International Journal of Applied Engineering Research
ISSN 0973-4562 Volume 10, Number 7 (2015) pp. 16769-16785

http://www.ripublication.com

Cosine Based Partition Algorithm For Clustering Breast Cancer Data

S. Anitha Elavarasi[1] and Dr. J. Akilandeswari[2]

[1]Assistant Professor, Department of Computer Science and Engineering,
Sona College of Technology, Salem, India
anithaelavarasi@sonatech.ac.in
[2]Professor and Head, Department of Information Technology,
Sona College of Technology, Salem, India
akilandeswari@sonatech.ac.in

Abstract

Clustering is an unsupervised learning technique for grouping objects which are similar to each other. Partition clustering algorithm splits the data points into k partitions, where each partition represents a cluster. Objects in real world are categorical in nature. Categorical data are not analyzed as numerical data because of the absence of implicit ordering. In this paper we are proposing a term frequency based cosine similarity for partition clustering algorithm. Similarity matrix is calculated by using both the standard cosine similarity and term frequency based cosine similarity. K-means algorithm is applied to the similarity matrix and cluster are formed and evaluated. Results are evaluated for breast cancer real life data set using cosine based k-means and term frequency based cosine k-means clustering algorithm for accuracy, error rate, precision, recall and F-measure. It is inferred that when the number of cluster increases accuracy increases and error rate decreases in finding the recurrence and no recurrence event of the breast cancer data set. The empirical result suggests that our methodology has more performance gain in finding the recurrence and no recurrence event of the breast cancer data set than the traditional algorithm.

Keywords: Clustering, Unsupervised learning, Cosine similarity, Categorical data.

1. INTRODUCTION

Data mining deals with extracting information from a data source and transform it into a valuable knowledge for further use. Clustering is one of the steps in data mining. Clustering deals with grouping objects which are similar to each other. Clustering process should exhibit high intra class similarity and low inter class

similarity [1]. Clustering algorithms are broadly classified into hierarchical, partition based, grid based and density based algorithms.

Partition based algorithm splits the data points into k partition, where each partition represents a cluster. The partition is done based on certain objective function satisfying the property of symmetry, non-negativity and transitivity. K-means [1] is a popular partition algorithm. The algorithm partitions the data into n partition initially. It then uses an iterative relocation technique to improve the partitioning by moving objects from one group to another. An advantage of this algorithm is that, it is relatively efficient. The shortcomings are (1) specify K in advance, (2) not suitable for non convex shape data, (3) cannot handle noise and (4) sensitive to outliers. The time complexity is O(tkn) where t is no of iterations, k is number of clusters to be generated and n is number of objects. Other examples for Partition algorithms are K-modes [2], Fast Genetic K-means algorithm (FGKA) [3], KANMI [4] and K-Histogram [5].

Distance or similarity measure plays vital role in the formation of final cluster. Distance measure should satisfy three main properties such as (1) Non negativity, (2) Symmetry and (3) triangular inequality. Some of the popular distance measures are Euclidean distance and Manhattan distance. Categorical data are not analyzed as numerical data because of the absence of implicit ordering [6,7]. Categorical data are used in health care, educational, marketing and biomedical field. Cosine similarity [6,7,8,9] is a popular method for information retrieval or text mining. It is used for comparing the documents (word frequency) and finds the closeness among the data points in clustering. In this paper performance of cosine based k-means clustering algorithm for categorical data is studied.

The paper is organized as follows: Section 2 deals with various partition algorithms its methodology and its performance on various data sets. Section 3 deals with the proposed methodology, sample walkthrough and algorithm for Term frequency based Cosine K-means for categorical data (TCKmeans). Section 4 describes the results and discussion and finally section 5 concludes the work.

2. RELATED WORK

In the literature we find two extension of K-means algorithm for clustering larger data set. (1) K Modes Algorithm extends the k means by using simple matching dissimilarity function suitable for categorical data [2]. Mode value is used instead of mean value and finally a frequency based method for updating mode value is used to reduce the clustering cost function. (2) K-prototype algorithm is designed to work for both numerical and categorical data by integrating both K-means and K-modes algorithm.

Methodology:

1. Choose K initial mode value.
2. Simple matching dissimilarity function used for categorical data is :

$$d_c(X,Y) = \sum_{j=1}^{m} \delta(x_j, y_j) \quad (1)$$

and

$$\delta(x_j, y_j) = \begin{cases} 0, x_j = y_j \\ 1, x_j \neq y_j \end{cases} \quad (2)$$

Where X,Y represents the categorical object and m refers to the categorical attribute.

3. Frequency based method is used to calculate the mode value.
4. Allocate an object to a cluster with minimum mode value. Update the mode for all iteration until end of the object.
5. Test the dissimilarity of object against current mode. If mode value of object belongs to different cluster rather than the current one, reallocate the object to the new cluster.
6. Repeat step 2,3,4and 5 until no such modification exists

K modes algorithm produce only local optima. The author compares the performance and scalability of K-modes with K-prototype algorithm for soya bean disease dataset and motor insurance dataset. Cluster performance is verified by using cluster accuracy and error rate for soya bean disease dataset. Scalability is verified against number of clusters for a given number of objects and number of objects for a given number of clusters using motor insurance dataset.

