MLK-Means - A Hybrid Machine Learning based K-Means Clustering Algorithms for Document Clustering

*P. PERUMAL ,
Department of CSE
Sri Ramakrishna Engineering College
Coimbatore, 641 022.
INDIA
perumalsrec@gmail.com

R. NEDUNCHEZHIAN
Department of IT, Sri Ramakrishna Engineering College
Coimbatore,641 022.
INDIA
rajuchezhian@yahoo.co.uk

ABSTRACT: - Document clustering is useful in many information retrieval tasks such as document browsing, organization and viewing of retrieval results. They are very much and currently the subject of significant global research. Generative models based on the multivariate Bernoulli and multinomial distributions have been widely used for text classification. In this work, address a new hybrid algorithm called MLK-Means for clustering TMG format document data, in which, the normal Euclidean distance based metric of the k-mean process is replaced by a machine learning technique. The results of the proposed algorithm were compared with the probabilistic model namely, von Mises-Fisher model-based clustering (vMF-based k-means) and the standard k-mean with L-2 normalized data method. In this proposed work, the MLK-Means algorithm has been implemented and its performance is compared with other algorithms mentioned above. The improvements in the proposed algorithm are more significant and comparable.

Key Words: Document Clustering; Model Based Clustering; Term Document Matrix; Text to Matrix Generator (TMG); k-means; Machine Learning; Bernoulli; Multinomial and von Mises-Fisher Clustering.

1. Introduction

Document Clustering

Document clustering is a kind of text data mining and organization technique that automatically groups related documents into clusters. Document clustering also referred to as text clustering is closely related to the concept of data clustering. Document clustering is a more specific technique for unsupervised document organization, automatic topic extraction and fast information retrieval or filtering [14, 15, 16]. Document clustering is a specialized data clustering problem, where the objects are in the form of documents. Document clustering aims to discover natural grouping among documents in such a way that document with in a cluster are similar (high intra cluster similarity) to one another and are dissimilar to documents in other clusters (low inter cluster similarity).

Document clustering algorithms mainly uses features like words, phrases, and sequences from the documents to perform clustering. Document clustering has been studied intensively because of its wide applicability in areas such as web mining and information retrieval. Document clustering has long been important problem in text processing systems. The goal of these document clustering is

systems is to automatically discover, in the absence of metadata or a pre-existing categorization, sensible topical organizations of the document.

1.2. Problem Specification

Generally, unsupervised algorithms were used in document clustering. There are different variants of k-means clustering algorithms derived exclusively for document clustering application. Recently, the spherical k-means algorithm, which has desirable properties for text clustering, has been shown to be a special case of a generative model based on a mixture of von Mises-Fisher (vMF) distributions. The standard k-mean clustering algorithm is having some weakness on clustering TMG format document data and document in general. In our previous evaluation work [16], we showed the results of k-means clustering algorithm on document clustering application. Often, the normal Euclidean distance based metric of the k-mean process which is used for clustering the word count based feature vectors leads to inaccurate clusters. So in this work, we propose a new model of k-mean algorithm in which the traditional Euclidean distance based clustering process is replaced by a SVM based machine learning (ML) technique. In addition to that, to get more performance in terms of speed, we will incorporate the Principal Component Analysis (PCA) in the k-mean process. Even though SVM is a supervised ML technique, in this work, we are deriving an unsupervised clustering algorithm by using this supervised machine learning technique.

1.3. Machine Learning

Machine learning (ML) evolves from artificial intelligence (AI). It combines AI heuristics with advanced statistical analysis. ML is contributing to a major category of data mining algorithms. AI was not commercially successful and is mostly used only for research. But ML is applied very widely as its entry price itself was lower than AI. It is capable of taking advantage of the improving price/performance ratios of computer. ML learns characteristics of data using fundamental statistics and uses AI heuristics to achieve its goal [22].

2. Material and Methods

2.1. Probabilistic K-Mean Clustering Algorithms

Model-based Partitional Clustering

The model-based k-means (mk-means) algorithm is a generalization of the standard k-means algorithm, with the cluster centroid vectors being replaced by probabilistic model. Let $X = \{x_1, ..., x_N\}$ be the set of data objects and $\wedge = \{\lambda_1, ..., \lambda_K\}$ the set of cluster models. The mk – means algorithm locally maximizes the log – likelihood objective function $\log P(X \mid \wedge) = \sum_{x \varepsilon X} \log p(x \mid \lambda_{y(x)})$,

where $y(x) = \arg\max_y \log p(x \mid \lambda_y)$ is the cluster identity of object x [14,15,16,23].

The traditional vector space representation is used for text documents, i.e., each document is represented as a high dimensional vector of "word"2 counts in the document. The dimensionality equals the number of words in the vocabulary used. Next, we briefly introduce the three generative models studied in our experiments.

