International Journal of Computational Intelligence Research
ISSN 0973-1873 Volume 13, Number 10 (2017), pp. 2379-2390

http://www.ripublication.com

A Critical Study on Cluster Analysis Methods to Extract Liver Disease Patterns in Indian Liver Patient Data

K.Swapna[1]and Prof. M.S. Prasad Babu[2]

[1]*Research Scholar, Dept. of CS & SE, Andhra University, Visakhapatnam, A.P, India.*

[2]*Professor, Dept. of CS & SE, Andhra University, Visakhapatnam, A.P, India,*

Abstract

Clustering is one of an important technique that can be used to find hidden patterns and structures from a large dataset. Liver diseases can be easily diagnosed by analyzing the levels of enzymes in the blood. In this paper, some clustering algorithms are considered for finding the patterns by generating the clusters on liver patient datasets. The clustering algorithms used in this work are 1.*k*-Means (KM) Clustering Algorithm, 2. Agglomerative Nesting (AGNES) Clustering Algorithm, 3.Density Based Spatial Clustering of Applications with Noise (DBSCAN) Clustering Algorithm, 4. Ordering Points to Identify the Clustering Structure (OPTICS) Clustering Algorithm and 5. Exception Maximization (EM) Clustering Algorithm. These algorithms are applied to a Indian liver dataset with 1083 records with ten attributes. This Dataset is formed by collecting 500 records from local hospitals and taking 583 records from ILPD dataset available in the UCI machine repository. Four quality parameters, namely Accuracy, Entropy, F-Measure and Purity are considered in clustering the above dataset.

Keywords: Clustering Algorithms, Data Mining, Liver Diagnosis, Validation, Gastroenterology.

1. INTRODUCTION

Clustering is a very popular Data Mining technique that can be used in the design of many automatic medical diagnoses tools. It is similar to a classification, but here the groups are not predefined before, where as in classification the groups are predefined.

Here data is divided into groups or clusters by taking Intra-class similarity as minimum and Inter–class similarity as a maximum. Liver diseases are very dangerous diseases and problems with liver patients are not easily discovered in the early stages. Liver functions normally and even if it is partially damaged, it is very difficult to recognize its dysfunction during the early prognosis. But an early prognosis of liver problems will increase patient's survival rate and this may be diagnosed by analyzing the levels of enzymes in the blood [2]. Automatic classification tools may be used to reduce the patient queue at the liver experts [3].classification tools require classification factor or class label to find the groups with some diagnosis. But clustering techniques can be used data without a class label for the analysis. It is in this contest clustering techniques are considered to find the patterns in the liver patient dataset. After that clustering, the model is used for labeling the data (annotation) and then that dataset is used for classification to diagnose the patient.

In this paper five clustering algorithms, namely, 1.*k*-Means (KM) Clustering Algorithm, 2. Agglomerative Nesting (AGNES) Clustering Algorithm, 3. Density Based Spatial Clustering of Applications with Noise (DBSCAN) Clustering Algorithm, 4. Ordering Points to Identify the Clustering Structure (OPTICS) Clustering Algorithm, and 5. Exception Maximization (EM) Clustering Algorithm, have been applied to the new liver patient dataset and compared their performances. Two liver patient datasets were used in this study, one is from ILPD dataset taking from University of California at Irvine (UCI) Machine Learning Repository and the second one is a new dataset (physically collected) from various pathological laboratories in southern India. In this experimentation, the two datasets are merged to form it into one single liver patient dataset. This paper concentrates on the performance of clustering algorithms with a Indian liver patient dataset.

2. RELATED WORKS

M. Vijaya Lakshmi [4] presented an overview of different clustering algorithms including different parameters and approaches followed in large datasets. S. Anita Elavarasi et al. [5] considered various partition clustering algorithms and found their performances on large datasets. K.Sasirekha, P.Baby[6] studied both Agglomerative hierarchical clustering and Divise hierarchical clustering algorithms. Michael Steinbach, George Karypis, Vipin Kumar[7] deduced that k-Means algorithm is performing better for Agglomerative hierarchical clustering with document clustering. Pradeeprai and Shubha Singh [8] reviewed different clustering techniques and shown that these algorithms may be used to discover highly correlated patterns from very large datasets. ManishVerma, MaulySrivastava etc [9] compared various clustering algorithms and concluded that EM algorithms and *k*-Means algorithms can be applied

on both small dataset and large datasets, whereas DBSCAN and OPTICS clustering algorithms shows poor performance on small data sets.