Fast Genetic K-means algorithm (FGKA) [3] is motivated from the Genetic K-means Algorithm (GKA). Both GKA and FGKA achieve global optimum, FGKA runs much faster than GKA. The goal of FGKA is to minimize the Total Within Cluster variation (TWCV). The algorithm starts with an initial population P0. Evolution of next population will be done with the selection, mutation and K-means operators (KMO) which are done in sequence until a termination condition is met. The objective of selection operation is to find the population having greatest fitness value and assign smaller fitness value for illegal strings. The objective of mutation operation is to achieve global optimum. It generates positive probability and makes the pattern to move closer to the cluster and provides a legal solution. KMO operation is carried in order to clearly separate all illegal string from the domain.

Methodology

1. Generation of Initial population (P0)
2. Repeat step 3 through step 6 until a termination condition is reached.
3. Apply objective function to minimize the Total Within-Cluster Variation (TWCV) defined by

$$TWCV = \sum_{n=1}^{N}\sum_{d=1}^{D} Xnd^2 - \sum_{k=1}^{K}\frac{1}{Zk}\sum_{d=1}^{D} SFkd^2 \quad (3)$$

4. where Xnd represent the dth feature of pattern Xn, SFkd is the sum of the dth

feature of all the patterns in Gk, D denotes the dimension and N denotes the pattern.

5. Apply proportional selection operator Pz defined as,

$$P_z = \frac{F(S_z)}{\sum_{z=1}^{Z} F(S_z)} \quad (4)$$

6. where Sz denoted the solution, F denotes the fitness value.
7. Selection of probability distribution (Pk) is done on Mutation operator.
8. K-means operator is added as the last step to speed up the convergence process.
9. Return the final cluster formation.

The advantage of FGKA over GKM is that it provides minimum TWCV, illegal strings are eliminated and the mutation operation is simplified. The author compares FGKA with GKA and K-means using two dataset fig2data and chodata. FGKA runs 20 times faster than GKA.

K-Histogram extends k means algorithm to categorical domain by replacing mean with histogram. Histograms are dynamically updated during clustering process [5]. The K-means algorithm cannot cluster categorical data in an efficient way. To make them work for categorical data two modifications are one. First mean value is replaced with histogram. Second new dissimilarity measure between categorical data is applied. Dissimilarity functions and cost measure applied for K histogram is given in equation 5 and 6.

Methodology:

1. Initialize the 'K' value.
2. Apply cost and dissimilarity function.

$$P(W,H) = \sum_{l=1}^{k}\sum_{i=1}^{n} w_{i,l} d\,(X_i H_l) \quad (5)$$

$$d(H,Y) = \frac{\sum_{j=1}^{m} h_j y_j}{n} \quad (6)$$

3. Allocate an object to a cluster whose histogram is near to it
4. Update the histogram after each assignment
5. Repeat the steps until no object change the cluster

Histogram can be used in computer vision application. Results of K histogram lie on the initial selection of histogram and the order in which data are processed. Hence it produces only local optimal results. The author compares K Histogram with K modes algorithm for Congressional vote dataset and Mushroom data set. The methodology is evaluated for: (1) Cluster error Vs Number of cluster (2) Number of objects Vs Number of cluster (3) Number of Iteration Vs Number of cluster

K-ANMI algorithm uses K-means algorithm with Average Normalized Mutual Information (ANMI) an entropy based objective function [4]. It uses the hash table as the basic data structure. The algorithm works in two phases: (1) Initialization phase:

K objects are selected to construct initial histogram. (2) Iteration phase: Object are moved to various cluster that maximizing the ANMI. Time complexity of this algorithm is $O(Ink^2rp^2)$ and space complexity is O(rkp+nr) where 'I' refers to iteration time, 'n' refers to size of the dataset, 'r' refers to number of attribute, 'k' refers to number of cluster, 'p' refers to number of histogram.

Methodology:

1. Initialize the 'K' value.
2. Iteratively changes the class label of each data object to improve the value of object function. Objective function is defined as

$$\emptyset^{ANIM}\left(\Lambda, \bar{\lambda}\right) = \frac{1}{r}\sum_{q=1}^{r} \emptyset^{NMI}\left(\bar{\lambda}, \lambda^{(q)}\right) \tag{7}$$

3. All objects have been checked for possible improvements. If at least one label was changed in a sweep, we initiate a new sweep.
4. The algorithm terminates when a full sweep does not change any labels.

Advantages of k-ANMI are (1) suitable for both categorical data clustering and cluster ensemble, (2) it could be easily deployed in clustering distributed categorical data (3) it is flexible in handling heterogeneous data that contains a mix of categorical and numerical attributes The limitations of k-ANMI are (1)it is a great research challenge to implement the k- ANMI algorithm in an efficient way such that it is scalable to large datasets and (2)Finding global or near optimal is limited. The author compares squeezer, GAClust, ccdByEnsemble, K modes and KANMI algorithm for Congressional vote dataset, Mushroom data set and Wisconsin Breast Cancer data set.

