

An Ensemble of Locally Reliable Cluster Solutions

Huan Niu ¹, Nasim Khozouie ², Hamid Parvin ^{3,4,*}, Hamid Alinejad-Rokny ^{5,6} ,
Amin Beheshti ^{7,*} and Mohammad Reza Mahmoudi ^{8,9,*}

¹ School of Information and Communication Engineering, Communication University of China, Beijing 100024, China; huanniu80@gmail.com

² Department of computer Engineering, Faculty of Engineering, Yasouj University, Yasouj 759, Iran; n_khozooyi2003@yahoo.com

³ Department of Computer Science, Nourabad Mamasani Branch, Islamic Azad University, Mamasani 7351, Iran

⁴ Young Researchers and Elite Club, Nourabad Mamasani Branch, Islamic Azad University, Mamasani 7351, Iran

⁵ Systems Biology and Health Data Analytics Lab, The Graduate School of Biomedical Engineering, UNSW Sydney, Sydney 2052, Australia; h.alinejad@ieee.org

⁶ School of Computer Science and Engineering, UNSW Australia, Sydney 2052, Australia

⁷ Department of Computing, Macquarie University, Sydney 2109, Australia

⁸ Institute of Research and Development, Duy Tan University, Da Nang 550000, Vietnam

⁹ Department of Statistics, Faculty of Science, Fasa University, Fasa 7461, Iran

* Correspondence: parvin@iust.ac.ir (H.P.); beheshti@gmail.com (A.B.); mohammadrezamahmoudi@duytan.edu.vn (M.R.M.)

Received: 28 October 2019; Accepted: 27 December 2019; Published: 10 March 2020



Abstract: Clustering ensemble indicates to an approach in which a number of (usually weak) base clusterings are performed and their consensus clustering is used as the final clustering. Knowing democratic decisions are better than dictatorial decisions, it seems clear and simple that ensemble (here, clustering ensemble) decisions are better than simple model (here, clustering) decisions. But it is not guaranteed that every ensemble is better than a simple model. An ensemble is considered to be a better ensemble if their members are valid or high-quality and if they participate according to their qualities in constructing consensus clustering. In this paper, we propose a clustering ensemble framework that uses a simple clustering algorithm based on kmedoids clustering algorithm. Our simple clustering algorithm guarantees that the discovered clusters are valid. From another point, it is also guaranteed that our clustering ensemble framework uses a mechanism to make use of each discovered cluster according to its quality. To do this mechanism an auxiliary ensemble named reference set is created by running several kmeans clustering algorithms.

Keywords: ensemble learning; ensemble clustering; kmedoids clustering; local hypothesis

1. Introduction

Clustering as a task in statistics, pattern detection, data mining, and machine learning is considered to be very important [1–5]. Its purpose is to assign a set of data points to several groups. Data points placed in a group must be very similar to each other, while they need to be very different from other data points located in other groups. Consequently, the purpose of clustering is to group a set of objects in such a way that objects in the same group (called a cluster) are more similar (in some sense) to each other than to those in other groups (clusters) and have the maximum difference with other objects within the other clusters [6]. It is often assumed in the definition of clustering that each data object must belong to a minimum of one cluster (i.e., the clustering of all data must be done rather than part of it) and a maximum of one cluster (i.e., clusters must be non-overlapping). Each group is known as

a cluster and the whole process of finding a set of clusters is known as clustering process. All clusters together are called a clustering result or abbreviately a clustering. A clustering algorithm is defined as an algorithm that takes a set of data objects and returns a clustering. Various categorizations have been proposed for various clustering algorithms, including hierarchical approaches, flat approaches, density-based approaches, network-based approaches, partially-based approaches, and graph-based approaches [7].

In consensus-based learning as one of the most important research topics in data mining, pattern recognition, machine learning, and artificial intelligence, we train several simple (often weak) learners to learn how to solve a single problem. In this learning method, instead of learning the data directly by a strong learner (which is usually slow), they try to learn a set of weak learners (which are usually fast) and combine their results with an agreement function mechanism (like a voting) [8]. In supervised learning, the evaluation of each simple learner is straightforward because of the existence of labels for data objects. But, it is not true in non-supervised learning, and consequently it is very difficult to find a solution without the use of side information to assess the weaknesses and strengths of a clustering result (or algorithm) on a dataset. Now, there are several methods for ensemble clustering to improve the strength and quality of clustering task. Each clustering in the ensemble clustering is considered to be a basic learner.

This study tries to solve all of the aforementioned sub-problems by defining valid local clusters. In fact, this study calls data around a cluster center in the clustering of kmedoids as valid local data clusters. In order to generate diverse clustering, a duplicate strategy of producing weak clustering results (i.e., using kmedoids clustering algorithm as base clustering algorithm) is used on non-appeared data in previously valid local clusters. Then, an intra-cluster similarity criterion is used to measure the similarity between the valid local clusters. By forming a weighted graph whose vertices are valid local clusters, the next step of the proposed algorithm of this study is done. The weight of an edge in the mentioned graph is the degree of similarity between the two valid local clusters sharing the edge. In the next step of the algorithm, minimizing the graph cut for partitioning this graph is applied to a number (which is predetermined) of the final cluster. In the final step, the output of these clusters and the average credibility and the agreement of the final agreement clusters are maximized. It should be noted that any other base clusters can also be used as base clusters (for example, the fuzzy c-means algorithm (FCM) can be used). Also, other conventional consensus function methods can also be used as an agreement function for combining the basic clustering results.

The second section of this study addresses literature. The proposed method is presented in the third section. In Section 4, experimental results are presented. In the final section, conclusions and future works are discussed.

2. Related Works

There are two very important problems in ensemble clustering: (1) How to create an ensemble of valid and diverse basic clustering results; and (2) how to produce the best consensus clustering result from an available ensemble. Of course, although each of these two problems has a significant impact on the other one, but these two are widely known and studied as two completely independent problems. That is why only one of these two categories has been addressed in research papers and it has been less seen that both issues are considered together.

The first problem, which is called ensemble generation, tries to generate a set of valid and diverse basic clustering results. This has been done through a variety of methods. For example, it can be generated by applying an unstable basic clustering algorithm on a given data set with a change in the parameters of the algorithm [9–11]. Also, it can be generated by applying different base clustering algorithms on a given data set [12–14]. Another way that we can create a set of valid and diverse base clustering results is to apply a basic clustering algorithm on various mappings from the given dataset [15–23]. In the next step, a set of valid and diverse base clustering results can be created by

applying a base cluster algorithm on various subsets (which can be generated with-replacement or without-replacement) from the given data set [24].

Many solutions have been proposed in order to solve the second problem. The first solution is an approach based on the co-occurrence matrix, in which first we store the number of each pair of data with the same number of the clusters, which contain them simultaneously, in a matrix called the co-occurrence matrix. Then, the final clusters agreement is obtained by considering this matrix the similarity matrix and applying a clustering method (usually a hierarchical clustering method).

This approach is known as the most traditional method [25–28]. Another approach is a graph cutting-based approach. In this approach, first, the problem of finding a consensus clustering becomes a graph partitioning problem. Then, the final clusters are obtained using the partitioning or graph cutting algorithms [29–32]. Four graph-based ensemble clustering algorithms are known as CSPA, HGPA, MCLA, and HBGF.

Another approach is the voting approach [16,17,33–35]. For this purpose, first, a re-labeling must be done. Re-labeling is done in order to align the labels of various clusters to match. Other important approaches include [36–41]: (1) an approach which considers primary clusters an interface space (or new data set) and partitions this new space using a basic clustering algorithm such as the expectation maximization algorithm [37], and (2) an approach which uses evolutionary algorithms to find the most consistent clustering as consensus clustering [36], the approach of using the kmodes clustering algorithm to find an agreement clustering [40,41] (note that kmodes clustering algorithm is equivalent version of kmeans clustering algorithm for categorical data).

Furthermore, an innovative clustering ensemble framework based on the idea of cluster weighting was proposed. Using a certainty criterion, the reliability of clusters was first computed. Then, the clusters with highest reliability values are chosen to make the final ensemble [42]. Bagherinia et al. have introduced an original fuzzy clustering ensemble framework in which effect of the diversity and quality of base clusters were studied [43]. In addition to Alizadeh et al. [44,45], a new study claims that edited NMI (ENMI), which is derived from a subset of total primary spurious clusters, performs better than NMI for cluster evaluation [46].

Moreover, multiple clusters have been aggregated considering cluster uncertainty by using locally weighted evidence accumulation and locally weighted graph partitioning approaches, but the proposed uncertainty measure depends on the cluster size [47].

The ensemble clustering methods are considered to be capable of clustering data of arbitrary shape. Therefore, one of the methods to discover clusters with arbitrarily shaped clusters is clustering ensemble. Consequently, we have to compare our method to some of these methods such as: CURE [48] and CHAMELEON [49,50]. A set of hierarchical clustering algorithms, which aims at extraction of data clustering with arbitrary-shape clusters, uses sophisticated techniques and involves a number of parameters. Two examples of these clustering algorithms are CURE [48] and CHAMELEON [49,50]. CURE clustering algorithm takes a number of sampled datasets and partitions them. After that, a predefined number of distributed sample points are chosen per partition. Then, single link clustering algorithm is employed to merge similar clusters. Because of the randomness of its sampling, CURE is an instable clustering algorithm. CHAMELEON clustering algorithm first transforms the dataset into a k-nearest neighbors graph, and divides it into m smaller subgraphs by graph partitioning methods. After that, the basic clusters represented by these subgraphs are clustered hierarchically. According to the experimental results described in document [49], CHAMELEON algorithm has higher accuracy than CURE algorithm and DBSCAN algorithm.

3. Proposed Ensemble Clustering

This section provides definitions and necessary notations. Then, we define the ensemble clustering problem. In the next step, the proposed algorithm is presented. Finally, the algorithm is analyzed in the last step.

