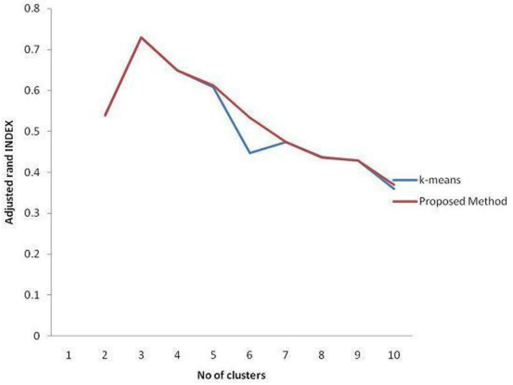

Fig. 2: Adjusted Rand Index for K-means and Proposed Clustering Algorithm using the Iris Dataset under Euclidean Distances

D. Rand index versus number of clusters (multivariate normal distribution datasets) (Euclidean distances)

Fig 3 shows the rand index against a number of clusters using the Euclidean distances when size/number of clusters was varied from n=2 to 10.The graph shows that in every number of cluster setting, the proposed method was higher than the existing method except at K=6,7,8,9. As the number of packets increased, the two schemes increased in rand index value until a peak of 0.62929 at K=9 for the proposed method and 0.63091 for K=9 for the existing algorithm. The rand index of 0.53939 and 0.55192 were obtained at the same number of clusters for existing and proposed methods at K=2. Hence, the clustering results of proposed algorithm on clustering multivariate normal distribution datasets, using Euclidean distances, is of better quality than the clustering with the existing algorithm.

Figure

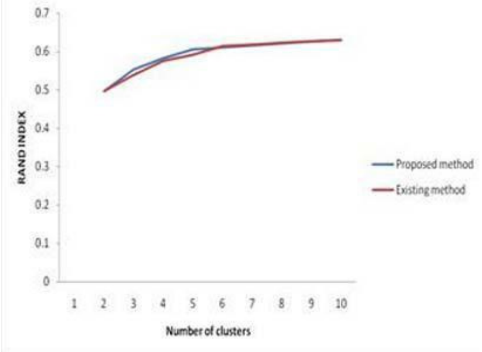

Fig. 3 : Rand Index for K-means and Proposed Clustering Algorithm using the Multivariate Normal Distribution Dataset under Euclidean Distances

E. Time in clustering the iris dataset at fixed number of clusters

Table III shows the time spent in clustering Iris dataset using the proposed method and the existing method. The simulation was done for 10 iterations and all the times were recorded at the end of each iteration. From table III, the existing algorithm was faster at some points, i.e. at iteration =2, 3, 4, 5, 6, 7 and proposed faster at iteration = 1, 2, 8, 9, 10. The average value for K-means is 0.0451s and 0.0439s from the proposed method, showing 2.7% decrease in speed. The range and mean value are also tabulated in Table III. At K=3 while clustering iris dataset for a fixed number of cluster K=3, the response time is faster for the proposed method than for the existing method.

TABLE III: ALGORITHM COMPARISON FOR IRIS DATASET AT K=3

TRIAL NO.	K-MEANS (SEC)	PROPOSED METHOD(SEC)
1	0.362	0.320
2	0.011	0.020
3	0.007	0.017
4	0.005	0.013
5	0.008	0.018
6	0.006	0.01
7	0.007	0.011
8	0.009	0.007

9	0.023	0.011
10	0.013	0.012

TABLE IV : COMPARATIVE RESULTS

ALGORITHM	AVERAGE VALUE(SEC)	RANGE(LOW-HIGH)(SEC)	MEAN VALUE (SEC)
K-MEANS	0.0451	(0.005-0.365)	0.0451
PROPOSED METHOD	0.0439	(0.010-0.320)	0.0439

4. Conclusion

The results from the performance evaluation showed that the proposed data clustering algorithm can be incorporated within a web based search engine to provide better performance. The response time results show that the time in retrieving documents will be reduced, while the accuracy and adjusted rand index show that the users queries will return consistent results that will meet their search criteria as compared to using the existing web search engines. The proposed model was able to reduce the problem of speed while increasing accuracy to some considerable level over the existing approach. Therefore, it will be suitable for web search engine designers to incorporate this model in an existing web based search engine so that web users can retrieve their documents at a faster rate and with higher accuracy.

References

[1] A. Jain, and M. Murty, "Data Clustering: A Review." ACM Computing Surveys, vol. 31, pp. 264-323. 1999.

[2] A. Moth‟d Belal, "A New Algorithm for Cluster Initialization". Proceedings of World Academy of Science, Engineering and Technology. Vol. 4, pp. 74-76. 2005.

[3] C.C. Hsu, and, Y.C. Chen," Mining of Fixed Data with Application of Catalogue Marketing". Expert Systems with Application, vol. 32, pp.12-23. 2007.

[4] C.M. Benjamin, K.W. Fung, and E. Martin, "Encyclopaedia of Data Warehousing and Mining". Montclair State University, USA. 2006.

[5] D. Boley, M. Gini, R. Cross, E. Hong(Sam),K. Hastings,G. Karypis, V. Kumar, B. Mobasher, and J. Moore," Partitioning-based Clustering for Web Document Categorization". Decision Support Systems.Vol. 27, pp. 329-341.1999.

[6] E.Z. Oren," Clustering Web Documents: A Phrase-Based Method for Grouping Search Engine Results". Ph.D. Thesis, University of Washington.1999.

[7] F. Glenn, "A Comprehensive Overview of Basic Clustering Algorithms" Technical Report, University [of Winsconsin, Madison.2001.

[8] O. Zamir, O. Etzioni, "Web Document Clustering" Department of Computer Science and Engineering, University of Washington, Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 46-54.1998.

[9] O.M. San, V.N. Huynh, and Y. Nakamori, "An Alternative Extension of the K-Means Algorithm for Clustering", International Journal of Applied Mathematics and Computer Science, vol. 14, pp.241-247.2004.

[10] P. Hae-Sang, L. Jong Seok, and J. Chen Hyuck, "A K-means like algorithm for k-medoid clustering and its performance", department of industrial and management engineering, Iostech San 31, Hyopa-Clong, 780-784.2006.

[11] S. Sambasivam and N. Theodosopoulos." Advanced Data Clustering Methods of Mining Web Documents". Issues in Informing Science and Information Technology. Vol. 3, pp. 563-579.

[12] Y.M. Cheung, "K*-means: A New Generalized k-means Clustering Algorithm". Pattern Recognition Letters, vol. 24, pp. 2883-2893.2003.

[13] Z. Huang, "Extensions to the k-Means Algorithm for Clustering Large Data Sets with Categorical Values". Data Mining and Knowledge Discovery, vol. 2, pp. 283-304.1998.