International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:3, No:2, 2009

A Survey: Clustering Ensembles Techniques

Reza Ghaemi , Md. Nasir Sulaiman , Hamidah Ibrahim , Norwati Mustapha

Abstract**—**The clustering ensembles combine multiple partitions generated by different clustering algorithms into a single clustering solution. Clustering ensembles have emerged as a prominent method for improving robustness, stability and accuracy of unsupervised classification solutions. So far, many contributions have been done to find consensus clustering. One of the major problems in clustering ensembles is the consensus function. In this paper, firstly, we introduce clustering ensembles, representation of multiple partitions, its challenges and present taxonomy of combination algorithms. Secondly, we describe consensus functions in clustering ensembles including Hypergraph partitioning, Voting approach, Mutual information, Co-association based functions and Finite mixture model, and next explain their advantages, disadvantages and computational complexity. Finally, we compare the characteristics of clustering ensembles algorithms such as computational complexity, robustness, simplicity and accuracy on different datasets in previous techniques.

Keywords**—**Clustering Ensembles, Combinational Algorithm, Consensus Function, Unsupervised Classification.

I. Introduction

IN contrast to supervised classification, clustering is inherently an ill-posed problem whose solution violates at least one of the common assumptions about scale invariance, richness, and cluster consistency. Different clustering solutions may seem equally plausible without a priori knowledge about the underlying data distributions. Every clustering algorithm implicitly or explicitly assumes a certain data model and it may produce erroneous or meaningless results when these assumptions are not satisfied by the sample data. Thus, the availability of prior information about the data domain is crucial for successful clustering, though such information can be hard to obtain, even from experts. Identification of relevant subspaces or visualization may help to establish the sample data's conformity to the underlying distributions or, at least, to the proper number of clusters [1].

The exploratory nature of clustering tasks demands efficient methods that would benefit from combining the strengths of many individual clustering algorithms. This is the focus of the research on clustering ensembles, seeking a combination of multiple partitions that provides improved overall clustering of the given data. Clustering ensembles can go beyond what is typically achieved by a single clustering algorithm in several respects:

Robustness: Better average performance across the domains and datasets.

Novelty: Finding a combined solution unattainable by any single clustering algorithm.

Stability and confidence estimation: Clustering solutions with lower sensitivity to noise, outliers, or sampling variations. Clustering uncertainty can be assessed from ensemble distributions.

Parallelization and Scalability: Parallel clustering of data subsets with subsequent combination of results. Ability to integrate solutions from multiple distributed sources of data or attributes (features) [1], [2].

Clustering ensembles can also be used in multiobjective clustering as a compromise between individual clusterings with conflicting objective functions. Fusions of clusterings using multiple sources of data or features become increasingly important in distributed data mining. Several recent independent studies [3], [4], [5], [6], [7], [8] have pioneered clustering ensembles as a new branch in the conventional taxonomy of clustering algorithms [9], [10]. Other related work includes [1] ,[2], [11], [12], [13].

Section 2 describes concept of clustering, clustering ensembles, its problems and section 3 explains representation of multiple partitions. In section 4, we summarize taxonomy of clustering combination approaches. In sections 4 and 5, we describe different consensus functions in clustering ensembles and compare their advantages, disadvantages and computational complexity.

II. Clustering Ensembles

In this section, we introduce clustering ensembles and its problems. Next we describe representation of multiple partitions.

Clustering analysis has been widely applied in many real world application domains such as data compression, data mining and pattern recognition. However, it is in fact an ill-posed combinatory optimization problem and no single clustering algorithm is able to achieve satisfactory clustering solutions for all types of data sets. Numbers of clustering algorithms exist so far and some of them often produce contradictory clustering solutions. There are many feasible approaches to improve the performance of clustering analysis. Among them is the clustering ensembles method [8]. The

Reza Ghaemi is PhD Student in Faculty of Computer Science and Information Technology Putra University of Malaysia (UPM), and Academic Member in IAUQ, Qoochan, Iran, rezaghaemi@ieee.org

Associate Prof. Dr. Md. Nasir Sulaiman, Associate Prof. Dr. Hamidah Ibrahim and Dr. Norwati Mustapha are academic members in Department of Computer Science, Faculty of Computer Science and Information Technology Putra University of Malaysia (UPM), Selangor, Malaysia, Phone:+6-038 9466585,Fax:+6-03-89466577,{nasir,hamidah,noewati}@fsktm.upm.edu.my

clustering ensembles method leverages the consensus across multiple clustering solutions and combines these multiple clustering solutions into a single consensus one. It has often been noted in the literature that the clustering ensembles method is able to improve the robustness and stability of clustering analysis [5]. Therefore, the clustering ensembles method has gained a lot of real world applications such as gene classification, image segmentation [16], video retrieval and so on [14], [15], [17].

Combination of multiple partitions can be viewed as a partitioning task itself. Typically, each partition in the combination is represented as a set of labels assigned by a clustering algorithm. The combined partition is obtained as a result of yet another clustering algorithm whose inputs are the cluster labels of the contributing partitions. We will assume that the labels are nominal values. In general, the clusterings can be "soft", i.e. described by the real values indicating the degree of pattern membership in each cluster in a partition. We consider only "hard" partitions below, noting however, that combination of "soft" partitions can be solved by numerous clustering algorithms and does not appear to be more complex [2].

Clustering ensembles usually are two stage algorithms. At the first, it stores the results of some independent runs of *K*-means or other clustering algorithms. Then, it uses the specific consensus function to find a final partition from stored results. Fig. 1 shows clustering ensembles architecture as following:

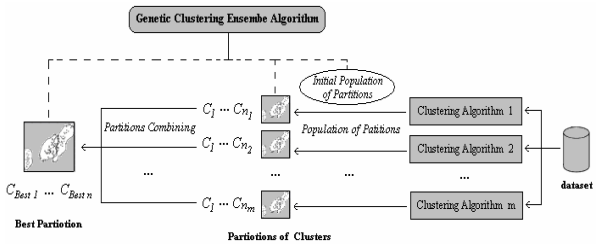

Fig. 1 Clustering Ensembles Architecture

The cluster ensembles design problem is more difficult than designing *classifier ensembles* since cluster labels are symbolic and so one must also solve a correspondence problem. In addition, the number and shape of clusters provided by the individual solutions may vary based on the clustering method as well as on the particular view of the data available to that method. Moreover, the desired number of clusters is often not known in advance. In fact, the "right" number of clusters in a dataset often depends on the *scale* at which the data is inspected, and sometimes equally valid (but substantially different) answers can be obtained for the same data [8].

