World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:8, No:8, 2014

A Comprehensive Review on Different Mixed Data Clustering Ensemble Methods

S. Sarumathi, N. Shanthi, S. Vidhya, M. Sharmila

Open Science Index, Computer and Information Engineering Vol:8, No:8, 2014 publications.waset.org/9999281.pdf

***Abstract*—**An extensive amount of work has been done in data clustering research under the unsupervised learning technique in Data Mining during the past two decades. Moreover, several approaches and methods have been emerged focusing on clustering diverse data types, features of cluster models and similarity rates of clusters. However, none of the single clustering algorithm exemplifies its best nature in extracting efficient clusters. Consequently, in order to rectify this issue, a new challenging technique called Cluster Ensemble method was bloomed. This new approach tends to be the alternative method for the cluster analysis problem. The main objective of the Cluster Ensemble is to aggregate the diverse clustering solutions in such a way to attain accuracy and also to improve the eminence the individual clustering algorithms. Due to the massive and rapid development of new methods in the globe of data mining, it is highly mandatory to scrutinize a vital analysis of existing techniques and the future novelty. This paper shows the comparative analysis of different cluster ensemble methods along with their methodologies and salient features. Henceforth this unambiguous analysis will be very useful for the society of clustering experts and also helps in deciding the most appropriate one to resolve the problem in hand.

***Keywords*—**Clustering, Cluster Ensemble Methods, Co-association matrix, Consensus Function, Median Partition.

I. Introduction

CLUSTERING is one of the most important bedrock processes in Data Mining. It holds a crucial role in the other fields like Spatial Data Extraction, Machine Learning process, Information retrieval, Pattern Recognition, World Wide Web and Image Processing. Data clustering deals with the process of grouping a set of objects in vector space based on their proximity [1]. The destination of the cluster analysis is to see resemblances between data according to the uniqueness and to group these related data objects together as clusters. A large number of clustering algorithms have been proposed from earlier periods. On the contrary, there is no single clustering method that is able to offer accurate and appropriate cluster results [2]. By putting a clustering algorithm to the data sets, it performs the similarity or dissimilarity measures among the instances in the dataset. But if two different clustering algorithms are applied to the same data set, and then it will provide diverse cluster solutions. Hence it will be very complicated to estimate the exact clustering results. This estimation is linked with the use of Cluster Validity Indexes which is used to determine the quality of clustering results [2]. Nevertheless, to overcome this severe concern, we can combine multiple clustering approaches in a single ensemble framework, which may permit the user to obtain the benefit of the strengths of individual clustering approaches. The solutions from diverse base clustering methods are combined to form a final partition [3]. This Meta level approach engages the following two main tasks, namely generation of a cluster ensemble and then development of a final partition usually denoted as the consensus function [3],[4]. The major challenge in clustering ensemble is the description of the most suitable consensus function, which is capable to heighten the effects of the single clustering algorithm.

II. Overview of Cluster Ensemble Technique

Cluster ensemble is a process for obtaining consensus results which can be formed by grouping up with several clustering solutions. The consensus result is based upon merging various partitions that contain well-defined rules. So the cluster ensemble is considered to be more robust. The visualization tool is used to determine the cluster membership, the number and boundaries. For generating most perfect clusters, it has ensemble clustering as a main approach; this may be possible by the individual clustering approach [4]. It contains two tasks, such as Generation step and Consensus step. The common fundamental structure of the cluster ensemble was exposed in Fig. 1 [3].

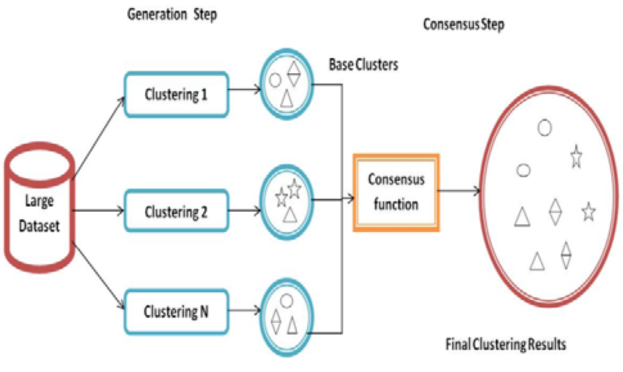

Fig. 1 Fundamental Process of Cluster Ensemble

Mrs.S.Sarumathi, Associate Professor, is with the Department of Information Technology, K. S. Rangasamy College of Technology, Tamil Nadu, India (phone: 9443321692; e-mail: rishi_saru20@rediffmail.com).
Dr.N.Shanthi, Professor and Dean, is with the Department of Computer Science Engineering, Nandha Engineering College, Tamil Nadu, India (e-mail: shanthimoorthi@yahoo.com).
Ms.S.Vidhya, PG Scholar, is with the Department of Information Technology, K. S. Rangasamy College of Technology, Tamil Nadu, India (phone: 9443960666; e-mail: vidhyapsubramani@gmail.com).
Ms.M.Sharmila, Assistant Professor, is with the Department of Information Technology, M.Kumarasamy College of Engineering, Tamil Nadu, India (phone: 9443581688; e-mail: sharmi28.it@gmail.com).

A. Generation Step

Basically, in generation step, there are no limits available for the partition that should be obtained. For producing various base cluster solutions, the creation process [2] various clustering algorithms or with various parameter initialization of the same algorithm, various representations of objects, projections of objects on various subspaces and objects based on their subsets as shown in Fig. 2. When there is a proper consensus function, then the generation step can able to produce a high quality consensus cluster even if it has a weak cluster algorithm.

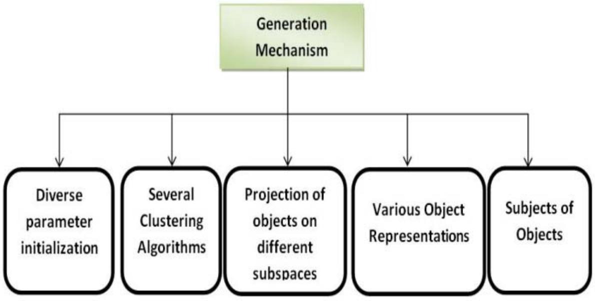
Generation Mechanism
Diverse parameter initialization
Several Clustering Algorithms
Projection of objects on different subspaces
Various Object Representations
Subjects of Objects

Fig. 2 Generation Steps of Primary Cluster Ensemble

B. Consensus Step

By using this consensus step, the consensus functions are produced and also useful for obtaining the ultimate data partition from result of various base clusters. The outcome of the single clustering algorithm has been improved by the consensus function. This includes two methods such as median partition and object co-occurrence. In the first method, it involves the partition in the cluster ensemble which exploits their resemblance with all other partitions. The improper analysis of the variation measures gives the complexity of this median partition. In the second method, it deals with an individual cluster for determining the number of occurrences of an object and in same cluster it can be evaluated that how many times two objects belong together. Although there are various queries that have been raised from these methods such as, among various clustering algorithms which one will be used? Which are the parameters that will be applicable? What are the exact variation measures? What are the best experimental methods used to find the solution for the problem or to reach near the solution? [2]. To give answers to these questions, there are set of clustering ensemble approaches are estimated over recent years.

