Preproceedings of the Federated Conference on
Computer Science and Information Systems pp. 127–135

A Novel Cluster Ensemble based on a Single Clustering Algorithm

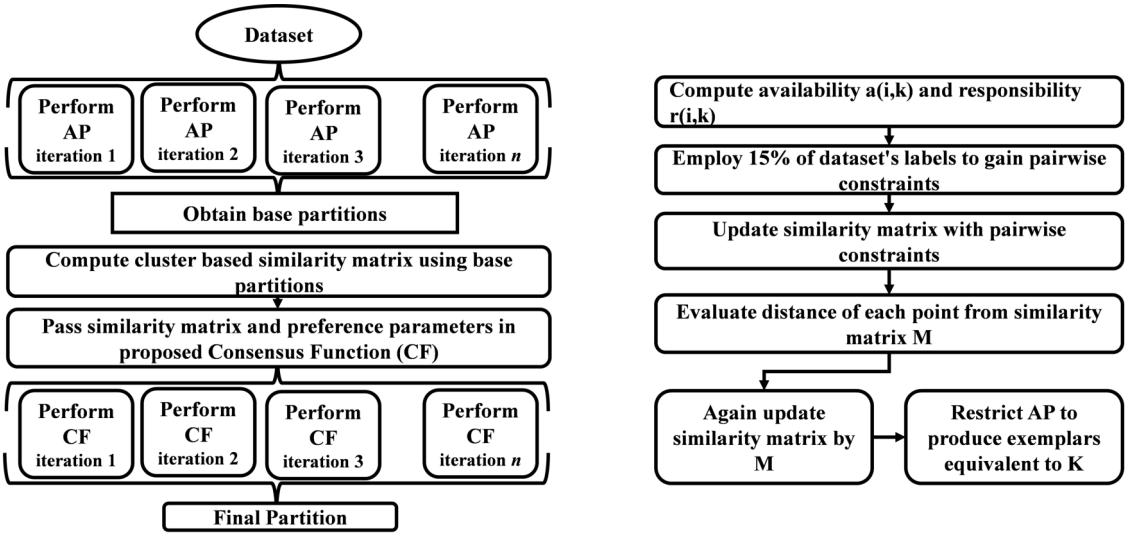
Tahseen Khan[1], Wenhong Tian[1], Mustafa R. Kadhim[1], and Rajkumar Buyya[2,1]
[1]School of Information and Software Engineering, University of Electronic Science and Technology of China, China
tahseen.khan240@gmail.com, tian_wenhong@uestc.edu.cn, mustafa892009@yahoo.com
[2]Cloud Computing and Distributed Systems (CLOUDS) Laboratory
School of Computing and Information Systems, The University of Melbourne, Australia
rbuyya@unimelb.edu.au

***Abstract*—In recent years, several cluster ensemble methods have been developed, but they still have some limitations. They commonly use different clustering algorithms in both stages of the clustering ensemble method, such as the ensemble generation step and the consensus function, resulting in a compatibility issue in terms of working functionality between different clustering algorithms. In addition, in a clustering ensemble method, the accuracy of the final results is a major concern. To deal with it, we propose a novel cluster ensemble method based on a single clustering algorithm (CES). In this method, we iterate a clustering algorithm affinity propagation (AP) ten times in the ensemble generation step to obtain multiple base partitions with a high level of diversity in each iteration due to its nature of producing a random number of clusters. Furthermore, with a few modifications, the same algorithm AP is used to propose a novel consensus function for combining these base partitions into a single partition. The proposed consensus function takes advantage of little side-information in the form of partial labels by using pairwise constraints with AP and number of clusters in a dataset. By employing this information, AP is limited to produce an actual number of cluster centres in a dataset rather than a random number of clusters, which considerably enhanced the accuracy of final outcomes. As a result, CES uses the same clustering functionality in both stages of proposed cluster ensemble method and produces the desired number of clusters in the final partition of a dataset which is significantly improving accuracy when compared to state-of-the-art cluster ensemble methods. Furthermore, as a result of these modifications, the CES outperforms AP in terms of accuracy and execution time. Experiments on real-world datasets from various sources show that CES improves accuracy by 5% on average compared to state-of-the-art cluster ensemble methods and by 55.54% compared to AP while consuming 44.60% less execution time.**

I. Introduction

CLUSTERING is an unsupervised learning technique that seeks to divide a collection of data objects into a set of related classes [1], [2], [3]. It is a crucial and challenging subject in data mining and machine learning, and it has been successfully applied in a wide range of fields, including image processing [4], recommender systems [5], text mining [6], and pattern recognition [7]. A variety of methods have been used in recent years to develop a large number of clustering algorithms [8]. Different algorithms may lead to very different clustering performances for a specific dataset. Each clustering algorithm has its own set of advantages and disadvantages. However, no single algorithm is appropriate for all datasets or applications. Even if a specific algorithm is provided, determining the best parameters for the clustering task can be difficult.

Traditionally, a single clustering algorithm has been used to generate a single clustering result, which has a high rate of inaccuracy. Cluster ensemble has recently emerged as a powerful tool for combining multiple different clustering results (generated by different clustering algorithms or the same algorithm with different iterations) into a potentially better, more robust, and single partition [9]. In detail, a cluster ensemble has mainly two stages: the first, known as the ensemble generation step, obtains multiple base partitions, and the second, known as the consensus function, combines these base partitions [10]. In theory, a functional clustering ensemble must produce reconcilable and well-grounded clustering results when compared to discrete clustering algorithms. However, there were some distinct and demanding issues to deal with while constructing an ensemble for clustering, and it was not as simple as this interpretation suggests. Cluster ensemble is gaining popularity, and several algorithms have been proposed in recent years [11], [12], [13] and [14]. Cluster ensembles can achieve more than a single clustering algorithm in terms of robustness, novelty, stability, and confidence estimation, as well as parallelization and scalability [13]. Despite its considerable success, the current research still faces major challenges. They all have the same flaw: the current cluster ensemble methods use different clustering algorithms in both stages, to obtain base partitions and a final partition, respectively. Furthermore, the use of different clustering algorithms in both stages of the current cluster ensemble architecture may generate compatibility issue related to working functionality. This has motivated us to use a single clustering algorithm in both stages of the new cluster ensemble architecture that significantly improved accuracy of the final outcomes. As a consequence, we propose a novel cluster ensemble method that employs the same clustering in both stages. Accordingly, multiple base partitions are obtained in the its first stage, the ensemble generation process, by executing an unsupervised clustering algorithm affinity propagation (AP) ten times, which provides a high level of diversity among base partitions in each iteration since it generates a random number of clusters [15]. In addition, it also captures all possible different infor-