The problem of clustering ensembles or clustering combination can be defined generally as follows: Given multiple clusterings of the dataset, find a combined clustering with better quality. Whereas the problem of clustering combination bears some traits of a classical clustering problem, it also has three major problems which are specific to combination design:

1. Consensus function: How to combine different clusterings? How to resolve the label correspondence problem? How to ensure symmetrical and unbiased consensus with respect to all the component partitions? [2], [18]
2. Diversity of clustering: How to generate different partitions? What is the source of diversity in the components? [2], [18]. Diversity of the individual clusterings of a give dataset can be achieved by a number of approaches. Applying various clustering algorithms [8], using one algorithm with different built-in initialization and parameters [4], [5], [19], projecting data onto different subspaces [8], [12], choosing different subsets of features [8], and selecting different subsets of data points [3], [6], [16] are instances of these generative mechanism [20].
3. Strength of constituents/components: How "weak" could each input partition is? What is the minimal complexity of component clusterings to ensure a successful combination? [1]

In order to optimally integrate clustering ensembles in a robust and stable manner, one needs a diversity of component partitions for combination. Generally, this diversity can be obtained from several sources [18], [21]:

1) Using different clustering algorithms to produce partitions for combination [3].
2) Changing initialization or other parameters of a clustering algorithm [11], [19].
3) Using different features via feature extraction for subsequent clustering [8], [22].
4) Partitioning different subsets of the original data [3], [16], [23].

The major hardship in clustering ensembles is consensus functions and partitions combination algorithm to produce final partition, or in other words finding a consensus partition from the output partitions of various clustering algorithms [2], [18].

Similar questions have already been addressed in the framework of multiple classifier systems. Combining results from many supervised classifiers is an active research area and it provides the main motivation for clusterings combination. However, it is not possible to mechanically apply the combination algorithms from classification (supervised) domain to clustering (unsupervised) domain. Indeed, no labeled training data is available in clustering; therefore, the ground truth feedback necessary for boosting the overall accuracy cannot be used. In addition, different clusterings may produce incompatible data labeling, resulting in intractable correspondence problems, especially when the numbers of clusters are different [1].

Unlike supervised classification, the patterns are unlabeled; therefore, there is no explicit correspondence between the labels delivered by different partitions. The combination of multiple clustering can also be viewed as finding a median partition with respect to the given partitions which is proven to be NP-complete [18].

III. Presentation of Multiple Partitions

Clustering ensembles need a partition generation procedure. Several methods are known to create partitions for clustering ensembles. For example, one can use: 1) different regular

clustering algorithms [8], 2) different initializations, parameter values or built-in randomness of a specific clustering algorithm [5], 3) weak clustering algorithms [19], 4) data resampling [3], [16]. All these methods generate ensemble partitions independently, in a sense that the probability to obtain the ensemble consisting of H partitions $\{\pi_1, \pi_2, \ldots, \pi_H\}$ of the given data D can be factorized in (1) [23]:

$$P(\{\pi_1, \pi_2, \ldots, \pi_H\}) \Big| D) = \prod_{t=1}^{H} P(\pi_t | D) \quad (1)$$

Hence, the increased efficacy of an ensemble is mostly attributed to the number of identically distributed and independent partitions, assuming that a partition of data is treated as a random variable p. Even when the clusterings are generated sequentially, it is traditionally done without considering previously produced clusterings as in (2):

$$P(\{\pi_t | \pi_{t-1}, \pi_{t-2}, \ldots, \pi_1; D) = P(\pi_t | D) \quad (2)$$

Strehl and Ghosh presented a presentation for multiple partitions. They assumed $X = \{x_1, x_2, \ldots, x_n\}$ denote a set of objects/samples/points. A partitioning of these n objects into k clusters can be represented as a set of k sets of objects $\{C_\ell \mid \ell = 1, \ldots, k\}$ or as a label vector $\lambda \in N^n$. A clusterer Φ is a function that delivers a label vector given a tuple of objects. Fig. 2 shows the basic setup of the cluster ensemble: A set of r labelings $\lambda^{(1,\ldots,r)}$ is combined into a single labeling λ (the *consensus labeling*) using a consensus function Γ [8].

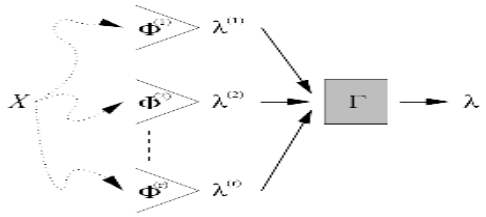

Fig. 2 Representation of multiple partitions by function Γ

Fern and Brodley [24] described a basic definition of graph partitioning. A weighted graph is represented by $G = (V, W)$, where V is a set of vertices and W is a nonnegative and symmetric $|V| \times |V|$ similarity matrix characterizing the similarity between each pair of vertices. The input to a graph partitioning problem is a weighted graph G and a number K. To partition a graph into K parts is to find K disjoint clusters of vertices $P = \{P_1, P_2, \cdots, P_K\}$, where $\cup P_k = V$. Unless a given graph has K, or more than K, strongly connected components, any K-way partition will cross some of the graph edges. The sum of the weights of these crossed edges is defined as the cut of a partition P: $Cut(P, W) = \sum W(i, j)$, where vertices i and j do not belong to the same cluster.

IV. Taxonomy of Clustering Combination Approaches

We summarize clustering combination approaches in Fig. 3. We focus on consensus function methods and describe them in next section.