III. Different Cluster Ensemble Methods

The below section describes on few different methods of cluster ensemble. The features and methodology of each process are described in brief.

A. Link-Based Cluster Ensemble (LCE)

High resolution Satellite Precipitation (SPE) is supported by the Precipitation Estimation from Remotely Sensed Imagery using an Artificial Neural Network Cloud Classification (PERSIANN-CSS) algorithm. This modified SPE with the inclusion of LCE (link-based cluster ensemble) [5] involves the following four steps: 1) segmentation of infrared cloud images into patches; 2) Cloud patches feature extraction; 3) clustering cloud patches using LCE; and 4) dynamic application of brightness temperature (Tb) and rain-rate relationship, derived using satellite observations [6].

The objective of the training mode is to obtain the parameters, such as the temperature-rain rate relationship curve and classification weights for each cluster. At first, the raw infrared images received from the GOES-12 satellite are regulated into cloud-top brightness temperature images. Then the images are segmented into patches by using a region growing method. The next step is feature extraction, in which the features like statistics (minimum, mean, standard deviation of brightness temperature of each patch at the corresponding thresholds [6]), geometry (area and shape index of each patch [6]),and texture (wavelet features, Grey-Level Co-occurrence Matrix i.e., GLCM) are extracted at the cloud patch temperature thresholds of 220 K, 235 K and 255 K respectively. Then, the patches are categorized into 100 clusters using the link-based cluster ensemble technique. Note that the SOM is utilized for clustering in the PERSIANN-CCS algorithm. Next, a Temperature-Rain-rate (T-R) curve is allocated to each cluster. For getting this T-R pixel pair (obtained from GOES-12 observations and Nexrad Stage IV rainfall) is redistributed with the help of Probability Matching Method (PMM) [7]. The resultant T-R transformation is fitted by a curve fitting method [6], [8].

In the testing mode, segmentation and feature extraction are done similar to training mode, with the difference being that the selected features of each patch are compared to the weights of each cluster in classification step and the most similar cluster is chosen. The rain-rate estimation of the patches is calculated based on the T-R curve of the chosen cluster and the infrared pixel values of the patch.

The link-based cluster ensemble technique comprises of three stages [9]: 1) M base clustering, 2) Computing the Cluster-association Matrix (CM), and 3) final clustering using a consensus function. In LCE, the data are segregated by various base clusters; the CM matrix is consumed by using Weighted Connected Triples (WCT). The outcome of consensus clustering can be used for spectral clustering (SPEC) by applying CM.

In the initial stage, by using single clustering the M base cluster can be achieved. For example, the K-Means can be done with various initializations and several clustering algorithms. Here we have observed various clustering algorithms such as K-Means, SOM, Agglomerative hierarchical clustering and base clustering as Fuzzy C-Means [10].

In the second stage, the CM can be produced. An association degree can be represented by every entry in the matrix, which is taken among each sample and also each cluster. The cluster association matrix [9] can be calculated as shown below,

$$\mathrm{CM}(\mathrm{x}_i,\mathrm{cl}) = \begin{cases} 1 & \text{if cl} = \mathrm{C}^* \\ \mathrm{sim}(\mathrm{cl},\mathrm{C}^*) & \text{otherwise} \end{cases} \quad (1)$$

Here $\mathrm{x_i}$ and cl are corresponding sample and a cluster and sim(cl,C*) is the similarity among cluster cl and clusterC*. Depend on connected-triple method [9], for every pair of clusters the three sub graph clusters with non-zero of two edges are measured. The similarity among two clusters are computed as,

$$\mathrm{sim}\left(\mathrm{C_i},\mathrm{C_j}\right)= \frac{\mathrm{WCT_{ij}}}{\mathrm{WCT_{max}}} \times \mathrm{DC} \quad (2)$$

The third stage, to attain the final clustering, a consensus function can be used. The consensus function is nothing but a graph based clustering in which the weighted bipartite graph is transmuted as CM and also SPEC is produced. A weighted bipartite graph is built in which every sample and cluster contains the vertices of the graph. The weight edges among vertices are calculated as,

$\mathrm{w_{ij}}$ = 0, when both verticesv$_\mathrm{i}$ and v$_\mathrm{j}$ are samples;

$\mathrm{w_{ij}}$ = 0, when both verticesv$_\mathrm{i}$ and v$_\mathrm{j}$ are clusters;

$\mathrm{w_{ij}}$=$\mathrm{CM}\left(\mathrm{v_i},\mathrm{v_j}\right)$, where vertexv$_\mathrm{i}$ is sample and v$_\mathrm{j}$ is a cluster.

After getting this graph, to present the final data partition the SPEC [11] can be applied.

B. Bipartite Graph-Based Consensus Maximization (BGCM)

In numerous applications, data integration is difficult to accomplish because of the different data source formats and inability to access the raw data. In these cases, it is a better approach to combine multiple information sources at a higher level than to integrate individual raw data, thus providing an improved solution, as multiple sources can provide complimentary knowledge. The process of combining multiple classification models is done by ensemble learning and studies reveal that incorporating unsupervised models into classification ensembles can progress prediction accuracy.

Supervised models (classifiers) have data that are pulled from the raw data and supervised models have data pulled from clustering models. Let's consider the common problem of combining the outputs of multiple supervised and unsupervised models to increase prediction accuracy. Assume we have a set of objects $X = \{x_1,\ldots,x_n\}$ from c classes and m models to provide information about the classification of X. The first r of them is (supervised) classifiers, and the remaining are (unsupervised) clustering models. The goal here is to envisage class label of x$_\mathrm{i}$Є X. This class label must agree with the base classifier predictions, as well as should satisfy the constraints enforced by the clustering models as much as possible. We refer to this problem as consensus maximization.

The challenging issue here is that it cannot be resolved by simple majority voting as the correspondence between the cluster ID and the class label is unknown, and the same cluster ID in diverse clustering models may represent different clusters. The other issue is while trying to get a global optimal prediction for the target objects to obtain maximum agreement among various models; the search space will become exponential.

To rectify this problem, we can summarize the base model outputs using a bipartite graph in a lossless manner. In this, bipartite graph, we define an optimization problem with deviations from the base classifier predictions and the discrepancies of the predicted class labels among nearby nodes. Then this optimization problem can be resolved using a coordinate descent method and obtaining a global optimal label assignment for the target objects.

Consider we have the outputs of r classification models and *(m-r)* clustering models for a set of objects $X = \{x_1,x_2,\ldots,x_n\}$. For simplicity, we assume that each object is allocated to only one class or cluster by each of the *m* models, and the number of clusters in each clustering model is *c*, the same as the number of classes. Here, the cluster ID *z* may not be related to class *z*. Each of one base model divides X into *c* groups, and there is a total of *v=mc* groups, where the first $s{=}rc$ groups are created by classifiers and the remaining *v-s* groups are created by clustering algorithms.