A. Multivariate Bernoulli Model

In a multivariate Bernoulli model [9], a document is represented as a binary vector over the space of words. The l-th dimension of a document vector x is denoted by x(l), and is either 0 or 1, indicating whether word w_l occurs or not in the document. The the number of occurrences is not considered, i.e., the word frequency information is lost.

With naïve Bayes assumption, the probability of a document x in cluster y is

$$P(x \mid \lambda_y) = \prod_l Py(w_l)^{x(l)} (1 - Py(w_l))^{1 - x(l)}$$

where $\lambda y = \{P_y(w_l)\}, P_y(w_l)$ is the probability of word w_l being present in cluster y, and $(1 - P_y(w_l))$ the probability of word wl not being present in cluster y. To avoid zero probabilities when estimating $P_y(w_l)$, one can employ the solution as [9]

$$P_y(w_l) = \frac{1 + \sum_x P(y \mid x, \wedge) x(l)}{2 + \sum_x P(y \mid x, \wedge)} \qquad (1)$$

where $P(y \mid x, \wedge)$ is the posterior probability of cluster y[25].

B. Multinomial Model

Based on the naive Bayes assumption, a multinomial model for cluster y represents a document x by a multinomial distribution of the words in the document (vocabulary)

$$P(x \mid \lambda_y) = \prod_l P_y(l)^{x(l)}$$

where $x(l)$ is the l-th dimension of document vector x, indicating the number of occurrences of the l-th word in document x. To accommodate documents of different lengths, we use a normalized (log)-likelihood measure

$$\log \tilde{P}(x \mid \lambda_y) = \frac{1}{|x|} \log P(x \mid \lambda_y) \qquad (2)$$

where $|x| = \sum_l x(l)$ is the size of the length of document x. The $Py(l)$'s are the multinomial model parameters and represent the word distribution in cluster y. They are subject to the constraint $\sum_l P_y(l) = 1$ and can be estimated by counting the number of documents in each cluster and the number of word occurrences in all documents in the cluster (Nigam, 2001). With Laplacian smoothing, i.e., with model prior $P(\lambda_y) = C.\prod_l P_y(l)$, the parameter estimation of multinomial models amounts to

$$P_y(l) = \frac{1 + \sum_x P(y \mid x, \wedge) x(l)}{\sum_l (1 + \sum_x P(y \mid x, \wedge) x(i))} = \frac{1 + \sum_x P(y \mid x, \wedge) x(l)}{|V| + \sum_l \sum_x P(y \mid x, \wedge) x(i)} \qquad (3)$$

where |V| is the size of the word vocabulary, i.e., the dimensionality of document vectors [23].

C. von Mises-Fisher Model

The von Mises-Fisher distribution is the analogue of the Gaussian distribution for directional data in the sense that it is the unique distribution of L2-normalized data that maximizes the entropy given the first and second moments of the distribution (Mardia, 1975). It has recently been shown that the spherical k-means algorithm that uses the cosine similarity metric (to measure the closeness of a data point to its cluster's centroid) can be derived from a generative model based on the vMF distribution under certain restrictive conditions (Banerjee & Ghosh, 2002; Banerjee et al., 2003). The vMF distribution for cluster j can be written as

$$P(d_i \mid \lambda_j) = \frac{1}{Z(k_j)} \exp(kj \frac{d_i^T \mu j}{\| \mu_j \|}), \qquad (4)$$

where d_i is a normalized document vector and the Bessel function $Z(k_j)$ is a normalization term. The parameter k measures the directional variance and the higher it is, the more peaked the distribution is. For the vMF-based k-means algorithm, we assume k is the same for all clusters, i.e., $k_j = k, \forall_j$. This results in the spherical k-means (Dhillon & Modha, 2001; Dhillon et al., 2001). The model estimation in this case simply amounts to $\mu_i = \frac{1}{n_j} \sum_{i:y_i=j} d_i$, where n_j is the number of documents in cluster j [23].

Here, the proposed work only concentrates on von Mises – Fisher model and improve the results significantly.

2.2.k-means and the Proposed MLK-Means clustering algorithm

One of the most popular heuristics for solving the k-means problem is based on a simple iterative scheme for finding a locally optimal solution. This algorithm is often called the k-means algorithm. There are a number of variants to this algorithm, so to clarify which version we are using, we will refer to it as the naïve k-means algorithm as it is much simpler compared to the other algorithms described here. This algorithm is also referred to as the Lloyd's algorithm [26].

The naive k-means algorithm partitions the dataset into 'k' subsets such that all records, from now on referred to as points, in a given subset "belong" to the same center. Also the points in a given subset are closer to that center than to any other center. The partitioning of the space can be compared to that of Voronoi partitioning except that in Voronoi partitioning one partitions the *space* based on

distance and here we partition the *points* based on distance [12, 21]. The algorithm keeps track of the centroids of the subsets, and proceeds in simple iterations. The initial partitioning is randomly generated, that is, we randomly initialize the centroids to some points in the region of the space. In each iteration step, a new set of centroids is generated using the existing set of centroids following two very simple steps. Let us denote the set of centroids after the i^{th} iteration by $C^{(i)}$. The following operations are performed in the steps [12, 21, 26] :

- Partition the points based on the centroids C(i), that is, find the centroids to which each of the points in the dataset belongs. The points are partitioned based on the Euclidean distance from the centroids.

