Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 10, October 2015, pg.353 – 363

SURVEY ARTICLE

Survey of Cluster Analysis and its Various Aspects

Harminder Kaur[1], Jaspreet Singh[2]
[1]CSE & RIMT Mandi Gobindgarh, India
[2]CSE & LCET Katani Kalan, India
[1] harminder.mann@ymail.com; [2] jsprtsekhon@gmail.com

Abstract: Cluster analysis or clustering is a technique storing logically similar objects together physically. This physical storage is referred as classes in clustering. The data available as input for clustering can be of various types e.g. image, text etc. This process is carried out by different algorithms such as k-mean, fuzzy-C etc. In this paper light is thrown out on various aspects related to cluster analysis. Topics covered are types of cluster, types of data and clustering methods available. Aim is to help the researchers to understand the basics of clustering in a single paper who are interested to work on the same.

Keywords- Clustering; types of cluster; k-mean; DBSCAN; Grid based

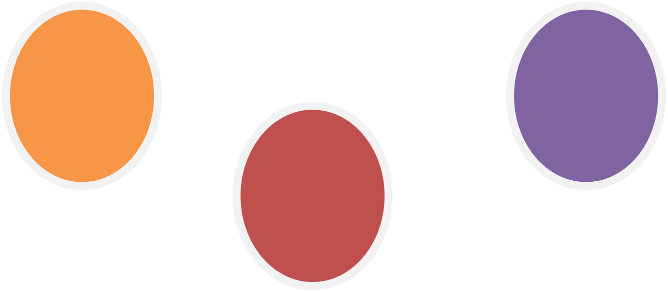
I. Introduction

Cluster analysis or clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)[1].It is a main task of exploratory data mining, and a common technique for statistical data analysis, used in many fields, including learning, pattern, image analysis, information retrieval, and bioinformatics.

Cluster analysis itself is not one specific algorithm, but the general task to be solved. It can be achieved by various algorithms that differ significantly in their notion of what constitutes a cluster and how to efficiently find them. Popular notions of clusters include groups with small distances among the cluster members, dense areas of the data space, intervals or particular statistical distributions. Clustering can therefore be formulated as a multi-objective optimization problem[3].The appropriate clustering algorithm and parameter settings (including values such as the distance function to use, a density threshold or the number of expected clusters) depend on the individual data set and intended use of the results. Cluster analysis as such is not an automatic task, but an iterative process of knowledge discovery or interactive multi-objective optimization that involves trial and failure[2],[4]. It will often be necessary to modify data preprocessing and model parameters until the result achieves the desired properties.

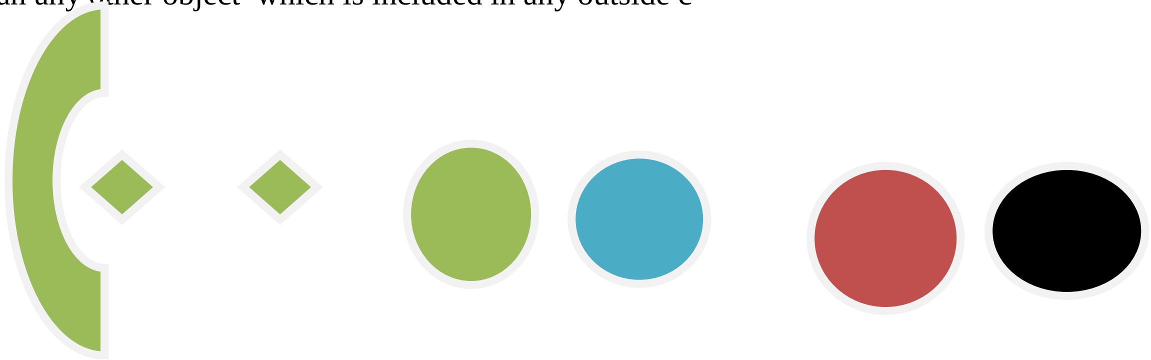
II. Types of clusters

a) *Well separated clusters*:- A cluster is collection of similar set of points but separating dissimilar points from it.