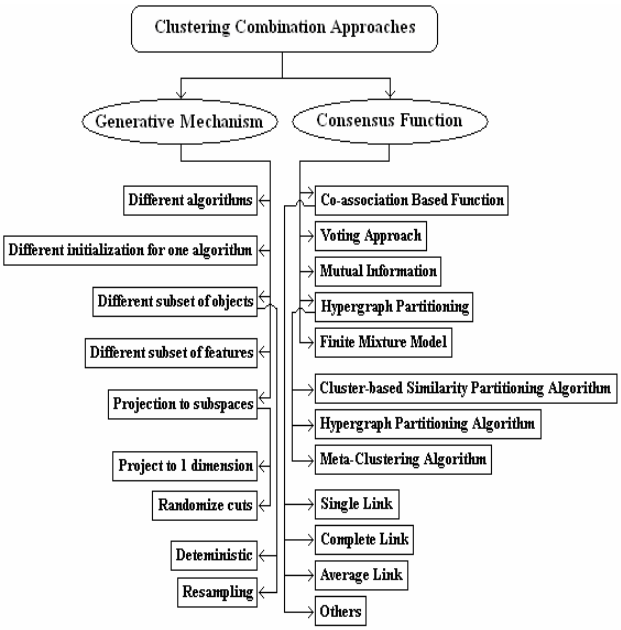

Fig. 3 Taxonomy of Clustering Combination Approaches

V. Consensus Functions in Clustering Ensembles

There are some types of consensus function such as: Hypergraph Partitioning, Voting Approach, Mutual Information Algorithm, Co-association based functions and Finite Mixture model. We introduce all of kind of consensus functions and some previous research works in them, separately.

A. Hypergraph Partitioning

The clusters could be represented as hyperedges on a graph whose vertices correspond to the objects to be clustered, so each hyperedge describes a set of objects belonging to the same clusters. The problem of consensus clustering is then reduced to finding the minimum-cut of a hypergraph. The minimum k-cut of this hypergraph into k components gives the required consensus partition [8], [22]. Hypergraph partitioning is NP-hard problem, but efficient heuristics to solve the k way min-cut partitioning problem are known, some with computational complexity on the order of $O(|\varepsilon|)$, where ε is the number of hyperedges [2], [18].

Strehl and Ghosh [8] used a knowledge reuse framework and they have considered three different consensus functions for ensemble clustering. The Cluster based Similarity Partitioning Algorithm (CSPA) induces a graph from a co-association matrix and clusters it using the METIS algorithm. The Hypergraph Partitioning Algorithm (HGPA) represents each cluster by a hyperedge in a graph where the nodes correspond to a given set of objects. Good hypergraph partitions are found using minimal cut algorithms such as HMETIS coupled with the proper objective functions, which also control partition size. Hyperedge collapsing operations are considered in another hypergraph based Meta Clustering Algorithm (MCLA). The MCLA uses these operations to

determine soft cluster membership values for each object. Computing complexity of CSPA, HGPA and MCLA are O(kN^2H), O(kNH), and O(k^2NH^2), respectively. They defined a mutual information based objective function that can select automatically the best solution from several algorithms and to build a supra consensus function as well. Their proposed algorithms improved the quality and robustness of the solution, but their proposed greedy approach is the slowest and often is intractable for large n.

Fern and Brodley [24] proposed another cluster ensemble method based on graph partitioning named Hybrid Bipartite Graph Formulation (HBGF) [8]. It constructs a bipartite graph from a set of partitions to be combined, modeling objects and clusters simultaneously as vertices, and later partitioning the graph by a traditional graph partitioning technique. Their approach retained all of the information provided by a given ensemble, allowing the similarity among instances (IBGF) and the similarity among clusters (CBGF) to be considered collectively in forming the final clustering. HBGF has high robust clustering performance against IBGF and CBGF and the reduction of HBGF is lossless. The computing complexity of HBGF is O(kN). Their proposed method's implementation is difficult.

Ng *et al*. [25] proposed a popular multiway spectral graph partitioning algorithm (SPEC) which seeks to optimize the normalized cut criterion. SPEC can be simply described as a graph $G = (V,W)$, it first computes the degree matrix D, which is a diagonal matrix such that $D(i, i) = \sum_j W(i, j)$. Based on D, it then computes a normalized weight matrix $L = D^{-1}W$ and finds L's K largest eigenvectors $u_1, u_2, \cdots, u_K$ to form matrix $U = [u_1, \cdots, u_K]$. The rows of U are then normalized to have unit length. Treating the rows of U as K-dimensional embeddings of the vertices of the graph, SPEC produces the final clustering solution by clustering the embedded points using K-means. Comparing to HBGF, SPEC has low robust clustering performance. It's computing complexity is O(N^3).

Karypis and Kumar [26] proposed a multilevel graph partitioning system named METIS, approaches the graph partitioning problem from a different angle. It partitions a graph using three basic steps: (1) coarsen the graph by collapsing vertices and edges; (2) partition the coarsened graph and (3) refine the partitions. In comparison to other graph partitioning algorithms, METIS is highly efficient and achieves competitive performance. Comparing to HBGF, METIS has low robust clustering performance. It's computing complexity is O(kNH).

B. Voting Approach

It calls also direct approach or relabeling. In the other algorithms there is no need to explicitly solve the correspondence problem between the labels of known and derived clusters. The voting approach attempts to solve the correspondence problem, then a simple voting produce can be used to assign objects in clusters to determine the final consensus partition. However, label correspondence is exactly what makes unsupervised combination difficult. The main idea is to permute the cluster labels such that best agreement between the labels of two partitions is obtained. All the partitions from the ensemble must be relabeled according to a fixed reference partition. The reference partition can be taken as one from the ensemble, or from a new clustering of the dataset. Also, a meaningful voting procedure assumes that the number of clusters in every given partition is the same as in the target partition. This requires that the number of clusters in the target consensus partition is known. The complexity of this process is $k!$, which can be reduced to O(k^3) if the Hungarian method is employed for the minimal weight bipartite matching problem [2], [3], [11], [18].

Fischer and Buhmann [6], [16], and also Dudoit and Fridlyand [3], have implemented a combination of partitions by relabeling and voting. Their works pursued direct relabeling approaches to the correspondence problem. A relabeling can be done optimally between two clusterings using the Hungarian algorithm. After an overall consistent relabeling, voting can be applied to determine cluster membership for each object. However, this voting method needs a very large number of clusterings to obtain a reliable result. Computing complexity of their proposed algorithm is O(k^3).

Fischer and Buhmann [6], [16] presented path based clustering with automatic outlier detection that captures the empirical observation that group structures in embedding spaces might appear as manifolds with considerable extension but are characterized by local homogeneity and connectivity. Path based clustering is applicable, even in situations when the parametric form of such a transformation is unknown. Two central applications of perceptual organization, edge grouping and texture segmentation, have been solved by path based clustering.

