Recursive Hierarchical Clustering Algorithm

Pavani Y. De Silva, Chiran N. Fernando, Damith D. Wijethunge, and Subha D. Fernando

***Abstract*—Ultimate objective of data mining is to extract information from large datasets, and to utilize the extracted information in decision making process. Clustering is the most generic approach of many unsupervised algorithms in data mining, which cluster data into samples so that objects with similar statistical properties cluster together. Hierarchical, partition, grid and spectral are such clustering algorithms coming under unsupervised approach. Many of these approaches produce clusters either according to a predefined value or according to its own algorithm or produces hierarchies letting the user to determine the preferred number of clusters. Selecting appropriate number of clusters for a given problem is a crucial factor that determine the success of the approach. This paper proposes a novel recursive hierarchical clustering algorithm which combine the core concepts of hierarchical clustering and decision tree fundamentals to find the optimal number of clusters that suits to the given problem autonomously.**

***Index Terms*—Decision tree, gain ratio, gini-gain, gini-index, hierarchical clustering.**

I. Introduction

Clustering is a foremost unsupervised learning technique comes under data mining. Clustering organizes the groups of elements into clusters according to their statistical similarities. The fundamentals of the concept of clustering techniques allows the components of a cluster being highly similar to each other, components of different clusters being highly different to each other, and these similarity and dissimilarity measure have distinct and practical meaning [1], [2]. These similarity and distinct measures of clusters have been determined using

Euclidean distance, Cosine similarity and Jaccard similarity [3] based on the datatype the problem involve with. Cosine similarity measures the distance among two sets of vectors. Distance between two sets of sets is measured using Jaccard distance while Euclidean distance measures the distance between two points [4].

Applications of clustering is varying from grouping numerical data points to analyzing large corpus of unlabeled data. For example, clustering is useful for fundamental tasks of marketing for identifying assembly of users with similar behavior using a large corpus of data containing past buying patterns and consumer behaviors. It is used in biology to create a taxonomy of living beings. Information retrieval uses clustering algorithms to group search results of a query based on an aspect of a query. Cluster analysis is used to find patterns in the climate of atmosphere and ocean [5]. Furthermore, clustering can also be used to find a group of genes that are more likely to respond to the same treatment, detecting communities in huge groups of people.

However, many of these approaches expect the number of clusters required to be predetermined, or based on the deviations among the clusters, some algorithms itself determined the number of clusters to be produced as outputs. Also, some clustering algorithms produce the distance or radius between clusters enabling the number of clusters to be determined by the user. Additionally, decision trees like approaches using entropy or gini indexes split the data into hierarchies letting the user to determine the level to prune the tree.

This paper proposes a technique for clustering that is efficient which concerns about the pitfalls and limitations of existing clustering algorithms, mainly in determining the number of clusters to be produced. The proposed recursive hierarchical approach takes the gini-index from the decision trees and enable to find the optimal number of clusters with suitable radius and gini-index.

II. Analysis on Existing Clustering Algorithms

The existing key clustering algorithms have been reviewed with the purpose of highlighting the key performances, strengthens and limitations.

A. K Means Clustering

K means clustering is one of the simplest unsupervised clustering algorithms. In K means clustering the number of clusters for which the dataset is intended to be alienated has to be known in advance. In this approach a fixed set of centroids is defined initially by clustering the data set into k number of clusters arbitrarily. Then an iterative approach is followed for clustering data point into clusters by calculating the similarity between data points and the centroids of the clusters based on Euclidean distance or similar distance until the clusters of the two consecutive iterations are static and would not change. K means clustering is fast, robust and easier to understand [3]. For the purpose of determining the optimal k value for clustering a dataset, an experimentation which compares the average distance to the centroid for increasing k values has been conducted. As per the findings, the average distance to the centroid will be decreasing by a very small value for increasing k values, when the number of clusters exceeds than the optimal number of clusters for a dataset [3].

Weaknesses associated with k means clustering includes with lower sample data sets it is difficult to cluster data accurately. Clustering algorithm requires a priory specification of the number of clusters using another algorithm such as Self Organizing Maps. Results are circular

Manuscript received October 23, 2017; revised January 25, 2018. This work was supported in part by Faculty of Information Technology, University of Moratuwa.