In bipartite graphs [12], we have the object nodes $x_1,\ldots x_n$ and group nodes $g_1,\ldots,g_v$. If the object is assigned to the group by one of the models, then both are said to be connected. A group that is attained by a classification model which link to the node that corresponds to ground-truth label. The affinity matrix $A_{n\times v}$is,

$$a_{ij} = \begin{cases} 1 & x_i \text{is assigned to group j by a model} \\ 0 & \text{otherwise .} \end{cases} \quad (3)$$

The predicted label can be indicated by indicator variable u_{iz}, for object x_i,

$$u_{iz} = \begin{cases} 1 & \text{the ensemble assigns } x_i \text{ to class } z, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

Then, $\vec{u}_i$ is a row, which has one nonzero entry and class that the consensus method assigns x_i. Placing all the predictions for the data set X together, we get a$n\times c$ matrix U. To have confidence information associated with the final label prediction, we substitute the constraint u_{iz}(which must be 0 or 1) with the weaker constraint that each variable belongs to interval [0, 1]. Now u_{iz}represents the conditional probability of object x_i belonging to class *z*. The conditional probabilities at each group g_i(summarized as $U_{n\times c}$ for object nodes and $Q_{v\times c}$for group nodes) are estimated as nuisance parameter.

For every entry of matrices, $u_{iz}$and q_{iz}, represents the probability of object x_i and group $g_j$belonging to class *z* correspondingly.

$$u_{iz} = \hat{P}(\delta_{iz} = 1|x_i) \quad \text{and} \quad q_{jz} = \hat{P}\left(\delta_{jz} = 1|g_j\right) \quad (5)$$

The indicator variables are δ_{iz} or δ_{jz},, which denotes x_i or $g_j$fits to class *z* when the value is 1. The initial $s{=}rc$ groups are attained by classifiers where the initial class label estimates are represented by $Y_{v\times c}$

$$y_{jz} = \begin{cases} 1 & \mathrm{g_j}\text{s predicted label is } z, j = 1,\ldots,s, \\ 0 & \text{otherwise.} \end{cases} \quad (6)$$

where k_j= $\sum_{z=1}^{c} y_{jz}$. The below optimization problem on the graph is used to express the consensus agreement.

$$\mathbf{P:}\min_{Q,U} \varphi(Q,U)=\left(\sum_{i=1}^{n}\sum_{j=1}^{v} a_{ij}\left\|\vec{u}_i-\vec{q}_j\right\|^2+\alpha\sum_{j=1}^{v} k_j\left\|\vec{q}_j-\vec{y}_{j.}\right\|^2\right)(7)$$

$$\text{s.t.}\quad \vec{u}_i \geq \vec{0}, |\vec{u}_{i.}| = 1,\ i = 1{:}n \tag{8}$$

$$\vec{q}_{j.} \geq \vec{0},\ \left|\vec{q}_j\right| = 1,\ j = 1{:}v, \tag{9}$$

Here ||.|| and |.| indicates vector's L2 and L1 norm correspondingly. The first term guarantees that if an object x_i is assigned to group $g_i$by one of the models, then their conditional probability estimates for the category label must be close. The second term enforces the constraint that the group g_i's consensus class label estimate should not deviate too much from its initial class label prediction. If $j = s +$ *1,...,v,*$g_i$is from an unsupervised model group with no such constraints, α is the shadow price payment for violating the constraints. So, k_j= 0 and the weight of the constraint is 0. Hence, $\overrightarrow{u_l}$and $\overrightarrow{q_j}$ are probability vectors, so that each component must be greater than or equal to 0, and the sum of the components must be 1.

We recommend, block coordinate descent method to solve this problem. We can stable the value of U, at the *t*th iteration, and then the main function will be adding up of all v quadratic components which correspond to$\overrightarrow{q}_j$. This is strictly convex and $\nabla q_j.\varphi(Q,U^{t-1}) = 0$ which corresponds to $\vec{q}_j$ gives the unique global minimum cost function.

$$\vec{q}_{j.}{}^{(t)} = \frac{\sum_{i=1}^{n} a_{ij}\vec{u}_{i.}{}^{(t-1)}+\alpha k_j\vec{y}_j}{\sum_{i=1}^{n} a_{ij}+\alpha k_j}. \tag{10}$$

Certainly the unique global minimum by fixing Q with respect to $\overrightarrow{u_l}$ is attained by,

$$\vec{u}_{i.}{}^{(t)} = \frac{\sum_{j=1}^{v} a_{ij}\vec{q}_{j.}{}^{(t)}}{\sum_{j=1}^{v} a_{ij}}. \tag{11}$$

In BGCM algorithm [12], the matrix form of update steps are provided. D_v = diag$\{(\sum_{i=1}^{n} a_{ij})\}\, v \times v$ and D_n = diag$\{\sum_{j=1}^{v} a_{ij}\}\, n \times n$ are normalized factors.K_v = diag$\{(\sum_{z=1}^{c} y_{jz})\} v \times v$ specifies the existence of constraints on group nodes. While performing each iteration, the probability can be estimated at each group node by associating their initial values *Y* and the node's neighboring object nodes, extracts their information. When updating U, it propagates the updated probability which estimates back to its neighboring object nodes. This is directed prove that $(Q^{(t)},U^{(t)})$ meets to a stationary point of optimization problem [13].

C. Improved Link-Based Cluster Ensemble (ILCB)

The Link-based method (LCE) is introduced in [15], [16] to attend and use cluster ensemble associations to their true potential. The base clustering results are modeled as a link network, which is used to systematically obtain the relations between and within the decisions. This is done through the link-based similarity measure named 'Weighted Connected Triple (WCT)' [14]. Then the disclosed relations are broken to refine the conventional meta-level matrix which is considered to be the center of several benchmark techniques. Researches show that the resulting technique executes constantly better compared to many state-of-the-art alternatives on both UCI benchmark and gene expression datasets. This has led to the recent improvement of LCE, thus leading to a new link-based similarity measure, Weight Triple Uniqueness (WTU) for underlying similarity assessment. WTU is as good as WCT and it also makes use of more information which is already available in a network for similarity measure estimation. Hence, the quality of information matrix is improved, thus improving the final clustering.

There are three major steps in the improve framework of Link-based method (LCE): (i) Creation of a cluster ensemble Π, (ii) aggregation of base clustering results, π_g Є Π, g = 1….M, into a meta-level data matrix Θ, and (iii) generation of the final data partitionπ^* with the help of the spectral graph partitioning (SPEC) algorithm.

1) Creating Cluster Ensembles

The approach proposed here is generic so that it can be coupled with diverse ensemble generation methods. Here we are taking Fixed-k and Random-k cluster ensemble for our investigation. Based on the original work [15], base clustering is formed using the classical k-means and each of these clustering is initialized with a random set of cluster prototypes.

Fixed-k: The data set X Є $\mathbb{R}^{N\times D}$with all D attributes is used to create each clusteringπ_g Є Π, in which the number of clusters is fixed to k = $\lceil\sqrt{N}\rceil$ k becomes 50 if $\lceil\sqrt{N}\rceil$> 50, in order to get a meaningful partition intuitively.

Random-k: The data set with all attributes is used to create eachπ_g, with the difference being that the number of clusters is randomly selected between$\{2,\dots,\lceil\sqrt{N}\rceil\}$. Both 'Fixed-k' and 'Random-k' generation strategies are originally introduced in the primary work [17].