Cosine similarity [7,8,9] is a popular method in Information retrieval. It is used for comparing the documents (word frequency) and finds the closeness among the data points in clustering. Its range lies between 0 and 1. The similarity between two terms X and Y are defined as follows:

$$Cosine\ Similarity(X, Y) = \frac{X \circ Y}{\|X\|\|Y\|} \tag{8}$$

The desirable property of Cosine similarity is that it is independent of document length. The limitation is that the terms are assumed to be orthogonal in space. The similarity value of zero implies data elements are dissimilar and the vale is 1 implies data elements are similar. Consider two documents X and Y with X = {Rose is a beautiful flower} and Y = {Rose is a flower}. Similarity between X and Y can be calculated as follows:

$$CosSim\ (X, Y) = \frac{1*1+1*0+1*0}{\|1*1+1*1+1*1\|\|1*1+1*1+1*1\|} = 0.82$$

Genetic Kmeans algorithm for mixed numerical and categorical data sets:

Genetic K-means algorithm (GKA) is the combination of genetic algorithm with K-means algorithm [10]. K-means is suitable for numeric data set. Genetic K-means clustering algorithm uses an enhanced cost function to handle the categorical data.

Methodology

1. Objective function: Defines the objective function (φ) for mixed data type

$$\varphi = \sum_{i=1}^{n} V(d_i, c_j) \tag{9}$$

where $V(d_i, c_j)$ distance of data d_i from the closest center c_j

2. Selection: Calculates the probability distribution function (P_z) using the fitness value $F(S_z)$.

$$Pz = \frac{F(Sz)}{\sum_{z=1}^{Z} F(Sz)} \tag{10}$$

where S_z denotes the solution.

3. Mutation: Mutation is performed to achieve global optimum using the probability distribution function P_k.

$$Pk = \frac{1.5 * d_{max}(X_n) - d(X_n, C_k) + 0.5}{\sum_{k=1}^{K}(1.5 * d_{max}(X_n) - d(X_n, C_k) + 0.5} \tag{11}$$

where $d(X_n,C_k)$ is the Euclidean distance between pattern X_n and the centroid C_k , $d_{max}(X_n)$ represents the maximum distance for the pattern X_n

4. Finally convergence is obtained by applying K-means operator.

The author tested the algorithm using the Heart Diseases dataset taken from the UCI repository and judge the quality of the clusters obtained. The algorithm runs for 100 iterations with the population size of 50, generation size of 100 and the mutation probability ranging from 0.001 to 0.1.GKM is compared with GKMODE, IGKA and FGKA. Results show the standard deviation for error rate is found to be 3.

Cluster Ensemble approach for mixed numerical and categorical data (CEMBC): Dataset with mixed data type are common in real life. Cluster Ensemble [11] is a method to combine several runs of different clustering algorithms to get a common partition of the original dataset. Many algorithms use similarity measures like Euclidean distance which gives good result for the numeric attribute. This will not work well for categorical attribute. In the cluster ensemble approach numeric data are handled separately and categorical data are handled separately. Then both the results are treated in a categorical manner using squeezer algorithm. The Squeezer algorithm yields good clustering result, good scalability and it handles high dimensional data set efficiently.

Methodology

1. Splitting of the given data set into two parts. One for numerical data and another for categorical data

2. Applying clustering algorithm for numerical data set
3. Applying clustering algorithm for categorical data set
4. Combining the result of step 2 and step 3 and applying squeezer algorithm to obtain the final cluster.

The author uses credit approval data set and cleve data set and compared the result of CEMBC with k prototype algorithm. Cluster accuracy and error rate was compared against number of cluster ranging from 2 to 10. Average cluster error was found to be 0.258 for k prototype and 0.191 for CEMBC algorithm.

G-ANMI algorithm uses genetic clustering algorithm with Average Normalized Mutual Information as an objective function for categorical data [12,13]. The basic genetic algorithm operation such as selection, crossover and mutation are used for clustering categorical data.

Methodology:

1. Randomly select initial population
2. Fitness function is evaluated using ANMI
3. cross over takes place by maximizing the ANMI
4. Procedure will be halted when best fitness function is greater that user specified threshold or there has be no change in the fitness value for further iteration

Advantages of G-ANMI are:(1) obtain better clustering results using less iterations and running times.(2) Easy and simple to implement (3)In larger data sets, even when population size is relatively small, G-ANMI can find better solutions at the cost of more iterations. Only drawback of G-ANMI is the computation of fitness value is very time-consuming. Four data sets (voting, breast cancer, zoo and mushroom) from UCI repository are used for performance evaluation.