Dudoit and Fridlyand [3] proposed two bagged clustering procedures to improve and assess the accuracy of a partitioning clustering method. The bagging is used to generate and aggregate multiple clusterings and to assess the confidence of cluster assignments for individual observations. As in prediction, the motivation behind the application of bagging to cluster analysis is to reduce variability in the partitioning results via averaging. The proposed bagged clustering procedures are illustrated using the *Partitioning Around Medoids* or PAM method of Kaufman and Rousseeuw (1990). As implemented in the R and S-Plus libraries cluster, the two main arguments of the PAM function are: a dissimilarity matrix and the number of clusters K. The PAM procedure is based on the search for K representative objects, or *medoids*, such that the sum of the dissimilarities of the observations to their closest medoid is minimized. They suspected that, as in prediction, the increase in accuracy observed with PAM is due to a decrease in variability achieved by aggregating multiple clusterings. Application of bagging to cluster analysis can substantially improve clustering accuracy and yields information on the accuracy of cluster assignments for individual observations. In addition, bagged clustering procedures are more robust to the variable selection scheme, i.e. their accuracy is less sensitive to the number and type of variables used in the clustering.

C. Mutual Information

The objective function for a clustering ensemble can be formulated as the mutual information (MI) between the

empirical probability distribution of labels in the consensus partition and the labels in the ensemble. Under the assumption of independence of partitions, MI can be written as sum of pair-wise MIs between target and given partitions. Using the classical definition of MI, one can easily compute its value for a candidate partition solution and the ensemble. However, such a definition does not offer a search algorithm for maximizing the consensus. An elegant solution can be obtained from a generalized definition of MI. Quadratic Mutual Information (QMI) or feature based approach can be effectively maximized by the *K*-means algorithm in the space of specially transformed cluster labels of given ensemble. It treats the output of each clustering algorithm as a categorical feature. The collection of *L* features can be regarded as an "intermediate feature space" and another clustering algorithm can be run on it. Computational complexity of the algorithm is O(*kNH*), but it may require a few restarts in order to avoid convergence to low quality local minima [2].

Topchy *et al*. [19] have developed a different consensus function based on information theoretic principles, namely using generalized mutual information (MI). It was shown that the underlying objective function is equivalent to the total intra-cluster variance of the partition in the specially transformed space of labels. Therefore, the *K*-means algorithm in such a space can quickly find corresponding consensus solutions. They proposed two different weak clustering algorithms as the components of the combination: 1) Clustering of random 1-dimensional projections of multidimensional data. This can be generalized to clustering in any random subspace of the original data space. 2) Clustering by splitting the data using a number of random hyperplanes. For example, if only one hyperplane is used then data is split into two groups. Computational complexity of this algorithm is low, O(*kNH*), but it may require a few restarts in order to avoid convergence to low quality local minima.

Luo *et al*. [20] proposed a consensus scheme via the genetic algorithm based on information theory. A combined clustering is found by minimizing an information theoretical criterion function using genetic algorithm. The searching capability of genetic algorithms has been used in this article for the purpose of appropriately deriving a consensus clustering from a clustering ensemble. The clustering metric that has been adopted is the sum of the entropy based dissimilarity of the consensus clustering from the component clusterings in the ensemble. The optimal correspondence can be obtained using the Hungarian method for minimal weight bipartite matching problem with O (k^3) complexity for k clusters.

Azimi *et al*. [21], proposed a new clustering ensemble method, which generates a new feature space from initial clustering outputs. Multiple runs of an initial clustering algorithm like *k*-means generate a new feature space, which is significantly better than pure or normalized feature space. Therefore, running a simple clustering algorithm on generated feature space can obtain the final partition significantly better than pure data. In this method is used a modification of *k*-means for initial clustering runs named as "Intelligent *k*-means", which is especially defined for clustering ensembles. Fast convergence and appropriate behavior are the most interesting points of the proposed method. The proposed method uses *k*-means for clustering data. The complexity of *k*-means is O(*kNId*) where *k* is the number of clusters and *N* is the number of samples and *I* is the number of iterations of *k*-means to converge in each execution and *d* is the number of features (dimensions). Therefore, the complexity of the proposed method is O($k!+kNIdd'$) , where, d' is the number of partitions, in the other words, the number of generated features. *k*! is the complexity time to generate spanning tree. Since *k* is a small number, *k*! can be neglected. Therefore, the complexity of the proposed method is very low. The proposed method has unsuitable accuracy and its implementation is difficult.

D. Co-association based functions

It calls also pair wise approach. The consensus function operates on the co-association matrix. Numerous hierarchical agglomerative algorithms (criteria) can be applied to the co-association matrix to obtain the final partition, including Single Link (SL), Average Link (AL), Complete Link (CL) and Voting *k*-means [4], [5]. Note that the computational complexity of co-association based consensus algorithms is very high, O(kN^2d^2) [18].

The co-association matrix values are used in fitness function. Therefore we explain the co-association function specially as in (3). Let *D* be a dataset of *N* data points in *d*-dimensional space. The input data can be represented as an *N* * *d* pattern matrix or *N* * *N* dissimilarity matrix, potentially in a nonmetric space. Suppose that $X = \{x_1, \ldots, x_B\}$ is a set of bootstrap samples or sub samples of input dataset *D*. A chosen clustering algorithm is run on each of the samples in *X* that results in *B* partitions $P = \{p_1,\ldots,p_B\}$. Each component partition in *P* is a set of clusters $P_i=\{C^i_1, C^i_2,\ldots,C^i_{k(i)}\}$, $X_i = C^i_1 \cup C^i_2 \ldots \cup C^i_{k(i)}$ and k(*i*) is the number of clusters in the *i*-th partition.

$$Co - associate\ (x, y) = \frac{1}{B} \sum_{i=1}^{B} \varphi(p_i(x), p_i(y)) \quad (3)$$

Where φ(a,b) = 1, if a = b and φ(a,b) = 0, if a ≠ b.

Similarity between a pair of objects simply counts the number of clusters shared by these objects in the partitions $\{p_1,\ldots, p_B\}$ [2], [18].