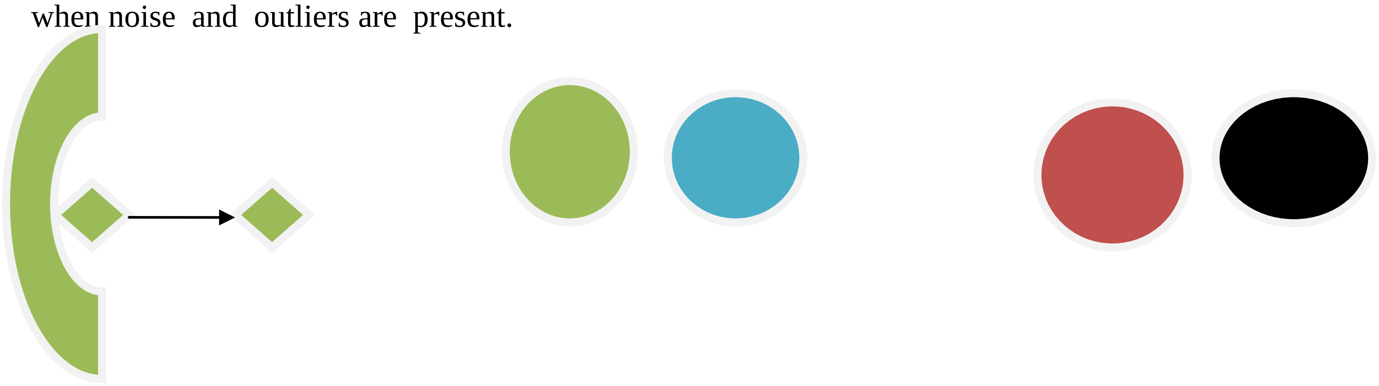


b) *Center based clusters*:- Centroid is a term used to measure the center of the cluster. Center based object is more nearer to cluster containing it than any other outside cluster.

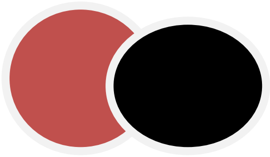
c) *Contiguous Clusters*:- Clusters containing objects which are closer to one or more its neighbors in the same cluster than any other object which is included in any outside c



d) Density based clusters:- A cluster is a dense region of points which is separated by low density regions from other regions of high density. It is used when the clusters are irregular or interwined and when noise and outliers are present.



e) Shared Property or conceptual clusters:- Clusters sharing common properties for representing a concept.



III. Types of Data in Cluster Analysis

- Interval-scaled variables
- Binary variables
- Nominal, ordinal, and ratio variables
- Variables of mixed types

a) Interval-scaled variables

Standardize data

$$s_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + ... + |x_{nf} - m_f|)$$

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + ... + x_{nf}).$$

where

-Calculate the mean absolute deviation:

Calculate the standardized measurement (*z-score*)

But using mean absolute deviation is more robust than using standard deviation

Similarity and Dissimilarity Between Objects

Distances are normally used to measure the similarity or dissimilarity between two data objects.

$$d(i,j)=\sqrt[q]{(|x_{i_1}-x_{j_1}|^q+|x_{i_2}-x_{j_2}|^q+...+|x_{i_p}-x_{j_p}|^q)}$$

Some popular ones include: *Minkowski distance*:

$$d(i,j)=|x_{i_1}-x_{j_1}|+|x_{i_2}-x_{j_2}|+...+|x_{i_p}-x_{j_p}|$$

where i = (x_{i1}, x_{i2}, …, x_{ip}) and j = (x_{j1}, x_{j2}, …, x_{jp}) are two p-dimensional data objects, and q is a positive integer

If $q = 1$, d is Manhattan distance, $q = 2$,

$$d(i,j)=\sqrt{(|x_{i_1}-x_{j_1}|^2+|x_{i_2}-x_{j_2}|^2+...+|x_{i_p}-x_{j_p}|^2)}$$

d is Euclidean distance:

Properties

- $d(i,j) \geq 0$
- $d(i,i) = 0$
- $d(i,j) = d(j,i)$
- $d(i,j) \leq d(i,k) + d(k,j)$

Also, one can use weighted distance, parametric Pearson product moment correlation, or other dissimilarity measures

e.g. Salary, height etc.

b) Binary Variables

i)A contingency table for binary data

Objects j

$$
\text{Objects i}\quad
\begin{array}{c|ccc}
 & 1 & 0 & sum \\
\hline
1 & a & b & a+b \\
0 & c & d & c+d \\
sum & a+c & b+d & p
\end{array}
$$

ii)Distance measure for symmetric binary variables:

$$d(i,j)=\frac{b+c}{a+b+c+d}$$

iii)Distance measure for asymmetric binary variables:

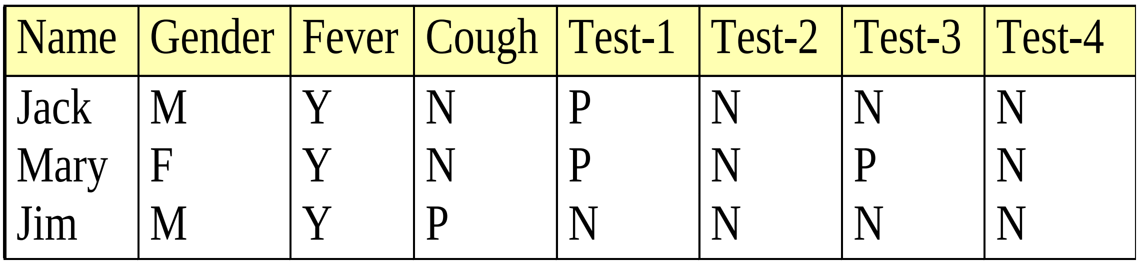
iv)Jaccard coefficient (*similarity* measure for *asymmetric* binary variables):

$$d(i,j)=\frac{b+c}{a+b+c}$$

$$\mathbf{0}\ \ sim_{Jaccard}(i,j)=\frac{a}{a+b+c}$$

v)Dissimilarity between Binary Variables

Example:



Name	Gender	Fever	Cough	Test-1	Test-2	Test-3	Test-4
Jack	M	Y	N	P	N	N	N
Mary	F	Y	N	P	N	P	N
Jim	M	Y	P	N	N	N	N

Gender is a symmetric attribute; the remaining attributes are asymmetric binary. Let the values Y and P be set to 1, and the value N be set to 0.

$$d(jack, mary) = \frac{0+1}{2+0+1} = 0.33$$
$$d(jack, jim) = \frac{1+1}{1+1+1} = 0.67$$
$$d(jim, mary) = \frac{1+2}{1+1+2} = 0.75$$

c) Nominal Variables

- A generalization of the binary variable in that it can take more than 2 states, e.g., red, yellow, blue, green
- Method 1: Simple matching

$$d(i,j) = \frac{p-m}{p}$$

  - *m*: # of matches, *p*: total # of variables
- Method 2: use a large number of binary variables
- creating a new binary variable for each of the *M* nominal states

d*) Ordinal Variables*

An ordinal variable can be discrete or continuous Order is important, e.g., rank can be treated like interval-scaled

$$r_{if} \in \{1,...,M_f\}$$

replace x_{if} by their rank .map the range of each variable onto [0, 1] by replacing *i*-th object in the *f*-th variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

compute the dissimilarity using methods for interval-scaled variables.

e) Ratio-Scaled Variables

Ratio-scaled variable: a positive measurement on a nonlinear scale, approximately at exponential scale, such as Ae^{Bt} or Ae^{-Bt} Methods:treat them like interval-scaled variables but it is n*ot a good choice because* the scale can be distorted. Apply logarithmic transformation $y_{if} = log(x_{if})$ and treat them as continuous ordinal data treat their rank as interval-scaled.

f) Variables of Mixed Types

A database may contain all the six types of variables symmetric binary, asymmetric binary, nominal, ordinal, interval and ratio.

$$d(i,j) = \frac{\Sigma_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\Sigma_{f=1}^{p} \delta_{ij}^{(f)}}$$

One may use a weighted formula to combine their effects

f is binary or nominal: $d_{ij}^{(f)} = 0$ if $x_{if} = x_{jf}$, or $d_{ij}^{(f)} = 1$ otherwise

f is interval-based: use the normalized distance .

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

f is ordinal or ratio-scaled .compute ranks r_{if} and and treat z_{if} as interval-scaled.