ROCK stands for RObust Clustering using linKs [14] is a Agglomerative hierarchy clustering. It uses links to measure similarity between data points. Initially each tuple is assigned as a separate cluster. Clusters are merged based on the closeness between clusters. Closeness is measured as the sum of the number of links between all pair of tuple. Scalability of the algorithm depends on the sample size The author uses Congressional Vote dataset and Mushroom data set from UCI repository and compared ROCK algorithm with traditional centroid based hierarchical algorithm. In vote dataset cluster of republican contains only 12% of democrat whereas traditional approach has 25% of democrat with □=0.73. For Mushroom data set ROCK use □ = 0.8 and number of desired cluster as 20.

QROCK [15] is a Quicker version of ROCK algorithm. Two major variations of QROCK from ROCK are: (1) Final clusters are formed as the connected components of certain graph. (2) Similarity threshold is specified instead of desired number of clusters. It uses three main ADT (1) merge (A, B): finds the union of A and B, (2) Initial(X): creates the component with element X, (3) find(X): returns the component with element X. The overall complexity is QROCK is $O(n^2)$ where n is the number of data point.

Squeezer [16] is a categorical data clustering algorithm. The main data

structures involved are Cluster Summary and Cluster Structure. Summary holds set of pair of attribute values and their corresponding support value. Cluster Structure (CS) holds the cluster and summary information. The advantages of Squeezer algorithm are (1) It produces high quality cluster result (2) It has good scalability (3) It makes only one scan over the dataset, so it is highly efficient when considering I/O cost. The disadvantage of Squeezer algorithm is, the quality of the cluster depends on the threshold value (s). Space complexity is O(n+k*p*m), where 'n' represents the size of the data set, 'm' represents number of attribute, 'k' represents final number of cluster and 'p' represent distinct attribute values Squeezer algorithm is compared with ROCK using Congressional vote dataset and Mushroom data set. Threshold values are assumed as 10 and 16 for vote and mushroom dataset respectively. The only parameter that affects the clustering result and speed of the algorithm is the threshold value (s).

NabSqueezer[17] is an improved squeezer algorithm. It attempt to improve the clustering accuracies of existing squeezer algorithms by giving greater weight to uncommon attribute value matches during similarity computations. It takes two scan over the data set. First scan computers frequency of occurrence (MSFVS) and assigns weight for each attribute value. In second scan, similarity computation and cluster formation are handled. Advantages of NabSqueezer are: (1) Outlier can be handled and (2) Number of cluster is not needed. The results of the system depends on tuning parameter‘s’. Experiments were conducted and results are compared against squeezer and K-modes algorithm using soya bean, mushroom and breast cancer data set.

In Scaling up top-K cosine similarity search, the author develops two algorithm (1) TOP DATA algorithm (Top-K cOsine similarity Pairs using DiagonAl TrAversal method). [18], exploit a diagonal traversal strategy and (2) TOP-MATA algorithm((Top-K cOsine similarity Pairs using MAx-first TrAversal method), a max-first traversal strategy for the search of top-K pairs. TOPMATA has the advantages of saving the computations for false-positive item pairs and can significantly reduce I/O costs. The limitation of the system is diagonal traversal strategy usually leads to more I/O costs.

3. PROPOSED METHODOLOGY

In this paper, term frequency based cosine similarity measure has been used for clustering categorical data. The most popular K-means clustering algorithm is chosen as an underlying algorithm. The core part of this methodology is the similarity matrix formation. Data from real world consist of noise and inconsistency. Data preprocessing ensures quality input to be given to the similarity computation process. **Similarity matrix formation** uses three functions: (1) FrequencyComputation : Frequency computation deals with calculating the rate of occurrence for each attribute available in the dataset. (2) Term Frequency based Cosine Similarity (TFCS): Frequency information are generated and stored in a multi dimensional array. TFCS deals with computing the similarity matrix using cosine similarity defined in equation 12. (3) Binary Cosine Similarity (BCS): deals with computing the similarity matrix using simple binary matching similarity defined in equation 14. Similarity matrix

generated is given as an input for the K-means algorithm. Clusters are formed and the results are evaluated.

3.1 DEFINITION

Let D be the data set with X_1 to X_n instance. Each instances has $A_1..A_n$ categorical attributes with D_1 to D_n domain respectively. Val(D_{ij}) finds the number of times a particular value occur in a domain. TSim[X,Y] represents a similarity matrix computation using TFCS. BSim[X,Y] represents a similarity matrix computation using BCS.

Definition 1: Frequency computation deals with calculating the rate of occurrence for each attributes present in the dataset. In other words it returns Val(Dij) (i.e number of times a particular value occur in a domain Di for an attribute Ai). It is represented by the term F_w.