3.1. Notifications and Definitions

Table 1 shows all the symbols used in this study.

Table 1. The used notifications and symbols.

Symbol	Description
D	A dataset
$D_{i:}$	i -th data object in dataset D
$L_{D_{i:}}$	Real label of the i -th data object in dataset D
D_{ij}	j -th feature from i -th data object
$ D_{:1} $	The size of data set D
$ D_{1:} $	The number of features in data set D
Φ	A set of initial clustering results
Φ^i	i -th clustering result in the ensemble clustering Φ
$\Phi_{:,k}^i$	k -th cluster in the i -th clustering result of the ensemble clustering Φ
Φ_{jk}^i	A Boolean indicating whether j th data point of the given dataset belongs to the k -th cluster in the i -th clustering result of the ensemble clustering Φ or not
C	The number of the consensus clusters in given dataset
$R_{:,k}^{\Phi_{:,k}^i}$	A valid sub-cluster from the cluster $\Phi_{:,k}^i$
$R_j^{\Phi_{:,k}^i}$	A Boolean indicating whether j th data point of the given dataset belongs to the valid sub-cluster of the k -th cluster in the i -th clustering result of the ensemble clustering Φ or not
γ	The neighboring radius parameter of the valid cluster in the proposed algorithm
$M^{\Phi_{:,i}}$	center point of cluster $\Phi_{:,i}$
$M_w^{\Phi_{:,i}}$	w -th feature from center point of cluster $\Phi_{:,i}$
π^*	Consensus clustering result
$sim(u, v)$	Similarity between two clusters u and v
$T_q(u, v)$	q -th hypothetical cluster between two center points of clusters u and v
$p_q(\pi^i, \pi^j)$	The center of q -th hypothetical cluster between two clusters u and v
B	The size of the ensemble clustering Φ
C_i	The number of the clusters in the i -th clustering result Φ^i
$G(\Phi)$	The graph defined on the ensemble clustering Φ
$V(\Phi)$	The nodes of the graph defined on the ensemble clustering Φ
$E(\Phi)$	The edges of the graph defined on the ensemble clustering Φ
λ	A clustering result similar to the real labels

3.1.1. Clustering

A set of C non-overlapping subsets of data set can be called as a clustering result (or abbreviately a clustering) or a partitioning result (or abbreviately a partitioning), if the subsets union is the entire data set and the intersection of each pair of subsets is null; any subset of a data set is called a cluster. A clustering is shown by Φ , a binary matrix, where $\Phi_{:,i}$, a vector of size $|D_{:1}|$, represents the i -th cluster; and $\Phi_{i:}^T$, a vector of size C , represents which cluster the i -th data point belongs to. Obviously, $\sum_{j=1}^C \Phi_{ij} = 1$ for any i in $\{1, 2, \dots, |D_{:1}|\}$; and $\sum_{i=1}^{|D_{:1}|} \Phi_{ij} > 0$ for any j in $\{1, 2, \dots, C\}$; and also we have $\sum_{i=1}^{|D_{:1}|} \sum_{j=1}^C \Phi_{ij} = |D_{:1}|$. The center of each cluster $\Phi_{:,i}$ is a data point shown by $M^{\Phi_{:,i}}$, and its j -th feature is defined as Equation (1) [51].

$$M_j^{\Phi_{:,i}} = \frac{\sum_{k=1}^{|D_{:1}|} \Phi_{ki} D_{kj}}{\sum_{k=1}^{|D_{:1}|} \Phi_{ki}} \quad (1)$$

3.1.2. A Valid Sub-Cluster from a Cluster

A valid sub-cluster from a cluster Φ_i is shown by R^{Φ_i} and k -th data point belongs to it if $R_k^{\Phi_i}$ is one. The $R_k^{\Phi_i}$ is defined according to Equation (2).

$$R_k^{\Phi_i} = \begin{cases} 1 & \sqrt{\sum_{j=1}^{|D_{1:}|} |M_j^{\Phi_i} - D_{kj}|^2} \leq \gamma \\ 0 & o.w. \end{cases} \quad (2)$$

where γ is a parameter. It should be noted that a sub-cluster can be considered to be a cluster.

3.1.3. Ensemble of Clustering Results

A set of B clustering results from a given data set is called an ensemble clustering and shown by $\Phi = \{\Phi^1, \Phi^2, \dots, \Phi^B\}$, where Φ^i represents i -th clustering result in the ensemble Φ . Obviously, Φ^k as a clustering has a number of C_k clusters, and therefore, it is a binary matrix of size $|D_{1:}| \times C_k$. The j -th cluster of k -th clustering result of the ensemble Φ is shown by $\Phi_{:j}^{k_1}$. Objective clustering or the best clustering is shown by Φ^* including C clusters.

3.1.4. Similarity Between a Pair of Clusters

There are different distance/similarity criteria between the two clusters. In this study, we define the similarity between the two clusters $\Phi_{:i}^{k_1}$ and $\Phi_{:j}^{k_2}$, which is shown by $\text{sim}(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})$, and defined as Equation (3).

$$\text{sim}(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2}) = \begin{cases} \frac{|\cap(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})|}{|\cup(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})|} + \frac{|\cup_{q=1}^9 T_q(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2}) - \cup(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})|}{\sqrt{\sum_{w=1}^{|D_{1:}|} |M_w^{\Phi_{:i}^{k_1}} - M_w^{\Phi_{:j}^{k_2}}|^2}} & \text{if } \sqrt{\sum_{w=1}^{|D_{1:}|} |M_w^{\Phi_{:i}^{k_1}} - M_w^{\Phi_{:j}^{k_2}}|^2} \leq 4\gamma \\ 0 & o.w. \end{cases} \quad (3)$$

where, $T_q(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})$ is calculated using the Equation (4):

$$T_q(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2}) = \left\{ k : \{1, 2, \dots, |D_{1:}|\} \mid \sqrt{\sum_{w=1}^{|D_{1:}|} |p_{qw}(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2}) - D_{kw}|^2} \leq \gamma \right\} \quad (4)$$

where, $p_{qw}(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})$ is a point whose w -th feature is defined as the Equation (5):

$$p_{qw}(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2}) = \frac{(q) \times M_w^{\Phi_{:i}^{k_1}} + (10 - q) \times M_w^{\Phi_{:j}^{k_2}}}{10} \quad (5)$$

Different T_q for all $q \in \{1, 2, \dots, 9\}$ for two arbitrary clusters (two circles depicted at corners) are depicted in the top picture in Figure 1. Indeed, each T_q is an assumptive region or an assumptive cluster or an assumptive circle. The term $|\cup_{q=1}^9 T_q(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2}) - \cup(\Phi_{:i}^{k_1}, \Phi_{:j}^{k_2})|$ in Equation (3) is the set of all data points in the grey (blue in online version) region at the picture presented by Figure 1. The term p_{qw} is center of the assumptive cluster T_q .

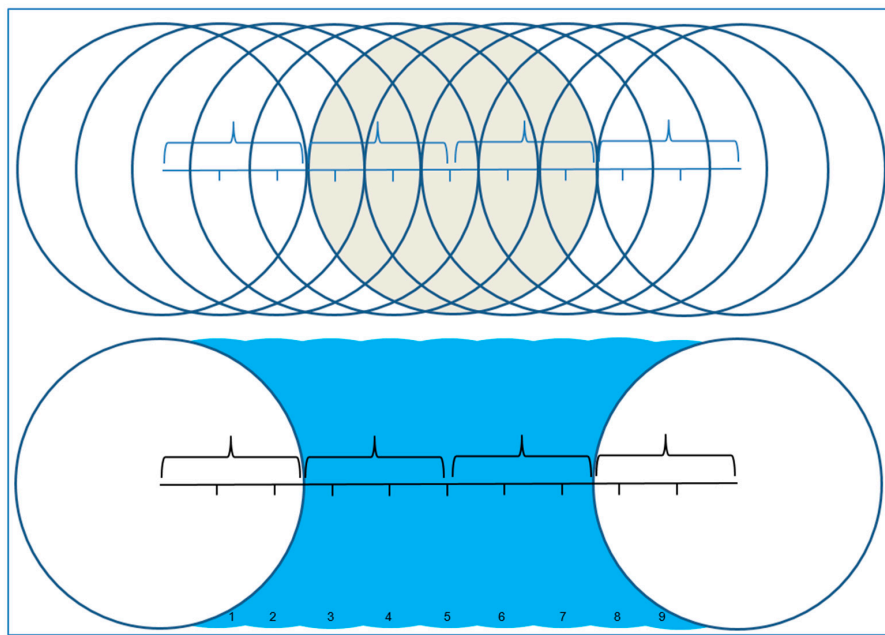


Figure 1. The assumptive regions (clusters) between a pair of clusters.

3.1.5. An Undirected Weighting Graph Corresponding to an Ensemble Clustering

A weighting graph corresponding to an ensemble Φ of clustering results is shown by $G(\Phi)$ and is defined as $G(\Phi) = (V(\Phi), E(\Phi))$. The vertex set of this graph is also the set of all of the valid subsets extracted out of all of the clusters in the ensemble members, namely, $V(\Phi) = \{R^{\Phi_1}_1, \dots, R^{\Phi_1}_{C_1}, R^{\Phi_2}_1, \dots, R^{\Phi_2}_{C_2} \dots R^{\Phi_B}_1, \dots, R^{\Phi_B}_{C_B}\}$. In this graph, the weight of each edge between a pair of the vertices in this graph, or the edge between a pair of the clusters is their similarity and it can be obtained in accordance with Equation (6).

$$E(v_i, v_j) = \text{sim}(v_i, v_j) \quad (6)$$

3.2. Problem Definition

3.2.1. Production of Multiple Base Clustering Results

The set of basic clustering results is generated based on the algorithm presented in Algorithm 1. In this pseudocode, the indices of the whole data set are first stored as T , and then step-by-step, the modified kmedoids algorithm [52] is applied and the result of the clustering is stored. The i th clustering result has at least C_i clusters where C_i is a positive random integer number in the interval $[2; \sqrt{|D_{:1}|}]$.