IV. Different Techniques of Data Clustering

a) Partitional or Representative-based Clustering

Given a dataset with n points in a d-dimensional space, **D** = {**x**i}$_{ni=1,}$ and given the number of desired clusters k, the goal of representative-based clustering is to partition the dataset into k groups or clusters, which is called a *clustering* and is denoted as C ={*C*1,*C*2, . . . ,*C*k}. Further, for each cluster *C*i there exists a representative point that summarizes the cluster, a common choice being the mean (also called the *centroid*) μ_i of all points in the cluster, that is,

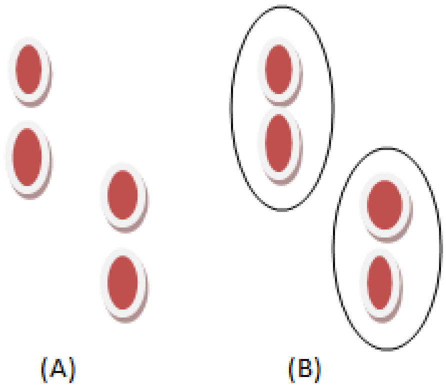
$$\boldsymbol{ui = 1/ni \sum\nolimits_{xj \in Ci} xj}$$

xj∈ *C*i **x**j where ni = |*C*i | is the number of points in cluster *C*i .

Clustering's is not practically feasible. In this chapter we describe two approaches for Representative-based clustering, namely the K-means and expect ratio[4]. The K-means algorithm, probably the first one of the clustering algorithms proposed, is based on a very simple idea: Given a set of initial clusters, assign each point to one of them, then each cluster center is replaced by the mean point on the respective cluster . [5]These two simple steps are repeated until convergence. A point is assigned to the cluster which is close in Euclidean distance to the point. Although K-means has the great advantage of being easy to implement, it has two big drawbacks.

First, it can be really slow since in each step the distance between each point to each cluster has to be calculated, which can be really expensive in the presence of a large dataset. Second, this method is really sensitive to the provided initial clusters, however, in recent years, this problem has been addressed with some degree of success.[5]

A brute-force or exhaustive algorithm for finding a good clustering is simply to generate all possible partitions of n points into k clusters, evaluate some optimization score for each of them, and retain the clustering that yields the best score.



(A) Original Points (B) Partitioning Clustering

b) Hierarchal clustering

Given n points in a d-dimensional space, the goal of hierarchical clustering is to create a sequence of nested partitions, which can be conveniently visualized via a tree or hierarchy of clusters, also called the cluster *dendrogram*. The clusters in the hierarchy range from the fine-grained to the coarse-grained – the lowest level of the tree (the leaves) consists of each point in its own cluster, whereas the highest level (the root) consists of all points in one cluster. Both of these may be considered to be *trivial* clustering's. At some intermediate level, we may find meaningful clusters. If the user supplies k, the desired number of clusters, we can choose the level at which there are k clusters. [6-8]There are two main algorithmic approaches to mine hierarchical clusters: agglomerative and divisive. Agglomerative strategies work in a bottom-up manner That is, starting with each of the n points in a separate cluster, they repeatedly merge the most similar pair of clusters until all points are members of the same cluster. Divisive strategies do just the opposite, working in a top-down manner. Starting with all the points in the same cluster, they recursively split the clusters until all points are in separate clusters.

i) Number of Hierarchical Clustering

The number of different nested or hierarchical clustering's corresponds to the number of different binary rooted trees or dendrograms with n leaves with distinct labels.[5] Any tree with t nodes has t −1 edges. Also, any rooted binary tree with m leaves has m−1 internal nodes. Thus, a dendrogram with m leaf nodes has a total of t = m+ m−1 =2m−1 nodes, and consequently t −1=2m−2 edges

ii) Agglomerative Hierarchical Clustering

In agglomerative hierarchical clustering, we begin with each of the n points in a separate cluster. We repeatedly merge the two closest clusters until all points are members of the same cluster.Formally, given a set of clusters C = {*C*1,*C*2, ..,*C*m}, we find the *closest* pair of clusters *C*i and *C*j and merge them into a new cluster *C*ij = *C*i ∪*C*j . Next, we update the set of clusters by removing *C*i and *C*j and adding *C*ij , as follows C =
C \ {*C*i,*C*j }∪ {*C*ij }.
We repeat the process until C contains only one cluster. Because the number of clusters decreases by one in each step, this process results in a sequence of n nested clusterings. If specified, we can stop the merging process when there are exactly k clusters remaining.