1(A): Proposed Cluster Ensemble (CES) **1(B): Proposed Consensus Function (CF)**

Fig. 1: 1(A) represents proposed cluster ensemble method, and 1(B) represents proposed consensus function

mation about a data set, which could help improve clustering efficiency. Following that, a similarity matrix is computed between these base partitions, which is known as cluster-based similarity [12]. The computed similarity matrix is then passed as a parameter in the novel consensus function proposed in the second stage of the cluster ensemble method, which uses the same clustering algorithm AP with some modifications. Furthermore, in proposed consensus function, we take advantage of pairwise constraints [16] that employs the concept of must-link (two objects must be in the same cluster) and cannotlink (two objects can not be in the same cluster) with the same clustering algorithm AP to provide a little supervision to the computed similarity matrix. The computed similarity matrix is then updated with this supervised little information, which aids in improving clustering efficiency. At this stage, the similarity matrix is again updated with the Gram matrix, which also enhances clustering efficiency. Furthermore, AP has a flaw in that it generates random number clusters as discussed above. As a result, AP is limited to producing a number of clusters equal to the number of classes in a dataset. This innovative improvement in AP has helped to dramatically increase the accuracy of the final outcomes when this proposed consensus function was used in the proposed cluster ensemble method. As a result, the proposed novel consensus function in cluster ensemble method integrates the base partitions into a single partition. We call our proposed method "A Novel Cluster Ensemble based on a Single Clustering Algorithm (CES)", because we use the same functionality in each stage of it, as shown in Figure 1(A). CES's key benefit is that it eliminates the complication of using two separate clustering paradigms in both stages, making it compatible, and improving clustering outcomes such as accuracy over stare-of-the-art cluster ensemble methods. In addition, when compared to AP, the innovative change significantly improves accuracy and execution time.

This paper makes the following key contributions:

- We propose a novel cluster ensemble method based on a single clustering algorithm , while conventional cluster ensemble methods use different clustering algorithms in both stages, resulting in compatibility issue in ensemble generation and consensus function.
- We propose a novel consensus function based on AP that integrates pair-wise constraints, Gram matrix, and limits AP to produce the actual number of clusters present in the dataset.
- The proposed cluster ensemble method outperforms AP in terms of accuracy and execution time.

The rest of the paper is organized as follows: Section II formulates the background of our work and defines consensus clustering problem. Section III provides details of the proposed framework with selected clustering algorithm AP. Section IV presents the experiments carried out for the framework on different real-world data sets and comparatively explains results. Finally, Section V concludes the paper and reveals the limitation of our work and ongoing work to overcome it.

II. Related Work

A clustering ensemble combines multiple base partitions obtained in ensemble generation step into a robust, accurate and single partition by using a consensus function [11]. The advantage of using cluster ensemble is that it increases the accuracy of the outcomes by taking individual solution biases into account. [17] was the first to propose three cluster ensembles. The first was the cluster-based similarity partitioning algorithm (CSPA), which was based on data point similarity S, with S modified according to whether data points are similar

or dissimilar. The hypergraph partitioning algorithm (HGPA) was the second, which was based on re-partitioning data using the given clusters. The final one was the meta-clustering algorithm (MCLA), which was based on clustering clusters and rendered each cluster by a hyperedge. [12] proposed the Adaptive Clustering Ensemble (ACE), which consisted of three stages: the first was to convert the base clusters into binary representations. The second stage was to find similar clusters based on cluster-based similarity, and the third was to obtain consensus function results by dealing with uncertain objects in order to achieve better final consensus clustering partitions of data. Furthermore, many proposed cluster ensembles has been proposed recently, for example, quadr mutual information consensus function (QMI), mixture model (EM) [13]. QMI is a consensus function based on quadratic mutual information, which is proposed and reduced to k-means clustering in the space of specially altered cluster labels. EM is unsupervised decision-making fusion method based on a probability model of the consensus partition in the space of contributing clusters. [11] proposed the weighted spectral cluster ensemble (WSCE) as a new cluster ensemble focused on group detection arena and graph based clustering concepts. Multiple base partitions are obtained using a new version of spectral clustering and combined into a single robust partition using a proposed consensus function in this method. [14] proposed a cluster ensemble method based on distribution cluster structure, with final results produced using a newly proposed distribution-based normalised hypergraph cut technique. [18] proposed two new cluster ensemble methods: ensemble clustering by propagating cluster-wise similarities with hierarchical consensus function (ECPCS HC) and ensemble clustering by propagating cluster-wise similarities with meta-cluster based consensus function (ECPCS MC). Some research has centred on the applications of cluster ensembles in different areas, for example, time series analysis has become a popular research topic in the field of pattern recognition, particularly for detecting manufacturing flaws. As a result, [19] proposed an automated alternative called control chart pattern recognition (CCPR) model based on consensus clustering. Furthermore, [20] proposed a cluster ensemble method for unsupervised pattern recognition that centred on the growth of damages in composites under solicitations.