Algorithm 1 The Diverse Ensemble Generation algorithm

Input: D, B, γ

Output: $\Phi, \text{Clusters}$

01. $\Phi = \emptyset$;

02. For $i = 1$ to B

03. C_i = a positive random integer number in $[2; \sqrt{|D_{:1}|}]$;

04. $[\Phi^i_1, \Phi^i_2, \dots, \Phi^i_{C_i}] = \text{FindValidCluster}(D, C_i, \gamma)$;

05. EndFor

06. $\text{Clusters} = [C_1, C_2, \dots, C_B]$

07. Return $\Phi, \text{Clusters}$

Each base clustering result is an output of a base locally reliable clustering algorithm presented in Algorithm 2. The mentioned method repeats until the number of the objects out of the so-far reliable clusters is less than C^2 . The final cluster centers obtained at any time are extracted using a repeated method to display different sub-sets of data and also ensure that multiple clustering results are used to describe the entire data. Here, we have to explain why it determines the final conditions. Many researchers, in the research background [53,54], argued that the maximum number of clusters in a data set should be less than $\sqrt{|D_{:1}|}$. Thus, as soon as the number of the objects in the dataset which is out of the so-far reliable clusters is less than C^2 , we assume that the dataset can be divided no longer into C clusters. Therefore, the loop ends in that case.

The time complexity of the kmedoids clustering algorithm (a version of the kmeans clustering algorithm) is $O(|D_{:1}|CI)$, so that I is the number of iterations. It should be noted that the kmeans or kmedoids clustering algorithm is a poor learner whose function is affected by many factors. For example, the algorithm is very sensitive to initial cluster centers. So that, selection of different initial cluster centers often leads to different clustering results. In addition, the kmeans or kmedoids clustering algorithm has the tendency to find spherical clusters in relatively uniform sizes that are not suitable for data with other distributions. Therefore, we will try to provide multiple clustering results generated by the kmedoids clustering algorithm in order to create an ensemble of good clustering results over the data set, by distributing different data, instead of using a strong clustering algorithm.

Algorithm 2 The FindValidCluster algorithm

Input: D, C, γ
Output: Φ, C
01. $\Phi = 0; \Pi = 0;$
02. $T = \{1, \dots, |D_{:1}|\};$
03. $counter = 0;$
04. While $((|T|) \geq C^2)$
05. $[\Pi_{:1}, \Pi_{:2}, \dots, \Pi_{:C}] = \text{KMedoids}(D_T, C);$
06. For $k = 1$ to C
07. If $(\sum_{i=1}^{|D_{:1}|} \Pi_{ik} \geq C)$
08. $counter = counter + 1;$
09. $\Phi_{:counter} = \Pi_{:k};$
10. $Rem = \{x \in \{1, \dots, |D_{:1}|\} | R_x^{\Pi_k} = 1\};$
11. $T = T - Rem;$
12. EndIf
13. EndFor
14. EndWhile
15. $C = counter;$
16. Return $\Phi, C;$

3.2.2. Time Complexity of Production of Multiple Base Clustering Results

The incremental method is called the kmedoids clustering algorithm, which is called the base locally reliable clustering algorithm presented in Algorithm 2. The time complexity of the base locally reliable clustering algorithm presented in Algorithm 2 is $O(|D_{:1}|IC)$, which is in the worst case $O(|D_{:1}|^{1.5}I)$ where I is number of the iterations the kmedoids clustering algorithm needs to converge. The algorithm of generating the ensemble of clustering results presented in Algorithm 1 is $O(|D_{:1}|I \sum_{i=1}^B C)$, which is the worst case $O(|D_{:1}|^{1.5}IB)$ where B is the number of basic clustering results generated. The outputs of the algorithm presented in Algorithm 1 have been the clustering set Φ and also the set of clusters' numbers in ensemble members.

3.3. Construction of Clusters' Relations

Class labels show specific classes in the classification, while cluster labels express only the data grouping features and are not comparable in cluster analysis in different clustering results. Therefore, different clustering labels must be aligned in the ensemble clustering. Additionally, since the kmeans and kmedoids clustering algorithms can only detect spherical and uniform clusters, a number of clusters in a same clustering result can inherently be a same cluster. Therefore, analysis of the relationship between clusters through a between-cluster similarity measure is needed.

Now, there are a large number of criteria proposed in the research background [49,55–57] to measure the similarity among the clusters. For example, in the chain clustering algorithm, the distance between the closest or farthest data object of the two clusters is used for measuring the cluster separation [56,57]. They are sensitive to noise because of their dependence on a few objects. In the center-based clustering algorithms, distance between the centers of clusters measures the lack of correlation between the two clusters. Although this measure is considered to be a computationally efficient and powerful measure to deal with noise, it cannot reflect the boundary between the two clusters.

The number of common objects created by the two clusters is used to represent their similarity in cluster grouping algorithms. This measure does not consider the fact that the cluster labels of some objects may be incorrect in a cluster. Therefore, some of these objects have a significant impact on the measurement. Additionally, since two clusters of a same clustering do not have any common objects, measurement cannot be used to measure their similarity. Although there are good practical implementations of different measures, they are not suitable for ensemble clustering. In the previous section, the basic clustering generated Φ with valid local labels are different, which means that the labels of each cluster are only partially valid. Therefore, measurement of the difference between the two clusters in our local labels, instead of all the labels, is needed. However, the overlap between the local spatial spaces of both clusters should be very small because of the basic clustering generation mechanism. Therefore, we consider an “indirect” overlap between the two clusters in order to measure their similarity.

Let us assume we have $\Phi_{:i}$ and $\Phi_{:j}$ as the two clusters, $M^{\Phi_{:i}}$ and $M^{\Phi_{:j}}$ as their cluster centers, and consequently, $p_5(\Phi_{:i}, \Phi_{:j})$ as the middle point of the two centers. We assume there is a hidden dense region between the reliable sections of the clusters pair $\Phi_{:i}$ and $\Phi_{:j}$, i.e., $R^{\Phi_{:i}}$ and $R^{\Phi_{:j}}$. We define 9 points $p_k(\Phi_{:i}, \Phi_{:j})$ for $k \in \{1, 2, \dots, 9\}$ at the equal distances on line connecting $M^{\Phi_{:i}}$ to $M^{\Phi_{:j}}$. We assume whatever the number of objects is larger in all of the valid local spaces, it is more likely that those clusters are the same. If all of the valid local spaces are dense and the distance between $M^{\Phi_{:i}}$ and $M^{\Phi_{:j}}$ is not greater than 4γ , the likelihood that those clusters are the same should be a high value, as shown in Figure 1. For clusters $\Phi_{:i}$ and $\Phi_{:j}$, we examine the following two factors in order to measure their similarity: (1) The distance between their cluster centers, (2) the possibility of the existence of a dense region between them. As we know, if the distance between their cluster centers is smaller, it is more likely that they may be the same cluster. Therefore, we assume that their similarity must be inversely proportional to this distance. Additionally, we know that, since the kmedoids clustering algorithm is a linear clustering algorithm, the spaces of both clusters are separated by the middle line between their cluster centers. If the areas around them contain a few objects, i.e., they are sparse, they can be clearly identified.

An example has been presented in Figure 2. It is observed that the distance between the centers of the clusters B and C is not greater than the distance between clusters A and B. But we find that the boundary between the clusters B and C is clearer than the boundary between the clusters A and B. Therefore, if the clarity between boundary and clusters is considered, there is more distance between clusters B and C than clusters A and B. Based on the above analyses, we assume that their similarity should be proportional to the probability of existence of dense regions between their centers. So similarity between two clusters is formally measured based on Equation (3).

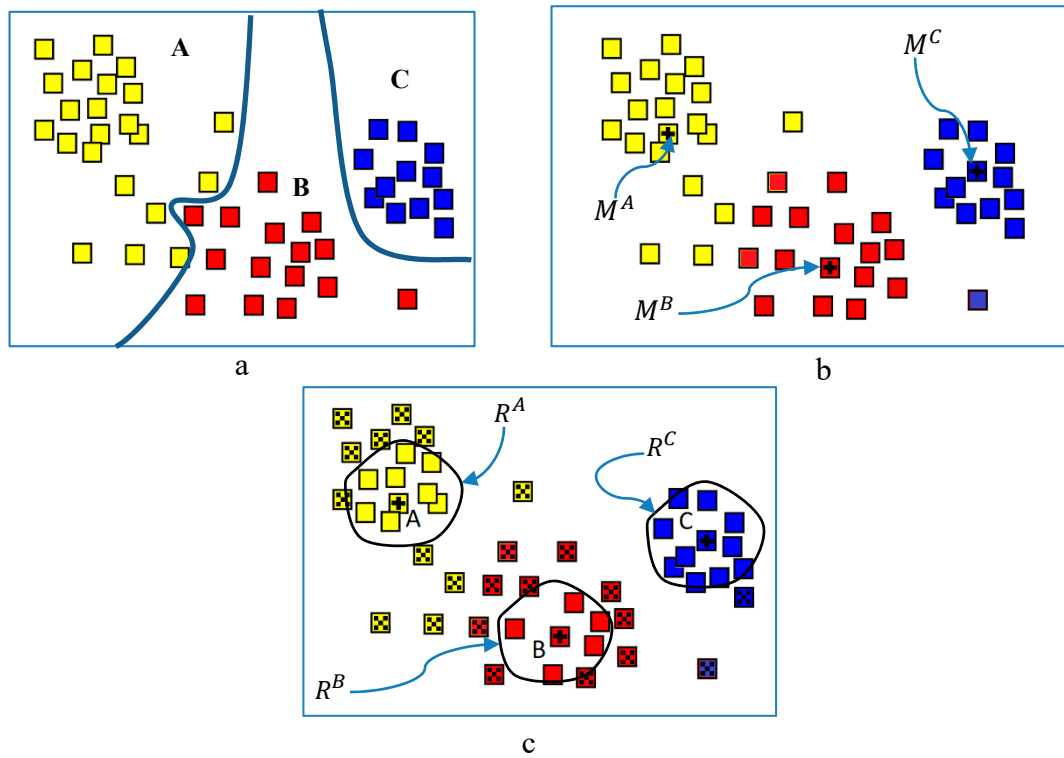


Figure 2. (a) An exemplary dataset with three clusters; (b) the assumptive cluster centers of applying kmedoids on the given dataset; and (c) removing unreliable data points in each cluster. In (a) and (b), dots mean the representative centers of clusters, and crosses means removed data points or unreliable data points.