The authors are with the University of Moratuwa, Sri Lanka (e-mail: pavanidesilva@gmail.com, pcnfernando@gmail.com, ddrwijethunge@gmail.com, subhaf@uom.lk).

doi: 10.18178/ijmlc.2018.8.1.654

or spherical in shape due to calculating the Euclidean distance. There is no knowledge on the variable which highly contributes to the clustering process in case of high dimensional data clustering. K means algorithm fails for categorical data. The use of exclusive assignment i.e. if there exists data which are highly overlapping then k-means will not be able to resolve the clusters distinctly [5]. Each iteration of this K means algorithm has complexity of O(kn), but the number of iterations are very large therefore Bradley-Fayyad-Reina (BFR) is proposed as an alternative to overcome this issue [3]. Storage requirement of the k means algorithm is $O((m+K)n)$,where m represents the number of data points and n is the dimension of the attributes.

B. Self-Organizing Maps (SOM)

Self-Organizing Maps are based on competitive learning methods. It is an unsupervised learning technique which non-linearly projects multi-dimensional data onto a two-dimensional plot which is known as vector quantization. In SOM, no human intervention is needed for the learning process and less information is needed to be known about the input data. SOM measures the similarity among data points in terms of statistical measures. SOM creates a network in a way that topological relationship within the data set are preserved [7].

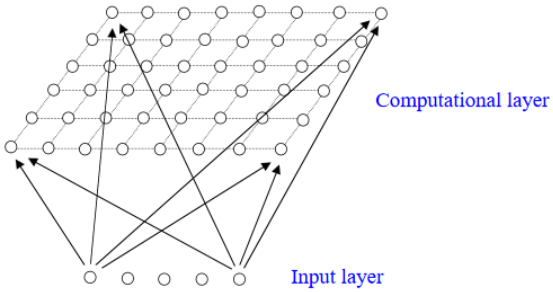

Fig. 1. Network architecture of self organizing maps.

Kohenan SOMs are a type of neural network. Network architecture of SOM is depicted in Fig. 1. Each computational layer node is connected to each input layer node. Nodes in the computational layer are connected only to represent the adjacency and there is no weighted connection between the nodes in that layer. Each node has a specific topological position and a vector of weights as of the same dimension as the weight vector. SOM is based on mainly three steps: competition, cooperation and synaptic adaption. Competition process is about finding the winning neuron for an input data. Winning neuron is the best matching unit and is calculated using the minimum Euclidean distance or the highest inner product of the weight vector and the input vector [8].

Then it finds the neighboring neurons affected by the winning neuron. It concentrates on the facts that search space or the neighboring function should decrease when the number of iterations increase and the effect of the winning neuron to the neighboring neuron should decrease when the distance between the winning neuron and the neighboring neurons increases. Finally, Synaptic adaption is adjusting the weights of the neighboring neurons based on the effect to the winning neuron [9].

Main advantage of Self Organizing Maps includes its ability of projecting multivariate data into a two-dimensional plot while providing intelligible, useful summary of data [7]. SOM is easy to understand and is simple. SOM can evaluate its own quality. The disadvantages associated with SOM include the possibility of distortions due to the fact that high dimensional data cannot be always represented in a two-dimensional plot. In order to overcome the distortion problem, training rate and neighborhood radius has been reduced slowly in many researches. Therefore, for successful memory map development, SOM needs a high number of training iterations [7]. Getting the right data is a problem associated with this algorithm because each attribute or the dimension of every data point needs to have a value. Missing data is one of the major problems we find in a dataset. SOM clusters data so that data points are surrounded by similar data points. But similar points are not always together or closer in this method [10].

C. Hierarchical Clustering

Hierarchical clustering is a clustering approach which constructs a tree of data points by considering the similarity between data points, while other clustering algorithms are flat clustering algorithms. Hierarchical clustering returns a set of clusters which are informative than flat clustering structures. The number of clusters need not to be specified in advance and these algorithms are directive. Hierarchical clustering iteratively merges clusters into either a single cluster or to individual data nodes. Dendrogram is the diagrammatic representation of the arrangement of clusters produced during hierarchical clustering process. To stop at a predefined number of clusters either the dendrogram has to be cut at a similarity measure [11]. There are two types of hierarchical clustering methods. Distance between the clusters or data points are calculated using Euclidean distance or Manhattan distance. Distance can be calculated using single linkage, average linkage and complete linkage. Single linkage defines the distance between two clusters as the shortest distance between two points in each cluster. While the distance between two clusters is defined as the longest distance between two points in each cluster in complete linkage. And the distance between two clusters is defined as the average distance between each point in one cluster to every point in the other cluster in average linkage [12]. Single linkage suffers from chaining because only two close points are needed to merge two clusters and the cluster may spread due to data points far away in the cluster. Complete linkage avoids chaining but is subjected to crowding. Clusters can be closer to other clusters than to the clusters to which distance is calculated [13].

Agglomerative hierarchical clustering

Bottom up hierarchical clustering approach is referred to as agglomerative clustering approach. This method treats the one data point as a single cluster and merges the clusters considering the similarity between individual clusters until a single cluster containing all the data points is constructed.