The following notations will be used consistently in this paper. Table I also contains several important notations with their definitions that were used in this article. We call a set of objects $D = \{x_1, x_2, ..., x_n\}$, where each object $x_i \in D$ is represented by a vector of N attribute values $x_i = (x_{i,1}, ..., x_{i,N})$. Let $\Gamma = \{\beta_1, \beta_2, ..., \beta_m\}$ be a cluster ensemble with m base partitions, where each base partition is an "ensemble member", and returns a set of clusters $\beta_h = \{\beta_1^h, \beta_2^h, ...\beta_n^h\}$, such that $\bigcup_{p=1}^{k_h} \beta_p^h = D$,where k_h is the number of h^{th} clustering. For each data point $x_i \in D, \beta^h(x_i)$ indicates cluster label in the g^{th} base partition to which data point x_i belongs to, i.e. $\beta^h(x_i) = \beta_h^p$, if $x_i \in \beta_h^p$. As a result, the problem is to find a new partition $\Gamma^* = \beta_1^*, \beta_2^*, ...\beta_K^*$, where K is the number of clusters in the final clustering result of the dataset D, which summarises the details from the cluster ensemble Γ [21].

TABLE I: Important notations used in this paper

Definition	Symbol/Notation
Dataset	D
Data object	$x_i \in D$, $1 \leq i \leq n$
Number of objects	n
Number of ensemble members	m
Ensemble member	β_i, $1 \leq j \leq m$
Similarities between objects	S_{ij}, $1 \leq i \leq n$, $1 \leq j \leq n$
Distance from similarity matrix	P_{ij}, $1 \leq i \leq n$, $1 \leq j \leq n$
euclidean distance	d_{euc}
Similarities between ensemble members	S_m
Preference parameter for ensemble members	p_m

III. A Novel Cluster Ensemble based on a Single Clustering Algorithm

Figure 1A depicts the proposed cluster ensemble method which consists of two steps: (1) an ensemble generation step in which multiple base partitions are obtained by running AP ten times; (2) a proposed consensus function using AP that combines these multiple partitions into a single robust partition. The proposed cluster ensemble method's operation is described in more detail below. Algorithm 1 presents the pseudo code of CES.

A. First Stage: Ensemble Generation Step

The first step is called ensemble generation, and our main goal is to generate m base clustering members. In algorithm 1, steps from 2 to 5 represent the ensemble generation step. Any clustering algorithm can be used to generate ensemble members as long as it produces as many different members as possible [12]. At this stage, different partitions of the same dataset can be created using independent runs of different clustering algorithms or the same clustering algorithm [22][9][18]. Then, in the following stage, a consensus function is used to obtain a final partition from the base partitions generated in the previous stage. Accordingly, we use unsupervised AP, as described in Section III-B1, and run it $(iter = 10)$ times to create multiple m ensemble members, such that $\beta_i \in \Gamma$, where $i \in (1, ..., n)$ and n are the number of data objects. The reason for AP's adoption is that it generates a random set of exemplars (clusters) in β_h, where β_h is an ensemble member, which provides a high level of diversity among ensemble members in each iteration and acquires all possible distinct information about a data set, which may help to increase clustering performance. In other words, in each iteration, AP offers distinct clusters, ensuring the foundation of ensemble

Algorithm 1: The pseudo code of our proposed cluster ensemble method CES

Input: data, No. of clusters K
Output: the clustering Outcomes Γ^*

1: $no\_classes \leftarrow K$, $random \leftarrow []$, $temp \leftarrow []$, $O \leftarrow []$, $s \leftarrow []$ $Z \leftarrow []$, $idx \leftarrow []$, $status \leftarrow []$, $availability \leftarrow a_{ik}$, $responsibility \leftarrow r_{ik}$
2: Calculate m base partitions β_i by executing AP ten times
3: $S_m \leftarrow Euclidean(\beta_i, \beta_i)$ /* where S_m is similarity matrix */
4: $p_m \leftarrow min(S_m)$ /* where p_m is preference parameter */
5: Pass S_m and p_m in proposed consensus function
/* Proposed Consensus Function (modified AP)) */
/* Execute consensus function ten times */
6: Compute a_{ik} and r_{ik}
7: $s \leftarrow .15(labels)$
8: **for** *i = 1 to length(s)* **do**
    for *j = i + 1 to length(s)* **do**
        if $(x_i, x_j) \in C$ **then**
            $status \leftarrow 0$
        else
            $status \leftarrow 1$
        /* where C denotes cannot-link constraints */
9: return *status*
10: S_{ij} & S_{ji} = *status* /* where $i \in (1,...,n)$, $j \in (1,...,n)$ */
11: $P_{ij} \leftarrow \frac{S_{1j}^2 + S_{i1}^2 + S_{ij}^2}{2}$ /* where $i \in (1,...,n)$, $j \in (1,...,n)$ */
12: $S_{ij} \leftarrow P_{ij}$ /* where $i \in (1,...,n)$, $j \in (1,...,n)$ */
13: $Z \leftarrow$ set of exemplars
14: $Z \leftarrow Sort(Z, descending)$
15: **if** $length(Z) < no\_classes$ **then**
    $no\_classes \leftarrow length(Z)$
16: $random \leftarrow Random(length(Z), no\_classes)$
17: $O \leftarrow Z[random]$
18: **for** *i = 1 to no_classes* **do**
    for *j =1 to length(Z)* **do**
        $temp \leftarrow Z[j]$
    if *temp = O(i)* **then**
        $idx \leftarrow temp$
19: return *idx*
20: $\Gamma^* \leftarrow idx$

clustering, which is that ensemble members should have a high level of diversity to capture all of a dataset's information. [12].

Definition 1: Let $X = (X_1, X_2, ...X_N)$ and $Y = (Y_1, Y_2, ...Y_N)$ are two points in euclidean N-space, then Euclidean Distance d_{euc} from point X to Y and Y to X is given by Equation (1) from [23]:

$$\begin{aligned} d_{euc}(X,Y) &= d_{euc}(Y,X) \\ &= \sqrt{(Y_1 - X_1)^2 + (Y_2 - X_2)^2 + ... + (Y_N - X_N)^2} \\ &= \sqrt{\sum_{i=1}^{N} (X_i - Y_i)^2} \end{aligned} \tag{1}$$

where X and Y represent two vectors in euclidean N-space that begin at the space's origin.