- Set a new centroid $c(i+1) \in C(i+1)$ to be the mean of all the points that are closest to $c(i) \in C(i)$ The new location of the centroid in a particular partition is referred to as the new location of the old centroid.

The algorithm is said to have converged when recomputing the partitions does not result in a change in the partitioning. In the terminology that we are using, the algorithm has converged completely when $C^{(i)}$ and $C^{(i-1)}$ are identical. For configurations where no point is equidistant to more than one center, the above convergence condition can always be reached. This convergence property along with its simplicity adds to the attractiveness of the k-means algorithm. The naïve k-means needs to perform a large number of "nearest-neighbour" queries for the points in the dataset. If the data is 'd' dimensional and there are 'N' points in the dataset, the cost of a single iteration is O (kdN). As one would have to run several iterations, it is generally not feasible to run the naïve k-means algorithm for large number of points. Sometimes the convergence of the centroids (i.e. $C^{(i)}$ and $C^{(i+1)}$ being identical) takes several iterations. Also in the last several iterations, the centroids move very little. As running the expensive iterations so many more times might not be efficient, we need a measure of convergence of the centroids so that we stop the iterations when the convergence criterion is met. Distortion is the most widely accepted measure.

Clustering error measures the same criterion and is sometimes used instead of distortion. In fact k-means algorithm is designed to optimize distortion. Placing the cluster center at the mean of all the points minimizes the distortion for the points in the cluster. Also when another cluster center is closer to a point than its current cluster center, moving the cluster from its current cluster to the other can reduce the distortion further. The above two steps are precisely the steps done by the k-means cluster. Thus k-means reduces distortion in every step locally. The k-Means algorithm terminates at a solution that is locally optimal for the distortion function. Hence, a natural choice as a convergence criterion is distortion. Among other measures of convergence used by other researchers, we can measure the sum of Euclidean distance of the new centroids from the old centroids. Here, we use clustering error/distortion as the convergence criterion for all variants of k-means algorithm [12, 21, 26].

Definition: *Clustering error* is the sum of the squared Euclidean distances from points to the centers of the partitions to which they belong. Mathematically, given a clustering ϕ, we denote by $\phi(x)$ the centroid this clustering associate with an arbitrary point x (so for k-means, $\phi(x)$ is simply the center closest to x). We then define a measure of quality for ϕ:

$$distortion_{\phi} = \frac{1}{N}\sum_{x} \left|x - \phi(x)\right|^2 \quad (5)$$

where |a| is used to denote the norm of a vector 'a'. The lesser the difference in distortion over successive iterations, the more the centroids have converged. Distortion is therefore used as a measure of goodness of the partitioning. In spite of its simplicity, k-means often converges to local optima. The quality of the solution obtained depends heavily on the initial set of centroids, which is the only non-deterministic step in the algorithm. Note that although the starting centers can be selected arbitrarily, k-means is fully deterministic, given the starting centers. A bad choice of initial centers can have a great impact on both performance and distortion. Also a good choice of initial centroids would reduce the number of iterations that are required for the solution to converge.

Many algorithms have tried to improve the quality of the k-means solution by suggesting different ways of sampling the initial centers, but none has been able to avoid the problem of the solution converging to a local optimum. For example, gives a discussion on how to choose the initial centers, other techniques using stochastic global optimizations methods (e.g. simulated annealing, genetic algorithms), have also been developed. None of these algorithms is widely accepted [12, 21, 26].

Problems with K-Means

- The algorithm is simple and has nice convergence but there are number of problems with this
- Selection of value of K is itself an issue and sometimes it's hard to predict before hand the number of clusters that would be there in the data.
- Experiments have shown that outliers can be a problem and can force the algorithm to identify false clusters.
- Experiments have shown that performance of algorithms degrade in higher dimensions and can be off by factor of 5 from optimum.