Definition 2: Term Frequency based Cosine Similarity computes the similarity matrix TSim[x,y] for the given data D. Let X and Y be the two instances with 'n' attributes. TFCS(X,Y) is defined as:

$$TFCS(X,Y) = \frac{\sum_{i=1}^{n} F_w(X_i) * F_w(Y_i)}{\sqrt{\sum_{i=1}^{n}(F_w(X_i))^2} * \sqrt{\sum_{i=1}^{n}(F_w(Y_i))^2}} \quad (12)$$

Definition 3: Simple matching coefficient computation returns a value 1 if both the attribute (Xi and Yi) returns the same value for a particular domain Di, otherwise it returns a value 0.

$$X_i * Y_i = \begin{cases} 1 & if\ X_i = Y_i \\ 0 & if\ X_i \neq Y_i \end{cases} \quad (13)$$

Definition 4: Binary matching cosine similarity computes the similarity matrix BSim[X,Y] for the given data D. Let X and Y are the two instances with n attribute. BCS(X,Y) is defined as:

$$BCS(X,Y) = \frac{\sum_{i=1}^{n} X_i * Y_i}{\sqrt{\sum_{i=1}^{n}(X_i)^2} * \sqrt{\sum_{i=1}^{n}(Y_i)^2}} \quad (14)$$

3.2 SAMPLE WALKTHROUGH

Let us consider the Balloon dataset shown in table 1, with 10 numbers of instances (A to J) and 4 number of attribute. Attribute information used in balloon data set are color, size, act and age. Domain of color =1 (i.e. yellow), Domain of size =2 (i.e. small and large), Domain of act =2 (i.e. stretch and dip) and Domain of age =2 (i.e. adult and child).

Table 1: Balloon data set

ID	Color	Size	Act	Age
A	yellow	small	stretch	adult
B	yellow	small	stretch	adult
C	yellow	small	stretch	child
D	yellow	small	dip	adult
E	yellow	small	dip	child
F	yellow	large	stretch	adult
G	yellow	large	stretch	adult
H	yellow	large	stretch	child
I	yellow	large	dip	adult
J	yellow	large	dip	child

The term frequencies for each element are represented in the array $O_{w.}$ The frequency for each attribute are: O_w[Yellow] = 10, O_w[Small] = 5, O_w[Large] = 5, O_w[Stretch] = 6, O_w[Dip] = 4, O_w [Adult]=6 and O_w[Child] = 4. Similarity computation of A with all other element is represented in table 2. The detailed computation of similarity for (A,C) and (A,E) for both cosine similarity and term frequency based cosine similarity are shown below.

$$Cosine\ Similarity\ (A,C) = \frac{1*1+1*1+1*1+1*0+0*1}{\|1*1+1*1+1*1+1*1\|\|1*1+1*1+1*1+1*1\|} = 0.75$$

$$Cosine\ Similarity\ (A,E) = \frac{1*1+1*1+1*0+1*0+0*1+0*1}{\|1*1+1*1+1*1+1*1\|\|1*1+1*1+1*1+1*1\|} = 0.50$$

$$TFCS\ (A,C) = \frac{10*10+5*5+6*6+6*0+0*4}{\|10*10+5*5+6*6+6*6\|\|10*10+5*5+6*6+4*4\|} = 0.86$$

$$TFCS\ (A,E) = \frac{10*10+5*5+6*0+4*0+0*6+0*4}{\|10*10+5*5+6*6+6*6\|\|10*10+5*5+4*4+4*4\|} = 0.71$$

Table 2: Similarity Computation

Similarity between ID	Cosine similarity	Term frequency based Cosine similarity
A,B	1.00	1.00
A,C	0.75	0.86
A,D	0.75	0.86
A,E	0.50	0.71
A,F	0.75	0.86
A,G	0.75	0.86
A,H	0.50	0.71
A,I	0.50	0.71
A,J	0.25	0.56

3.3 ALGORITHM

Input: Given Dataset with 'N' categorical attribute

Output: 'k' clusters

Algorithm : TCKmeans

begin
Initialize n as total number of tuple
read the dataset one by one
//Computer frequency
call FrequencyComputation();
//Similarity Matrix formation
TSim(X,Y) = TFCS(X,Y);
BSim(X,Y) = BCS(X,Y)/N;
//Cluster the data with max similarity - K means pseudo code
Initialize k cluster randomly
repeat
using similarity matrix select object that are most similar;
assign to the respective cluster Cluster[k++];
update the cluster;
until no change;
return final cluster formed
end

Function : FrequencyComputation()
begin
Assign initial occurrence of an attribute A_i of domain D_{ij} be zero
while not EOF do
for each attribute A_i of domain D_{ij}
calculate the occurrence of each attribute as Fw_{ij}
end for
end while
return occurrence
end

Function :TFCS(X,Y)
begin
Vector representation of string (X,Y)
multiplying each vector by its frequency of occurrence Fw
for i=1 to vectorlength do
$XYvector = F_w(X_i) * F_w(Y_i)$
$Xvector = F_w(X_i) * F_w(X_i)$
$Yvector = F_w(Y_i) * F_w(Y_i)$
compute result using $TSim(X,Y) = \frac{XYvector}{\sqrt{Xvector}\sqrt{Yvector}}$

end for
return TSim(X,Y);
end

Function : BCS()
begin
Conversion of string into vector
Multiplying each vector by its occurrence of (X,Y)
for i=1 to vectorlength do
$XY = X_i * Y_i$
$Xvector = X_i * X_i$
$Yvector = Y_i * Y_i$
compute result using
BSim(X,Y) = $\frac{XY}{\sqrt{Xvector}\sqrt{Yvector}}$
end for
return BSim(X,Y);
end

4. RESULT AND DISCUSSION

4.1 ENVIRONMENT:

Experiments were conducted on a Intel core i5 processor with 2.4GHz and 6GB DDR3 memory,1000 GB HDD running windows operating system. Program for similarity computation and k-means algorithm was written in java language.