Considering a set of N data points, a distance matrix of $N * N$ points is created and the basic algorithm of agglomerative hierarchical clustering can be described as follows:

a) Initialize with N clusters, where a single data point is represented by one cluster

b) Determine the pair of clusters with the least distance that is the closest pair of clusters and unify them into a single cluster, so that it results with one cluster less than the original number of clusters.
c) Calculate pairwise distances (similarities) between the clusters at current that is the newly formed cluster and the priory available clusters.
d) Repeat steps *b* and *c* until all data samples are merged into a single cluster of size *N* [14-16].

Complexity of agglomerative clustering algorithm is $O(n^3)$ in general case. Therefor this makes it less appropriate for larger datasets due to its high time consumption and high processing time [17].

Decisive hierarchical clustering

This is a top down approach of constructing a hierarchical tree of data points. The entire data sample set is considered as one cluster initially and division of clusters considering the 'second flat hierarchical clustering' approach. This method is more complex than the bottom up approach because a second method is used to split the clusters. In this case in most of the time k means clustering approach is used as the flat clustering algorithm. Most of the time divisive approach produce more accurate results than bottom up approaches [18].

Algorithm of divisive hierarchical clustering is as follows.

a) Primarily initiate the process with one cluster containing all the samples
b) Select a cluster with the widest diameter as the largest cluster.
c) Detect the data point in the cluster found in step b) with the minimum average similarity to the other elements in that cluster.
d) The data sample found in c) is the first element to be added to the fragment group
e) Detect the element in the original group which reports the highest average similarity with the fragment group;
f) If the average similarity of element detected in e) with the fragment group is greater than its average similarity with the original group then assign the data sample to the fragment group and go to Step e; otherwise do nothing;
g) Repeat Step b – Step f until each data point is separated into individual clusters [19].

Complexity of divisive hierarchical clustering algorithm is $O(2n)$ [17].

Hierarchical clustering algorithms does not have any problems with choosing initial points and getting stuck in local minima. These hierarchical clustering algorithms are expensive in storage and computational complexity. Since all merges are final and as we do not have any control over the algorithm it might produce erroneous results in noisy and high dimensional data. Above identified problems could be suppressed to some extent by initially clustering the data using partitive clustering algorithms [6]. In hierarchical clustering there is no way to detect the optimal number of clusters. Number of clusters can be identified by cutting the dendrogram at a similarity measure so that the similarity among the clusters of that point is the y axis value at the cut. Furthermore hierarchical clustering algorithms concerns only about the inter cluster and intra cluster variations in clustering data.

D. Decision Tree

Decision tree is a classification algorithm which constructs a tree considering the underlying rules used primarily for classification and prediction. There exist numerous decision tree methods namely ID3, CART, C4.5, PUBLIC, CN2, SPRINT etc. [20]. This paper analyses only ID3, C4.5 and CART decision tree development methods in detail as they are the key approaches under Decision Trees.

In a top down recursive approach decision trees analyses the attributes in the internal nodes of the tree and predicts the downward attributes based on the attributes of the node and concludes the classification process using the leaf nodes. It breaks down the data set into smaller subsets while incrementally building a decision tree. Decision trees can handle both categorical and numerical data and uses either the depth first greedy approach or the breadth first search to find the suitable cluster [21].

Iterative Dichotomized algorithm (ID3 algorithm)

ID3 algorithm used entropy and information gain to build the decision tree. Entropy calculated the homogeneity of a sample dataset. To build a decision tree the entropy is calculated by using one attribute factor. Secondly the entropy is again calculated by using two attribute factors, and so on. The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain. Attribute factor with the highest information gain is used to split the data set at the first level. Algorithm continues iteratively until pure node is received.

Algorithm of decision tree can be explained as follows;

a) Calculate the entropy of the target split of data using (1).

$$E(S) = \sum_{i=1}^{c} - \mathrm{Pi} \log_2 Pi \quad (1)$$

where *i* is the target class of the split of data, *Pi* is the probability of the target class and *E*(*S*) is the entropy of the target split.

b) Split the dataset on different attributes and calculate the entropy for each branch. Calculated entropy of each branch is added proportionally to get the entropy of the split.
c) Resulting entropy is subtracted from the previous entropy to get the information gain. Information gain is calculated using (2).

$$\text{Gain } (T, X) = \text{Entropy } (T) - \text{Entropy } (T, X) \quad (2)$$

where *T* and *X* are attributes.

d) Choose attribute with the highest information gain as the decision node. Then divide the dataset by its branches and repeat the same process on every branch.
e) Splits with 0 entropy need not be further divided while splits with entropy higher than 0 needs to be further into leaf nodes

Computational complexity of ID3 algorithm is the linear function of product of a number of examples, the characteristic number, and node number [22]. ID3 algorithm does not reveal high accuracy rates for noisy data sets and highly detailed datasets. Therefor preprocessing of data is important before training the ID3 algorithm. Main drawback of this algorithm is it usually favors the attribute with unique values using the 'Gain' measurement [21]. This algorithm id highly sensitive to the noise.