A. The Standard K-means Algorithm

Inputs : X= $\{x_1,...,x_k\}$ (the document vectors to be clustered)
n (the number of clusters)

Outputs: C= $\{c_1,\ldots, c_n\}$ (the Cluster Centroids)

m: X $\rightarrow$ {1..n} (the cluster membership)

Procedure k-means {

Randomly initialize C

For each $x_i \in$ X {

$m(x_i)$=argmin distance(x_i, c_j)
$j \in \{1..n\}$

}

While m has changed {

For each $i \in \{1..n\}$ {

Recomputed Ci as the centroid of {x|m(x) = i}

}

For each $x_i \in$ X {

$m(x_I)$ = argmin distance(x_i , c_j)
$j \in \{1..n\}$

}

}

}

B. The Proposed MLK-Means Algorithm

Inputs : X= $\{x_1,...,x_k\}$ (the document vectors to be clustered)

n (the number of clusters)

Outputs : C=$\{c_1,\ldots, c_n\}$ (the Cluster Centroids)

m: X $\rightarrow$ {1..n} (the cluster membership)

Procedure MLK-means

{

1. Randomly initialize C
2. (λi , u_i) $\leftarrow$ pca(C) where λi – eigenvalues and u_i - eigenvectors
3. u_iC is the will be the dimensionality reduced representation of C

4. Create a network to learn the dimensionality reduced inputs and map it to the original output class of the documents

5. Learn the document space using the Centroids $u_i C$ and their corresponding class labels {1..n}

6. For each $x_i \in X$, calculate $u_i xi$ and predict the membership m ($u_i x_i$) of each calculate $u_i x_i$ using the trained network where $i \in \{1..k\}$

7. While m has changed {

For each $i \in \{1...n\}$ {

Recomputed Ci as the centroid of {x|m(x) = i}
$(\lambda i , u_i) \leftarrow pca(C)$
Calculate new $u_i C$
}

Again, for Each $x_i \in X$, calculate $u_i xi$ and predict the membership m(x_i) of each calculate $u_i x_i$ using the trained network where $i \in \{1..k\}$

}

}

Instead of using all the attributed, the proposed algorithm uses principal components of the attributes in the clustering process. Generally, while using a dimensionality reduction technique such as PCA, the entire data set is transformed in to the dimensionality reduced form. But that operation is not cost effective. Particularly, the document clustering is nothing but dealing with a very high dimensional data. So, applying PCA to the whole data set in one step will consume lot of time. So, in the proposed MLK-mean algorithm, first the principal components of the initially guessed centroids were only calculated. Since there were only C records corresponding to the C centers, the PCA process will not consume much time (the PCA on centroids will give corresponding Eigen values and eigenvectors).Thus, after fining the PCA of assumed centroids, a SVM is trained with the principal components of those centroids. The whole dataset is reduced to a lower dimension using the eigenvectors of the centroids calculated in the previous step. This operation will consume only negligible time since there is only a multiplication operation involved in it.

The dimensionality reduced form of data is now classified using the trained network. Re-compute the centroids using the newly classified data (using all the attributes corresponding to the class membership – average of the attributes of the each class). Now this is the newly calculated centroids. Now we can repeat the whole procedure again from the beginning to find more optimum centroids and class members. In k-mean, generally Euclidean distance as used in distance calculation during finding the membership of a record. But, if we use that kind of distance calculation in a document classification problem which is dealing with a huge dimensional data set, then this kind of distance metrics will not lead to accurate result. Several previous works highlights this weakness of k-mean clustering. So, in proposed MLK-mean, the traditional distance based membership calculation function is replaced with SVM and yields a new kind of machine learning based k-mean.

C. Principle Component Analysis (PCA)

A data set $\mathbf{x}_i$, $(i = 1,\ldots,n)$ is summarized as a linear combination of orthonormal vectors (called principal components) [27]:

$$\boldsymbol{f}(\mathbf{x},\mathbf{V}) = \mathbf{u} + (\mathbf{x}\mathbf{V})\mathbf{V}^T \qquad (6)$$

where $\boldsymbol{f}(\mathbf{x},\mathbf{V})$ is a vector valued function, $\mathbf{u}$ is the mean of the data $\{\mathbf{x}_i\}$, and $\mathbf{V}$ is an $d \times m$ matrix with orthonormal columns. The mapping $\mathbf{z}_i = \mathbf{x}_i\mathbf{V}$ provides a low-dimensional projection of the vectors $\mathbf{x}_i$ if $m < d$.

The PCA estimates the projection matrix $\mathbf{V}$ minimizing

$$R_{emp}(\mathbf{x},\mathbf{V}) = \frac{1}{n}\sum_{i=1}^{n}\left\|\mathbf{x}_i - \boldsymbol{f}(\mathbf{x}_i,\mathbf{V})\right\|^2 . \quad (7)$$

The first principal component is an axis in the direction of maximum variance. Consequently, Principle Component Analysis (PCA) replaces the original variables of a data set with a smaller number of uncorrelated variables called the *principle components*. If the original data set of dimension D contains highly correlated variables, then there is an effective dimensionality, d < D, that explains most of the data. The presence of only a few components of d makes it easier to label each dimension with an intuitive meaning. Furthermore, it is more efficient to operate on fewer variables in subsequent analysis [19].