There are three main concerns with this intuitively appealing approach. First, it has a quadratic complexity in the number of patterns O(N^2). Second, there are no established guidelines concerning which clustering algorithm should be applied, e.g. single linkage or complete linkage. The shapes of the clusters embedded in the space of clustering labels may or may not reflect cluster shapes in the original space. Third, an ensemble with a small number of clusterings may not provide a reliable estimate of the co-association values [2].

Fred [4] proposed to summarize various clustering results in a co-association matrix. The rational of the his approach is to weight associations between sample pairs by the number of times they co-occur in a cluster from the set of data partitions produced by independent runs of clustering algorithms, and propose this co-occurrence matrix as the support for consistent

clusters development using a minimum spanning tree like algorithm. The validity of this majority voting scheme is tested in the context of *k*-means based clustering, a new algorithm - *voting-k-means* - being presented. It can simultaneously handle the problem of initialization dependency and selection of the number of clusters. The proposed technique does not entail any specificity towards a particular clustering strategy.

Further work by Fred and Jain [5] also used co-association values, but instead of a fixed threshold, they applied a hierarchical (single link) clustering to the co-association matrix. They explored the idea of evidence accumulation for combining the result of multiple clusterings. The proposed strategy followed a split-and-merge approach. The final clusters were obtained by applying a MST based clustering algorithm. Their proposed method is able to identify arbitrary shaped clusters in multidimensional data. Their method performed poorly, however, in situations of touching clusters. One drawback of the co-association consensus function is its quadratic computational complexity in the number of objects $O(N^2)$.

Kellam *et al*. [13] have looked at several different methods for clustering from both the statistical and Artificial Intelligence communities and their application to viral gene expression profiles. They have compared the results of each method, firstly by using a comparison metric known as *Weighted-Kappa* which scores the differences between resulting clusters, and secondly in the context of finding known biological relationships amongst genes. They have found that whilst each method performs relatively well - the number of features found being very high - certain methods appear more geared towards certain features than others. They have also introduced an algorithm for generating robust clusters where all of the methods agree. Due to this algorithm taking the consensus of all methods, a proportion of genes will not be allocated to a cluster. However, it does allow the remaining allocated genes to be clustered with greater confidence than relying on the results of any one method. An $n \times n$ agreement matrix was generated with each cell containing the number of agreements amongst methods for clustering together the two variables represented by the indexing row and column indices. This matrix was then used to cluster variables based upon their cluster agreement (as found in the matrix). Their resulting algorithm, *Clusterfusion*, works by taking in the agreement matrix in order to generate a list, which contains all the pairs where the appropriate cell in the agreement matrix contains a value equal to the number of methods being combined. The proposed method's implementation is difficult and its computing complexity is $O(N^2)$.

E. Finite Mixture Model

The main assumption is that the labels are modeled as random variables drawn from a probability distribution described as a mixture of multinomial component densities. The objective of consensus clustering is formulated as a maximum likelihood estimation problem as in (4). To find the best fitting mixture density for a given data Y we must maximize the likelihood function with respect to the unknown parameters Θ as in (5) :

$$LogL\ (\Theta \left| Y \right.) = Log \prod_{i=1}^{N} p(y_i \left| \Theta \right.) = \sum_{i=1}^{N} Log \sum_{m=1}^{M} a_m\ p_m(y_i | \Theta_m) \quad (4)$$

$$\Theta^* = \arg_{\Theta} MaxLogL \quad (\Theta \left| Y \right.) \quad (5)$$

Expectation Maximization Algorithm (EM) is used to solve this maximum likelihood problem.

Analoui and Sadighian [27] have proposed a probabilistic model of consensus using a finite mixture of multinomial distributions in a space of clustering, a combined partition is found as a solution to the corresponding maximum likelihood problem using the genetic algorithm. The excellent scalability of this algorithm and comprehensible underlying model are particularly important for clustering of large datasets. They calculated a correlation matrix that show correlation between samples and found the best samples that can be in the center of clusters. Next, they employed a genetic algorithm to produce the most stable partitions from an evolving ensemble (population) of clustering algorithms along with a special objective function. The objective function evaluates multiple partitions according to changes caused by data perturbations and prefers those clustering that are least susceptible to those perturbations. A critical problem in this clustering manner is how to adjust initial parameters that they approach this problem by create correlation's matrix and genetic algorithm. The complexity of the proposed algorithm is $O(N^2)$. Their proposed model has unsuitable accuracy.

Topchy *et al*. [2] offered a probabilistic model of consensus using a finite mixture of multinomial distributions in the space of cluster labels. A combined partition is found as a solution to the corresponding maximum likelihood problem using the Expectation Maximization Algorithm (EM). The likelihood function of an ensemble is optimized with respect to the parameters of a finite mixture distribution. Each component in this distribution corresponds to a cluster in the target consensus partition. Their approach completely avoids solving the label correspondence problem. The excellent scalability of this algorithm and comprehensible underlying model are particularly important for clustering of large datasets. EM consensus function need to estimate at least *kHN* parameters. Therefore, accuracy degradation will inevitably occur with increasing number of partitions when sample size is fixed [2]. Their approach able to handle missing data, in this case missing cluster labels (or labels determined to be unknown) for certain patterns in the ensemble (for example, when bootstrap method is used to generate the ensemble). Their approach can operate with arbitrary partitions with varying numbers of clusters, not necessarily equal to the target number of clusters in the consensus partition. The optimal correspondence can be obtained using the Hungarian method for minimal weight bipartite matching problem with $O(k^3)$ complexity for k clusters.

Topchy *et al*. [1] have introduced a unified representation for multiple clusterings and formulate the corresponding categorical clustering problem. They proposed a probabilistic model of consensus using a finite mixture of multinomial distributions in a space of clusterings. A combined partition is found as a solution to the corresponding maximum likelihood

problem using the EM algorithm. They also have defined a new consensus function that is related to the classical intraclass variance criterion using the generalized mutual information definition (QMI) and reduced to the *k*-means clustering in the space of specially transformed cluster labels. Finally, they have demonstrated the efficacy of combining partitions generated by weak clustering algorithms that use data projections and random data splits. A simple explanatory model is offered for the behavior of combinations of such weak clustering components.

VI. COMPARSION OF TECHNIQUES

In table I, we summarize some previous research works related to different types of consensus functions. Next, we compare them based on robustness, accuracy, simplicity and computing complexity.