2) Aggregating Base Clustering Results

Now, after obtaining the cluster ensemble Π, the aggregation of these base clustering results into a single information matrix Θ Є$[0,1]^{N\times P}$, is performed. Here P represents the total number clusters in the ensemble which are under examination. A matrix entry $\Theta(x_i,cl)$is calculated for each clustering π_gЄ *Π* and their corresponding cluster C_1^g,…,$C_{k_g}^g$. This matrix entry basically symbolizes the association degree that the sample x_i Є X has with each cluster cl Є $\{C_1^g,\dots,C_{k_g}^g\}$ and this can be predicted as follows,

$$\Theta(x_i,cl)=\begin{cases}1 & if\ cl = C_*^g(x_i)\\ sim\left(cl, C_*^g(x_i)\right) & otherwise\end{cases} \tag{12}$$

Here $C_*^g(x_i)$ represents the cluster label to which sample $x_i$has been assigned. Also, $sim(C_x,C_y)$Є [0,1] represents the similarity among any two clusters $C_x, C_y \in \pi_g$, which can be found using the link-based algorithm presented next.

Weighted Triple Uniqueness (WTU) Algorithm: It has been primarily developed to assess the similarity among any pair of clusters $C_x, C_y \in \Pi$. 'Connected-Path' algorithm, which is introduced in [18] for the task of alias for detection is used to develop uniqueness measure and WCT is based on this measure. Given a graph $\mathrm{U}Q_{ij}^{k}$of any two objects i and j (denoted by vertices $v_i, v_j \in$ V) can be derived from each joint neighbor k (denoted by the vertex $v_k \in V$) as follows:

$$\mathrm{U}Q_{ij}^{k} = \frac{f_{ik} + f_{jk}}{\sum_m f_{mk}} \tag{13}$$

Here f_{ik}= frequency of the link between objects i and k occurring in data; f_{jk}= frequency of the link between objects j and k; f_{mk}= frequency of the link between object k and any object m

WTU is measured as an expansion to the WCT initially proposed by the LCE model. Along with maintaining the efficiency, it uses the additional information that is already available within a network, thus increasing quality of similarity measure compared to the similarity measure derived by WCT. In the beginning of WTU evaluation, the ensemble Π is denoted as a weighted graph G = (V,W), here V represents the set of vertices in which each cluster is indicated in Π and W is a set of weighted edges among clusters. The weight $|w_{xy}| \in [0,1]$ allotted to the edge $w_{xy} \in$ W among $C_x, C_y \in$ V, is calculated as,

$$|w_{xy}| = \frac{L_x \cap L_y}{L_x \cup L_y} \tag{14}$$

Here $L_z \subset$ the set of samples which belong to cluster$C_z \in \Pi$. Note G is an undirected graph such that $|w_{xy}|$corresponding to$|w_{yx}|$,$\forall C_x, C_y \in$ V.

3) Generating Final Data Partition

After the creation of the matrix Θ, the spectral graph partitioning (SPEC) algorithm [19] is used for the generation of final data partition. This technique is primarily introduced by [20] in the Hybrid Bipartite Graph Formation (HBGF) framework. Specifically SPEC is used to divide a bipartite graph into K clusters, with the graph being transformed from the matrix $\Theta' \in \{0,1\}^{N\times P}$ (a crisp variation of Θ). Taking this into consideration, HBGF can be measured as the baseline model of LCE, where a more distinguished information matrix is exploited to increase the solution accuracy. The procedure of generating the final data partition $\pi^*$from Θ is summarized as follows.

Initially, the matrix Θ is used to construct a weighted bipartite graph $G = (V, W)$, here $V = V^X \cup V^C$ is a set of vertices denoting equally the samples $V^X$and clustersV^C, and a set of weighted edges is represented by W. The edge w_{ij}contain the weight $|w_{ij}|$and connecting vertices $v_i, v_j \in$ V, can be expressed as,

$|w_{ij}|$=0 when $v_i, v_j \in V^X$ or $v_i, v_j \in V^C$.

$W \in [0,1]^{(N+P)\times(N+P)}$ can be represented as

$$W = \begin{bmatrix} 0 & \Theta \\ \Theta^T & 0 \end{bmatrix} \tag{15}$$

The matrix $U = [u_1 u_2 \dots u_K]$is produced by largest eigenvectors $u_1, u_2, \dots. u_K$ of $W$where in columns the eigenvectors are stacked. After that, the other matrix can be formed $U^* \in [0,1]^{(N+P)\times K}$by normalizing every U's.

$$U_{ss}^* = \frac{U_{ss}}{\sqrt{\sum_{s=1}^{K} U_{ss}^2}}, \tag{16}$$

Here $s = 1,..(N+P)$.The final partition can be generated $\pi^* = \{C_1^*, \dots, C_K^*\}$ of K clusters by k-means using samples in$[0,1]^K$.

D.Co-Association Based Cluster Ensemble (CABCE)

A Co-association tree (CA-tree for short), which is built using the base cluster labels, is similar to a dendogram [21]. A cut of this CA tree at a given threshold provides a preliminary partition of the data set into several disjoint groups that are similar to the preclusters. After that, we compute the co-association matrix and get the final clustering using the representatives of these groups. Based on the fact that the size of a node and all descendants of the node, the name CA-Tree is derived. The advantages of CA-tress are: (i) applicable to multi view clustering (ii) able to incorporate past clustering results (iii) reduced complexity compared to *O*(N^2) and *O(N).*

1) Co-Association Matrices

Consider that X = $\{x_1, x_2, \dots, x_N\}$is a data set that has N patterns. Let $P = \{C_1, C_2, \dots, C_k\}$ represent a crispclustering (partition) of X, where k denotes the number of clusters in P. $C_1, C_2, \dots, C_k$ clusters are disjoint nonempty subsets of X, with their union being X. It is feasible to attain many different partitions of X. Let a cluster ensemble comprise H clustering's of X:$P_1, P_2, \dots, P_H$. These are named the base clusterings of the ensemble. Let each base clustering have a different number of clusters and we can use $k_h$to denote the number of clusters in $P_h (1 \le h \le H)$. We also describe an H-element label vector for each x_i as

$$\boldsymbol{\lambda}_i = [\lambda_{i1}\lambda_{i2} \;....\; \lambda_{iH}]$$

with its *h*th element being the cluster label of x_i inP_h. We use the $d(\lambda_i, \lambda_j)$, dissimilarity is the Hamming distance between two label vectors $\lambda_i$and λ_j (i.e., the number of different cluster labels).