3. Results and Discussion

To evaluate the algorithms, a suitable and standard data set is needed. We decided to use some of the same datasets which were originally used in a previous reference work [23]. The datasets were originally from TREC collections (http://trec.nist.gov). Datasets tr11, tr23, tr41, and tr45 were originally derived from TREC-5, TREC-6, and TREC-7 collections. (NIST Text REtrieval Conferences - TREC). The dataset la12 was also originally derived from TREC. We used TMG format of these datasets which is available in several internet resources. We selected these data sets for evaluation because of their standard.

3.1. Metrics Considered for Evaluation

Validating clustering algorithms and comparing performance of different algorithms is complex because it is difficult to find an objective measure of quality of clusters. In order to compare results against external criteria, a measure of agreement is needed. Since we assume that each record is assigned to only one class in the external criterion and to only one cluster, measures of agreement between two partitions can be used [14, 15, 16]. An important aspect of cluster analysis is the evaluation of clustering results. Halkidi, M. et al. (2001) made a comprehensive review of clustering validity measures available in the literature and classified them into three categories. In this section we briefly review the commonly used document clustering evaluation measures and the evaluation of search results clustering in the literature [28].

The first is the external evaluation method, which evaluates the results of clustering algorithm based on a pre-classified document set. There are several ways of comparing the clusters with the pre-defined classes: Rand Index, purity and mutual information.

3.1.1. Rand Index

The Rand index or Rand measure is a commonly used technique for measure of such similarity between two data clusters. Given a set of n objects S = {O1, ..., On} and two data clusters of S which we want to compare: X = {x1, ..., xR} and Y = {y1, ..., yS} where the different partitions of X and Y are disjoint and their union is equal to S; we can compute the following values [18]:

- a is the number of elements in S that are in the same partition in X and in the same partition in Y,
- b is the number of elements in S that are not in the same partition in X and not in the same partition in Y,
- c is the number of elements in S that are in the same partition in X and not in the same partition in Y,
- d is the number of elements in S that are not in the same partition in X but are in the same partition in Y.

Intuitively, one can think of a + b as the number of agreements between X and Y and c + d the number of disagreements between X and Y. The rand index, R, then becomes,

$$R = \frac{a+b}{a+b+c+d} = \frac{a+b}{\binom{n}{2}} \quad (8)$$

The rand index has a value between 0 and 1 with 0 indicating that the two data clusters do not agree on any pair of points and 1 indicating that the data clusters are exactly the same [14, 15, 16, 18].

3.1.2. Purity

Purity can be computed by assigning class labels for each cluster by majority voting, then for a single cluster calculating the ratio of the correctly labelled

documents to the total number of documents in the cluster. Let there be k clusters (the k in k-means) of the dataset D and size of cluster Cj be |Cj |. Let |Cj |class=i denote number of items of class i assigned to cluster j [13]. Purity of this cluster is given by

$$purity(C_j) = \frac{1}{|C_j|} \max_i (|C_j|_{class=i}) \quad (9)$$

The overall purity of a clustering solution could be expressed as a weighted sum of individual cluster purities.

$$purity = \sum_{j=1}^{k} \frac{|C_j|}{|\mathcal{D}|} purity(C_j) \quad (10)$$

In general, larger value of purity means better the solution [15, 16].

3.1.3. Mutual Information

Here, use the mutual information between an element (document) and its features (terms). In this algorithm, for each element *e*, construct a frequency count vector $C(e) = (c_{e1}, c_{e2}, \ldots, c_{em})$, where *m* is the total number of features and c_{ef} is the frequency count of feature *f* occurring in element *e*. In document clustering, *e* is a document and c_{ef} is the term frequency of *f* in *e [13]*. Construct a mutual information vector $MI(e) = (mi_{e1}, mi_{e2}, \ldots, mi_{em})$, where mi_{ef} is the mutual information between element *e* and feature *f*, which is defined as:

$$mi_{ef} = log \frac{\frac{c_{ef}}{N}}{\frac{\sum_i c_{if}}{N} \times \frac{\sum_j c_{ej}}{N}} \quad (11)$$

where $N = \sum_i \sum_j c_{ij}$ is the total frequency count of all features of all elements. Compute the similarity between two elements e_i and e_j using the *cosine coefficient* of their mutual information vectors [29]:

$$sim(e_i, e_j) = \frac{\sum_f mi_{e_i f} \times mi_{e_j f}}{\sqrt{\sum_f mi_{e_i f}^2 \times \sum_f mi_{e_j f}^2}} \quad (12)$$

3.2. Performance in Terms of CPU time

In the following table we present the outputs of time study made on a Windows XP laptop equipped with Intel core 2 duo CPU at 2GHz and 2GB RAM. The Matlab implementations of the algorithms were used for evaluation. As shown in the following tables and graphs, the performance in terms of CPU time was very good in the proposed MLK-means clustering algorithm.