3. CLUSTERING ALGORITHMS USED IN THIS WORK

3.1 *k*-Means Algorithm

k- Means clustering is one of the partition clustering algorithms, proposed by Stuart Lloyd in 1957. It is the easiest and most commonly used clustering algorithm [10]. In this algorithm, a number of clusters *k* (defined) is assumed initially in the first step and then reset *k*- value in the subsequent steps. Based on the *k*-value divide the data set into k partitions randomly. Then consider a centroid for each partition or a cluster and calculate the distance from the centroid to each data object. Based on the distances rearrange the new clusters and re-compute the new centroids. This procedure is repeated until no elements cannot be moved from the clusters and final clusters are tabulated. The following are the steps involved in ***k***-means clustering algorithm:

Algorithm:

Input: *k* : number of clusters;

$D = \{t_1, t_2, t_3, ...t_n\}$: Dataset containing n objects and every object has m dimensions $\{t_{p1}, t_{p2}, t_{p3}, ...t_{pm}\}$;

E- $\tilde{E} = \tau$: Minimum error/threshold error value

Output: Set of *k* clusters.

Method:

1. Based on k value, choose initial cluster centers $C_1, C_2, C_3 ..... C_k$ arbitrarily from the dataset D.

2. For each object in D, calculate the Euclidean distance d between each data object (t_p) and the cluster center C_i.

$$d(C_i, t_p) = (\sum_{f=1}^{m} (t_{pf} - C_{if})^{1/2}$$

The Euclidean distance between two objects t_p and t_q can be computed by

$$d(t_p, t_q) = (\sum_{f=1}^{m} (t_{qf} - t_{pf})^{1/2}$$

3. Based on distance *d,* assign each object to a nearest center, *then* calculate

new centroid for new cluster and square error

$$E = \sum_{i=1}^{k} \sum_{t \in C_i} | \ t - C_i \ |^2$$

Where E is the sum of the squared error of all objects in the database.

4. Repeat the step 2 and 3until the difference between the successive squared errors$(E- \tilde{E})$less than threshold value = £

5. Similarly apply the procedure until no reassignment takes place

6. Final clusters are tabulated

3.2 Agglomerative Nesting (AGNES) Algorithm

AGNES is an agglomerative hierarchical clustering algorithm, proposed by SAHN et al in 1973. This algorithm uses dendrogram technique, a tree data structure, wherein each level shows a cluster for that level. AGNES Algorithms start with each object as a separate cluster itself, and successively merge groups according to a distance measure following greedy-like bottom-up merging method. The clustering may stop when all objects enters in a single group or at any other point the user wants. It can be broken at different levels to yield different clustering of the data. It is very useful for viewing the data at different levels of details [14]. This hierarchical clustering algorithm uses Average link (AL) for finding the distances.

Algorithm:

Input: $D = \{t_1, t_2, t_3, ...t_n\}$ n is the total number of objects

$P = \{d_{ij}\}_{nxn}$: Proximity Matrix showing the distance(eq(1)) between elements

n: Number of Clusters $\{C_0, C_1, C_2, ... C_{(n-1)}\}$

n_i: Number of objects in cluster c_i and m_i = Mean of cluster C_i.

L(k) is the level of the k^{th} clustering.

τ = Threshold average distance similarity between the centriods of all clusters.

Output :DE =Dendrogram represented as a set ordered triples.

Method:

1. Begin with the disjoint clustering having level L(0) = 0 and k = 0
2. Find the average distance between every pair of clusters in the current clustering using the equation

$$Avg\ \ distance(c_i, c_j) = avg\ \ x_{t \in c_{i}, t' \in c_j} \mid t - t' \mid$$

i.e. The average distance between each point in one cluster to every point in the other cluster

3. Increment the sequence number: k = k +1. Merge clusters C_i and C_j into a single cluster to form the next clustering at level m. Set the level of this clustering to $L(k) = d(C_i, C_j)$
4. *Update the proximity matrix* P, by deleting the rows and columns corresponding to clusters C_i and C_j and adding a row and column corresponding to the newly formed cluster.
5. *If* all objects are in one cluster**,** *stop. Else, go to* step 2.