For every base clustering we describe an N × N matrix $S^{(h)}$ = $[s_{ij}^{(h)}]$as,

$$s_{ij}^{(h)} = \begin{cases} 1, & \lambda_{ih} = \lambda_{jh} \\ 0, & otherwise \end{cases} \tag{17}$$

The cluster ensemble contains the overall co-association matrix which is represented as $S^* = \left[s^*_{ij}\right]$

$$s^*_{ij} = \frac{1}{\mathrm{H}} \sum_{1 \le h \le H} s^{(h)}_{ij} \quad (18)$$

2) CA-Tree

a) Core Groups

In a co-association matrix, multiple patterns, share the same label vector often, thus being the primary source of size reduction. A set of patterns with the same label vector is identical to the cluster ensemble algorithms. Since they come from the same cluster in each base clustering, we can expect that they are assigned to the same cluster in the final clustering in spite of the actual cluster ensemble algorithm used. Hence, these patterns can be noted as a single entry, yielding a smaller co-association matrix and reduced computational time and memory. To achieve this, we are introducing the concept of core groups, which are defined as subsets of X provided that satisfy the condition that the two patterns $x_i$and $x_j$can belong to the same core group if and only if$\lambda_i = \lambda_j$. Let the set of the entire different label vectors in X can be denoted as Λ. As each core group consists of a unique vector, the number of core groups is defined as$\|\Lambda\|$, which is the cardinality of Λ. It is simple to comprehend that$\|\Lambda\| \le N$. The number of core groups is based on the distribution of patterns, algorithms and parameters that are used to generate the base clustering's. Note that the concept of core groups has been formerly mentioned in [22], though the method given there for identification of the core groups is only intended to work with a particular cluster ensemble algorithm and is very difficult from our approach.

b) CA-Tree Construction

If we assume that the patterns with similar label vectors are more likely to be assigned to the same cluster in the final clustering, then the computational complexity can be further reduced. This guides to the use of groups that comprises of several similar core groups as the units for building the co-association matrix, instead of the individual core groups. Such an approach yields in even lesser groups than $\|\Lambda\|$ and further reduces the computation amount and memory requirement. This results in the question of how to form these groups from the core groups. Since our goal is to decrease the computational complexity than the quadratic complexity of the original co-association matrix, we intentionally exclude any option that begins with a matrix of pairwise similarity or dissimilarity among the original patterns or the core groups. As an alternative, we incrementally grow a tree structure with each base clustering of data that has few similarities to a dendogram which is formed with hierarchical clustering algorithms. A threshold is applied to the tree to extract a set of nodes after processing all the base clustering's. Each threshold represents a group of core groups and a co-association matrix can be built with these groups as the input for final clustering extraction.

Each node of the tree comprises of one or more core groups of X. For a given node *z*, X(*z*) can be defined as the union of these core groups (i.e., the elements of X(*z*) are the original patterns) and G(*z*) can be defined as the set of label vectors that are associated with these core groups. Then we are initializing the tree with a single root node where X(*z*) = X. For each additional base clustering P, we are adding a child to *z* for each different cluster label of X(*z*) in P, only if X(*z*) of a leaf node z belongs to more than one cluster in P. The goal is to make sure that X(*z*) of a leaf node z always remains as a core group.

A Single scan through all the patterns in X can be used to perform all the processing within the outermost loop. Thus, the complexity of this algorithm is *O*(NH), without including the generation of the base clustering. As the algorithm concludes, each (*z*) will be a partial label vector that is shared among z and all its descendants. If z is considered as a leaf node, then X(*z*) will be a single core group with G(*z*) containing a single label vector, which will be *(z).* The largest possible branching factor in the tree is $k_{h_{(max)}}$ ≡ max(k_h) (1≤ $h \le H$), and the largest possible depth is *H.* We make a note that the exact tree generated is based on the ordering of the clustering. Nevertheless, we are not expecting the variation of clustering results among different orderings to be more significant than the variation among different ensembles that are generated under the same conditions.

c) Determining Nodes Sizes and Representatives

If we can define the "size" (in the space of label vectors) of a node *z* to denote the consistency of the label vectors in G(*z*), then we can easily redraw the tree more to look like a dendogram, with the vertical axis being the node size. We can cut the tree at a particular level by applying a threshold on the node size.

The distance between a label vector **λ** and a node *z* can be defined as the largest distance between **λ** and all label vectors in G*(z).*

$$d(\boldsymbol{\lambda}, z) = \max_{\lambda \in G(z)} d(\lambda, \lambda) \quad (19)$$

For each node z, the label vector **λ** in G*(z)* that decreases *d(**λ**,z)* is selected as the node representative $\lambda_R\ (z)$

$$\lambda_R(z) = arg \min_{\lambda \in G(z)} d(\lambda, z). \quad (20)$$

Next, we declare the size of a node z in the space of the cluster label as

$$D_r(z) = d(\lambda_R(z), z). \quad (21)$$

The subscript "r" denotes that its meaning is similar to radius. Conversely, one issue with this definition is that its computation occupies all the possible label vectors in G(z). For those nodes which lay several levels above the leaf nodes, their G(z) may comprise of many label vectors, thus the computation of (19)-(21) quadratic to $\|G(z)\|$and, hence too time consuming. This is incompatible with our overall objective to reduce computational complexity. So we use $D_r(z)$to represent the estimated upper bound of$D_r(z)$.The definitions (19)-(21) are hence substituted with the following estimations based on the estimated D_r.First, we need to select a subset $Z(z)$of the descendants of (19) is substituted with

$$d(\lambda,\mathbf{z})=\min\{H,\max_{z\in Z(z)}[d(\lambda,\lambda_R(\mathbf{z}))+D_r(\mathbf{z})]\} \qquad (22)$$

The node representative is now determined as,

$$\lambda_R(z)= arg\,\min_{\lambda\in G(z)} d(\lambda,\mathbf{z}) \qquad (23)$$

where we consider only the node representative of the nodes in $Z(z)$

$$G(z)=\{\lambda_R(z)|z\in Z(z)\} \qquad (24)$$

The size of z is now estimated as,

$$D_r(z)=d(\lambda_R(z),z). \qquad (25)$$

If z is a leaf node, we simply set $D_r(z)$=0 and $\lambda_R(z)$=(z). The actual computation of (22)-(25) for all nodes is performed in a bottom-up order; this guarantees that the processing of a node occurs only after processing all its descendants. During this process, we can also determine G(z) and X(z) according to

$$G(z)=\cup_{z,parent(z)=z} G(z) \qquad (26)$$

$$X(z)=\cup_{z,parents(z)=z} X(z) \qquad (27)$$

Now, let us go with the procedure for determining$Z(z)$. The purpose of $Z(z)$is to decrease the computational complexity for node size and representative determination from proportional to $\|G(z)\|^2$ to proportional to $\|Z(z)\|^2$ per node. So, we include a parameter $n_{des}$to control$\|Z(z)\|$. We initialize $Z(z)$to grasp only the immediate children of z and iteratively substitute the largest node in $Z(z)$with its immediate children. This process is sustained until $\|Z(z)\|\geq n_{des}$ or all the nodes in $Z(z)$are leaf nodes. Overall, $\|Z(z)\|$ satisfies the condition that

$$\|Z(z)\|\leq\min(\|G(z)\|,k_{h(max)}+n_{des}). \qquad (28)$$

The complexity of this step is consequently the number of nodes times the upper bound of $\|Z(z)\|^2$or $O\left(N.H.(k_{h(max)}+n_{des})^2\right)$, as there are no more than 2N nodes. The factor$H$ is obtained from the need to compute the Hamming distances between label vectors. The set of nodes extraction from the tree is very similar to the partition extraction from the dendogram in hierarchical clustering algorithms. We determine Z(τ), which is the extracted set of nodes given a threshold τ, according to

$$Z(\tau)=\left\{z\ \middle|\ D_r(z)\leq\tau \quad \text{and} D_r(parent(z))>\tau\right\} \qquad (29)$$