Table 1: Accuracy in Terms of CPU time

Data Set Used and its size (rows x Columns)	Time Taken for Clustering (Average of Three runs)		
	von Mises-Fisher based k-means	k-means with L2-normalized data	MLK-means
Tr11 (414 x 6424)	0.4530	1.219	0.2660
Tr12 (313 x 5799)	0.2500	0.828	0.2340
Tr23 (204 x 5831)	0.1720	0.531	0.1720
Tr31 (927 x 10127)	1.0470	2.094	0.3280
Tr41 (690 x 8261)	0.3593	1.906	0.2810
Tr45.mat (690 x 8261)	0.4210	1.75	0.2810
La2.mat (3075 x 31472)	1.2920	4.5	0.4690
La12.mat (6279 x 31472)	2.9060	8.718	0.8590
Avg	0.862538	2.69325	0.3613

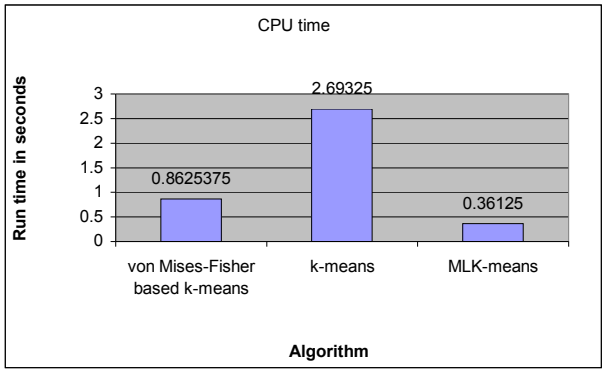

Fig. 1. Algorithm vs CPU time

3.3. The performance of Clustering with Different Datasets

3.3.1. Clustering Accuracy in Terms of Rand Index

Table 2: Accuracy in Terms of Rand Index with Different Data Sets

Data Set Used and its size (rows x Columns)	Clustering Accuracy in Terms of Rand Index (Average of Three runs)		
	von Mises-Fisher based k-means	k-means with L2-normalized data	MLK-means
Tr11 (414 x 6424)	0.8387	0.8484	0.8415
Tr12 (313 x 5799)	0.8308	0.8314	0.8248
Tr23 (204 x 5831)	0.6932	0.6924	0.7532
Tr31 (927 x 10127)	0.8184	0.7859	0.8162
Tr41 (690 x 8261)	0.8847	0.8743	0.8654
Tr45.mat (690 x 8261)	0.8821	0.8741	0.8904
La2.mat (3075 x 31472)	0.8161	0.7742	0.8086
La12.mat (6279 x 31472)	0.8248	0.7869	0.8397
Avg	0.8236	0.8084	0.8300

As shown in the above table and the following figure, the performance of proposed MLK-means clustering was very good and little bit higher than that of the probabilistic model - von Mises-Fisher based k-means clustering.

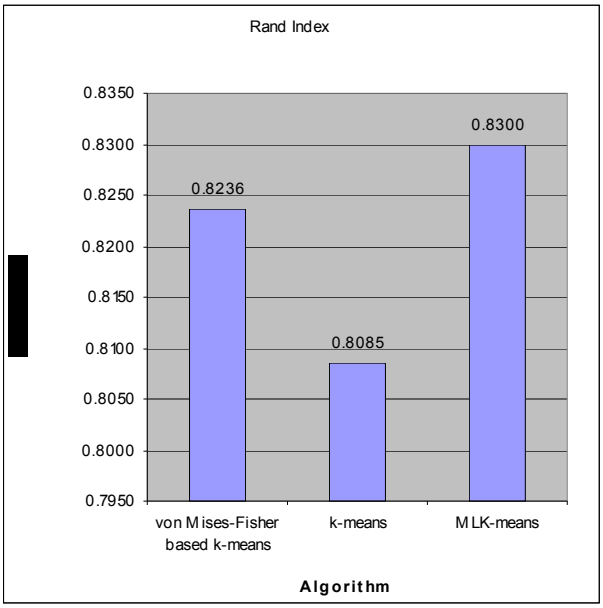

Fig. 2. Rand Index - Comparison Graph

3.3.2. Clustering Accuracy in Terms of Mutual Information

As shown in the following tables and graphs, the performance in terms of mutual information measure was also good in the case of proposed MLK-means clustering algorithm.