Classification 4.5 algorithm (C4.5)

This algorithm is the successor of ID3 algorithm and uses 'Gain ratio' as the splitting criteria instead of 'information gain' [23]. Information gain is normalized using 'split information' in C4.5 algorithm [24]. C4.3 uses gain ratio to choose the best splitting attribute instead of information gain. Refer (3) for Gain ratio of an attribute. Split information is defined in (4), where p_j is the probability of j the split and v is the total splits of a target dataset.

$$\text{Gain ratio (A)} = \frac{\text{Gain / Gini difference (A)}}{\text{Split Info } (A)} \quad (3)$$

$$\text{Split(A)} = \sum_{j=1}^{v} \text{pjlog2}(pj) \quad (4)$$

Advantage of C4.3 includes handling both discrete and continuous attributes. It defines a threshold for handling continuous attributes and ruptures the list into two as above the threshold and below or equal the threshold. This algorithm can deal with datasets that have patterns and contain unknown values. It avoids missing values when calculating gain and entropy [21].

Classification and Regression Trees (CART algorithm)

CART algorithm deals with both categorical and continuous data in developing a decision tree. Furthermore it also handles missing values in the dataset. It uses the 'Gini Index' to select the best split criteria. Analogous to ID3 and C4.5 algorithms it splits the data set into binary branches [24]. 'Gini index' is a measure of impurity of a data sample set, and defined in (5), where P_j is the probability of j the split of a dataset.

$$\text{Gini Index} = 1 - \sum_j Pj2 \quad (5)$$

CART trees are developed to a maximum size without the use of a stopping criterion. Cost complexity pruning is used to prune back the tree split by split [25]. CART algorithm yields high accuracy when compared to other two algorithms [26], refer Table I.

Decision trees decrease the cost of predicting the class of data with the addition of each data point. It can be used for both categorical and numerical data.

TABLE I: COMPARISON OF THE ACCURACY OF DECISION TREE ALGORITHMS [26]

Decision Tree Algorithms	Correctly classified instances	Incorrectly classified instances	Unclassified instances
ID3	50%	47.5%	2.50%
C4.5	54.17%	45.83%	0%
CART	55.83%	44.17%	0%

However, when dealing with categorical data with multiple levels, the information gain is biased in favor of the attributes with the most levels. Its more complex when dealing with linked outcomes.

E. Detecting the Optimal Number of Clusters

Elbow method

Elbow method uses the concept of any partitive clustering algorithm – to minimize the within cluster variation.

a) Compute clustering algorithm (e.g., k-means clustering) for different values of *k*. For instance, by varying *k* from 1 to 10 clusters

b) For each *k*, calculate the total within-cluster variation, using (6).

$$\sum_{k=1} kW(Ck) \quad (6)$$

where Ck is the *kth* cluster and $W(Ck)$ is the within-cluster variation.

c) The curve of Within Cluster Variation is then plotted according to the number of clusters k.

d) The location of a bend (knee) in the plot is generally considered as an indicator of the appropriate number of clusters as shown in Fig. 2.

Fig. 2. Graphical representation of the results of Elbow method.

Average Silhouette Method

This method measures the quality of clustering. It defines how well each object lies within the cluster. A high average silhouette width indicates a better clustering approach. This method observes average silhouette width for different number of clusters graphically and identify the optimal number of clusters through the number of clusters which maximize the average silhouette width, refer Fig. 3.

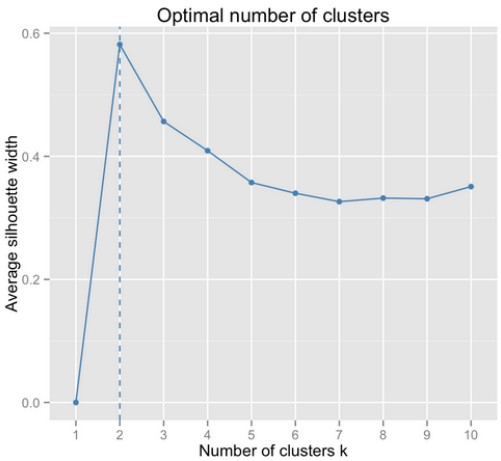

Fig. 3. Graphical representation of the results of Average Silhoette method.