TABLE I
SUMMARIZED CLUSTERING ENSEMBLES TECHNIQUES

Authors	Advantages	Disadvantages	Computing Complexity
Karypis & Kumar [26] 1998	• Coarsen the graph by collapsing vertices and edge • Partition the coarsened graph Refine the partitions	• Comparing to HBGF, low robust clustering performance of METIS • Low computational cost	O(kNH)
Ng *et al*. [25] 2002	• A popular multiway spectral graph partitioning algorithm (SPEC) that seek to optimize the normalized cut criterion	• Comparing to HBGF, low robust clustering performance of SPEC • High computational cost	O(N^3)
Strehl & Gosh [8], [22] 2002	• Knowledge reuse; to influence a new cluster based on different set of features • Control size of partitions • Low computational cost of HGPA • Improving the quality and robustness of the solution • Allowing one to add a stage that selects the best consensus function without any supervisory information by objective function	• High computational cost of CSPA, MCLA • The proposed greedy approach is slowest and often is intractable for large n	O(kN^2H) O(kNH) O(k^2NH^2)
Fern & Brodley [24] 2004	• Low computational cost • High robust clustering performance against instance based and cluster based approaches • Comparing to IBGF and CBGF, the reduction of HBGF is lossless	• Retaining all of the information of an ensemble • Difficult implementation	O(kN)
Fischer & Buhmann [6], [16] 2003	• Extracting arbitrary shaped structures from the data • Automatic outlier detection • Avoiding the dependency on small fluctuation in the data • High stability	• To need a very large number of clusterings to obtain a reliable result • High computational cost	O(k^3)
Dudoit & Fridlyand [3] 2003	• Reduce variability in the partitioning results via averaging • Improving clustering accuracy by using bagging to cluster analysis • More robust to the variable selection scheme by bagged clustering procedures	• The increase in accuracy observed with PAM is due to a decrease in variability achieved by aggregating multiple clusterings • High computational cost	O(k^3)
Topchy *et al*. [19] 2003	• Using two different weak clustering algorithms: clustering of multidimensional data ; clustering by splitting the data using a number of random hyperplanes • Low computational cost	• Requiring a few restart in order to avoid convergence to low quality local minima	O(kNH)
Luo *et al*. [20] 2006	• Using a consensus scheme via the genetic algorithm • Improving accuracy and robustness	• High computational cost	O(k^3)
Azimi *et al*. [21] 2007	• Generating a new feature space from initial clustering outputs better than pure or normalized feature space • Using a modification of *k*-means for initial clustering named Intelligent *k*-means • Fast convergence • Low computational cost	• Difficult implementation • Unsuitable accuracy	O($k!+kNIdd'$)
Fred [4] 2001	• Using a minimum spanning tree for consistent clusters development • Handling the problem of initialization dependency and selection of the number of clusters by voting-k-means algorithm • Not entail any specificity toward a particular clustering strategy by the proposed technique	• Not corresponding to known number of classes with number of concluded clusters • High computational cost • Fixed threshold	O(N^2)

Kellam *et al*. [13] 2001	• Using a comparison metric known as weighted-kappa and finding known biological relationships amongst gene • Generating robust clusters	• Difficult implementation • High computational cost	$O(N^2)$
Fred & Gain [5] 2002	• Applying a Minimum spanning tree (MST) based clustering algorithm on co-association matrix based on a voting mechanism • The ability of the proposed method to identify arbitrary shaped clusters in multidimensional data	• High computational cost • Poorly performing of method in situations of touching clusters	$O(N^2)$
Topchy *et al*. [23] 2004	• Using a finite mixture of multinomial distributions in the space of cluster labels • The excellent scalability of algorithm • Comprehensible underlying model • Completely avoiding from to solve the label correspondence problem • The ability to handle missing data in this case missing cluster labels for certain patterns in the ensembles • Operating with arbitrary partitions with varying numbers of clusters	• High computational cost for minimal weight bipartite matching problem	$O(k^3)$ $O(kNH)$
Analoui & Sadeghian [27] 2006	• Using the genetic algorithm for producing the most stable partitions • Using a correlation matrix for finding the best samples • The excellent scalability of the proposed genetic algorithm • Comprehensible model for clustering of large datasets • Selecting of clusters with at least perturbation from multiple partitions by objective function	• High computational cost • Unsuitable accuracy	$O(N^2)$

Table I presents the advantages and disadvantages of the previous related research works in clustering ensembles techniques. We investigate their abilities and compare them based on robustness, simplicity, comprehensibility and scalability.

Many proposed algorithms in the previous research works are which based on Hypergraph Partitioning technique contain high robustness, however several algorithms such as METIS, SPEC are compared to HBGF have lower robustness. In the proposed algorithms, researchers confront some difficulties in implementing are their algorithms due to lacking of simplicity in the algorithm. In addition, their proposed approaches often require more time to process the algorithm, and their proposed approaches are often intractable. In this technique, the proposed algorithms contain comparatively low computing complexity. Minimum computing complexity can be found in the algorithm proposed by Fern & Brodley, $O(kN)$, and maximum computing complexity can be found in this algorithm proposed by Ng *et al*., $O(N^3)$.

Many of the proposed algorithms in the previous research works are which based on Voting Approach contains high robustness and high stability. In this technique, the implementation of the proposed algorithms is simpler than other techniques. However, the proposed algorithms contain high computing complexity, $O(k^3)$.

Many of the proposed algorithms in the previous research works are which based on Mutual Information technique achieved while comparatively high robustness. Minimum computing complexity achieved the proposed algorithm by Topchy *et al*., $O(kHN)$, and maximum computing complexity achieved the proposed algorithm by Luo *et al*., $O(k^3)$.

Many of the proposed algorithms in the previous research works are which based on Co-association based functions technique produced high robust clusters and generally they achieved comparatively high robustness. Implementation of many proposed algorithms in this technique is difficult and complicated, in the other hand; the simplicity in the proposed algorithms is not obvious. Nevertheless some of these algorithms emerged fast convergence. Many of the proposed algorithms have high computing complexity, $O(N^2)$.

Many of the proposed models in the previous research works are which based on Finite Mixture Model technique are comprehensible models and their proposed algorithms have comparatively high scalability. Computing complexity in many of the proposed algorithms is high and the maximum computing complexity is $O(k^3)$.