iii) Divisive Hierarchical Clustering

Divisive clustering starts with a single cluster that contains all data points and recursively splits the most appropriate cluster. The process repeats until a stopping criterion (frequently, the requested number k of clusters) is achieved.

iv) Among the most used variations of the hierarchical clustering based on different distance measures are:

i) Average linkage clustering

The dissimilarity between clusters is calculated using average values. The average distance is calculated from the distance between each point in a cluster and all other points in another cluster. The two clusters with the lowest average distance are joined together to form the new cluster.

ii). Centroid linkage clustering

This variation uses the group centroid as the average. The centroid is defined as the center of a cloud of points.

iii). Complete linkage clustering (Maximum or Furthest-Neighbor Method)The dissimilarity between 2 groups is equal to the greatest dissimilarity between a member of cluster i and a member of cluster j. This method tends to produce very tight clusters of similar cases.

iv). Single linkage clustering (Minimum or Nearest-Neighbor Method): The dissimilarity between 2 clusters is the minimum dissimilarity between members of the two clusters. This method produces long chains whichform loose, straggly clusters.

v) Ward's Method: Cluster membership is assigned by calculating the total sum of squared deviations from the mean of a cluster. The criterion for fusion is that it should produce the smallest possible increase in the error sum of squares[9].

c)Density based algorithm

Clusters can be thought of as regions of high density, separated by regions of no or low density. Density here is considered as the number of data objects in the "neighborhood".

The most popular one is probably *DBSCAN (Density-Based Spatial Clustering of Applications with Noise*,. The algorithm finds, for each object, the neighborhood that contains a minimum number of objects. Finding all points whose neighborhood falls into the above class, a cluster is defined as the set of all points transitively connected by their neighborhoods. *DBSCAN* finds arbitrary-shaped clusters while at the same time not being sensitive to the input order. Besides, it is incremental, since every newly inserted point only affects a certain neighborhood. [9],[10]On the other hand, it requires the user to specify the radius of the neighborhood and the minimum number of objects it should have; optimal parameters are difficult to determine.

i) Computational Complexity

The main cost in DBSCAN is for computing the ϙ-neighborhood for each point. If the dimensionality is not too high this can be done efficiently using a spatial index structure in *O*(nlogn) time. When dimensionality is high, it takes *O*(n2) to compute the neighborhood for each point. Once *N*ϙ(**x**) has been computed the algorithm needs only a single pass over all the points to find the density connected clusters. Thus, the overall complexity of DBSCAN is *O*(n2) in the worst-case. The major feature of this algorithm is, Discover clusters of arbitrary shape and has a capability to handle noise data in a single scan. The several interesting studies on density based algorithm are DBSCAN, GDBSCAN, OPTICS, DENCLUE and CLIQUE. The two global parameters in density are *Eps:* Maximum radius of the neighbourhood and *MinPts:* Minimum number of points in an Eps neighbourhood of that point.

ii) Density Reachability - A point "p" is said to be density reachable from a point "q" if point "p" is within ε distance from point "q" and "q" has sufficient number of points in its neighbours which are within distance ε. Density Connectivity - A point "p" and "q" are said to be density connected if there exist a point "r" which has sufficient number of points in its neighbours and both the points "p" and "q" are within the ε distance. This is chaining process. So, if "q" is neighbour of "r", "r" is neighbour of "s", "s" is neighbour of "t" which in turn is neighbour of "p" implies that "q" is neighbour of "p".

iii) Density Based Spatial Clustering of Applications with Noise (DBSCAN)

Density Based 0Spatial Clustering of Application with Noise(DBSCAN) relies on a *density-based* notion of cluster: A *cluster* is defined as a maximal set of density-connected points. In spatial database it discovers clusters of arbitrary shape with noise

iv) Steps Involved in DBSCAN

Arbitrary select a point ***p*** Reclaim all the points density-reachable from ***p*** with respect to ***Eps*** and ***MinPts***. If point ***p*** is a core object, a cluster is formed. If point ***p*** is a border object, no points are density-reachable from ***p*** and DBSCAN visits the next point of the database. Continue the process until all of the points processed. ***Core Object:*** The object with at least MinPts objects within a radius 'Eps-neighborhood' ***Border Object:*** object that on the border of a cluster.