Table 3: Accuracy in Terms of Mutual Information Measure with Different Data Sets

Data Set Used and its size (rows x Columns)	Clustering Accuracy in Terms of Mutual Information (Average of Three runs)		
	von Mises-Fisher based	k-means with L2-normalized data	MLK-means
Tr11 414 x 6424	0.2150	0.5727	0.4121
Tr12 313 x 5799	0.2133	0.4038	0.5024
Tr23 204 x 5831	0.2015	0.3199	0.4468
Tr31 927 x 10127	0.2102	0.5163	0.4703
Tr41 690 x 8261	0.2346	0.6365	0.5836
Tr45.mat 690 x 8261	0.4825	0.6105	0.6564
La2.mat 3075 x 31472	0.4841	0.3717	0.3810
La12.mat 6279 x 31472	0.5008	0.4741	0.4718
Avg	0.3177	0.4882	0.4906

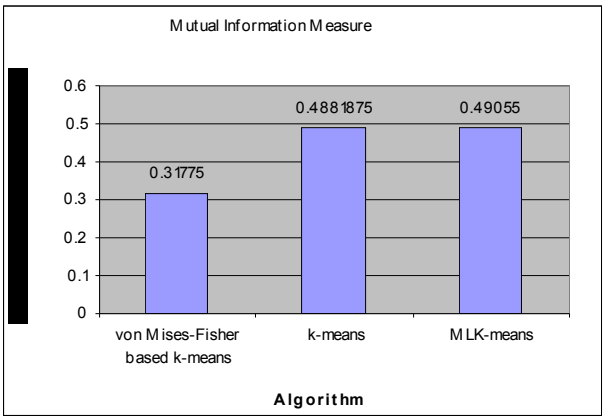

Fig. 3. Mutual Information - Comparison Graph

As shown in the following tables and graphs, the performance in terms of purity measure was also good in the case of proposed MLK-means clustering algorithm.

Table 4: Accuracy in Terms of Purity with Different Data Sets

Data Set Used and its size (rows x Columns)	Clustering Accuracy in Terms of Purity Measure (Average of Three runs)		
	von Mises-Fisher based k-means	k-means with L2-normalized data	MLK-means
Tr11 414 x 6424	0.2150	0.7352	0.5295
Tr12 313 x 5799	0.2133	0.4716	0.6772
Tr23 204 x 5831	0.2015	0.6783	0.7171
Tr31 927 x 10127	0.2102	0.7553	0.6799
Tr41 690 x 8261	0.2346	0.7942	0.7548

Tr45.mat 690 x 8261	0.6804	0.7495	0.8041
La2.mat 3075 x 31472	0.7161	0.6055	0.5929
La12.mat 6279 x 31472	0.7051	0.7321	0.6675
Avg	0.3970	0.6902	0.6779

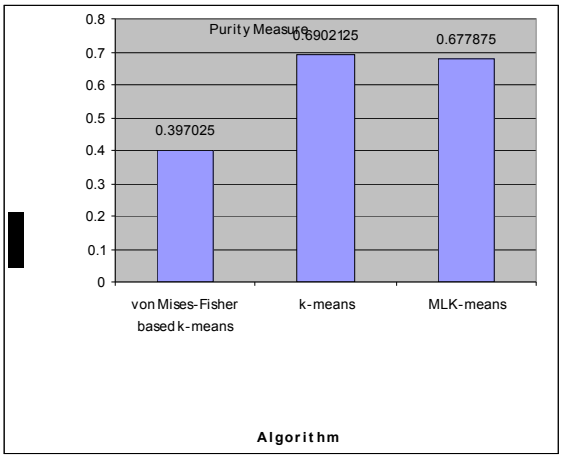

Fig. 4. Purity – Comparison Graph

4. conclusion and Scope For Further Enhancements

In this paper, we have proposed improved k-mean based unsupervised clustering model using supervised machine learning technique. The proposed MLK-means clustering algorithm has been successfully implemented and evaluated with suitable datasets. The arrived results were more significant and comparable. We used different kinds of metrics to evaluate the performance of the proposed MLK-means clustering algorithm. If we compare performance of these three algorithms in terms of CPU time, Rand Index, Mutual Information, and Purity Measure, it was obvious that results of the proposed MLK-means algorithm is significantly better than all the other previous methods. As in the case of standard k-means, this implementation of the MLK-means algorithm also starts with some randomly initiated centriods. Future work may address the ways to find better methods to start with optimum centroids to achieve better results. For that we may use any fast clustering method to estimate the initial centroids and then apply the proposed method with those centroids to achieve improved results. Our future work will address these new ideas.

5. Acknowledgement

We thank our Director, Principal and the management of Sri Ramakrishna Engineering College for providing lab facility to implement this work.

References:

[1] Aas, K., Eikvil, L.Text Categorisation: A Survey. *Technical report*, Norwegian Computing Center, P.B.114 Blindren, N-0314 Oslo, Norway, June 1999.

[2] Can, Fazli; Ozkarahan, Esen A. Similarity and Stability Analysis of the Two Partitioning Type Clustering Algorithms. *Journal of the American Society for Information Science*, 36(1):3-14, 1985.

[3] Jain, A.K. and Dubes, R.C. *Algorithms for Clustering Data*, Prentice-Hall advanced reference series. Prentice- Hall Inc., Upper Saddle River, NJ, 1988.

[4] Jain, A.K., Murty, M.N., and Flynn, P.J. Data Clustering – Survey. *ACM Computing Surveys*, Vol.31, No.3, pp. 264 – 323, 1999.