Thus, the lower the d_{euc} value between two sets of observations, the more similar they are and the more likely they are in the same cluster. As a result, we use this method to combine the m base partitions found in Section III-A. We use the Euclidean distance, as discussed above in Equation (1), to compute similarities between pairs of ensemble members. The similarities between ensemble members is known as cluster-based similarity. So, as shown in Equation (2), the S_m similarities for m ensemble members can be computed:

$$S_m = \sqrt{\sum_{i=1}^{m} (\beta_i - \beta_j)^2} \tag{2}$$

for all $i \in \{1,...,m\}$ and $j \in \{1,...,m\}$. As a consequence, the base partitions are derived as similarities between m ensemble members, and these base partitions are then grouped into a single partition using the proposed consensus function in Section III-B2. For this, we pass S_m and $p_m = min(S_m)$ in the proposed consensus function parameter, which is proposed using AP.

B. Second Stage: Consensus Function

The consensus function, which is responsible for achieving the final partition of the data by using base partitions generated during the ensemble generation step, is another important component of the cluster ensemble method. We propose a very effective and efficient consensus function, as explained in the sections below, because the consensus function has a direct impact on the performance of the cluster ensemble method. In algorithm 1, steps from 6 to 20 represent the consensus function step. The main idea behind proposing a new consensus function is to compute cluster-based similarities between pairs of ensemble members or clusters rather than computing similarities between data objects [12]. The proposed consensus function's operation is discussed further below. In Section III-B1, we describe some information about the traditional clustering algorithm AP, and then in Section III-B2, we show how it is improved and used in proposing the consensus function.

1) Affinity Propagation (AP): Affinity Propagation (AP)[15] is a clustering algorithm that works on the principle of message passing between data objects. Unlike other

clustering algorithms such as k-medoids or k-means, AP does not seek to determine the number of clusters before running the algorithm. AP, like k-medoids, seeks "exemplars," or members of the input set that are representative of clusters. In other words, rather than taking the number of clusters K as input, AP takes the collection of real-valued similarities S_{ik}, which indicate how well data object at index k is suited to be an exemplar for data object i for two data objects $(x_i, x_k) \in D$. In addition, AP accepts real numbers S_{kk} as input, with the possibility of selecting high similarity data objects as exemplars (number of clusters), referred to as preference p. The exemplars are influenced not only by p but also by message passing. This value can be changed to generate a different number of clusters. Moreover, this value can be a median of the input collection of real-valued similarities that yields a moderate number of clusters or a minimum of these that yields the fewest clusters. Additionally, two real-valued messages which are the 'responsibility' r_{ik} from data object x_i to x_k that depicts how well deserved the data object x_k is to serve as the exemplar of data object xi and the 'availability' a_{ik} from data object x_k to x_i that depicts how suitable it would be for data object x_i to select x_k as its exemplar, are computed. r_{ik} and a_{ik} can be considered as log-probability ratios. Initially, availabilities a_{ik} were set to zero: $a_{ik} = 0$. The responsibilities r_{ik} are then computed using Equation (3).

$$r_{ik} \leftarrow S_{ik} - \max_{k' \, s.t. \, k' \neq k} \{a_{ik'} + S_{ik'}\} \tag{3}$$

Because a_{ik} is set to 0 in the first iteration, r_{ik} has been assigned the difference of s_{ik} and the largest of the similarities between the data object at index i and the other candidates. As a result, if some data objects are assigned to exemplars in subsequent iterations, their availabilities a_{ik} fall below zero, as shown by the Equation (4). And these negative availabilities will have an effect on the similarities $S_{ik'}$ in Equation (3), and the corresponding exemplar will be removed from the competition. And in the Equation (3), for $i = k$, the responsibilities become r_{kk}, which is equivalent to input preference and point at indexed k or i is chosen as an exemplar. This condition allows other candidate exemplars to compete to be an exemplar for a data object and updates availabilities using Equation (4) below.

$$a_{ik} \leftarrow \min \left\{ 0, r_{kk} + \sum_{i' \, s.t. \, i' \notin \{i,k\}} \max \left\{ 0, r_{i'k} \right\} \right\} \tag{4}$$

Thus, in Equation (4), availabilities a_{ik} are assigned to the sum of self-responsibility r_{kk} and positive responsibilities received by the candidate exemplar at index k from other data objects. Only positive responsibilities are added here because it is required for a good exemplar. If self responsibility becomes negative, the availability of data objects at index k can be increased, and self-availability a_{kk} is updated using Equation (5).

$$a_{kk} \leftarrow \sum_{i' \, s.t. \, i' \notin k} \max\{0, r_{i'k}\} \tag{5}$$

As a result, these messages are exchanged between two data objects with pre-computed similarities. At any point, availabilities and responsibilities can be combined to identify a potential exemplar. As a result, $(a_{ik} + r_{ik})$ should be the maximum to determine which data object at index i should be chosen as an exemplar. And knowing $i = k$ leads to knowing the data object that is an exemplar for the data object at index i.