3.3 DBSCAN algorithm

DBSCAN (Density-based spatial clustering of applications with noise**)**is a density based clustering algorithm, proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and XiaoweiXuin 1996. This approach is used to create clusters with minimum size and minimum density. Density is defined as a number of points within a certain distance of each other. This handles the arbitrary shaped clusters and outliers with two major parameters r and MinPts. [11].This algorithm is effectively useful when the user-defined parameters r and MinPts are taken arbitrarily.

Algorithm:

Input: $D = \{t_1, t_2, t_3, ...t_n\}$a data set containing n objects; r : radius parameter, and MinPts : minimum points in the neighborhood: It is the density threshold.

Output*:* Set of density-based *k* clusters.

Method:

1. Mark all objects as unvisited; randomly select an unvisited object t_i; Mark t_i as visited;
2. If the r-neighborhood of t_i has at least MinPts objects .Create a new cluster C, and add t_i to C;
3. Let N be the set of objects in the *r*-neighborhood of t_i each point t_j in N
4. If t_j is unvisited, mark t_j as visited; The *r*-neighborhood of t_j has at least MinPts points, add those points to N;
5. t_j is not yet a member of any cluster, add t_j to C;
6. Visit the next point of the database; continue the process until all of the points have been processed.

7. If the objects are not in a clusters
8. mark t_i as noise; until no object is unvisited;

3.4. OPTICS algorithm

OPTICS (Ordering Points to Identify the Clustering Structure) is a density based clustering algorithm. It was first introduced by Mihael et al.[15] in 1999 to find the density-based clusters in spatial data. Clusters are found by giving input parameters r and *MinPts*. In this algorithm user has a choice of varying the Parameters values so that an acceptable cluster may be obtained. This is actually advantage of OPTICS algorithm over DBSCAN algorithm. Actually, this is a problem associated with many other clustering algorithms also. Such parameter settings are usually empirically set and it is very difficult to determine for real-world high-dimensional datasets. The details of OPTICS algorithm is given below

Algorithm:

Input: $D = \{t_1, t_2, t_3, ...t_n\}$a data set containing n objects; r: radius: For maximum distance for density measure, and MinPts: number of points in the neighborhood: It is the density threshold.

Output*:* Set of density-based *k* clusters.

Method:

1. This order selects an object that is density-reachable with respect to the lowest evaluate so that clusters with higher density will be finished first. Based on this idea, two values need to be stored for each object-core-distance and reach ability-distance.
2. The *core-distance* of an object p is the smallest r value that makes $\{t_i\}$ a core object. If t_i is not a core object, the core-distance of p is undefined.
3. The reach ability-distance of an object t_j with respect to another object t_i is the greater value of the core-distance of t_j and the Euclidean distance between t_i and t_j If $t_i$is not a core object, the reach ability -distance between p and t_j is undefined.
4. The reach ability-distance of $t_j^2$with respect to t_i is the Euclidean distance from t_i to t_j because this is greater than the core-distance of t_i.
5. Extraction of all density-based clustering with respect to any distance$r$that is smaller than the distance r used in generating the order.

3.5 Expectation Maximization (EM) Algorithm

Expectation maximization algorithm (EM) is one of the statistical model based clustering algorithm. It was first introduced by Liu, Rubin and wu1998 .It is an iterative algorithm for finding maximum likelihood or maximum a posteriori (MAP) estimates of parameters in statistical models, where the model depends on unobserved latent variables. The EM iteration alternates between performing an expectation (E) step, which computes the expectation of the log-likelihood evaluated using the current estimate for the parameters, and maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameters-estimates are then used to determine the distribution of the latent variables in the next E step [9].

- *Expectation*: Fix model and estimate missing labels.
- *Maximization*: Fix missing labels (or a distribution over the missing labels) and find the model that maximizes the expected log-likelihood of the data.

Algorithm:

Input*:* k number of clusters, D is a data set containing n objects

Output*:* A set of *k* clusters.

Method:

1**.** Make an initial guess of the parameter vector: This involves randomly selecting objects to represent the cluster means or centers (as in *k*-means partitioning), well as making guesses for the additional parameters.