4.2 DATA SET:

Real life dataset, such as breast cancer is obtained from UCI machine learning repository. It was obtained from the University Medical Centre, Institute of Oncology, Ljubljana, Yugoslavia. [19]. Number of instance is 286 and number of attribute is 9. It can be classified into two class (1) no-recurrence-events with 201instances and (2) recurrence-events with 85instances.

4.3 MEASURE FOR CLUSTER VALIDATION:

The cluster validation is the process of evaluating the cluster results in a quantitative and objective manner. Cluster Accuracy 'r' is defined a

$$r = \frac{\sum_{i=1}^{k} a_i}{n} \tag{15}$$

where
'n' refers number of instance in the dataset, 'a_i' refers to number of instance occurring in both cluster i and its corresponding class and 'k' refers to final number of cluster.
(2) Error rate 'E' is defined as

$$E = 1 - r, \tag{16}$$

where 'r' refers to the cluster accuracy.

Precision and recall can be viewed in two contexts as information theory and classification. Based on information theory precision is the fraction of retrieved documents that are relevant to the query [20]. Recall is the fraction of the documents that are relevant to the query that are successfully retrieved. Based on classification precision (P) is the ratio of true positive (TP) to that of sum of true positive and false positive (FP). In other words recall (R) is the ratio of true positive to that of sum of true positive and false negative (FN). F-measure (F) combines precision and recall [8].

$$P = \frac{TP}{TP+FP} \tag{17}$$

$$R = \frac{TP}{TP+FN} \tag{18}$$

$$F = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{19}$$

4.4 PERFORMANCE ANALYSIS ON ACCURACY VS NO OF CLUSTER:

Experiments were conducted by varying the no of cluster from 2 to 9. Accuracy deals with how many elements are properly identified to the correct cluster. Accuracy Vs No of cluster indicates as the number of cluster increases accuracy also increases. As the number of cluster increases accuracy for the standard cosine based K means algorithm lies between 72% and 71%. In case of term frequency based K-means algorithm accuracy ranges between 75% and 71%. Out of 8 iterations term frequency based K-means algorithm out performs standard cosine based K-means algorithm for 5 iterations is shown in Figure 1.

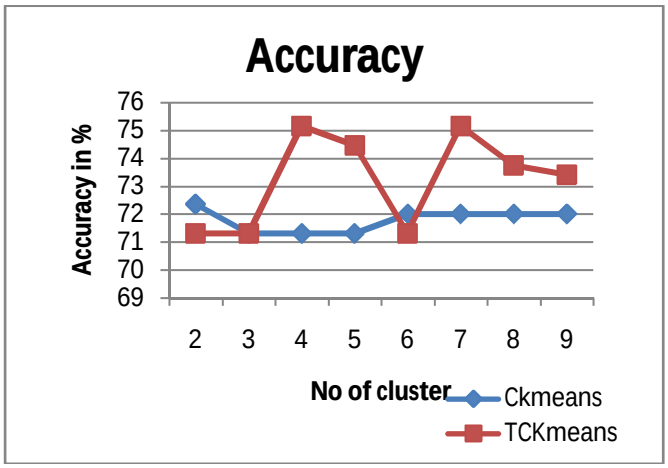

Fig1: Computation of accuracy Vs Numbero of cluster

4.5 PERFORMANCE ANALYSIS ON ERROR RATE VS NO OF CLUSTER:

Error rate is defined as the number of elements that are wrongly identified in a cluster. Error rate Vs No of cluster indicates as the number of cluster increases error rate decreases. As the number of cluster increases error rate lies in the range of 27% to

28% for the standard cosine based K means algorithm. In case of term frequency based cosine K-means algorithm shows a reduced error rate in the ranges of 24% to 28%. Out of 8 iterations term frequency based cosine K-means algorithm has reduced error rate compared with standard cosine based K-means algorithm for five iterations is shown in Figure 2.