2) Proposed Consensus Function: In proposed consensus function, we take advantage of little side-information such as pairwise constraints [16], which are made up of two constraints: must-link and cannot-link. It has helped to increase the precision in accuracy. We assume that partial class information is provided in the form of pairwise constraints showing whether two objects are members of the same ($must-link$ constraint) or different ($cannot-link$ constraint) clusters. The cluster information is expressed via a set $\Psi \subset D \times D$ $m_l = \{x_i, x_j\}$ where $\Psi = M \cup C$, and

$$\begin{aligned} M &= \{(x_i, x_j) \in D \times D : x_i \text{ and } x_j \in \text{same cluster}\} \\ C &= \{(x_i, x_j) \in D \times D : x_i \text{ and } x_j \in \text{different clusters}\} \\ &\text{where } i, j \in (1, 2, ..., n) \end{aligned} \tag{6}$$

Let us say we have pairwise constraints for some data objects and want to incorporate this side-information into our model. The first question is where we can use this side-information. One approach could be to directly connect the hidden variables corresponding to data points that must be in the same cluster via a function that applies the constraints, and to connect the hidden variables corresponding to cannot-link data objects via a suitable function [24]. Another approach could be to manipulate the similarities between the data objects. If two data objects are in the same cluster, we can maximise their similarities and minimise them if they are in different clusters. As a result, we can conclude that clustering performance is directly related to the similarities between data objects.

Definition 2: Let us suppose there two data objects such that $(x_i, x_j) \in D$ where $i \in (1, 2, ..., n)$, $j \in (1, 2, ..., n)$, the similarities between these objects S_{ij} or S_{ji} will be adjusted according to Equation (7) below.

$$\begin{aligned} (x_i, x_j) \in M \Rightarrow S_{ij} = 1 \,\&\, S_{ji} = 1 \\ \text{and } (x_i, x_j) \in C \Rightarrow S_{ij} = 0 \,\&\, S_{ji} = 0 \end{aligned} \tag{7}$$

As a result, this adjustment in similarities can increase more supervision to improve clustering performance because it increases the probability of similar constraints being in the same cluster as much as possible. As discussed in section III-B1, AP takes as input a collection of similarities between data objects and a preference that can be the median or minimum of the input similarities; unlike other algorithms such as k-means and k-medoids, it does not take the number of exemplars K as input. In addition, after exchanging real-valued messages, it generates a random number of exemplars to compute a_{ik} and r_{ik}, which may affect its clustering performance. So, to solve this problem, we use the number of exemplars K as an input parameter in AP. After that, real-valued messages a_{ik} and r_{ik}

are computed. At this point, we include the concept of pairwise constraints, and 15% of the actual labels were enforced to know constraints for each pair of data objects, and similarities are updated as a result. From Section III-A, we already have S_m and p_m in the AP's parameter. Therfore, S_m is iteratively updated with 1 (if they are in the same cluster) or 0 (if they are not) (if they are in different clusters), for two data objects $(x_i, x_j) \in D$, where $i \in (1,2,...,n)$ and $j \in (1,2,...,n)$.

Definition 3: Let S_{ij} comes from distances between data objects, then there are $x_i \in R^m$, then a matrix P_{ij} from distance matrix S_{ij} can be defined as:

$$P_{ij} = \frac{S_{1j}^2 + S_{i1}^2 + S_{ij}^2}{2} \tag{8}$$

for $i \in \{1,\ldots,n\}$ and $j \in \{1,\ldots,n\}$. P_{ij} is a positive semi-definite matrix of rank at most two which is known as Gram Matrix.

After adjusting similarities with constraints, new similarities are again updated with Gram Matrix as shown in Equation (9).

$$S_m \Leftarrow P_{ij} \tag{9}$$

This motive has come with enhancement in clustering accuracy when this consensus function has been utilized in our proposed cluster ensemble method CES. Finally, a good set of exemplars is obtained by using the updated similarities, as shown in Equation (9). At this point, we solve the previously discussed unsupervised AP problem, which generates a random number of exemplars. We use side-information such as the number of exemplars K passed as input to AP and restrict it to generate exemplars equivalent to K by iterating the obtained fine set of exemplars. As a result, AP clustering accuracy and execution time are dramatically improved. Thus, as shown in Figure 1B, we present a novel consensus function that is used in our cluster ensemble method CES, as shown in Figure 1A. Finally, a single robust dataset partition is produced in Γ^* equivalent to the number of clusters in the dataset.

TABLE II: Real-world data sets taken from different sources

S.No	Dataset	number of objects	Features	Classes
1.	aerosol	905	892	3
2.	alphabet	814	892	3
3.	aquarium	922	892	3
4.	banana	840	892	3
5.	basket	892	892	3
6.	blog	943	892	3
7.	book	896	892	3
8.	heartdisseaseh	294	13	5
9.	glass	214	10	6
10.	heap	155	19	2
11.	wing	856	899	3
12.	water	922	899	3

IV. Performance Evaluation

A. Experimental Design

The proposed clustering ensemble method CES is compared to several representative clustering ensemble methods on a variety of real-world data sets using representative assessment criteria to assess its performance. Our method is tested in ten separate runs. We choose a standard evaluation criterion, such as micro-precision, to assess its performance, which compares real labels to predicted labels to assess clustering approaches' accuracy [29]. [25] has evaluated the consensus cluster's accuracy in terms of true labels using micro-precision. This assessment criteria is also taken into account by [26]. As a result, we have used the only considered evaluation criterion to compare the CES approach to other clustering approaches in order to further evaluate its performance. The following are the remaining paragraphs in this section: The datasets used for comparisons will be addressed first. Then we will go over the assessment criteria and the steps of the experiment in detail.

We choose a variety of real-world data sets to implement the experimental study of the proposed CES approach, which are described in Table II. The twelve real-world data sets, which include different samples, features, and classes, were gathered from various sources, including the UCI repository and the Microsoft Research Asia Multimedia (MSRA-MM) image dataset obtained from Microsoft [30]. These data sets are also used in classification due to the availability of class labels, but class labels are not used in clustering for the evolutionary process of clustering [31]. We use micro-precision to assess the accuracy of the consensus cluster with respect to the true labels. If a data set has K classes and n objects, the micro-precision m_p is defined as in Equation (10):

$$m_p = \sum_{i=1}^{K} \left[\frac{a_i}{n}\right] \tag{10}$$

where a_i represents the number of items in consensus cluster i, and $0 \leq m_p \leq 1$ represents the best possible consensus clustering that is analogous to class labels. As a result, we can assume that the higher the m_p value, the better the clustering performance.