2. Iteratively refine the parameters (or clusters) based on the following two steps:

2.1. *Expectation Step:* Assign each object ***xi*** to cluster *Ck* with the probability

$$p(x_i \in C_k) = p(C_k \,/\, x_i) = \frac{p(C_k)\, p(x_i \,/ C_k)}{p(x_i)}$$

Where $p(x_i/C_k) = N(m_k, E_k(\boldsymbol{x_i}))$ follows the normal (i.e., Gaussian) distribution. Around mean, m_k, with expectation, E_k. In other words, this step calculates the Probability of cluster membership of object x_i, for each of the clusters. These probabilities are the "expected" cluster memberships for object x_i

2.2. *Maximization Step*: Use the probability estimates from above to re-estimate (or Refine) the model parameters.

$$m_k = \frac{1}{n}\sum_{i=1}^{n} \frac{x_i\, p(x_i \in C_k)}{\sum_j p(x_i \in c_j)}$$

This step is the "maximization" of the likelihood of the distributions given

Repeat *Expectation* -step and *Maximization* -step until convergence.

3.8 Cluster Validation

Entropy, F -Measure, and Purity are the most frequently used external quality measures in addition to the interpretability of the result.

Entropy: Entropy provides a measure of ring randomness. It specifies whether the particular data is constantly falling into same cluster or not. The Entropy of a clustering is

H (Ω) =∑ H (w) (N_w/N)Where Ω= {w_1, w_2…..w_k.} is the set of clusters, H (w) is a single clusters Entropy N_w is the number of points in cluster N is the total number of points.

F-Measure: F-measure provides a measure of Accuracy. It is based on recall and precision measures used in evaluation of an information retrieval system

$$F - Measure = \frac{2 * (precision * recall)}{precision + recall)}$$

$$precision = \frac{TP}{(TP + FP)} \qquad Recall = \frac{TP}{(TP + FN)}$$

Purity*:* Purity measures the quality of the clusters.

$$Purity = \frac{TP}{TP + TN + FP + FN}$$

Where TP=#True Positive, TN=#True Negative, FP=#False Positive, FN=#False Negative.

4. RESULTS AND DISCUSSIONS

Performance of clustering algorithms is evaluated with Indian liver datasets. First dataset (ILPD) is taken from University of California at Irvine (UCI) Machine Learning Repository [12] which contains medical records of 583 patients and each

record contains 10 important parameters required for diagnosis process .Second dataset is our dataset contains 500liverpatients' records and 13 attributes. For the experimentation we have merged the two datasets it will be 1083 records and common 10 attributes are used .The liver dataset as shown in table [1].

Table I. Attributes In Liver Dataset

Attributes	**Information(Normal Value)**
Age	Age of the patient
Gender	Gender of the patient
TB(LFT)	Total_ Bilirubin (0.22-1.0 mg/dl)
DB(LFT)	Direct_ Bilirubin (0.0-0.2 mg/dl)
Alkphos (LFT)	Alkaline Phosphotase (110-310U/L)
SGPT(LFT)	Alamine Aminotransferase (5-45U/L)
SGOT(LFT)	Aspartate Aminotransferase (5-40U/L)
TP(LFT)	Total Protiens (5.5-8gm/dl)
ALB(LFT)	Albumin(3.5-5 gm/dl)
A/G Ratio(LFT)	Albumin and Globulin Ratio (>=1)
Selector	field used to split the data into two sets

In Indian liver data set attributes are Simple blood tests the function tests are Total- Bilirubin, Direct_Bilirubin, Alkphos, SGPT, SGOT, Total Proteins, Albumin andA\G ratio other attributes are Age, Gender. According to the attributes this data set gives the output in two clusters. Cluster 1 is liver patients; cluster 2 is Non liver patients;

Table II. Performance of Clustering Algorithms

Clustering Algorithms	**Accuracy**	**Entropy**	**F-Measure**	**Purity**
k-Mean	64.28	0.1462	0.5642	0.7423
AGNES	61.32	0.1621	0.5252	0.7033
DBSCAN	51.39	0.2821	0.4283	0.6064
OPTICS	54.67	0.2431	0.4442	0.6226
EM	59.65	0.2241	0.5062	0.6843

In this critical study, five clustering algorithms are considered and they are *k*-Mean, AGNES, DBSCAN, OPTICS, and EM algorithms, clusters are generated using the selected algorithms and those clusters are interpreted and validated by the experts in Gastroenterologists. The clusters are evaluated using the quality and Validation

measures such as Accuracy, Entropy, F-measure and Purity. The results are shown in table II.