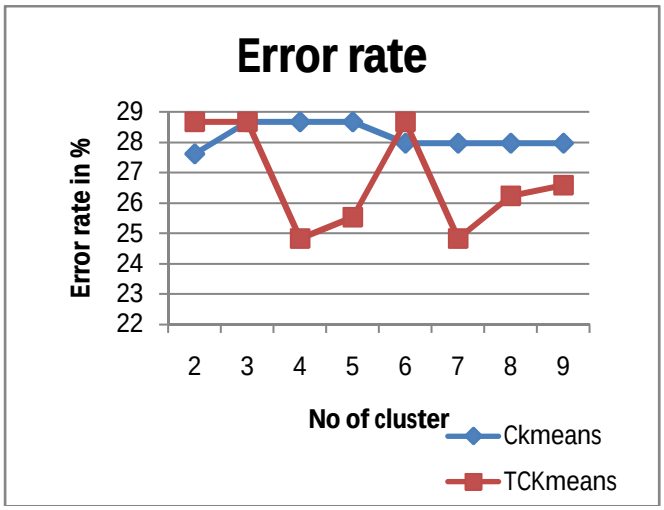

Fig2: Computation of error rate Vs Number of cluster

4.6 PERFORMANCE ANALYSIS ON PRECISION VS NO OF CLUSTER:

Figure 3 shows the computation of Precision Vs Number of clusters. It indicates as the number of cluster increases precision decreases gradually. The average precision for TCKmeans is 0.56 instead of 0.54 for CKmeans.

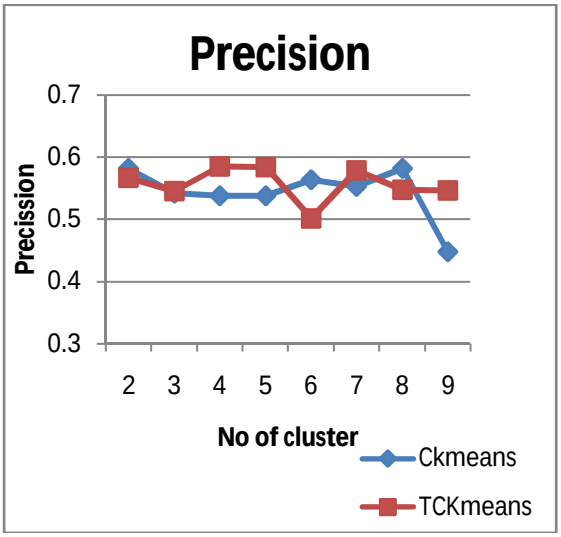

Fig3: Computation of precision Vs Number of cluster

4.7 PERFORMANCE ANALYSIS ON RECALL VS NO OF CLUSTER:

Figure 4 shows the computation of Recall Vs Number of cluster. It indicates as the number of clusters increases recall decreases gradually. The average recall is 0.22 for TCKmeans and 0.20 for CKmeans

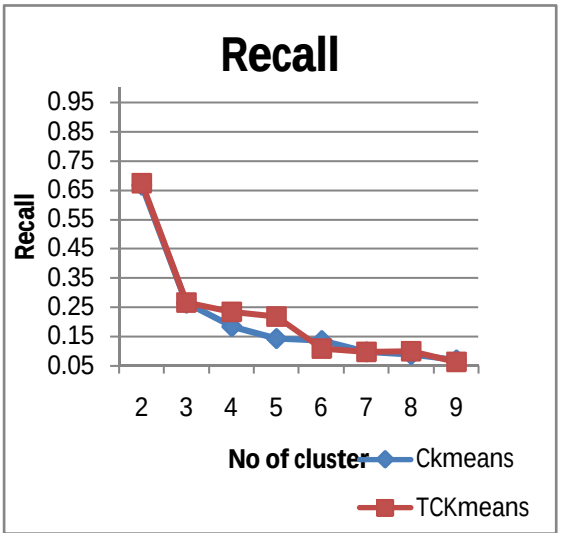

Fig4: Computation of recall Vs Number of cluster

4.8 PERFORMANCE ANALYSIS ON F-MEASURE VS NO OF CLUSTER:

Figure 5 shows the computation of F-measure Vs No of clusters for TFCKmeans and CKmeans. It indicates Fmeasure is inversely propositional to number of clusters. As the number of clusters increases F-measure decreases gradually. The average F-measure is 0.28 for TCKmeans and 0.26 for CKmeans..