Matlab R2019a was used to design the experiment. Our experiment is divided into two phases: generating ensemble members for these real-world datasets using the clustering algorithm AP, and obtaining consensus function results using the proposed consensus function described in Section III-B2. To begin, a similarity matrix is computed using pairwise euclidean distance and the number of objects n and features f in a dataset, yielding a $n \times n$ similarity matrix S. The preference parameter p is then set to $p = \min(S) / iter \times 0.3$, where $iter$ denotes the iteration number for this step, which is set to 10 to produce m ensemble members. The value $iter \times 0.3$ is used to generate various base partitions and has an impact on clustering performance. The similarity matrix Sm is computed using these acquired base partitions and the preference parameter is set to $p_m = \min(S_m)) / iter \times .09$ after receiving m base partitions after 10 execution of unsupervised

TABLE III: Comparison of Accuracy evaluated using micro-precision between CES and other cluster ensemble methods

Dataset	**CES**	CSPA	HGPA	MCLA	WSCE	EM	QMI	ECPCS MC	ECPCS HC
aerosol	**54.03**	50.28	50.28	50.28	51.27	39.67	50.61	53.26	51.05
alphabet	**51.97**	47.30	47.30	47.30	47.30	37.59	48.40	47.91	48.16
aquarium	**70.17**	**70.17**	**70.17**	**70.17**	69.63	36.23	70.07	65.73	69.96
banana	**47.98**	42.74	42.74	42.74	44.29	39.40	44.17	43.57	43.21
basket	**56.28**	56.05	56.05	56.05	**56.28**	37.89	55.83	52.58	**56.28**
blog	**73.59**	73.49	73.49	73.49	72.64	35.42	73.49	66.49	73.49
book	**57.70**	57.48	57.48	57.48	57.59	36.27	57.48	56.70	57.37
heartdisseaseh	**66.33**	63.95	63.95	63.95	50.00	30.27	54.08	55.10	57.82
glass	**65.42**	35.51	35.51	35.51	58.88	45.79	45.79	52.34	52.80
heap	**79.35**	54.84	54.84	54.84	77.42	59.35	59.35	59.35	58.71
wing	**62.03**	61.92	61.92	61.92	61.68	37.38	6168	57.59	61.68
water	**57.16**	56.94	56.94	56.94	56.29	36.66	56.62	55.86	57.05
Avg	**61.83**	55.89	55.89	55.89	58.61	39.33	56.46	55.54	57.30

TABLE IV: Accuracy and Execution time (seconds) between CES and AP

(a) Comparison of Accuracy between CES and AP

Dataset	AP	CES
aerosol	20.99	54.03
alphabet	15.36	51.97
aquarium	15.08	70.17
banana	18.21	47.98
basket	27.35	56.28
blog	19.72	73.59
book	22.99	57.70
heartdisseaseh	39.80	66.33
glass	53.74	65.42
heap	59.35	79.35
wing	22.90	62.03
water	14.43	57.16
Avg	27.49	61.83

(b) Comparision of Execution time between CES and AP

Datasets	AP(Avg)	AP(Max)	CES(Avg)	CES(Max)
aerosol	5.6822	6.2422	2.6765	2.7853
alphabet	1.9892	2.8138	1.8246	1.8878
aquarium	2.2208	3.1873	2.2521	2.3075
banana	3.5394	4.6083	1.9807	2.0751
basket	5.0143	5.5720	1.9964	2.0163
blog	1.9467	2.9702	2.2170	2.2603
book	1.8947	2.2981	2.1448	2.2040
heartdisseaseh	0.3843	0.6017	0.4923	0.5328
glass	0.3037	0.5951	0.2521	0.2709
heap	0.1677	0.4678	0.1893	0.2066
wing	1.9915	2.8967	1.9963	2.0228
water	4.7009	5.4571	2.2588	2.3218
Avg	2.4863	3.1425	1.6901	1.7409

TABLE V: Comparison of Accuracy between CES and other work with common datasets and evaluation criteria micro-precision

Study	Dataset	Accuracy
CES	blog	73.59
[25]		71.14
CES	aquarium	70.17
[25]		68.56
CES	glass	65.42
[26]		61.21
CES	glass	65.42
[27]		64.40
CES	glass	65.42
[28]		47.20

AP. These parameters, as well as the number of classes K, are passed as input parameters into the proposed consensus function for further calculations to determine final partitions of a dataset in K clusters. The introduced consensus function is also executed with $iter = 10$. The primary goal of this experiment is to evaluate the performance of CES and to see how effective our algorithm is when compared to other traditional clustering ensemble methods such as (CSPA, HGPA, MCLA [17]), (EM, QMI [13], WSCE [11], (ECPCS MC, ECPCS HC [18] by micro-precision. CES also outperforms AP in terms of accuracy and execution time due to innoative changes.

B. Results and Discussions

The accuracy of CES and other traditional cluster ensemble techniques are tested on real-world data sets derived from different sources measured by micro-precision is shown in Table III. Table IV shows the accuracy and execution time evaluated between AP and CES. The experimental results are explained in two parts: (1) comparisons on real-world data sets for accuracy between CES and other cluster ensemble methods, and (2) comparison of accuracy and execution time between AP and CES.