III. RECURSIVE HIERARCHICAL CLUSTERING ALGORITHM

In normal agglomerative and decisive hierarchical clustering algorithms, the distance between the data points is used for merging the data points into clusters. If we consider about agglomerative hierarchical clustering, a dendrogram is created by merging two individual data points at a single time by considering the pairwise distances of the data points existing at the moment. Merging of data points or clustered

points continues until one cluster is formed. Detecting the best number of clusters is done using experimentation and graphical methods.

In our algorithm, a method has been proposed to improve the accuracy of clustering images by combining core concepts of decision tree and hierarchical clustering. Local optima for the number of clusters is detected using a novel hierarchical clustering algorithm.

The proposed recursive hierarchical clustering algorithm considers each data point as an individual data point, calculates the pairwise distances and merges the data points with minimum pairwise distance. Each merged cluster is considered as an individual data point in the next iteration. Distance is calculated using one from minimum, complete and average linkage methods where minimum distance calculates the closest distance between two clusters, complete linkage calculates the maximum distance between two clusters and average linkage calculates the average distance between two clusters. The tree is developed by iteratively merging data points or clusters until a single cluster is formed. Decisive clustering considers the primary data set as one cluster and divides the data into clusters iteratively until a single data point is considered as one cluster.

Refer (7) for pairwise distance among data points or data clusters, calculated using Euclidean distance

$$Dij2 = \sum_{v=1}^{n} (Xvi - Xvj)\ 2 \quad (7)$$

where Dij the Euclidean distance between the two data is is points, Xvi and Xvj are the two data points where v is the dimension. The resulting dendrogram will be as in Fig. 4. X axis of the dendrogram represent the individual data points while Y axis of the dendrogram will represent the similarity among the data points or the pairwise distances among the data points or data clusters.

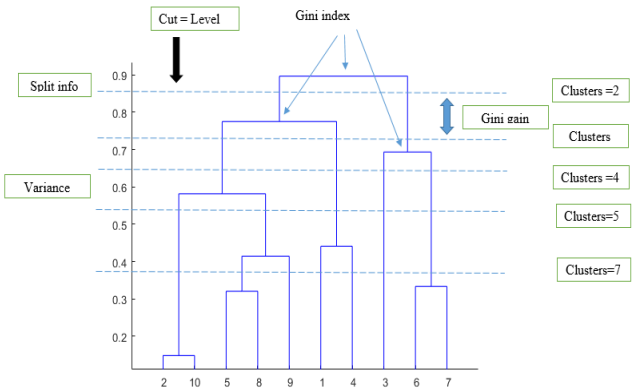

Fig. 4. High level representation of the dendrogram of recursive hierarchical clustering algorithm.

In Recursive Hierarchical clustering algorithm, we combine the approach of hierarchical clustering and decision tree to detect the optimal number of clusters. The completed dendrogram is used to calculate the Gini index (Gini Impurity) of each split at each level. Dendrogram is assumed as a decision tree and the gini index of each split is calculated recursively. Gini index of a split is a measure of impurity. Using the gini indexes of each split gini index of a level is calculated. Maximum gini index out of gini indexes of splits in the same level is chosen as the gini index of a level. Gini index of a leaf node is calculated using (8).

$$\text{Gini Index} = 1 - \sum_j Pj2 \quad (8)$$

where j is class and P_j is probability of class (proportion of samples with class j). Gini index of a split is calculated as in (9).

$$\text{Gini index}_{\text{split}}(N) = \frac{S_{\text{left}}}{S}\text{gini}(N_{\text{left}}) + \frac{S_{\text{right}}}{S}\text{gini}(N_{\text{right}}) \quad (9)$$

where S_{left} the number of samples in the left node, S_{right} is the number of samples in the right node, S is the total number of samples, N_{left} is the left child and N_{right} is the right child.

$$\text{Gini level} = \text{Max (Gini split of same level)} \quad (10)$$

Gini Difference between two consecutive levels is calculated using gini indexes of two levels. Gini difference is a measure of information gain or the drop of impurity. We choose the two levels with the highest gini difference or of highest drop calculated using (11).

$$\text{Gini diff} = \text{Gini level } (i) - \text{Gini level } (i+1) \quad (11)$$

To choose between the upper level and the lower level of the highest gini drop we use "Gain Index" (Gain ratio). Even though lower gini index reveals higher purity of a split, number of data points in a cluster and the number of clusters is a tradeoff. At the leaf nodes the purity of the split is higher but each cluster contains only individual elements. Gain ratio is a modification of information gain that reduces its biasness. Gain ratio takes number and size of branches into account when choosing a split. Gain ratio overcomes the tradeoff of Gini index through taking intrinsic information into account. Therefore we calculate the Gain ratio for each level using (12) and (13).