It is those nodes in Z(τ) that are used to build the co-association matrix. We also define $N_z(\tau)$=$\|Z(\tau)\|$. Let us index the nodes in Z(τ) as$z_1(\tau),z_2(\tau),\ldots,z_{Nz(\tau)}(\tau)$. A clear-cut method for computing the elements of the resulting co-association matrix, denoted as $S^*(\tau)$, is just to use the pairwise similarities among the node representatives.

$$s^*_{ij}=1-\frac{1}{H}d[\lambda_R(z_i(\tau)),\lambda_R(z_j(\tau))] \qquad (30)$$

d) Node Reduction

In many cases, several of the core groups contain very few patterns. These core groups are usually located in regions between actual clusters in the feature space or the low-density regions. Conversely, the core groups in high-density regions are more likely to comprise of more patterns. Hence, we believe it is possible to further reduce $N_Z(\tau)$ by sustaining only the important nodes (i.e., nodes that contain substantial numbers of patterns) for building the co-association matrix. Rather than specifying the number of nodes to keep, we define a parameter γ(0<γ≤1) such that we retain enough nodes, in decreasing number of patterns order, to include at least a total of γN patterns.

As the final clustering is now equated from the retained nodes, an issue remains regarding assigning the final cluster labels for the patterns in the excluded nodes. We first take out a sub tree that consists of only the retained nodes and their ancestors. For each excluded node *z*, we do a search beginning at the root of this sub tree. At each step, we choose the child that is most similar to z. This search goes on until we reach a leaf node, which is a retained node integrated in the final clustering. The final cluster label of this leaf node is then assigned to *z* as well. The complexity of this step is $O(N_Z^*(\tau).H.k_{h(max)})$. Conversely, the complexity of selecting the nodes to retain is$(N_Z^*(\tau).\log N_Z^*(\tau))$, as this involves sorting all the nodes in Z(τ) according to their number of patterns.

E. Weighted Clustering Ensemble (WCE)

In the weighted clustering ensemble module [23], a weighted consensus function has three clustering validation criteria to estimate the contribution of each partition that are received from the initial clustering analysis module. Then these candidate consensus partitions are fed to the agreement function which uses Pairwise majority voting mechanism, to form a final agreed partition.

1) Weighted Consensus Function

In this partition the Pairwise similarity among objects is considered as the basic concept of weighted consensus function. From the weighted partitions, a Pairwise similarity matrix can be obtained and those weights are derived by evaluating the various clustering quality. To generate candidate consensus partitions, the dendogram [24] is raised which is depend on similarity matrix.

a) Partition Weighting Scheme

For N objects consider X={$\mathrm{x}_n\}_{n=1}^{N}$ is a data set and for X there are M partitions are available $\mathbf{P}$={$P_m\}_{m=1}^{M}$, which contain a diverse M partition in cluster number that has been gathered from the starting cluster analysis. The weight $w_m^{\pi}$for each $P_m$is allocated by partition weighting scheme in which π is the criterion for clustering validation. The weight vector $\mathbf{w}^{\pi}$={$w_m^{\pi}\}_{m=1}^{\mathrm{M}}$ which is depend on criterion π, for $\mathbf{P}$ partition collection. The weight of the partition weighting scheme can be defined by,

$$w_m^{\pi} = \frac{\pi(P_m)}{\sum_{m=1}^{M}\pi(P_m)} \tag{31}$$

Here $w_m^{\pi} > 0$ and $\sum_{m=1}^{M} w_m^{\pi}$=1. The index value for clustering validity is $\pi(P_m)$. Instinctively, the weight about a partition may be prompted their influence based on the combination of their cluster quality evaluated by its validation criterion π.There are various features are available for clustering quality to evaluate the influence of the partition. When compared with all present clustering validation criteria, we can make use of three criteria which is useful for producing weights from several views such as Modified Huber's T index (MHT) [25], Dunn's Validity Index (DVI) [25], and Normalized Mutual Information (NMI) [26].

The Partition $P_m$for MHT index [25] is expressed by,

$$\text{MHT }(P_m) = \frac{N(N-1)}{2}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} A_{ij}Q_{ij} \tag{32}$$

where the proximity matrix of objects is $A_{ij}$and from the partition P_m, Q is the N×N cluster distance matrix that has been derived, in which each element Q_{ij} denotes the distance among the centers of clusters i.e., x_i andx_j. Naturally a partition that has high MHT value specifies that the partition has a compact and clustering structure has well-separated. Whereas, this condition will favors for a partition that containing more clusters.

The Partition P_m for DVI [25] is expressed by,

$$\text{DVI }(P_m) = \min_{i,j}\left\{\frac{d(C_i^m, C_j^m)}{\max_{k=1,\ldots,K_m}\{diam(C_k^m)\}}\right\} \tag{33}$$

Here the clusters are represented by $C_i^m, C_j^m,$ andC_k^min P_m, the dissimilarity metric among the clusters $C_i^m$and C_j^m is $d(C_i^m, C_j^m$ and in P_m the diameter of cluster C_k^m is $diam(C_k^m)$. Based upon the MHT index, the DVI also calculates the compactness and separation properties to find the clustering quality. Then the number of clusters in a partition is unresponsive. Due to the usage of a single linkage distance and the diameter information based on the cluster is less robust for this index.

The NMI [26] is suggested to evaluate the consistency among two partitions, i.e., the quantity of information that has been shared among two partitions. The summation of NMI can be done among the partition $P_m$and every single of other partitions $P_o$is estimated for the NMI index for a partition P_m

$$\overline{NMI}(P_m, P_o) = \frac{\sum_{i=1}^{K_m}\sum_{j=1}^{K_o} N_{ij}^{mo}\log\left(\frac{NN_{ij}^{mo}}{N_i^a N_j^b}\right)}{\sum_{i=1}^{K_m} N_i^m\log\left(\frac{N_i^m}{N}\right)+\sum_{j=1}^{K_o} N_j^o\log\left(\frac{N_j^o}{N}\right)} \tag{34}$$

$$\text{NMI }(P_m) = \sum_{o=1}^{M}\overline{NMI}\ (P_m, P_o)$$

where the data set of N objects has two partitions $P_m$and P_o which can be divided into K_m and $K_o$clusters respectively. N_{ij}^{mo} is the number of shared objects among two various clusters C_i^m Є P_m and C_j^o Є $P_o$in which N_i^m and N_j^oobjects is C_i^m and C_j^o. Naturally a well-accepted partition is indicated by a high NMI value in which the data set reflects the intrinsic structure. This condition is based upon highly correlated partitions and favors for a similar number of objects that contained in the cluster.

b) Weighted Similarity Matrix

A binary membership indicator matrix $H_m = \{0,1\}^{N\times K_m}$ is calculated for each partition P_m, where K_m represents the number of clusters in each partition P_m.In this H_m, a row denotes one datum and a column represents a binary coding vector for a particular cluster in the partition P_m. If the column entity value is one, it means that the related objects are grouped into the same cluster. Here, we make use of H_m to derive an N×N binary similarity matrix $S_m = \{0,1\}^{N\times N}$, which encodes the Pairwise similarity among any two objects in a partition can be denoted as,

$$S_m = H_m H_m^T. \tag{35}$$

In the above (35), the element $(S_m)_{ij}$ denotes the inner product between the rows i and j of matrix H_m. Hence, if $(S_m)_{ij} = 1$, objects i and j are grouped into the same cluster. Else they are grouped in different clusters.