Based on the similarity criterion, we generated a *weighted undirected graph* (WUG), denoted by $G(\Phi) = (V(\Phi), E(\Phi))$, to show the relationship between these clusters. In the mentioned graph $G(\Phi)$, $V(\Phi)$ is the set of vertices that represent clusters in the ensemble Φ . Therefore, each vertex is also seen as a cluster in the ensemble Φ . The $E(\Phi)$ is the weights of the edges between vertices, i.e., clusters.

For a pair of clusters, their similarity is used as the weight of the edge between them, i.e., the weight is calculated according to Equation (6); and the more similarity between them, the more likely they show a same cluster. After obtaining the WUG, the problem of determining the cluster relationship can be transferred to a normal graph partitioning problem [58]. Therefore, a partition of vertices in the graph $G(\Phi)$ is obtained and it is denoted by CC which is of the size $\sum_{i=1}^B C_i \times C$. The $CC_{ji} = 1$ if the R_i^p belongs to the i th consensus cluster where $\sum_{i=1}^{p-1} C_i + q = j$. We want to obtain such a partitioning by minimizing an objective function where vertices are very similar in the same subsets and are very different from the vertices in other subsets. In order to solve the optimization problem, we apply a normalized spectral clustering algorithm [59] to obtain a final partition of CC. The vertices in the same subsets are used to represent a cluster. Therefore, we define a new ensemble with aligned clusters denoted by Λ based on Equation (7).

$$\Lambda_{ir}^k = \begin{cases} 1 & \left(\sum_{t=1}^{k-1} C_t + j = p \right) \wedge \left(R_i^{\Phi^k} = 1 \right) \wedge (CC_{pr} = 1) \\ 0 & o.w. \end{cases} \quad (7)$$

The time complexity to make the cluster relationship will be $O(|D_{:1}|(\sum_{t=1}^B C_t)^2)$.

3.4. Extraction of Consensus Clustering Result

After securing the ensemble of the aligned (or relabeled) clustering results out of the main ensemble of the clustering results, Λ , where Λ_{ij}^k is a matrix of size $|D_{:1}| \times C$ for any $k \in \{1, 2, \dots, B\}$, is now available for the extraction of consensus clustering result. According to the ensemble Λ , the consensus function can be rewritten according to Equation (8).

$$\pi_{ij}^* = \begin{cases} 1 & \forall p \in \{1, 2, \dots, C\} : \bar{\Lambda}_{ij} \geq \bar{\Lambda}_{ip} \\ 0 & o.w. \end{cases} \quad (8)$$

where $\bar{\Lambda}_{ij} = \sum_{k=1}^B \Lambda_{ij}^k$. The complexity of the final cluster generation time is $O(|D_{:1}|CB)$.

3.5. Overall Implementation Complexity

The general complexity of the proposed algorithm is equal to $O(|D_{:1}|I(\sum_{t=1}^B C_t) + |D_{:1}|(\sum_{t=1}^B C_t) + |D_{:1}|(\sum_{t=1}^B C_t)^2 + |D_{:1}|CB)$. We observe that the time complexity is linear proportional to the number of objects, and for ensemble learning, the greater number of base clusters, i.e., $\sum_{t=1}^B C_t$, does not mean the ensemble performance is better. Therefore, we can control the equation $\sum_{t=1}^B C_t \ll |D_{:1}|$, so that the proposed algorithm is suitable for dealing with large-scale data sets. However, if there is enough computational resource, we can increase C_t up to $|D_{:1}|^{0.5}$ and consequently assuming $\sum_{t=1}^B C_t \approx |D_{:1}|^{0.5}$, the general complexity of the proposed algorithm is $O(|D_{:1}|^{1.5}I + |D_{:1}|^2)$. But, if there is not enough computational resource, the complexity of the proposed algorithm is linear with the data size.

4. Experimental Analysis

In this section, we test the proposed algorithm on four artificial datasets and five real-world datasets and evaluate its efficacy using (1) external validation criteria and (2) time assessment costs.

4.1. Benchmark Datasets

Experimental evaluations have been performed on nine benchmark datasets. The details of these datasets are shown in Table 2. The cluster distribution of the artificial 2D datasets has been shown in Figure 3. The real-world datasets are derived from the UCI datasets' repository [60].

Table 2. Description of the benchmark datasets; the number of data objects ($|D_{:1}|$), the number of attributes ($|D_{1:}|$), the number of clusters (C).

	Dataset	$ D_{:1} $	$ D_{1:} $	C
Artificial dataset	Ring3 (R3)	1500	2	3
Artificial dataset	Banana2 (B2)	2000	2	2
Artificial dataset	Aggregation7 (A7)	788	2	7
Artificial dataset	Imbalance2 (I2)	2250	2	2
UCI dataset	Iris (I)	150	4	3
UCI dataset	Wine (W)	178	13	3
UCI dataset	Breast (B)	569	30	2
UCI dataset	Digits (D)	5620	63	10
UCI dataset	KDD-CUP99	1,048,576	39	2

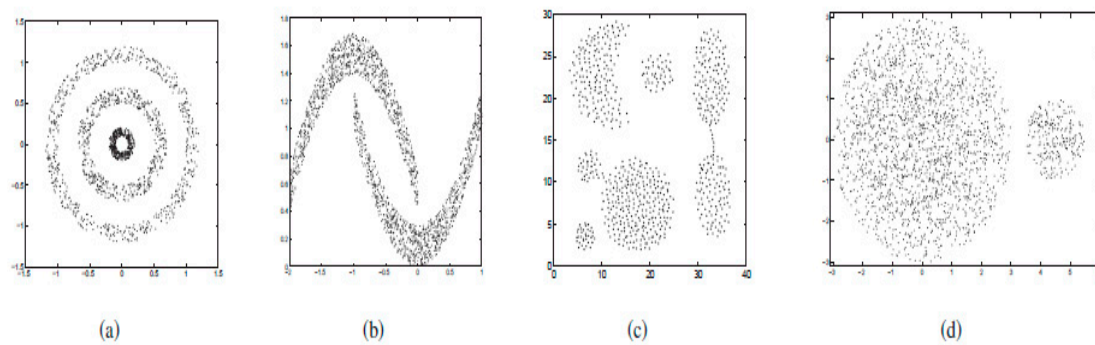


Figure 3. Distribution of four artificial datasets: (a) Ring3, (b) Banana2, (c) Aggregation7, and (d) Imbalance2.

4.2. Evaluation Criteria

Two external criteria have been used to measure the similarity between the output labels predicted by different clustering algorithms and the correct labels of the benchmark datasets. Let us denote the clustering similar to the real labels of dataset by λ . It is defined according to Equation (9).

$$\lambda_{ij} = \begin{cases} 1 & L_{D_i} = j \\ 0 & o.w. \end{cases} \quad (9)$$

where L_{D_i} is the real label of the i th data point.

Given a dataset D and two partitioning results over these objects, namely π^* (the consensus clustering result) and λ (the clustering similar to the real labels of dataset), the values between π^* and λ can be summed up in a probable table. It is presented in Table 3, so that n_{ij} shows the number of common data objects in the groups $\pi_{:i}^*$ and $\lambda_{:j}$.

Table 3. Reminder for the default table to compare two partitioning results.

		λ				
		$\lambda_{:1}$	$\lambda_{:2}$	\dots	$\lambda_{:C}$	$b_i = \sum_{k=1}^C n_{ik}$
π^*	$\pi_{:1}^*$	n_{11}	n_{12}	\dots	n_{1C}	b_1
	$\pi_{:2}^*$	n_{21}	n_{22}	\dots	n_{2C}	b_2
	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
	$\pi_{:C}^*$	n_{C1}	n_{C2}	\dots	n_{CC}	b_C
$d_i = \sum_{k=1}^C n_{ki}$		d_1	d_2	\dots	d_C	

The adjusted rand index (ARI) is defined based on Equation (10).

$$ARI(\pi^*, \lambda) = \frac{\sum_{ij} \binom{n_{ij}}{2} - \left[\sum_i \binom{b_i}{2} \sum_i \binom{d_i}{2} \right]}{\binom{n}{2}} \quad (10)$$

$$\frac{\frac{1}{2} \left[\sum_i \binom{b_i}{2} + \sum_i \binom{d_i}{2} \right] - \left[\sum_i \binom{b_i}{2} \sum_i \binom{d_i}{2} \right]}{\binom{n}{2}}$$

where the variables are defined in Table 3. Normalized mutual information (NMI) [61] is defined based on Equation (11).

$$NMI(\pi^*, \lambda) = \frac{2 \sum_i \sum_j n_{ij} \log \frac{n_{ij}n}{b_i b_j}}{\sum_{ij} b_j \log \frac{b_i}{n} - \sum_j d_j \log \frac{d_j}{n}} \quad (11)$$

The more similar the clustering result π^* (i.e., the consensus clustering result) and the ground truth clustering (the real labels of dataset), i.e., λ , the more the value of their NMI (and ARI).