$$\text{Gain ratio (A)} = \frac{\text{Gain / Gini difference (A)}}{\text{Split Info (A)}} \quad (12)$$

$$\text{Split Info A (D)} = \sum_{j=1}^{v} \frac{|Dj|}{|D|} * \log_2 \frac{|Dj|}{D} \quad (13)$$

where Dj is the number of elements of a class and D is the total number of elements. Higher the gain ratio better the split. Therefore we chose the level with higher gain ratio from the higher and lower levels of the maximum gini difference.

Using the Recursive Hierarchical clustering algorithm, we select the optimal number of clusters for a dataset.

Recursive Hierarchical clustering algorithm
Input = Data set of`d' number of data points
.
Until d==1
Calculate pairwise distances among the data points or clusters using Euclidean distance.
Merge two data points or data clusters of the least pairwise distance.
End Until
Compose the dendrogram of the hierarchical tree.
For each split {
Calculate the gini index
If (split is leaf node)

$$\text{Gini Index} = 1 - \sum_j Pj2 \quad (8)$$

Else

$$\text{Gini index}_{\text{split}}(N) = \frac{S_{\text{left}}}{S}\text{gini}(N_{\text{left}}) + \frac{S_{\text{right}}}{S}\text{gini}(N_{\text{right}}) \quad (9)$$

For all levels of the dendrogram {
Calculate Gini index of a level

Gini index of a level = Gini $_{level}$ = Max (Gini $_{split}$ of same level) (10)

Calculate Gini Ratio of a split.

$$\text{Gain ratio (A)} = \frac{\text{Gain / Gini difference (A)}}{\text{Split Info (A)}} \quad (11)$$

$$\text{Split Info A (D)} = \sum_{j=1}^{v} \frac{|Dj|}{|D|} * \log_2 \frac{|Dj|}{D} \quad (12)\}$$

Calculate gini differences between consecutive levels.

Select the two levels which results in the highest gini difference.

Best split = Split with higher Gain ratio and the highest gain difference

IV. Experiments and Results

Experiments have been conducted on clustering many datasets using multiple clustering techniques and the results have been cross validated against the results of recursive hierarchical clustering algorithm. Clustering results of the 'Fisher's Iris dataset with K means clustering, hierarchical clustering, Self-Organizing Maps are summarized and discussed here.

In K means clustering we have to predefine the number of clusters as an input to the algorithm. This is identified as a pitfall of K means clustering because there exists a need for identifying the possible number of clusters using another clustering approach such as SOM to detect the probable distribution of clusters for better information gain.

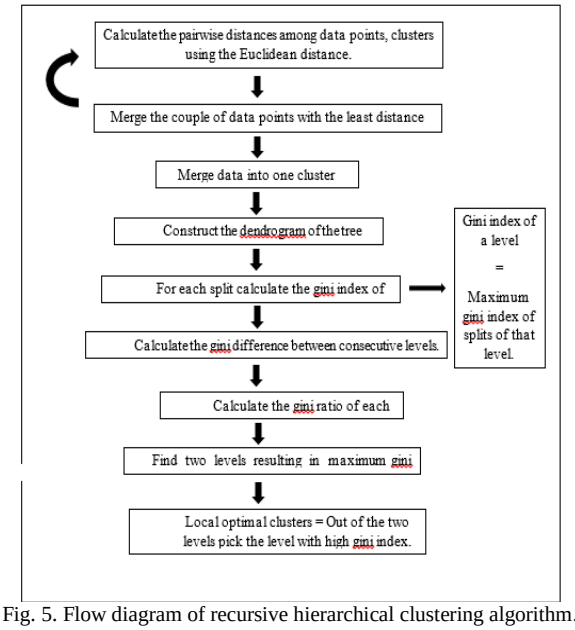

Fig. 5. Flow diagram of recursive hierarchical clustering algorithm.

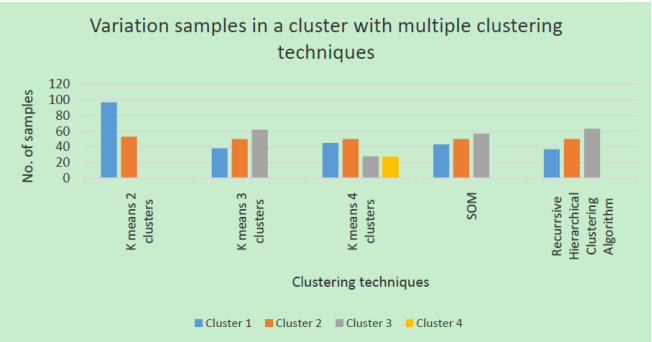

Fig. 6. Results of cluster samples variation using multiple clustering techniques.