In these research works, the experiments were performed on the datasets from the UCI benchmark repository, including: "Iris", "Wine", "Soybean", "Galaxy", "Thyroid", "Biochem", "Pending", "Yahoo", "Glass", "Isolet6" and some the real world dataset, including: "08X" and some the artificial datasets, including: "3-circle", "Smile", "Half-rings", "2-Spirals", "2D2k", "8D5k", "EOS", "HRCT", "MODIS" [2], [4], [5], [8], [18], [19], [20], [21], [24], [27].

In table II, we show the experiments performed by pervious algorithms and compare their mean error rate (%) of clustering accuracy. In table 2, the mean error rate of some different consensus functions are reported: Co-association function and Average Link (CAL), Co-association function and *K*-means (CK), Hypergraph Partitioning Algorithm (HGPA), Cluster based Similarity Partitioning Algorithm (CSPA), Meta Clustering Algorithm (MCLA), Expectation Maximization Algorithm (EM) and Mutual Information (MI).

TABLE II
COMPARSION OF ACCURACY IN DIFFERENT CLUSTERING ENSEMBLES TECHNIQUES

Authors	Dataset	Error rate of Clustering	Description
Strehl & Gosh [8], [22] 2002	2D2k 8D5k Pending Yahoo	0.68864 0.98913 0.63918 0.41008	Feature Distributed Clustering (FDC) results
Fern & Brodley [24] 2004	EOS Glass HRCT ISOLAT6	0.263 , 0.264 0.263 , 0.247 0.321, 0.342 0.398 , 0.360 0.396 , 0.378 0.401 , 0.393 0.312 , 0.295 0.279 , 0.266 0.314 , 0.289 0.805 , 0.792 0.834 , 0.781 0.830 , 0.793	Random Sup sample and Random Projection on : IBGF, CBGF, HBGF.
Fischer & Buhmann [6], [16] 2003	BSDS100	Mean error for PBC with agglomerative optimization = 16.2% LCE; 22.4%GCE The Bootstrap improve the errors to = 12.4% LCE; 17.0%GCE	Path Based clustering (PBC) Method Two error measure for human segmentation: LCE & GCE (Local Consistency Error & General Consistency Error)
Dudoit & Fridlyand [3] 2003	Leukemia Melnoma	Improvements in accuracy of at least 15% for a majority of models and up to 70% for BagClus1 applied to model 2	To quantify the improvement of bagging over a single application of PAM
Topchy *et al*. [19] 2003	Iris 2-Spirals Half-rings	5.25% ; by Hypergraph methods 0% ; by Single Link Algorithm 5.25% ; by Average Link Algorithm	Misassignment rate (error) of the consensus partition as a measure of performance of clustering combination
Luo *et al*. [20] 2006	Wine Iris 3-Circles Smile	0.3101, 0.3033, 0.3440, 0.3045 0.3533, 0.2133, 0.2733, 0.2533 0.3405, 0.2753, 0.2719, 0.2853 0.3872, 0.1730, 0.3421, 0.2444	Using different generator functions including: Gini, Entropy, Peak, Ge, separately
Azimi *et al*. [21] 2007	Thyroid Iris Wine Soybean	15.99% , 48.89% , 39,03% 6.89% , 4.63% , 6.09% 10.62% , 10.35% , 9.01% 5.97% , 13.78% , 15.51%	Using function for labeling: $f_3(w(i,j)) = w(i,j) / \min w(i,j)$
Fred [4] 2001	Iris	0.67 : 2 Clusters 0.75 : 3 Clusters	Consistency indexes ranging
Fred & Jain [5] 2002	Half-rings Spiral Random Iris	No of Clusters = 2,2,2 No of Clusters = 1,1 No of Clusters = 1,1,3 No of Clusters = 2,3,9	Threshold = 0.4,0.5,0.6,0.7
Topchy *et al*. [23] 2004	Galaxy Biochem Half-rings 2-Spirals Iris	21.1% by *k*-means algorithm 47.4% by *k*-means algorithm 25.6% by *k*-means algorithm 43.5% by *k*-means algorithm 15.1% by *k*-means algorithm	Mixture Model
Topchy *et al*. [23] 2004	Galaxy Biochem Half-rings 2-Spirals Iris	14.43% Average of error rates 42.63% Average of error rates 27.73% Average of error rates 43.2% Average of error rates 10.92% Average of error rates	For EM Algorithm
Topchy *et al*. [1] 2005	Biochem Galaxy 2-Spirals Half-rings Iris	42.63% , 44.59% , 42.86% 14.43% , 15.70% , 50% , 14.65% 43.20% , 43.35% , 43.11% , 46.15% , 42.20% 22.73% , 34.9% , 26.52% , 40.89% , 24.54% 10.92% , 12.99% , 9.38% , 41.36% , 11.06%	EM , QMI , MCLA EM , QMI , HGPA , MCLA EM , QMI , CSPA, HGPA , MCLA EM , QMI , CSPA , HGPA , MCLA EM , QMI , CSPA , HGPA , MCLA
Analoui & Sadighian [27] 2006	Any Dataset	Initial Value Fitness = 0.021556 Highest Value Fitness = 0.04096	Using Genetic Algorithm

As table II shows, important research works in clustering ensembles techniques have empirical results on some different real world and artificial datasets. Researchers encounter accuracy problem. It's clear that many of the proposed algorithms achieved highest accuracy on "Iris" and "Wine" datasets and also, many of the proposed algorithms achieved lowest accuracy on "Biochem" and "2-Spirals" by Topchy *et al*. [1], [2], [19]. Generally, most of the clustering ensembles techniques need to improve their accuracy.