At last, for all the partitions in $\mathbf{P}$, a linear combination of similarity matrix $S_m$with their weight w_m^{π}as

$$S^{\pi} = \sum_{m=1}^{M} w_m^{\pi} S_m. \tag{36}$$

Thus in our algorithm, the three weighted similarity matrices S^{MHT},$S^{DVI}$and S^{NMI} are derived correspondingly.

c) Candidate Consensus Partition Generation

The dendogram based similarity partitioning algorithm (DSPA) is used to produce a candidate consensus partition from a weighted similarity matrix S^{π}. DSPA algorithm employs an average-link hierarchical clustering algorithm which converts the weighted similarity matrix into a dendogram [24] where its horizontal axis indexes all the data in a given data set and its vertical axis gives the lifetime of all possible cluster formation. The lifetime of a cluster is expressed as an interval between the moments from which the cluster is created and the moment that it gets disappeared by merging with other clusters. Note that dendogram produced in this way is very different from that formed by the similarity matrix without being weighted [27]. Hence, the number of clusters in a candidate consensus partition P^{π} can be determined automatically by cutting the dendogram derived from S^{π}in order to form clusters at the longest lifetime. With the DSPA algorithm, we attain three candidate consensus partitions P^{π},$\pi = \{MHT, DVI, NMI\}$in our algorithm.

IV. Comparison of Cluster Ensemble Methods

This part demonstrates [28] the comparison of the above described various cluster ensemble methods depend upon various parameters. The main idea of this distinction is not to estimate the finest clustering method where it is used to distinguish the methods depend upon their features which make the users to solve their problem by choosing the suitable cluster ensemble method. In Table I, we shortened the above cluster ensemble methods which relate to their features and limitations.

TABLE I
Summarized Cluster Ensemble Methods

Clustering Ensemble Methods	Ensemble Size	Type of Consensus Function used	Dimensionality (size of the dimensions used in the datasets)	Types of Dataset used	Algorithm used to build Base Clustering	Features
LCE	Fixed	SPEC	Large	Image Dataset	SOM, K-Means, Agglomerative hierarchical clustering. Fuzzy C-means	Improved precipitation Estimation
BGCM	Fixed	Graph based consensus maximization	Small & Large	Mixed Dataset (numerical and categorical)	SVM Logistic Regression K-Means Spectral Clustering	Generates a consolidated classification solution for the target set based on both classification and clustering, Handles imbalanced class distribution, Retains all the information without loss
LBCE	Fixed	SPEC spectral graph partitioning	Small & Large	Mixed Dataset (numerical and categorical)	K-Means	Form accurate cluster ensemble, Automated and data-driven setting of DC
CABCE	Fixed	EAC-AL (Evidence Accumulation Clustering with Average Linkage)	Small & Large	Mixed Dataset (numerical, categorical, and image)	K-Means	Scalable to very large datasets, Reduced computational Complexity, improved efficiency
WCE	Fixed	DSPA Dendogram based similarity partitioning	Small & Large	Mixed Dataset (numerical and categorical)	K-Means Hierarchical Clustering	Does not require parameter tuning, Low computational complexity

A. Ensemble Size

The Ensemble is the technique of combining the cluster partitions in order to enhance the single clustering algorithms so that it can able to produce the effective resultby integrating the various base clustering solutions to construct final partition. The ensemble size can be obtained which indicates the number of clusters. This size has two types such as fixed size and variable size. In fixed size the length of the cluster is explained earlier. While in variable size, the ensemble size contains no limitations.

B. Types of Consensus Function Used

The consensus function includes two types such as ObjectCo-occurrences and Median Partition methods. First type contracts with computing the number of Co-occurrences of an object in a distinct cluster and the second type contracts with the partition that exploits the similarity with all partitions in the cluster.

C. Dimensionality

This property signifies the ability of the datasets used for the experimental analysis of the ensemble techniques. The Capacity of the datasets is ordered in small and large by analyzingthrough the number of data points, classes, attributes values, patterns and features occurring in the dataset.

D. Type of Datasets Used

Datasets which are used for the experimental setup included of three types such as Categorical Datasets and Numerical Datasets and categorical numerical &mixed datasets. Afirst typeconsists of only a group of numerical data points; the second type includes the text data points which are related to the particular domain, whereas the third type of datasets deals witha mixture of the first and second type.

E. Algorithm Used for Base Clustering

Single clustering algorithm contains several sets of parameter initializations that is executed by selecting Base clustering algorithm. The generation of cluster ensembles this is based on base clustering. Except from this at various clustering algorithms can also be used as a base clustering to execute heterogeneous ensemble creation.

Thus the following Table I represents the main features of each ensemble approach. We can easily examine their abilities and compare them based on the types of consensus Ensemble size, dimensionality, function, types of dataset, and the Algorithm used for base cluster generation.

V. Conclusion

Cluster Ensembles have been recently emerged as an offshoot for solving the issues caused by individual clustering solutions. This efficient technique was mainly bloomed as an outstanding method to enhance the stability, individuality, accuracy and robustness of unsupervised learning solutions. This Ensemble methodology is really helpful and acts as a keystone for pointing out and compensating possible issues happening in single clustering algorithms. Consequently, this comprehensive study exposes some different mixed data Cluster Ensemble approaches, including their functioning process and salient features of each method. Henceforth the original outcome of this paper is to express the systematic workflow of each technique and the comparative table reveals the individual method's features and limitations. This review makes better perceptive for the readers and also helpful for the community of clustering followers to indulge in more research activities to innovate several efficient Cluster Ensemble methods in future.

References

[1] S. Sarumathi, N. Shanthi, M. Sharmila. "A Review: Comparative Analysis of Different Categorical Data Clustering Ensemble Methods". International Journal of Computer, Information Science and Engineering Vol:7 No:12, 2013.

[2] Sandro Vega-pons & Jose reuizShulcloper. "A Survey of Clustering Ensemble algorithms". International Journal of Pattern Recognition and Artificial Intelligence Vol. 25, No. 3 337_372 2011.

[3] NatthakanIam-On, TossaponBoongoen, Simon Garrett, & Chris Price. "A Link based cluster ensemble approach for categorical data clustering". IEEE Transactions on knowledge and data engineering, Vol. 24, No. 3, 2012.

[4] HarunPirim, DilipGautam, Tanmay ,Bhowmik, Andy D. Perkins, BurakEkşioglu, &AhmetAlkan,. "Performance of an ensemble clustering algorithm on biological datasets". Mathematical and Computational Applications, Vol. 16, No. 1, pp. 87-96 2011.

[5] MajidMahrooghy, Nicolas H. Younan, Valentine G. Anantharaj, James Aanstoos, and ShantiaYarahmadian, "On the Use of a Cluster Ensemble Cloud Classification Technique in Satellite Precipitation Estimation", IEEE journal of selected topics in applied earth observations and remote sensing, vol. 5, no. 5, october 2012.