[5] Han, J. W. and Kamber, M. *Data Mining: Concepts and Techniques*, 2nd edition, Morgan Kaufmann Publishers, March 2006.

[6] Hussein, N. A Fast Greedy K-Means Algorithm. Master's Thesis, University of Amsterdam, Netherlands, 2002.

[7] J.L.Neto, A.D.Santos, C.A.A. Kaestner, and A.A. Freitas. Document Clustering and Text Summarization. *4th International Conference On Practical Applications of Knowledge Discovery and Data Ming*, London, 2000.

[8] Maarek. S, Ronald Fagin, Israel Z. Ben-Shaul, Dan Pelleg. Ephemeral Document Clustering for Web Applications, *IBM Research Report RJ,* 10186, April, 2000.

[9] McCallum and K. Nigam. A comparison of event models for naive Bayes text Classification. *AAAI Workshop on Learning for Text Categorization*, 1998.

[10] Mei-Ling Shyu, Shu-Ching Chen et. al. Affinity-Based Probabilistic Reasoning and Document Clustering on the WWW,

COMPSAC, pp.149-154, 2000.

[11] Michael Steinbach, George Karypis, Vipin Kumar. A Comparison of Document Clustering Techniques. *Proc. Text Mining Workshop, KDD 2000*, 2000.

[12] Mumtaz, K. et al. A Novel Density based improved k-meansClustering Algorithm – Dbkmeans. *International Journal on Computer Science and Engineering*, Vol. 02, No. 02, pp. 213-218, 2010.

[13] Patrick Pantel, and Dekang Lin. Efficiently Clustering Documents with Committees. In: *PRICAI*, Vol. 2417, Springer, p. 424-433, 2002.

[14] P. Perumal, R. Nedunchezhian. Performance Evaluation of Three Model-Based Documents Clustering Algorithms. *European journal of Scientific Research*, Vol.52 No.4 (2011), pp.618-628, 2011.

[15] P. Perumal, R. Nedunchezhian. Improving the Performance of Multivariate Bernoulli Model based Documents Clustering Algorithms using Transformation Techniques. *Journal of Computer Science*, 7 (5): 762-769, 2011.

[16] P.Perumal and R. Nedunchezhian. Performance Analysis of Standard k-Means Clustering Algorithm on Clustering TMG format Document Data. *International Journal of Computer Applications in Engineering Sciences*, Vol I, Issue 4, 2011.

[17] Radecki, and Tadeusz. Probabilistic Methods for Ranking Output Documents in Conventional Boolean Retrieval Systems. *Inf. Process. Manage*. 24(3): 281- 302, 1998.

[18] Raghuvira Pratap A et al. An Efficient Density based Improved K- Medoids Clustering algorithm. *International Journal of Advanced Computer Science and Applications*, Vol. 2, No. 6, p. 49-54, 2011.

[19] Ramarajan and Punithavalli. M. Taxonomically Clustering Organisms Based on the Profiles of Gene Sequences using PCA. *Journal of Computer Science* 2 (3): 292-296, 2006.

[20] Ravichandra Rao. Data Mining and Clustering Techniques. *DRTC Annual workshop on Semantic Web,* Paper K: 1-12, 2003.

[21] Sandhia Valsala et at. A Study of Clustering and Classification Algorithms Used in Data mining. *International Journal of Computer Science and Network Security*, vol.11 No.10, October 2011.

[22] Soman, K.P., Shyam Diwakar, V. Ajay. *Insight into Data Mining Theory and Practice*, PHI publication, 2006.

[23] S. Zhong, S and J. Ghosh, J. A comparative study of generative models for document clustering. *SDM Workshop on Clustering High Dimensional Data and Its Applications*, May 2003.

[24] S.M. Rüger, S.E. Gauch. Feature Reduction for Document Clustering and Classification. *Technical report*. Computing Department, Imperial College London, UK, 2000.

[25] Turenne, Nicolas; Roussillon, Francois. Evaluation of Four Clustering Methods Used in Text Mining. *Proceedings of ECML Workshop on Text Mining*, 1998.

[26] Tapas Kanuho et al. The analysis of a simple k-means clustering algorithm. *Proceedings of the sixteenth annual symposium on Computational geometry,* 2000

[27] Vidyabanu, R. and Nagaveni, N. A Model Based Framework for Privacy Preserving Clustering Using SOM. *International Journal of Computer Applications*, Volume 1 – No. 13, pp. 17-21, 2010.

[28] Xiaoxia Wang and Max Bramer. Exploring web search results clustering. Research and development in intelligent systems. XXIII: proceedings of AI -2006, *the 26th International conference on Innovative techniques and applications of AI*, pp. 393-397, 2006.

[29] Salton, G. and McGill, M. J. 1983. *Introduction to Modern Information Retrieval*. McGraw Hill.