4.3. Compared Methods

In order to investigate the performance of the proposed algorithm, we compare it with the state-of-the-art clustering ensemble algorithms including: (1) Evidence accumulation clustering (EAC) along with single link clustering algorithm as consensus function (EAC + SL) and (2) average link clustering algorithm as consensus function (EAC + AL) [9], (3) weighted connection triple (WCT) along with single link clustering algorithm as consensus function (WCT + SL) and (4) average link clustering algorithm as consensus function (WCT + AL) [27], (5) weighted triple quality (WTQ) along with single link clustering algorithm as consensus function (WTQ + SL) and (6) average link clustering algorithm as consensus function (WTQ + AL) [27], (7) combined similarity measure (CSM) along with single link clustering algorithm as consensus function (CSM + SL) and (8) average link clustering algorithm as consensus function (CSM + AL) [27], (9) cluster based similarity partitioning algorithm (CSPA) [29], (10) hyper-graph partitioning algorithm (HGPA) [29], (11) meta clustering algorithm (MCLA) [29], (12) selective unweighted voting (SUW) [17], (13) selective weighted voting (SWV) [17], (14) expectation maximization (EM) [37], and (15) iterative voting consensus (IVC) [40].

In addition, we compared the proposed method with other “strong” base clustering algorithms including: (1) the normal spectral clustering algorithm (NSC) [59], (2) “density-based spatial clustering of applications with noise” algorithm (DBSCAN) [62], (3) “clustering by fast search and find of density peaks” algorithm (CFSFDP) [63]. The purpose of this comparison is to test whether the proposed method is a “strong” clustering or not.

4.4. Experimental Settings

A number of settings for the different ensemble clustering algorithms are listed in the following so as to ensure that they are reproducible. The number of clusters in each base clustering result is randomly set in the proposed ensemble clustering algorithm where the kmedoids clustering algorithm is also used to produce their basic clustering results. In each of the state-of-the-art clustering algorithms, the number of clusters per each base clustering result in the ensemble is set according to the method they used; and the kmeans clustering algorithm is also used to produce their basic clustering results. The parameter B is always set to 40. For these compared methods, we determine their parameters based on their authors’ suggestions. The quality of each clustering algorithm is reported as an average over 50 independent runs. A Gaussian kernel has been employed for the NSC algorithm and a value in a range greater than or equal to 0.1 and less than or equal to 2 with step size 0.1 is chosen as the kernel parameter σ^2 . In these parameters, the best clustering result has been selected for comparison.

DBSCAN and CFSFDP algorithms also require the input parameter ε . We estimated the value of ε using the average distance between all data points and their average point, denoted by \overline{ASE} . However, each of these algorithms may require different values of ε . Therefore, we have evaluated each of these algorithms with ten different values from the following set $\left\{ \frac{\overline{ASE}}{1}, \frac{\overline{ASE}}{2}, \frac{\overline{ASE}}{3}, \frac{\overline{ASE}}{4}, \frac{\overline{ASE}}{5}, \frac{\overline{ASE}}{6}, \frac{\overline{ASE}}{7}, \frac{\overline{ASE}}{8}, \frac{\overline{ASE}}{9}, \frac{\overline{ASE}}{10} \right\}$ and the best clustering result is used for comparison.

4.5. Experimental Results

4.5.1. Comparison with the State of the Art Ensemble Methods

Different consensus functions have been first used to extract the final clustering result out of the output ensemble of the Algorithm 1. As it is clear, the clustering results in the output ensemble of

the Algorithm 1 are not complete; therefore, to apply EAC on the output ensemble of the Algorithm 1, edited EAC (EEAC) is needed [64,65]. CSPA, HGPA, and MCLA are also applied on the output ensemble of the Algorithm 1. EM has been also used to extract the final clustering result out of the output ensemble of the Algorithm 1. Therefore, seven methods including PC + EEAC + SL, PC + EEAC + AL, PC + CSPA, PC + HGPA, PC + MCLA, PC + EM, and the proposed mechanism presented in Section 3.4 have been used as different consensus functions to extract the final clustering result out of the output ensemble of the Algorithm 1. Here, PC stands for the proposed base clustering presented in Algorithm 1. Experimental results of different ensemble clustering methods on different datasets in terms of ARI and NMI have been presented respectively in Figures 4 and 5. The last seven bars stand for performances of the seven mentioned methods. All these results have been summarized in last seven rows of Table 4. The proposed consensus function presented in Section 3.4 is the best method and the PC + EEAC + SL consensus function is the second. According to Table 4, the PC + EEAC + AL and PC + MCLA consensus functions are the third and fourth methods. Therefore, the proposed mechanism presented in Section 3.4 is considered to be our main consensus function.

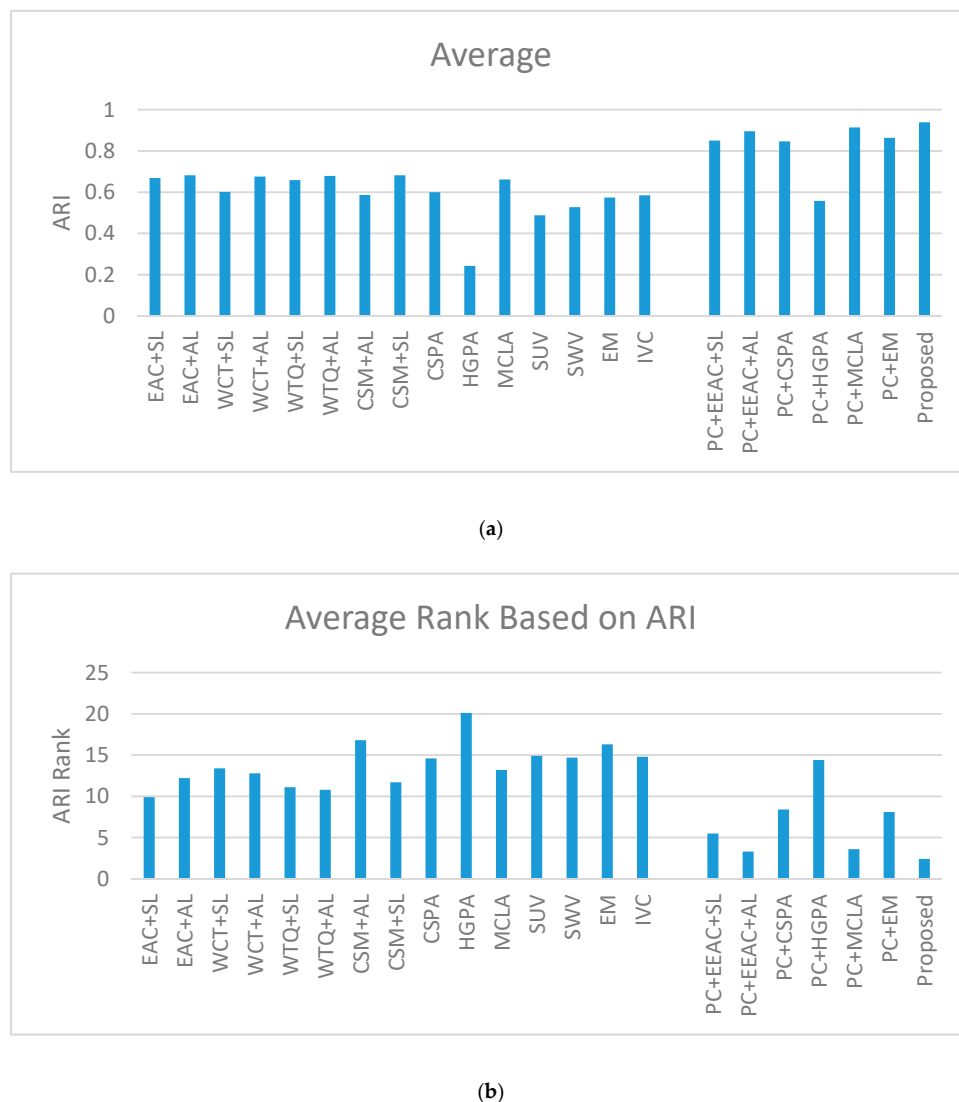


Figure 4. Experimental results of applying different consensus functions considering the output ensemble of the Algorithm 1 on different datasets in terms of (a) adjusted rand index (ARI) and (b) rank of adjusted rand index (ARI Rank).