K means clustering differentiating the number of clusters is tested. The results or the distribution of data of the Fisher's Iris Data set within two, three and four clusters and are compared with the results obtained using recursive hierarchical clustering algorithm and graphically depicted in Fig. 6. Recursive hierarchical clustering detects the optimal cluster number as 3 for Fishers Iris dataset. The distribution of samples among clusters using recursive hierarchical clustering algorithm is almost similar to the K means clustering algorithm of 3 clusters.

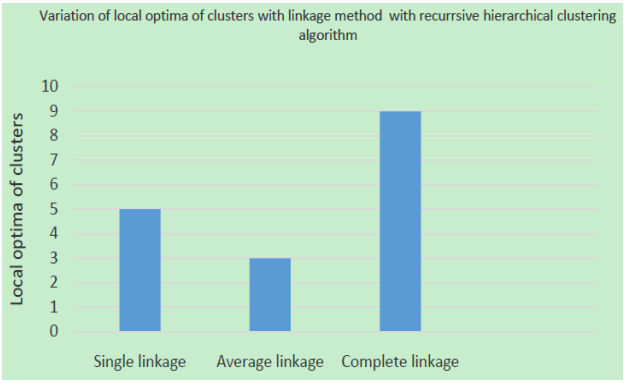

Fig. 7. Variation of the optimal clusters with linkage variation.

Optimal number of clusters of a dataset differs with the change of the linkage method in recursive hierarchical clustering algorithm. Fishers Iris dataset was used to conduct experiments by varying the linkage method and the results are summarized in Fig. 7.

SOM uses statistical measures such as mean and variance for the competitive learning process. Competitive learning is based on these measures. Euclidean distance is the bottom line for calculating mean and variance. Therefor the data becomes meaningless when it is encoded into binary format.

K means algorithm also clusters data based on the Euclidean distance. There is no explicit control of the algorithm and the only control is the ability to define the number of clusters explicitly. Even though we explicitly define the number of clusters, optimal number of clusters is derived by considering the intra cluster variation among the data samples of the clusters and the inter cluster variations between the clusters.

Decision trees focus on the gini index and information gain in developing the decision tree. Decision tree considers about the number of samples contained within the clusters but it does not consider about the variance of data of a single branch that is the spread of data. Therefor decision trees gives control over the number of samples within a cluster and not over the spread or the variance of data in the clusters.

Hierarchical clustering concerns about the variance of clusters. The sigma value of the dendrogram gives the variance in the clusters. But this algorithm does not consider about the information gain of data.

Recursive Hierarchical Clustering algorithm concerns all factors of information gain, gini index and variance or the spread of data in clusters. This algorithm finds the clusters with similar variances or spreads of data. Each cut along the y axis of the dendrogram detects clusters of a sigma (value of y axis at the cut) variation. Then we choose the level with the highest information gain to detect the optimal number of clusters. This algorithm combines the features of the hierarchical clustering and the decision tree. This can be used

to cluster both label data and real data.