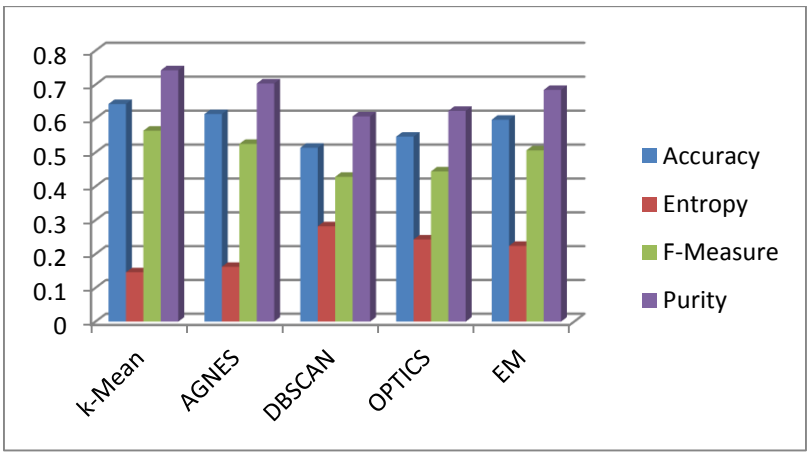

Figure 1: Performance Evaluation of Clustering Algorithms

5. CONCLUSIONS

In this study five selected clustering algorithms namely *k*-Mean, AGNES, DBSCAN, OPTICS, and EM were considered and applied to the Indian liver dataset, Accuracy, Entropy, *F*-measure and Purity parameters are measured and finally it is found that *k*-Means clustering algorithm shown high performance, low performance in Entropy, high performance in purity, and high performance in *F*-measure, compared to other selected clustering algorithms and produced better clusters. So the result of this paper concludes that *k*-Means clustering algorithm is very suitable for Indian liver dataset to diagnoses the disease.

ACKNOWLEDGEMENTS

We sincerely to the expert Gastroenterologists Dr.Srinivas Rao and Dr.Srinivas Baba for their highly valuable contribution and cooperation.

REFERENCES

[1] Rong-Ho Lin. An intelligent model for liver disease diagnosis. Artificial Intelligence in Medicine 2009; 47:53—62.

[2] Schiff's Diseases of the Liver, 10th Edition Copyright ©2007 Lippincott Williams &Wilkinsby Schiff, Eugene R.; Sorrell, Michael F.; Maddrey, Willis.

[3] BendiVenkataRamana, Prof.M.SurendraPrasadBabu, Prof. N. B. Venkateswarlu "A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis" International Journal of Database Management Systems(IJDMS),Vol.3,No.2 May 2011

[4] M.Vijayalakshmi ,and M.Renuka Devi "A Survey of Different Issue of Different clustering Algorithms Used in Large Data sets" International Journal of Advanced Research in Computer Science and Software Engineering Volume 2, Issue 3, March 2012 ISSN: 2277 128X.

[5] S.Anitha Elavarasi "A Survey on Partition Clustering algorithms "International Journal of Enterprise Computing and Business System (Online)http://www.ijecbs.com Vol. 1 Issue 1 Jan2011.

[6] K.Sasirekha, P.Baby, Agglomerative Hierarchical Clustering Algorithm-" International Journal of Scientific and Research Publications, Volume 3, Issue 3, March 2013 ISSN 2250-315.

[7] Michael Steinbach, George Karypis ,Vipin Kumar "A Comparison of Document Clustering Technique.

[8] Pradeeprai, Shubha Singh "A Survey of Clustering Techniques" International Journal of Computer.

[9] Manish Verma, MaulySrivastava, NehaChack, AtulKumarDiswar, NidhiGupta "A Comparative Study of Various Clustering Algorithms in Data Mining" published in International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com Vol. 2, Issue 3, May-Jun 2012, pp.1379-1384 1379.

[10] NohaAbu-Zeid, RashaKashif and Osama Mohamed Badawy '' ImmuneBased Clustering for Medical Diagnostic Systems" 2012 International Conference on Advanced Computer Science Applications and Technologies.

[11] Margaret H.DunhamS.sridhar "Data mining Introductory and advanced topics.

[12] The Indian liver patient dataset (ILPD)is from UCI machine repository in the area of life science. The ILPD data set is available in following hyper link http://archive.ics.uci.edu/datasets/ILPD+(indian+liver+patient+Datase)

[13] Jiawei Han and Micheline Kamber "Data Mining Concepts and Techniques"

[14] SanjoyDasgupta, Philip M. Long, Performance guarantees for hierarchical clustering.

[15] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: ordering points to identify the clustering structure," in Proceedings of ACM SIGMOD Conference, pp. 49–60, June 1999.