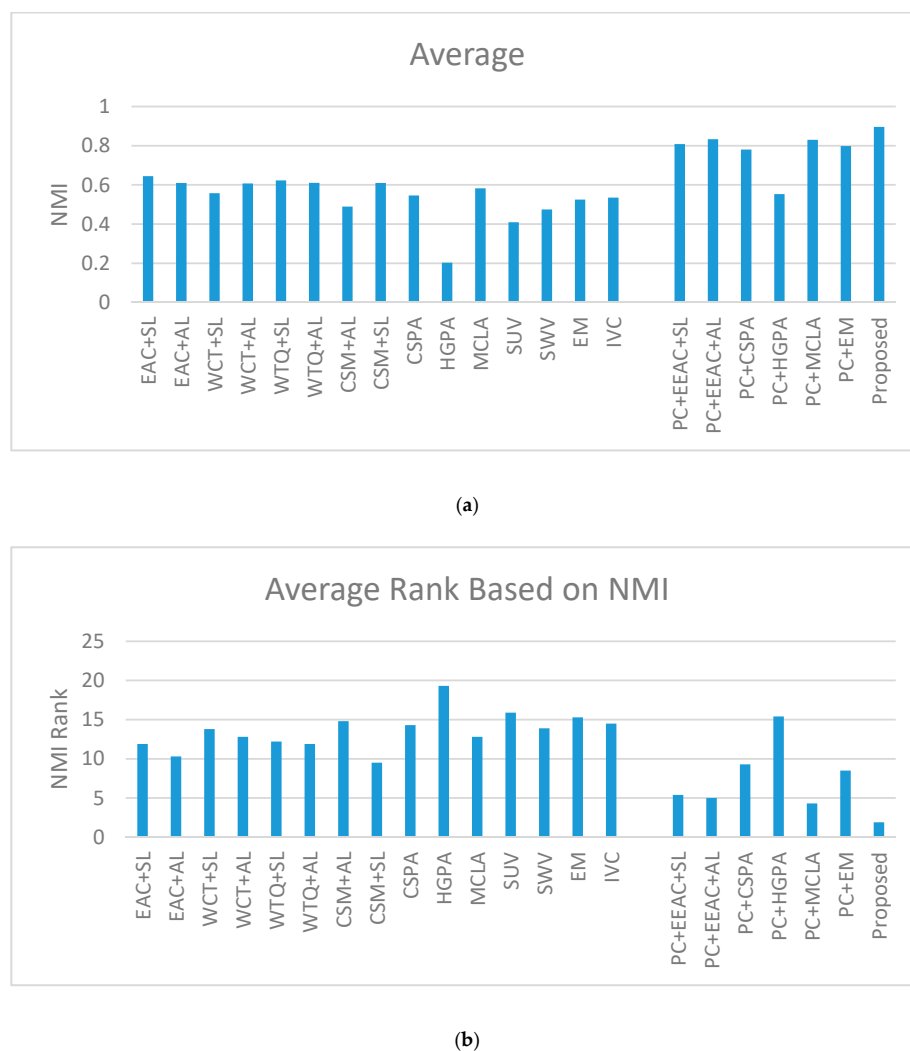


Figure 5. Experimental results of applying different consensus functions considering the output ensemble of the Algorithm 1 on different datasets in terms of (a) normalized mutual information (NMI) and (b) rank of normalized mutual information (NMI Rank).

Based on the ARI and NMI criteria, the comparison of the performances of different ensemble clustering algorithms on the artificial and real-world benchmark datasets is shown respectively in Figures 4 and 5, and they are summarized in Table 4. As shown in Figures 4 and 5, we observe that the proposed ensemble clustering algorithm has a high clustering accuracy in the benchmark synthetic and real-world datasets compared to other existing ensemble clustering algorithms. According to the experimental results, the proposed ensemble clustering algorithm can detect different clusters in an effective way and increase the performance of the state-of-the-art ensemble clustering algorithms.

Also, as shown in Table 4, the efficiency of the proposed ensemble clustering algorithm is significantly better than other ensemble clustering algorithms in the artificial datasets. However, accuracy improvement of the proposed ensemble clustering algorithm in the real-world datasets is marginally better than other ensemble clustering algorithms. The main reason for this is that the complexity of the real-world datasets is larger than the complexity of the artificial datasets. Considering the performance of each method on different datasets to be a variable, a Friedman test is performed. It is proved (with a p -value about 0.006) that there is a significant difference between our variables. It is shown through the post-hoc analysis that the difference is mostly because of the difference between PC+MCLA and the proposed method with a p -value of about 0.047. Therefore, the difference between the proposed method performance and the performance of the most effective method aside from the

proposed method, i.e., PC+MCLA, contributing mostly in Friedman test p -value (i.e., 0.006), is with a p -value equal to 0.047 which is still considered to be significant.

Table 4. The summary of the results presented in Figure 5. The column L - D - W indicates the number of datasets on which the proposed method *Loses to-Draws with-Wins against* a rival validated by paired t -test [66] with the confidence level of 95%.

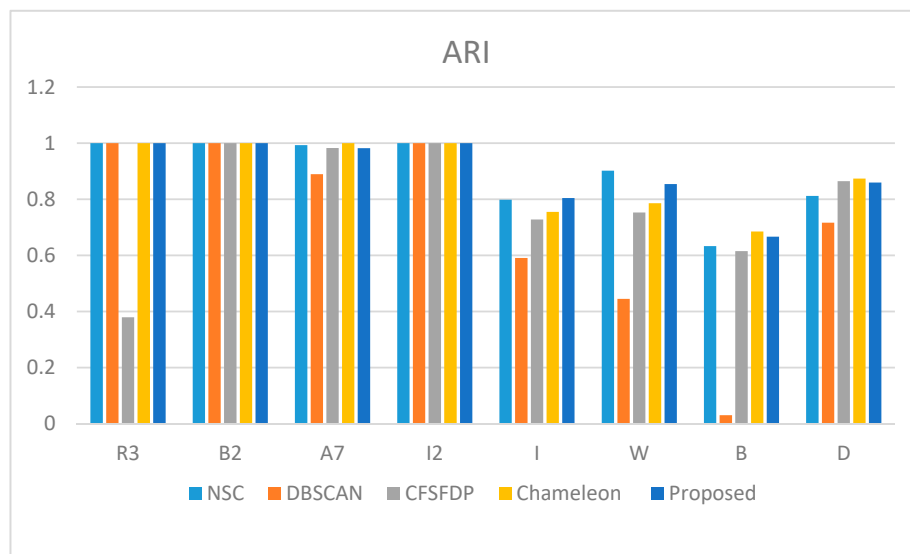
	ARI		NMI	
	Average \pm STD	L-D-W	Average \pm STD	L-D-W
EAC+SL	66.87 \pm 3.39	0-2-6	64.43 \pm 2.43	1-0-7
EAC+AL	68.19 \pm 2.65	0-2-6	60.97 \pm 2.75	0-2-6
WCT+SL	60.11 \pm 3.23	0-1-7	55.78 \pm 2.69	1-1-6
WCT+AL	67.58 \pm 3.13	0-1-7	60.72 \pm 2.32	0-1-7
WTQ+SL	65.87 \pm 2.89	0-1-7	62.28 \pm 3.03	0-1-7
WTQ+AL	67.88 \pm 2.58	1-1-6	61.02 \pm 2.94	1-2-5
CSM+AL	58.67 \pm 3.68	1-0-7	48.90 \pm 2.72	1-0-7
CSM+SL	68.21 \pm 2.48	1-0-7	60.99 \pm 2.71	1-1-6
CSPA	59.97 \pm 2.55	1-0-7	54.55 \pm 2.43	0-0-8
HGPA	24.24 \pm 2.36	0-0-8	20.29 \pm 2.26	0-0-8
MCLA	66.21 \pm 3.29	0-2-6	58.26 \pm 2.54	0-1-7
SUV	48.82 \pm 3.08	1-0-7	40.93 \pm 2.31	1-1-6
SWV	52.76 \pm 2.83	1-0-7	47.43 \pm 3.25	1-1-6
EM	57.42 \pm 2.89	0-0-8	52.48 \pm 2.97	0-0-8
IVC	58.48 \pm 3.02	0-0-8	53.52 \pm 2.37	0-0-8
PC+EEAC+SL	85.01 \pm 3.31	1-1-6	80.82 \pm 2.40	1-2-5
PC+EEAC+AL	89.51 \pm 2.07	1-0-7	83.29 \pm 1.42	1-2-5
PC+CSPA	84.63 \pm 2.75	0-1-7	78.02 \pm 2.76	0-0-8
PC+HGPA	55.77 \pm 2.98	0-0-8	55.26 \pm 3.52	0-0-8
PC+MCLA	91.42 \pm 2.09	1-0-7	83.03 \pm 2.25	1-2-5
PC+EM	86.37 \pm 2.44	1-1-6	79.80 \pm 2.35	0-0-8
Proposed	93.86 \pm 1.37		89.58 \pm 2.15	

4.5.2. Comparison with Strong Clustering Algorithms

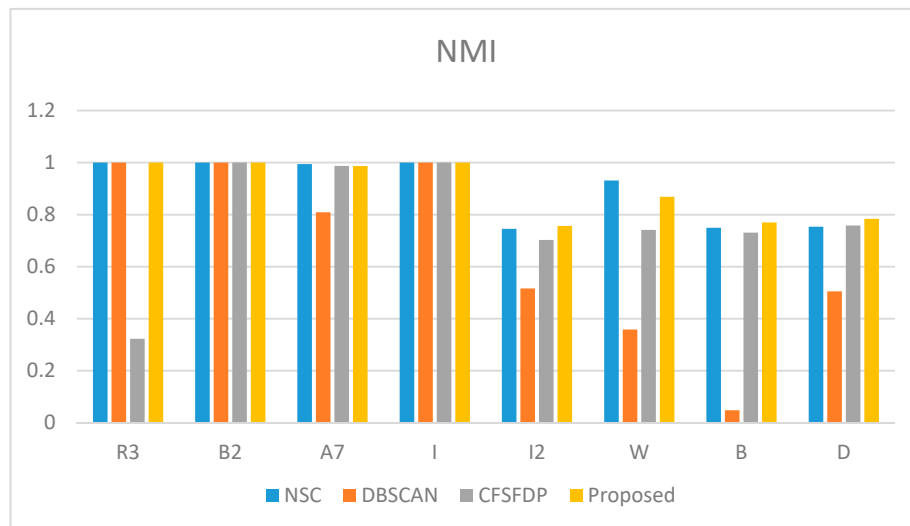
The results of the proposed ensemble clustering algorithm compared with four “strong” clustering algorithms on the different benchmark datasets are depicted in Figure 6. In Figure 6, the last two columns indicate the mean and standard deviation of the clustering validity of each algorithm on the different datasets. We observe that the clustering validity of the proposed ensemble clustering algorithm is superior or close to the best results of the other four algorithms. According to the results of these experiments, the proposed algorithm can compete with “strong” clustering algorithms; therefore, it approves that “a number of weak clusters equal to a strong clustering”.