v) Pros and Cons of Density-Based Algorithm

The main advantage density-based clustering Algorithm does not require a-priori specification and able to identify noisy data while clustering. It fails in case of neck type of dataset and it does not work well in case of high dimensionality.

d)Grid based algorithm

The Grid-Based type of clustering approach uses a multi resolution grid data structure. It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The grid-based clustering approach differs from the conventional clustering algorithms in that it is concerned not with the data points but with the value space that surrounds the data points. In general, a typical grid-based clustering algorithm consists of the following five basic steps:

- Creating the grid structure, i.e., partitioning the data space into a finite number of cells.
- Calculating the cell density for each cell.
- Sorting of the cells according to their densities.
- Identifying cluster centers.
- Traversal of neighbor cells.

STING (A Statistical Information Grid Approach)

In STING the spatial area is divided into rectangular cells. There are many levels of cells corresponding to different There are many levels of cells corresponding to different levels of resolution.[11] Each cell is partitioned at high level into a number of smaller cells in the next lower level. The statistical info of each cell is calculated and stored beforehand and is used to answer queries. By using the parameters of lower level the parameters of higher level cell scan be easily calculated. STING uses a top-down approach to answer the spatial data queries.[10]

The Merits and Demerits are as follows

Merits

Fast processing time.

Good cluster quality.

No distance computations

Clustering is performed on summaries and not individual objects; complexity is usually O(#- populated-grid-cells) and not O(#objects) Easy to determine which clusters are neighboring.

• Shapes are limited to union of grid-cells

Demerits

• All the cluster boundaries are either horizontal or vertical, and no diagonal boundary is detected.

V. Conclusion

The overall goal of the data mining process is to separate the information from a large data set and transform it into an understandable form for further use. Clustering is an important task in data analysis and data mining applications. Clustering is the task of grouping a set of objects so that objects in the same group are more similar to each other than to those in other groups (clusters).Clustering can be done by the different algorithms such as hierarchical-based, partitioning-based, grid-based and density-based algorithms. Hierarchical-based clustering is the connectivity based clustering. Partitioning-based algorithm is the centroid based clustering. Density based clusters are defined as area of higher density then the remaining of the data set. Grid based clustering, partition the space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed.

REFERENCES

[1] Pavel Berkhin, "A Survey of Clustering Data Mining Techniques", pp .25-71, 2002.

[2] Amandeep Kaur Mann, .Review paper on Clustering Techniques., Global Journal of Computer Science and Technology Software & Data Engineering.

[3] K.Kameshwaran et al," Survey on Clustering Techniques in Data Mining" / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (2) , 2014, 2272-2276

[4] Osama Abbu Abaas "Comparision between data clustering algorithm" ,The international Arab journal of Information Technology,Vol. 5, No.3,july 2008

[5] Ramandeep Kaur & Gurjith Singh Bhathal, .A Survey of Clustering Techniques., International Journal on Computer Science and Engineering Vol. 02, No. 09, 2010, 2976-2980.

[6] Megha Gupta, Vishal Shrivastava "Review of various Techniques in Clustering" International Journal of Advanced Computer Research (ISSN (print):2249-7277 ISSN (online):2277-7970) Volume-3 Number-2 Issue-10 June-2013

[7] Wael M.S. Yafooz,Siti Z.Z. Abidin,Nasiroh Omar, Rosenah A. Halim," Dynamic Semantic Textual DocumentClustering Using Frequent Terms and Named Entity",201 3 IEEE 3rd International Conference on System Engineering and Technology, 19 - 20 Aug. 2013, Shah Alam, Malaysia

[8] Aastha Joshi , Rajneet Kaur,"A Review: Comparative Study of Various Clustering Techniques in Data Mining", International Journal of Advanced Research in Computer Science and Software Engineering Volume 3, Issue 3, March 2013

[9] Dr. Manju Kaushik, Mrs. Bhawana Mathur," Comparative Study of K-Means and Hierarchical Clustering Techniques",International journal of hardware and software research in Engineering Volume 2,Issue 6,June 2014

[10] T. Soni Madhulatha," An Overview On Clustering Methods" IOSR Journal of Engineering Apr. 2012, Vol. 2(4) pp: 719-725 ISSN:

[11] Pradeep Rai, Shubha Singh "Survey of Clustering Techniques" International Journal of Computer Applications (0975 – 8887)Volume 7– No.12, October 2010