[6] Y. Hong, K. L. Hsu, S. Sorooshian, and X. G. Gao, "Precipitation estimation from remotely sensed imagery using an artificial neural network cloud classification system," J. Appl. Meteorol., vol. 43, pp. 1834–1852, 2004

[7] J. C. Grzegorz and W. F. Krajewski, "Comments on the window probability matching method for rainfall measurements with radar," J. Appl. Meteorol., vol. 36, pp. 243–246, 1997.

[8] M. Mahrooghy, V. G. Anantharaj, N. H. Younan, W. A. Petersen, F. J.Turk, and J. Aanstoos, "Infrared satellite precipitation estimate using waveletbased cloud classification and radar calibration," in Proc. 2010 IEEE Int. Geoscience and Remote Sensing Symp. (IGARSS), 2010, pp. 2345–2348.

[9] N. Iam-on, T. Boongoen, and S. Garrett, "LCE: A link-based cluster ensemble method for improved gene expression data analysis," Bioinformatics, vol. 26, pp. 1513–1519, 2010

[10] R. Duda, P. E. Hart, and D. G. Stork, Pattern Classification. New York: Wiley-Interscience, 2000.

[11] D. A. Neumann and V. Norton, "Clustering and isolation in the consensus problem for partitions," J. Classification, vol. 3, pp. 281–297, 1986.

[12] Jing Gao, Feng Liang, Wei Fan, Yizhou Sun, and Jiawei Han, "A Graph-Based Consensus Maximization Approach for Combining Multiple Supervised and Unsupervised Models" , IEEE transactions on knowledge and data engineering, vol. 25, no. 1, january 2013.

[13] D.P. Bertsekas, Non-Linear Programming, second ed. Athena Scientific, 1999.

[14] NatthakanIam-On, TossaponBoongoen, "Improved Link-Based Cluster Ensembles", IEEE World Congress on Computational Intelligence June, 10-15, 2012.

[15] N Iam-on, "LCE: A link-based cluster ensemble method for improved gene expression data analysis," Bioinformatics, vol. 26, no. 12, pp. 1513– 1519, 2010

[16] N. Iam-On, T. Boongoen, S. Garrett, and C. Price, "A link-based approach to the cluster ensemble problem," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 12, pp. 2396–2409, 2011.

[17] N. Iam-On, T. Boongoen, and S. Garrett, "Refining pairwise similarity matrix for cluster ensemble problem with cluster relations," in Proceedings of Eleventh International Conference on Discovery Science, 2008,pp. 222–233.

[18] T. Boongoen, Q. Shen, and C. Price, "Disclosing false identity through hybrid link analysis," AI and Law, vol. 18, no. 1, pp. 77–102, 2010.

[19] A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," Advances in Neural Information Processing Systems, vol. 14, pp. 849–856, 2001.

[20] Xiaoli Z. Fern and Carla E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," in Proceedings of International Conference on Machine Learning, 2004, pp. 36–43.

[21] Tsaipei Wang, "CA-Tree: A Hierarchical Structure for Efficient and Scalable Coassociation-Based Cluster Ensembles", IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 41, no. 3, june 2011.

[22] T.-Y. Lv, S.-B. Huang, X.-Z. Zhang, and Z.-X. Wang, "Combining multiple clustering methods based on core group," in Proc. 2nd Int. Conf.Semantics, Knowl., Grid, 2006, pp. 29–34.

[23] Yun Yang and Ke Chen, "Temporal Data Clustering via Weighted Clustering Ensemble with Different Representations", IEEE transactions on knowledge and data engineering, vol. 23, no. 2, february 2011.

[24] A. Jain, M. Murthy, and P. Flynn, "Data Clustering: A Review," ACM Computing Surveys, vol. 31, pp. 264-323, 1999.

[25] M. Halkidi, Y. Batistakis, and M. Varzirgiannis, "On Clustering Validation Techniques," J. Intelligent Information Systems, vol. 17, pp. 107-145, 2001

[26] A. Strehl and J. Ghosh, "Cluster Ensembles—A Knowledge Reuse Framework for Combining Multiple Partitions," J. Machine Learning Research, vol. 3, pp. 583-617, 2002.

[27] A. Fred and A. Jain, "Combining Multiple Clusterings Using Evidence Accumulation," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 6 pp. 835-850, June 2005.

[28] Sarumathi S, Shanthi N, Sharmila M. " A Comparative Analysis of Different Categorical Data Clustering Ensemble Methods in DataMining" International Journal of Computer Applications Vol 81, No.4 November 2013.

Mrs. S.Sarumathi received B.E. degree in Electronics and Communication Engineering from Madras University, Madras, Tamil Nadu India in 1994 and the M.E. degree in Computer Science and Engineering from K.S.Rangasamy College of Technology, Namakkal, Tamil Nadu, India in 2007. She is doing her Ph.D. programme under the area Data Mining in Anna University, Chennai. She has a teaching experience of about 16 years. At present she is working as Associate professor in Information Technology department at K.S.Rangasamy College of technology. She has published 5 papers in the reputed International Journals and 2 papers in the reputed National journals. And also she has presented papers in three International conferences and four national Conferences. She has received many cash awards for producing cent percent results in university examination. She is a life member of ISTE.

Dr.N.Shanthi received the B.E. degree in Computer Science and Engineering from Bharathiyar University, Coimbatore, Tamil Nadu, India in 1994 and the M.E. degree in Computer Science and Engineering from Government College of Technology, Coimbatore, Tamil Nadu, and India in 2001. She has completed the Ph.D. degree in Periyar University, Salem in offline handwritten Tamil Character recognition. She worked as a HOD in department of Information Technology, atK.S.Rangasamy College of Technology, Tamil Nadu, India since 1994 to 2013, and currently working as a Professor& Dean in the department of Computer Science and Engineering at Nandha Engineering College Erode. She has published 30 papers in the reputed International journals and 9 papers in the National and International conferences. She has published 2 books. She is supervising 14 research scholars under Anna University, Chennai. She acts as the reviewer for 4 international journals. Her current research interest includes Document Analysis, Optical Character Recognition, and Pattern Recognition and Network security. She is a life member of ISTE.

Ms. M.Sharmilareceived B.Tech degree and M.Tech degree in Information Technology from K.S.Rangasamy College of Technology, affiliated to Anna University Chennai, Tamil Nadu, India in 2012 and 2014 respectively. At present she is working as an Assistant Professor in Information Technology department at M.Kumarasamy College of Engineering,Karur. She has published 5 international journals and presented three papers in National level technical symposium. She is an active member of ISTE. Her Research interests include Mining Medical data, Opinion Mining and Web mining. Most of her current work involves the development of efficient cluster ensemble algorithms for extracting accurate clusters in large dimensional database.

Ms.S.Vidhyaholds a B.Tech degree in Information Technology from N.P.R College of Engineering and Technology, affiliated to Anna University of Technology, Tiruchirappalli, Tamil Nadu, India in 2013. Now she is an M.Tech student of Information Technology department in K.S.Rangasamy College of Technology. She has published 1 International Journal and presented three papers in National level technical symposium. Her Research interest includes Data Mining, Wireless Networks.