Parameter analysis: How to set the parameter γ is considered to be an important issue for the proposed ensemble clustering algorithm. The selection of this parameter regulates the number of basic clusters produced through each base clustering result. The number of base clusters generated by the proposed ensemble clustering algorithm increases exponentially with decreasing γ . However, the accuracy of clustering task does not increase with decreasing γ , therefore the value of γ must increase to a certain extent. According to the empirical results, if the number of basic clusters in a clustering result is very large or small, it can be considered to be a bad clustering. Therefore, the value γ is selected in such a way that the number of clusters in each clustering result is less than $\sqrt{|D_{:1}|}$ and more than $\frac{\sqrt{|D_{:1}|}}{2}$.

Time Analysis: In the end, the efficiency of the proposed ensemble clustering algorithm is evaluated on the KDD-CUP99 dataset. The γ is set to 0.14. The proposed ensemble clustering algorithm is implemented on Matlab 2018. The runtime of the proposed ensemble clustering algorithm in terms of different numbers of objects are shown in Table 5. We observe that the number of the basic clusters in the base clustering results increases with increasing the number of objects.



(a)



(b)

Figure 6. Experimental results of different strong clustering methods comparing with the results of the proposed ensemble clustering algorithm on different datasets in terms of (a) ARI, and (b) NMI.

Table 5. The computational cost of the proposed ensemble clustering algorithm in terms of the number of data points.

$ X $	$\sum_{i=1}^B c_i$	Time (in Sec.)
10K	91	11.23
20K	213	51.29
30K	225	80.11
40K	232	114.06
50K	233	138.91
60K	242	178.71
70K	245	197.62
80K	353	331.02
90K	461	516.96
100K	472	576.58

4.5.3. Final Decisive Experimental Results

In this subsection, a set of six real-world datasets has been employed to evaluate the efficacy of the proposed method in comparison with some recently published papers. Three of these six datasets are the same Wine, Iris, and Breast datasets presented in Table 2. To make our final conclusion fairer and more general, three additional datasets, whose details are presented in Table 6, are used as benchmark in this subsection.

Table 6. Description of the benchmark datasets; the number of data objects ($|D_{:1}|$), the number of attributes ($|D_{1:}|$), the number of clusters (C).

Source	Dataset	$ D_{:1} $	$ D_{1:} $	C
UCI dataset	Glass (Gl)	214	9	6
UCI dataset	Galaxy (Ga)	323	4	7
UCI dataset	Yeast (Y)	1484	8	10

Based on the NMI criterion, the comparison of the performances of different ensemble clustering algorithms on the real-world benchmark datasets is shown in Table 7. According to the experimental results, the proposed ensemble clustering algorithm can detect different clusters in an effective way and increase the performance of the state-of-the-art ensemble clustering algorithms.

Table 7. The NMI between the consensus partitioning results, produced by different ensemble clustering methods, and the ground truth labels validated by paired t-test [66] with 95% level of confidence.

Evaluation Measure	Dataset Number						T-Test Results Wins Against-Draws with-Loses to
	B	I	Gl	Ga	Y	W	
NMI [9] + ItoU	95.73–	82.89–	41.38–	21.71–	34.45–	91.83±	5-1-0
MAX [44] + ItoU	96.39±	83.21–	42.63–	20.57–	33.89–	91.29–	5-1-0
APMM [45] + ItoU	95.16–	82.10–	41.98–	24.01–	34.12–	91.78±	5-1-0
ENMI [46] + ItoU	96.51±	84.66–	42.65–	24.84–	35.58–	92.27+	4-1-1
Proposed	97.28	86.05	44.79	29.44	38.20	92.13	

5. Conclusions and Future Works

The kmedoids clustering algorithm, as a fundamental clustering algorithm, has been widely considered to be a low-computational one. However, it is also considered to be a weak clustering method, because its performances are affected by many factors, such as unsuitable selection of the initial cluster centers and dissimilar distribution of data. This study is carried out aiming to propose a new ensemble clustering algorithm using multiple kmedoids clustering algorithms. The proposed ensemble clustering method has the advantages of the kmedoids clustering algorithm, including its high speed. Meanwhile, it does not have its major weaknesses, i.e., the inability to detect non-spherical and non-uniform clusters. Indeed, the new ensemble clustering algorithm can improve stability and quality of kmedoids clustering algorithm and prove “aggregating several weak clustering results is better than or equal to a strong clustering result.” This study tries to solve all of the ensemble clustering problems by defining valid local clusters. In fact, this study calls data around a cluster center in the clustering of kmedoids as valid local data clusters. In order to generate a diverse clustering result, a strategy of sequential application of a weak clustering algorithm (i.e., using kmedoids clustering algorithm as base clustering algorithm) is used on non-appeared data in previously valid local clusters. Empirical analysis compares the proposed ensemble clustering algorithm with several existing ensemble clustering algorithms and three strong fundamental clustering algorithms running on a set

of artificial and real-world benchmark datasets. According to the empirical results, the performance of the proposed ensemble clustering algorithm is much more effective than the state-of-the-art ensemble clustering methods. In addition, we examined the efficiency of the proposed ensemble clustering algorithm, which is even suitable for dealing with large-scale datasets. This method works because it concentrates on finding local structures in valid small-cluster and then merging them. It also benefits from instability of kmedoids clustering algorithm to make a diverse ensemble. The main limitation of the paper is its usage of base kmedoids clustering algorithm. It seems that other base kmedoids clustering algorithms such as fuzzy cmeans clustering algorithm can be a better alternative and it will be discussed in future work.

Author Contributions: H.N. and H.P. designed the study. N.K., H.N. and H.P. wrote the paper. M.R.M. and H.P. revised the manuscript. H.N. and H.P. provided the data. M.R.M., H.N. and H.P. carried out tool implementation and all of the analyses. H.N. and H.P. generated all figures and tables. M.R.M., H.N. and H.P. performed the statistical analyses. H.A.-R. and A.B. provided some important suggestions on the paper idea during preparation but were not involved in the analyses and writing of the paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Han, J.; Kamber, M. *Data Mining: Concepts and Techniques*; Morgan Kaufmann: San Francisco, CA, USA, 2001.
2. Shojafar, M.; Canali, C.; Lancellotti, R.; Abawajy, J.H. Adaptive Computing-Plus-Communication Optimization Framework for Multimedia Processing in Cloud Systems. *IEEE Trans. Cloud Comput. (TCC)* **2016**, *99*, 1–14. [[CrossRef](#)]
3. Shamshirband, S.; Amini, A.; Anuar, N.B.; Kiah, M.L.M.; Teh, Y.W.; Furnell, S. D-FICCA: A density-based fuzzy imperialist competitive clustering algorithm for intrusion detection in wireless sensor networks. *Measurement* **2014**, *55*, 212–226. [[CrossRef](#)]
4. Agaian, S.; Madhukar, M.; Chronopoulos, A.T. A new acute leukaemia-automated classification system. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2016**, *6*, 303–314. [[CrossRef](#)]
5. Khoshnevisan, B.; Rafiee, S.; Omid, M.; Mousazadeh, H.; Shamshirband, S.; Hamid, S.H.A. Developing a fuzzy clustering model for better energy use in farm management systems. *Renew. Sustain. Energy Rev.* **2015**, *48*, 27–34. [[CrossRef](#)]
6. Jain, A.K.; Dubes, R.C. *Algorithms for Clustering Data*; Prentice Hall: Englewood Cliffs, NJ, USA, 1988.
7. Jain, A.K. Data clustering: 50 years beyond K-means. *Pattern Recognit. Lett.* **2010**, *31*, 651–666. [[CrossRef](#)]
8. Zhou, Z. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA, 2012.
9. Fred, A.; Jain, A. Combining multiple clusterings using evidence accumulation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 835–850. [[CrossRef](#)]
10. Kuncheva, L.; Vetrov, D. Evaluation of stability of k-means cluster ensembles with respect to random initialization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2006**, *28*, 1798–1808. [[CrossRef](#)]
11. Zhang, X.; Jiao, L.; Liu, F.; Bo, L.; Gong, M. Spectral clustering ensemble applied to SAR image segmentation. *IEEE Trans. Geosci. Remote Sens.* **2008**, *46*, 2126–2136. [[CrossRef](#)]
12. Gionis, A.; Mannila, H.; Tsaparas, P. Clustering aggregation. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 1–30. [[CrossRef](#)]
13. Law, M.; Topchy, A.; Jain, A. Multi-objective data clustering. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 27 June–2 July 2004.
14. Yu, Z.; Chen, H.; You, J.; Han, G.; Li, L. Hybrid fuzzy cluster ensemble framework for tumor clustering from bio-molecular data. *IEEE/ACM Trans. Comput. Biol. Bioinform.* **2013**, *10*, 657–670. [[CrossRef](#)]
15. Fischer, B.; Buhmann, J. Bagging for path-based clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* **2003**, *25*, 1411–1415. [[CrossRef](#)]
16. Topchy, A.; Minaei-Bidgoli, B.; Jain, A. Adaptive clustering ensembles. In Proceedings of the 17th International Conference on Pattern Recognition, Cambridge, UK, 26 August 2004.
17. Zhou, Z.; Tang, W. Clusterer ensemble. *Knowl.-Based Syst.* **2006**, *19*, 77–83. [[CrossRef](#)]