As a result, it is concluded that, when compared to other clustering ensemble methods, CES has achieved promising results in accuracy assessment on all datasets, as shown in Table III. Although CSPA, HGPA, MCLA, and CES achieved comparable accuracy of 70.17% in the dataset aquarium, WSCE, ECPCSHC, and CES also achieved comparable accuracy of 56.28% in the dataset basket, CES outperformed state-of-the-art clustering ensemble methods WSCE, ECPCSMC and ECPCSHC by 5.21%, 6.29% and 4.53% on average respectively. Furthermore, CES has also outperformed all cluster ensemble methods by 5% on average. The use of the same clustering functionality in both cluster ensemble steps may boost the stability of clustering results, resulting in a significant improvement in clustering accuracy. We see

a significant improvement in high-dimensional data sets with noises, such as aerosol, alphabet, aquarium, banana, basket, blog, book, wing, and water, because we limit AP to produce the actual number of clusters in the proposed consensus function. Furthermore, the clustering accuracy has been compared to state-of-the-art cluster ensemble methods that use common data sets and evaluation criterion micro-precision shown in Table V. The clustering ensemble approach HCEKG by [25] has achieved approximately 71.14% and 68.56% clustering accuracy with the blog and aquarium datasets, respectively, whereas our CES has obtained 73.59% and 70.17% indicating 3.33% and 2.29% improvement respectively. [26] has achieved 61.21% accuracy with glass dataset while CES has achieved 65.42%, indicating a 6.45% improvement. With the glass dataset, [27] has achieved 64.4% accuracy, while CES has achieved 65.42%, indicating a 1.56% improvement. [28] has obtained 47.20% accuracy with basket dataset, while CES has obtained 65.42% with a 27.85% improvement.

CES has significantly improved in terms of accuracy and execution time when compared to AP. Table IVa clearly shows that CES achieved a significant improvement in clustering accuracy and execution time when compared to AP. Furthermore, CES has achieved an average accuracy of 61.83% across all twelve datasets, whereas AP has achieved an average accuracy of 27.49% with a 55.54% improvement. When it comes to execution time, CES has significantly outperformed AP as shown in Table IVb. We have measured execution time on various real-world datasets with low and high dimensions, including (heartdisseaseh, 13), (glass, 10), (heap, 19), and (aerosol, 892), (alphabet, 892), (aquarium, 892), (banana, 892), (basket, 892), (blog, 892), (book, 892), (wing, 899) and (water, 899). When considering the maximum time in 10 iterations, CES has consumed 3.4569 seconds, 0.926 seconds, 0.8798 seconds, 2.5332 seconds, 3.5332, 3.5557 seconds, 0.79099 seconds, 0.0941 seconds, 0.0689 seconds, 0.3242 seconds, 0.2612 seconds, 0.8739 seconds, and 3.1353 seconds less than AP. Finally, CES took 1.4016 seconds less than AP on all real-world datasets; additionally, our method has consumed 44.60% less execution time than AP. When it comes to average time, AP outperforms on some of the datasets, but only by a small margin. Nonetheless, when the average performance of average time consumed on all datasets is considered, CES has consumed 32.02% less time than AP. The proposed cluster ensemble method, depicted in Figure 1(A), has quadratic time complexity, i.e., in $O(n^2)$ time, whereas the proposed consensus function, depicted in 1(B), has time complexity of order $O(n^2)$ i.e., $O(n^2+n)$ time.

V. Conclusions and Future Work

In this paper, we proposed a new cluster ensemble method (CES), which is capable of dealing with limitations of traditional cluster ensemble methods which use different clustering algorithms to obtain base partitions in the ensemble generation step and to obtain a single partition in the consensus function that might create a compatibility issue in terms of working functionality in cluster ensemble architecture. Furthermore, the accuracy of the final results was a big worry to cope with. We tested our proposed framework on ten real-world benchmark datasets. The results showed that the proposed clustering ensemble method outperformed state-of-the-art clustering ensemble methods such as the CSPA, HGPA, MCLA, WSCE, EM, QMI, ECPCS MC, and ECPSCS HC algorithms on average. There are several strengths to the proposed cluster ensemble method; firstly, the same clustering functionalities in both of its stages lead the framework more compatible that significantly improves accuracy over state-of-art cluster ensemble methods. Second, it employs a newly proposed consensus function to combine base partitions into a single partition that uses information of cluster centers present in a data set to limit AP to produce a actual number of clusters rather than random number of clusters, resulting in a significant improvement in accuracy and execution time when compared to AP.

The proposed cluster ensemble method has several advantages that researchers can take advantage of. clustering is useful for extracting useful knowledge from large amounts of data. Cluster ensemble is the preferred option for reclustering previously obtained knowledge or hidden patterns from the clustering algorithm in knowledge reuse. The proposed cluster ensemble method can be used to reuse clustering algorithm knowledge and recluster it using the same clustering algorithm. As a result, it avoids the overheads associated with including another clustering algorithm for the consensus function.

As part of future work, we will further enhance the accuracy of CES and compare it to advanced cluster ensemble methods and datasets. We will optimise CES such that its time complexity will be comparable to other cluster ensemble methods. We will explore other cluster algorithms like AP features such as density peaks [32] that help in increasing accuracy significantly.

References

[1] Chang-Dong Wang, Jian-Huang Lai, and S Yu Philip. Multi-view clustering based on belief propagation. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):1007–1021, 2015. doi: 10.1109/TKDE.2015.2503743.

[2] Cosmin Marian Poteraş, Marian Cristian Mihăescu, and Mihai Mocanu. An optimized version of the k-means clustering algorithm. In *2014 Federated Conference on Computer Science and Information Systems*, pages 695–699, 2014. doi: 10.15439/2014F258.

[3] Cosmin M. Poteraş and Mihai L. Mocanu. Evaluation of an optimized k-means algorithm based on real data. In *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, pages 831–835, 2016. doi: 10.15439/2016F231.

[4] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905, 2000. doi: 10.1109/34.868688.

[5] Dimitrios Rafailidis and Petros Daras. The tfc model: Tensor factorization and tag clustering for item recommendation in social tagging systems. *IEEE Transactions on Systems, Man and Cybernetics:Systems*, 43(3):673–688, 2012. doi: 10.1109/TSMCA.2012.2208186.

[6] Dnyanesh G Rajpathak and Satnam Singh. An ontology-based text mining method to develop d-matrix from unstructured text. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, 44(7):966–977, 2013. doi: 10.1109/TSMC.2013.2281963.