REFERENCES

[1] F. Long, H. Zhang, and D. D. Feng, "Fundamentals of content-based image retrieval," *Multi-media Information Retrieval and Management*, Springer Berlin Heidelberg, (2003), pp. 1-26.
[2] C. H. C. Leung and Y. Li, *Semantic Image Retrieval*, 2015.
[3] A. Alzu'bi, A. Amira, and N. Ramzan, "Semantic content-based image retrieval: A comprehensive study," *Journal of Visual Communication and Image Representation*, vol. 32, pp. 20-54, 2015.
[4] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," CVPR, 2009.
[5] Attribute Datasets. (April 10, 2017). [Online]. Available: https://www.ecse.rpi.edu/homepages/cvrl/database/AttributeDataset.htm.
[6] J. Deng *et al*., "ImageNet: A large-scale hierarchical image database," in *Proc. 2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248-255.
[7] M. R. Anderberg, "Cluster analysis for application," Academic Press, 1973.
[8] P. N. Tan, "Cluster analysis: Basic concepts and algorithms," *Introduction to Data Mining*, pp. 48-559, 2006.
[9] O. Maimon and L. Rokach, "Data mining and knowledge discovery handbook," Springer Science Business Media Inc, pp. 321-352, 2005.
[10] A. Huang, "Similarity measures for text document clustering," NZCSRSC 2008, April 2008, Christchurch, New Zealand.
[11] L. V. Bijuraj, "Clustering and its Applications," in *Proc. National Conference on New Horizons in IT – NCNHIT*, 2013.
[12] D. M. Blei. (September 5, 2007). Clustering and the k-means Algorithm. [Online]. Available: http://www.cs.princeton.edu/courses/archive/spr08/cos424/slides/clustering-1.pdf
[13] P. Stefanovic and O. Kurasova, "Visual analysis of self-organizing maps," *Nonlinear Analysis: Modelling and Control*, vol. 16, no. 4, pp. 488–504, 2011.
[14] S. Krüger. *Self-Organizing Maps*. [Online]. Available: http://www.iikt.ovgu.de/iesk_media/Downloads/ks/computational_neuroscience/vorlesung/comp_neuro8-p-2090.pdf
[15] T. Kohonen, *Self-Organizing Maps*, 3rd edition, Springer Ser. Inf. Sci., Springer-Verlag, Berlin, 2001.
[16] A. K. Mann and N. Kaur, "Survey paper on clustering techniques," *International Journal of Science, Engineering and Technology Research*, vol. 2, issue 4, April 2013.
[17] R. Tibshirani. (September 14, 2009). Distances between clustering, Hierarchical clustering. [Online]. Available: http://www.stat.cmu.edu/~cshalizi/350/lectures/08/lecture-08.pdf
[18] S. Sayad. Hierarchical clustering. [Online]. Available: http://www.saedsayad.com/clustering_hierarchical.htm
[19] R. Tibshirani. (January 29, 2013). Clustering 2: Hierarchical clustering. [Online]. Available: http://www.stat.cmu.edu/~ryantibs/datamining/lectures/05-clus2-marked.pdf
[20] M. S. Yang," A Survey of hierarchical clustering," *Mathl. Comput. Modelling*, vol. 18, no. 11, pp. 1-16, 1993.
[21] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Computing Surveys*, vol. 31, no. 3, September 1999.
[22] Hierarchical Clustering Algorithms. [Online]. Available: http://home.dei.polimi.it/matteucc/Clustering/tutorial_html/hierarchical.html
[23] K. Sasirekha and P. Baby, "Agglomerative hierarchical clustering algorithm — a review," *International Journal of Scientific and Research Publications*, vol. 3, issue 3, March 2013.
[24] M. Roux, "A comparative study of divisive hierarchical clustering algorithms," 2015.
[25] N. Rajalingam and K. Ranjini, "Hierarchical clustering algorithm — A comparative study," *International Journal of Computer Applications*, vol. 19, no. 3, pp. 0975–8887, April 2001.
[26] T. M. Lakshmi, A. Martin, R. M. Begum, and V. P Venkatesan, "An analysis on performance of decision tree algorithms using student's qualitative data," *I. J. Modern Education and Computer Science*, vol. 5, pp. 18-27, June 2013.

Pavani Y. De Silva was born in Colombo, Sri Lanka in October 1992. She received her BSc. (Hons.) degree in information technology with first class honors from University of Moratuwa, Sri Lanka in 2017. She works as a software engineer at IFS R & D International (Pvt) Ltd. Her current research interests include machine learning, data mining, big data analytics and data science.

Chiran N. Fernando was born in Sri Lanka in November 1991. He graduated from University of Moratuwa, Sri Lanka with a BSc. (Hons.) degree in information technology in 2017. He has worked at Virtusa Polaris Pvt. Ltd as a software engineer intern. He currently works as a software engineer at WSO2 Lanka (Pvt) Ltd. His current research interests are machine intelligence, deep learning and artificial neural networks.

Damith D. Wijethunge was born in September 1991 in Kandy, Sri Lanka. He has earned his bachelor's honours degree in the field of information technology from University of Moratuwa in 2017. He has worked at Virtusa Polaris Pvt Ltd, SquareMobile Pvt Ltd as a software engineer intern. Currently he is working at DirectFN Ltd, Sri Lanka as a software engineer. He is interested in image processing, deep learning, artificial neural network, big data analytics and data mining.

Subha D. Fernando graduated from University of Kelaniya, Sri Lanka with BSc. Special (Hons) Statistics and computer science in 2004 and from Nagaoka University of Technology with master of engineering (M. Eng.), Management information systems science in 2010. She completed her PhD degree in computational intelligence from Nagaoka University of Technology in 2013. She is the head of the Department of Computational Mathematics, Faculty of Information Technology, University of Moratuwa, Sri Lanka. She is the president of Sri Lanka Association for Artificial Intelligence. Her current research interests are machine learning, deep learning, artificial neural networks, multi-agent-systems and intelligent informatics.