18. Hong, Y.; Kwong, S.; Wang, H.; Ren, Q. Resampling-based selective clustering ensembles. *Pattern Recognit. Lett.* **2009**, *30*, 298–305. [\[CrossRef\]](#)
19. Fern, X.; Brodley, C. Random projection for high dimensional data clustering: A cluster ensemble approach. In Proceedings of the International Conference on Machine Learning, Washington, DC, USA, 21–24 August 2003.
20. Zhou, P.; Du, L.; Shi, L.; Wang, H.; Shi, L.; Shen, Y.D. Learning a robust consensus matrix for clustering ensemble via Kullback-Leibler divergence minimization. In *25th International Joint Conference on Artificial Intelligence*; AAAI Publications: Palm Springs, CA, USA, 2015.
21. Yu, Z.; Li, L.; Liu, J.; Zhang, J.; Han, G. Adaptive noise immune cluster ensemble using affinity propagation. *IEEE Trans. Knowl. Data Eng.* **2015**, *27*, 3176–3189. [\[CrossRef\]](#)
22. Gullo, F.; Domeniconi, C. Metacenter-based projective clustering ensembles. *Mach. Learn.* **2013**, *98*, 1–36. [\[CrossRef\]](#)
23. Yang, Y.; Jiang, J. Hybrid Sampling-Based Clustering Ensemble with Global and Local Constitutions. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *27*, 952–965. [\[CrossRef\]](#)
24. Minaei-Bidgoli, B.; Parvin, H.; Alinejad-Rokny, H.; Alizadeh, H.; Punch, W.F. Effects of resampling method and adaptation on clustering ensemble efficacy. *Artif. Intell. Rev.* **2014**, *41*, 27–48. [\[CrossRef\]](#)
25. Fred, A.; Jain, A.K. Data clustering using evidence accumulation. In Proceedings of the 16th International Conference on Pattern Recognition, Quebec City, QC, Canada, 11–15 August 2002; pp. 276–280.
26. Yang, Y.; Chen, K. Temporal data clustering via weighted clustering ensemble with different representations. *IEEE Trans. Knowl. Data Eng.* **2011**, *23*, 307–320. [\[CrossRef\]](#)
27. Iam-On, N.; Boongoen, T.; Garrett, S.; Price, C. A link-based approach to the cluster ensemble problem. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 2396–2409. [\[CrossRef\]](#)
28. Iam-On, N.; Boongoen, T.; Garrett, S.; Price, C. A link-based cluster ensemble approach for categorical data clustering. *IEEE Trans. Knowl. Data Eng.* **2012**, *24*, 413–425. [\[CrossRef\]](#)
29. Strehl, A.; Ghosh, J. Cluster ensembles: A knowledge reuse framework for combining multiple partitions. *J. Mach. Learn. Res.* **2002**, *3*, 583–617.
30. Fern, X.; Brodley, C. Solving cluster ensemble problems by bipartite graph partitioning. In Proceedings of the 21st International Conference on Machine Learning, Banff, AB, Canada, 4–8 July 2004.
31. Huang, D.; Lai, J.; Wang, C.D. Ensemble clustering using factor graph. *Pattern Recognit.* **2016**, *50*, 131–142. [\[CrossRef\]](#)
32. Selim, M.; Ertunc, E. Combining multiple clusterings using similarity graph. *Pattern Recognit.* **2011**, *44*, 694–703.
33. Bouslim, C.; Ostendorf, M. Combining multiple clustering systems. In *European Conference on Principles and Practice of Knowledge Discovery in Databases*; Springer: Berlin/Hidelberg, Germany, 2004.
34. Hore, P.; Hall, L.O.; Goldgo, B. A scalable framework for cluster ensembles. *Pattern Recognit.* **2009**, *42*, 676–688. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Long, B.; Zhang, Z.; Yu, P.S. Combining multiple clusterings by soft correspondence. In Proceedings of the 4th IEEE International Conference on Data Mining, Houston, TX, USA, 27–30 November 2005.
36. Cristofor, D.; Simovici, D. Finding median partitions using information theoretical based genetic algorithms. *J. Univers. Comput. Sci.* **2002**, *8*, 153–172.
37. Topchy, A.; Jain, A.; Punch, W. Clustering ensembles: Models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2005**, *27*, 1866–1881. [\[CrossRef\]](#)
38. Wang, H.; Shan, H.; Banerjee, A. Bayesian cluster ensembles. *Stat. Anal. Data Min.* **2011**, *4*, 54–70. [\[CrossRef\]](#)
39. He, Z.; Xu, X.; Deng, S. A cluster ensemble method for clustering categorical data. *Inf. Fusion* **2005**, *6*, 143–151. [\[CrossRef\]](#)
40. Nguyen, N.; Caruana, R. Consensus Clusterings. In Proceedings of the Seventh IEEE International Conference on Data Mining, Omaha, NE, USA, 28–31 October 2007; pp. 607–612.
41. Huang, Z. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data Min. Knowl. Discov.* **1998**, *2*, 283–304. [\[CrossRef\]](#)
42. Nazari, A.; Dehghan, A.; Nejatian, S.; Rezaie, V.; Parvin, H. A Comprehensive Study of Clustering Ensemble Weighting Based on Cluster Quality and Diversity. *Pattern Anal. Appl.* **2019**, *22*, 133–145. [\[CrossRef\]](#)
43. Bagherinia, A.; Minaei-Bidgoli, B.; Hossinzadeh, M.; Parvin, H. Elite fuzzy clustering ensemble based on clustering diversity and quality measures. *Appl. Intell.* **2019**, *49*, 1724–1747. [\[CrossRef\]](#)

44. Alizadeh, H.; Minaeibidgoli, B.; Parvin, H. Cluster ensemble selection based on a new cluster stability measure. *Intell. Data Anal.* **2014**, *18*, 389–408. [CrossRef]
45. Alizadeh, H.; Minaei-Bidgoli, B.; Parvin, H. A New Criterion for Clusters Validation. In *Artificial Intelligence Applications and Innovations (AIAI 2011)*; IFIP, Part I; Springer: Heidelberg, Germany, 2011; pp. 240–246.
46. Abbasi, S.; Nejatian, S.; Parvin, H.; Rezaie, V.; Bagherifard, K. Clustering ensemble selection considering quality and diversity. *Artif. Intell. Rev.* **2019**, *52*, 1311–1340. [CrossRef]
47. Rashidi, F.; Nejatian, S.; Parvin, H.; Rezaie, V. Diversity Based Cluster Weighting in Cluster Ensemble: An Information Theory Approach. *Artif. Intell. Rev.* **2019**, *52*, 1341–1368. [CrossRef]
48. Zhou, S.; Xu, Z.; Liu, F. Method for Determining the Optimal Number of Clusters Based on Agglomerative Hierarchical Clustering. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 3007–3017. [CrossRef]
49. Karypis, G.; Han, E.-H.S.; Kumar, V. Chameleon: A hierarchical clustering algorithm using dynamic modeling. *IEEE Comput.* **1999**, *32*, 68–75. [CrossRef]
50. Ji, Y.; Xia, L. Improved Chameleon: A Lightweight Method for Identity Verification in Near Field Communication. In *Proceedings of the 2016 International Symposium on Computer, Consumer and Control (IS3C)*, Xi'an, China, 4–6 July 2016; pp. 387–392.
51. MacQueen, J.B. Some methods for classification and analysis of multivariate observations. In *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*; University of California Press: Berkeley, CA, USA, 1967; Volume 1, pp. 281–297.
52. Kaufman, L.; Rousseeuw, P.J. *Clustering by Means of Medoids*, in *Statistical Data Analysis Based on the L_1 —Norm and Related Methods*; Dodge, Y., Ed.; North-Holland: Amsterdam, The Netherlands, 1987; pp. 405–416.
53. Bezdek, J.C.; Pal, N.R. Some new indexes of cluster validity. *IEEE Trans. Syst. Man Cybern. Part B* **1998**, *28*, 301–315. [CrossRef]
54. Pal, N.R.; Bezdek, J.C. On cluster validity for the fuzzy c-means model. *IEEE Trans. Fuzzy Syst.* **1995**, *3*, 370–379. [CrossRef]
55. Guha, S.; Rastogi, R.; Shim, K. Cure: An efficient clustering algorithm for large databases. In *Proceedings of the Conference on Management of Data (ACM SIGMOD)*, Seattle, WA, USA, 1–4 June 1998; pp. 73–84.
56. Sneath, P.H.A.; Sokal, R.R. *Numerical Taxonomy*; Freeman: San Francisco, CA, USA; London, UK, 1973.
57. King, B. Step-wise clustering procedures. *J. Am. Stat. Assoc.* **1967**, *69*, 86–101. [CrossRef]
58. Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888C905.
59. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems*; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2002; Volume 14.
60. UCI Machine Learning Repository. 2016. Available online: <http://www.ics.uci.edu/ml/ML-Repository.html> (accessed on 19 February 2016).
61. Press, W.; Teukolsky, S.A.; Vetterling, W.T.; Flannery, B.P. Conditional Entropy and Mutual Information. In *Numerical Recipes: The Art of Scientific Computing*, 3rd ed.; Cambridge University Press: New York, NY, USA, 2007.
62. Ester, M.; Kriegel, H.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD-96: Proceedings: Second International Conference on Knowledge Discovery and Data Mining*; Simoudis, E., Han, J., Fayyad, U.M., Eds.; AAAI Press: Menlo Park, CA, USA, 1996; pp. 226–231.
63. Rodriguez, A.; Laio, A. Clustering by fast search and find of density peaks. *Science* **2014**, *344*, 1492–1496. [CrossRef] [PubMed]
64. Parvin, H.; Minaei-Bidgoli, B. A clustering ensemble framework based on elite selection of weighted clusters. *Adv. Data Anal. Classif.* **2013**, *7*, 181–208. [CrossRef]
65. Parvin, H.; Minaei-Bidgoli, B. A clustering ensemble framework based on selection of fuzzy weighted clusters in a locally adaptive clustering algorithm. *Pattern Anal. Appl.* **2015**, *18*, 87–112. [CrossRef]
66. Dietterich, T.G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* **1998**, *7*, 1895–1924. [CrossRef]