VII. Conclusion and Future works

Clustering ensembles have emerged as a prominent method for improving robustness, stability and accuracy of unsupervised classification solutions. So far, many contributions have been done to find consensus clustering. Firstly, we introduced clustering ensembles and research area and showed different representation of multiple partitions. There are several challenges for clustering ensemble that one of the major problems in clustering ensembles is the consensus function. We summarized clustering combination approaches and focused on consensus function method including: Hypergraph partitioning, Voting approach, Mutual

information, Co-association based functions and Finite mixture model. We investigated some of the most important previous research works in each approach and compared their advantages, disadvantages and computational complexity. The comparison results show that robustness in all of the techniques is high. There are difficulties in implementing the algorithms especially in Hypergraph partitioning and Co-association based functions techniques, thus, simplicity in their algorithms is necessary. Voting approach has the simplest implementation of algorithms between all of these techniques and ideas in this technique can contribute to researchers in implementing their algorithms. High scalability can be found in Finite mixture model technique that more research works can be done. It is clear researchers can do more to investigate scalability in other techniques. One of the most important problems in Hypergraph partitioning technique is to require more time to process the algorithm and also their algorithms are intractable, so it can be a necessary research in future. In Co-association based functions technique, there is fast convergence that it can be useful for researcher to achieve high speed, but it causes the algorithms can not achieve results with high accuracy. As an alternative, researchers can utilize genetic algorithms to achieve better results. Researchers can control generation, crossover and mutation operators in genetic algorithms and this causes their algorithms can not achieve fast convergence and so they achieve better results. Hypergraph partitioning and mutual information have lowest computational complexity between all of the clustering ensembles techniques and other techniques need to improve their computational complexity. Thus, improving computational complexity in voting approach, Finite mixture model and Co-association based functions can be an important investigation in future. We compared accuracy on different datasets in previous techniques. Some of the most important research works in clustering ensembles techniques have empirical results on some different real world and artificial datasets and it shows researchers encounter accuracy problem. Generally, most of the clustering ensembles techniques need to improve their accuracy, therefore improving of accuracy can be an important research in future.

REFERENCES

[1] A. Topchy, A. K. Jain and W. Punch, "Clustering ensembles: Models of consensus and weak partitions," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 27, no. 12, pp. 1866-1881, Dec. 2005.

[2] A. Topchy, A. K. Jain and W. Punch, "A mixture model for clustering ensembles," *Proceedings of the SIAM International Conference on Data Mining*, Michigan State University, USA, 2004.

[3] S. Dudoit and J. Fridlyand, "Bagging to improve the accuracy of a clustering procedure," *Bioinformatics oxford university*, vol. 19, no. 9, pp. 1090-1099, Nov. 2003.

[4] A. L. N. Fred, "Finding consistent cluster in data partitions," *Springer-Verlag Berlin Heidelberg*, MCS, pp. 309-318, 2001.

[5] A. L. N. Fred and A. K. Jain, "Data clustering using evidence accumulation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 835-850, 2002.

[6] B. Fischer and J. M. Buhmann, "Path-based clustering for grouping of smooth curves and texture segmentation," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no.4, Apr. 2003.

[7] Y. Qian and C. Suen, "Clustering combination method," *Proceeding International Conference Pattern Recognition*, vol. 2, 2000.

[8] A. Strehl and J. Ghosh, "Cluster ensembles – A knowledge reuse framework for combining multiple partitions," *Journal of Machine Learning Research*, pp.583-617, Feb. 2002.

[9] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Transaction on Neural Networks*, vol. 16, no. 3, May 2005.

[10] X. Z. Fern and C. E. Brodley, "Random Projection for high dimensional data clustering: A cluster ensemble approach," *Proceedings of the 20th International Conference on Machine Learning (ICML)*, Washington DC., pp.186-193, 2003.

[11] W. Gablentz and M. Koppen, "Robust clustering by evolutionary computation," *Proceeding Fifth Online World Conference Soft Computing in Industrial Applications (WSC5)*, 2000.

[12] P. Kellam, X. Liu, N. Martin, C. Orengo, S. Swift and A. Tucker, "Comparing, contrasting and combining clusters in viral gene expression data," *Proceedings of 6th Workshop on Intelligent Data Analysis*, 2001.

[13] Y. C. Chiou and L. W. Lan, "Genetic clustering algorithms," *EJOR European Journal of operational Research*, vol. 135, pp. 413-427, Nov. 2001.

[14] A. K. Jain, M. N. Murty and P. Flynn, "Data clustering: A Review," *ACM Computing Surveys*, vol. 31, no. 3, pp. 264-323, Sep. 1999.

[15] B. Fischer and J. M. Buhmann, "Bagging for path-based clustering," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 25, no.11, Nov. 2003.

[16] Y. Hong, S. Kwong, Y. Chang and Q. Ren, "Unsupervised feature selection using clustering ensembles and population based incremental learning algorithm," *Pattern Recognition Society*, vol. 41, no. 9, pp. 2742-2756, Dec. 2008.

[17] J. Azimi, M. Mohammadi, A. Movaghar and M. Analoui, "Clustering ensembles using genetic algorithm," *IEEE The international Workshop on computer Architecture for Machine perception and sensing*, pp. 119-123, Sep. 2006.

[18] A. Topchy, A. K. Jain and W. Punch, "Combining multiple weak clusterings," *Proceeding of the Third IEEE International Conference on Data Mining*, 2003.

[19] H. Luo, F. Jing and X. Xie, "Combining multiple clusterings using information theory based genetic algorithm," *IEEE International Conference on Computational Intelligence and Security*, vol. 1, pp. 84-89, 2006.

[20] J. Azimi, M. Abdoos and M. Analoui, "A new efficient approach in clustering ensembles," *IDEAL LNCS*, vol. 4881, pp. 395-405, 2007.

[21] A. Strehl and J. Ghosh, "Cluster ensembles - A knowledge reuse framework for combining partitionings," *Proceeding of 11th National Conference on Artificial Intelligence*, Alberta, Canada ,pp. 93 98, 2002.

[22] A. Topchy, B. Minaei Bidgoli, A. K. Jain and W. Punch, "Adaptive clustering ensembles," *Proceeding International Conference on Pattern Recognition (ICPR)*, pp. 272-275, Cambridge, UK, 2004.

[23] X. Z. Fern and C. E. Brodley, "Solving cluster ensemble problems by bipartite graph partitioning," *Proceedings of the 21st International Conference on Machine Learning*, Canada, 2004.

[24] A. Ng, M. Jordan and Y. Weiss, "On spectral clustering: Analysis and an algorithm," *NIPS 14*, 2002.

[25] G. Karypis and V. Kumar, "A fast and high quality multilevel scheme for partitioning irregular graphs," *SIAM Journal on Scientific Computing*, pp. 359–392, 1998.

[26] M. Analoui and N. Sadighian, "Solving cluster ensemble problems by correlation's matrix & GA," *IFIP International Federation for Information Processing*, vol. 228, pp. 227-231, 2006.