[7] Feiping Nie, Shaojun Shi, and Xuelong Li. Auto-weighted multi-view co-clustering via fast matrix factorization. *Pattern Recognition*, 102:107207, 2020. doi: 10.1016/j.patcog.2020.107207.

[8] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010. doi: 10.1016/j.patrec.2009.09.011.

[9] Ana LN Fred and Anil K Jain. Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):835–850, 2005. doi: 10.1109/TPAMI.2005.113.

[10] Pan Su, Changjing Shang, and Qiang Shen. A hierarchical fuzzy cluster ensemble approach and its application to big data clustering. *Journal of Intelligent & Fuzzy Systems*, 28(6):2409–2421, 2015. doi: 10.3233/IFS-141518.

[11] M. Yousefnezhad and D. Zhang. Weighted spectral cluster ensemble. In *2015 IEEE International Conference on Data Mining*, pages 549–558, Nov 2015.

[12] Tahani Alqurashi and Wenjia Wang. Clustering ensemble method. *International Journal of Machine Learning and Cybernetics*, 10(6):1227–1246, 2019. doi: 10.1007/s13042-017-0756-7.

[13] Alexander Topchy, Anil K Jain, and William Punch. Clustering ensembles: Models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12):1866–1881, 2005. doi: 10.1109/TPAMI.2005.237.

[14] Zhiwen Yu, Xianjun Zhu, Hau-San Wong, Jane You, Jun Zhang, and Guoqiang Han. Distribution-based cluster structure selection. *IEEE Transactions on Cybernetics*, 47(11):3554–3567, 2016. doi: 10.1109/TCYB.2016.2569529.

[15] Brendan J Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. doi: 10.1126/science.1136800.

[16] Kiri Wagstaff and Claire Cardie. Clustering with instance-level constraints. *AAAI/IAAI*, 1097:577–584, 2000. doi: 10.5555/645529.658275.

[17] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research*, 3(Dec):583–617, 2002. doi: 10.1162/153244303321897735.

[18] D. Huang, C. Wang, H. Peng, J. Lai, and C. Kwoh. Enhanced ensemble clustering via fast propagation of cluster-wise similarities. *IEEE Transactions on Systems, Man and Cybernetics: Systems*, pages 1–13, 2018. 10.1109/TSMC.2018.2876202.

[19] Siavash Haghtalab, Petros Xanthopoulos, and Kaveh Madani. A robust unsupervised consensus control chart pattern recognition framework. *Expert Systems With Applications*, 42(19):6767–6776, 2015. doi: 10.1016/j.eswa.2015.04.069.

[20] Emmanuel Ramasso, Vincent Placet, and Mohamed Lamine Boubakar. Unsupervised consensus clustering of acoustic emission time-series for robust damage sequence estimation in composites. *IEEE Transactions on Instrumentation and Measurement*, 64(12):3297–3307. doi: 10.1109/TIM.2015.2450354.

[21] Tossapon Boongoen and Natthakan Iam-On. Cluster ensembles: A survey of approaches with recent extensions and applications. *Computer Science and Review*, 28:1–25, 2018. doi: 10.1016/j.cosrev.2018.01.003.

[22] Ashraf Mohammed Iqbal, Abidalrahman Moh'd, and Zahoor Khan. Semi-supervised clustering ensemble by voting. *arXiv preprint arXiv:1208.4138*, 2012.

[23] Teh Ying Wah Ali Seyed Shirkhorshidi, S. Aghabozorgi and Andrew R. Dalby. A comparison study on similarity and dissimilarity measures in clustering continuous data. *PLOS One*, 10:1–20, 2015. doi: 10.1371/journal.pone.0144059.

[24] Inmar Givoni and Brendan Frey. Semi-supervised affinity propagation with instance-level constraints. In *Artificial Intelligence and Statistics*, pages 161–168. doi: 10.1.1.158.678.

[25] Jie Hu, Tianrui Li, Hongjun Wang, and Hamido Fujita. Hierarchical cluster ensemble model based on knowledge granulation. *Knowledge-Based Sytems*, 91:179–188, 2016. doi: 10.1016/j.knosys.2015.10.006.

[26] Hongjun Wang, Hanhuai Shan, and Arindam Banerjee. Bayesian cluster ensembles. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 4(1):54–70, 2011. doi: 10.1002/sam.10098.

[27] Hongjun Wang, Jianhuai Qi, Weifan Zheng, and Mingwen Wang. Semi-supervised cluster ensemble based on binary similarity matrix. In *2010 2nd IEEE International Conference on Information Management and Engineering*, pages 251–254. IEEE, 2010. doi: 10.1109/ICIME.2010.5478054.

[28] Bo Liu, Hong-Jun Wang, Yan Yang, and Xiao-Chun Wang. The method of cluster ensemble based on minimum redundancy feature subset. In *Proceedings of the 2012 International Conference on Electronics, Communications and Control*, pages 2320–2323. IEEE Computer Society, 2012. doi: 10.5555/2417502.2418206.

[29] Zhi-Hua Zhou and Wei Tang. Clusterer ensemble. *Knowlwdge-Based Ssytems*, 19(1):77–83, 2006. doi: 10.1016/j.knosys.2005.11.003.

[30] Hao Li, Meng Wang, and Xian-Sheng Hua. Msra-mm 2.0: A large-scale web multimedia dataset. In *2009 IEEE International Conference on Data Mining Workshops*, pages 164–169. IEEE, 2009. doi: 10.1109/ICDMW.2009.46.

[31] Emrah Hancer. A new multi-objective differential evolution approach for simultaneous clustering and feature selection. *Engineering Application of Artificial Intelligence*, 87:103307, 2020. doi: 10.1016/j.engappai.2019.103307.

[32] Alex Rodriguez and Alessandro Laio. Clustering by fast search and find of density peaks. *Science*, 344(6191):1492–1496, 2014. doi: 10.1126/science.1242072.