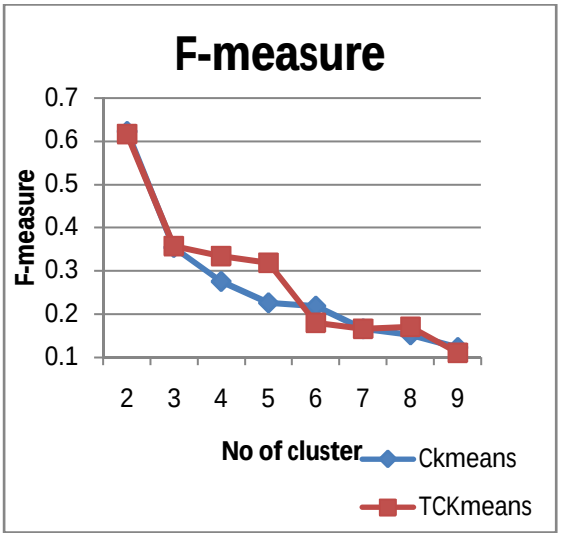

Fig5: Computation of F-measure Vs Number of cluster

5. CONCLUSION

Objects in real world consist of categorical data. Categorical data are not analyzed as numerical data because of the absence of inherit ordering. Results of Standard K-means algorithm indicate poor performance for categorical data which paves way for new measure for computing similarity among objects. Cosine similarity is used in the field of information retrieval. The desirable property of cosine similarity is independent of document length. The performance of cosine based k-means clustering algorithm for categorical data (TCKmeans) is evaluated by using three functions such as Frequency Computation, Term Frequency based Cosine Similarity (TFCS) computation, and Binary Cosine Similarity (BCS) computation. Results are evaluated for breast cancer data set using standard cosine based k-means and term frequency based cosine k-means. Inferences from the proposed system are

1. The number of cluster increases the accuracy increases for term frequency based cosine K-means algorithm (i.e. accuracy ranges between 75% and 71%). The Average accuracy observed for TCKmeans is 74% whereas Ckmeans is 71%.
2. The number of cluster increases, the error rate decreases 29% to 24% for term frequency based cosine K-means algorithm. The average error rate observed for TCKmeans is 26% whereas Ckmeans is 28%.
3. The average precision for TCKmeans is 0.56 instead of 0.54 for CKmeans.
4. The average recall for TCKmeans is 0.22. and average F-measure for TCKmeans is 0.28.
5. TCKmeans achieves 62.5% enhanced result in finding the recurrence and no recurrence event of the breast cancer data set than CKmeans algorithm.

Algorithm can be improved further by using normalized frequency value or TF-IDF score instead of frequency value while computing the similarity matrix. In addition techniques to reduce the time complexity can be dealt for further enhancement.

Reference

[1] Han, Jiawei, Micheline Kamber, and Jian Pei. *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.

[2] Huang, Zhexue. "Extensions to the k-means algorithm for clustering large data sets with categorical values." *Data mining and knowledge discovery* 2 no 3 (1998): 283-304.

[3] Lu, Y., Lu, S., Fotouhi, F., Deng, Y., & Brown, S. J. (2004, March). FGKA: A fast genetic k-means clustering algorithm. In *Proceedings of the 2004 ACM symposium on Applied computing* (2004) 622-623.

[4] He, Zengyou, Xiaofei Xu, and Shengchun Deng. "k-ANMI: A mutual information based clustering algorithm for categorical data." *Information Fusion* 9 no 2 (2008): 223-233.

[5] He, Zengyou, et al. "K-histograms: An efficient clustering algorithm for categorical dataset." *arXiv preprint cs/0509033* (2005).

[6] http://reference.wolfram.com/language/ref/CosineDistance.html
[7] http://en.wikipedia.org/wiki/Cosine_similarity
[8] Boriah, Shyam, Varun Chandola, and Vipin Kumar. "Similarity measures for categorical data: A comparative evaluation." *red* 30.2 (2008):
[9] Leskovec, Jure, Anand Rajaraman, and Jeffrey David Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.
[10] Roy, Dharmendra K., and Lokesh K. Sharma. "Genetic k-Means clustering algorithm for mixed numeric and categorical data sets." *International Journal of Artificial Intelligence & Applications* 1 no 2 (2010): 23-28.
[11] He, Zengyou, Xiaofei Xu, and Shengchun Deng. "A cluster ensemble method for clustering categorical data." *Information Fusion* 6 no 2 (2005): 143-151.
[12] Deng, Shengchun, Zengyou He, and Xiaofei Xu. "G-ANMI: A mutual information based genetic clustering algorithm for categorical data." *Knowledge-Based Systems* 23.2 (2010): 144-149.
[13] Agresti, Alan. *An introduction to categorical data analysis*. Vol. 135. New York: Wiley, 1996.
[14] Guha, Sudipto, Rajeev Rastogi, and Kyuseok Shim. "ROCK: A robust clustering algorithm for categorical attributes." *Data Engineering, 1999. Proceedings., 15th International Conference on*. IEEE, (1999) 512-521.
[15] Dutta, Mala, A. Kakoti Mahanta, and Arun K. Pujari. "QROCK: A quick version of the ROCK algorithm for clustering of categorical data." *Pattern Recognition Letters* 26.15 (2005): 2364-2373.
[16] He, Zengyou, Xiaofei Xu, and Shengchun Deng. "Squeezer: an efficient algorithm for clustering categorical data." *Journal of Computer Science and Technology* 17.5 (2002): 611-624.
[17] He, Zengyou, Xiaofei Xu, and Shenchun Deng. "Improving categorical data clustering algorithm by weighting uncommon attribute value matches."*Computer Science and Information Systems* 3.1 (2006): 23-32.
[18] Zhu, Shiwei, et al. "Scaling up top-K cosine similarity search." *Data & Knowledge Engineering* 70.1 (2011): 60-83.
[19] http://www.ics.uci.edu/mlearn/MLRepository.html
[20] http://en.wikipedia.org/wiki/Precision_and_recall