Clustering Ensemble Method

Tahani Muqbil Alqurashi

A thesis submitted in fulfilment
of the requirements for the degree of
Doctor of Philosophy

School of Computing Sciences
University of East Anglia
January, 2017

Abstract

Clustering is an unsupervised learning paradigm that partitions a given dataset into clusters so that objects in the same cluster are more similar to each other than to the objects in the other clusters. However, when clustering algorithms are used individually, their results are often inconsistent and unreliable. This research applies the philosophy of Ensemble learning that combines multiple partitions using a consensus function in order to address these issues to improve a clustering performance.

A clustering ensemble framework is presented consisting of three phases: Ensemble Member Generation, Consensus and Evaluation. This research focuses on two points: the consensus function and ensemble diversity. For the first, we proposed three new consensus functions: the Object-Neighbourhood Clustering Ensemble (ONCE), the Dual-Similarity Clustering Ensemble (DSCE), and the Adaptive Clustering Ensemble (ACE). ONCE takes into account the neighbourhood relationship between object pairs in the similarity matrix, while DSCE and ACE are based on two similarity measures: cluster similarity and membership similarity.

The proposed ensemble methods were tested on benchmark real-world and artificial datasets. The results demonstrated that ONCE outperforms the other similar methods, and is more consistent and reliable than *k-means*. Furthermore, DSCE and ACE were compared to the ONCE, CO, MCLA and DICLENS clustering ensemble methods. The results demonstrated that on average ACE outperforms the state-of-the-art clustering ensemble methods, which are CO, MCLA and DICLENS.

On diversity, we experimentally investigated all the existing measures for determining their relationship with the ensemble quality. The results indicate that none

of them are capable of discovering a clear relationship and the reasons for this are: (1) they all are inappropriately defined to measure the useful difference between the members, and (2) none of them have been used directly by any consensus function. Therefore, we point out that these two issues need to be addressed in future research.

Publications

Some of the results presented in this thesis has been reported in the following publications:

Journal Papers

- **Tahani Alqurashi and Wenjia Wang**. A novel adaptive clustering ensemble method. *International Journal of Machine Learning and Cybernetics*, Springer, 2016. Note: Accepted subject to minor revision.

Conference Papers

- **Tahani Alqurashi and Wenjia Wang**. Object-neighbourhood clustering ensemble method. *In Proceedings of the Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 142 – 149. Springer, 2014.
- **Tahani Alqurashi and Wenjia Wang**. A new consensus function based on dual-similarity measurements for clustering ensemble. *In Proceedings of the International Conference on Data Science and Advanced Analytics (DSAA)*, pages 149 – 155. IEEE/ACM, 2015.

Other Publication

- **Tahani Alqurashi and Wenjia Wang**. A Graph based Methodology for Web Structure Mining-with a Case Study on the Webs of UK Universities. *In Proceedings of the 4th International Conference on Web Intelligence, Mining and Semantics (WIMS)*, page 4. ACM, 2014.

Acknowlegments

First and foremost, I would like to thank Almighty Lord, who has made everything possible. A special thanks goes to Dr Wenjia Wang, who was not only my supervisor but also my mentor. He has been supportive since my Masters degree and throughout the long road to the end of my thesis. He gave me confidence in my work and was always present for guidance and help whenever I needed it. Thank you Wenjia, for giving me the opportunity to work with you. And big thanks go to all the staff in the Computing Sciences School — it was a great honour to work with you. I also would like to thank the reviewers of my papers for their critical comments and suggestions.

This thesis is also the result of many people who supported me emotionally: Mum and Dad, thanks for your support, which helped me to overcome my fears and thank you for encouraging me to achieve my dreams. Thanks for being a role model to me. Yaser, you are the source of endless love and kindness, your love helped me overcome difficult times. Thank you for being my best friend, and a supportive husband. Sultan and Yasmin, you are the best kids I could ask for, thank you for your patience and I know you both suffered so much during my study for not having the time you needed. My sisters and brothers, I wish I were with you during the last four years. I thank you for believing in me and I hope I made you proud of your sister.

Lastly, I would like to show my deepest gratitude to my sponsor, Umm Al-Qura University and the Ministry of Education in Saudi Arabia, for the full scholarship that has been given to me.

Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Meaning**
X	Dataset
x	Object
c	Cluster
P	Parttition/Member
Γ	Set of ensemble members
m	Number of ensemble members
n	Number of object in X
k	Number of clusters in X
Φ	Clustering ensemble
CF	Consensus function
Dis	Distance function
S	Similarity measure
CO	Co-assoication matrix
ONCE	Object-Neighbourhood Clustering Ensemble
$\mathcal{E}$-ONCE	$\mathcal{E}$-Object Neighbourhood Clustering Ensemble
DSCE	Dual-Similarity Clustering Ensemble
ACE	Adaptive Clustering Ensemble
MCLA	Meta-Clustering Algorithm
DICLENS	Divisive Clustering Ensemble
X_f	Friedman test
F	Iman-Dave port test/ ANOVA test
CD	Critical difference
P^*	Final clustering result of the ensemble
P^t	Ground-truth partition of the dataset
NMI	Normailsed Mutual Information
ARI	Adjust Rand Index

Av	Average linkage method
Si	Single linkage method
Cm	Complete linkage method
B	The average neighbourhood similarity matrix
W	The overall similarity matrix
Z	The set of object pair common neighbourhood
S_c	Cluster similarity
S_x	Membership similarity
α_1	Merging threshold
α_2	Certainty threshold
θ_1	Membership matrix of the newly formed clusters
λ	Number of clusters in θ_1
$\overleftarrow{C}$	The set of all the newly formed clusters in θ_1
Var	Variance
p_{c_g}	Cluster certainty
DV	Diversity Measure
p	Pairwise Diversity Measure
np	Non-pairwise Diversity Measure
EOD	Ensemble Output Dependent diversity measure
EOI	Ensemble Output Independent diversity measure
cc	Correlation coefficient
H	Highest quality
L	Lowest quality
DV^+	The positive effect of diversity
DV^-	The negative effect of diversity
$Q(\Gamma)$	Average member quality
Std	Standard deviation

Chapter 1

Introduction

1.1 Background

In the context of machine learning, an ensemble is generally defined as "a machine learning system that is constructed with a set of individual models working in parallel, whose outputs are combined with a decision fusion strategy to produce a single answer for a given problem" [106].

The ensemble method was first introduced and well-studied in the supervised learning field. Due to its successful application in classification tasks over the past decades, researchers have attempted to apply the same paradigm to the unsupervised learning field, particularly to clustering problems. However, this may be challenging for the following two obvious reasons. Firstly, in unsupervised learning, as there is normally no prior knowledge about the underlying structure or about any particular properties that we want to find or what we consider as good solutions about the data [55, 95], different clustering algorithms often produce different clustering results for the same data. Secondly, according to the "no free lunch" theorem [108], there is no single clustering algorithm that performs consistently well in finding the correct underlying structure for different data, and there are no clear guidelines in the literature for choosing individual clustering algorithms for a given problem.

Conceptually speaking, a clustering ensemble, which is also referred to as consensus ensemble or clustering aggregation, can be simply defined in the same manner as for classification. In other words, it is the process of combining multiple clustering models (partitions) into a single consolidated partition [94]. In principle, an effective clustering ensemble should be able to produce better results than that of the individual clustering algorithms in terms of quality and consistency. From the clustering point of view, the quality is measured either using external information (class label) or internal information. If the external information is available the quality is defined by some degree of similarity between the clustering results and the known labels of the data (class label). If not, the quality is defined as how well the clustering result fits the data using only internal information [95]. The consistency is defined as the ability that the clustering ensemble method has to produce similar performances on a multiple number of test datasets [32].

However, the transmission from supervised learning to unsupervised learning is not as straightforward as this conceptual definition because there are some unique and challenging issues when building an ensemble for clustering. Of these issues, the key and tricky one is how to combine the clusters that are generated by the individual clustering models (members) in an ensemble, as this cannot be done through simple voting or averaging as in classification. Instead, it requires more complicated aggregating strategies and mechanisms. Therefore, developing an effective aggregation strategy as well as efficient is essential for building a successful clustering ensemble.

1.2 Research Motivation

This thesis focuses on two central points, which are the consensus function and the diversity of the clustering ensemble. This section explains the motivation behind them.

1.2.1 Consensus Function

A consensus function is the main component of a clustering ensemble method. It combines a number of members to produce single improved clustering results, compared to the individual member in the ensemble. In the past decade, a number of researchers have studied clustering ensemble methods [94, 26, 97, 27, 98].

One simple, popular consensus approach focuses on combining members by mapping them onto a new representation, that contains similarity information. This similarity information can be estimated from members at object level or at cluster level. Generally, solving the problem of clustering the data through similarity information is not a new concept; it is a widely used concept in clustering analysis, and it is in fact the core of some of the most popular cluster algorithms such as *k-means* and the hierarchical clustering algorithm. It is simple and easy to understand and implement.

In the similarity-based consensus function approach, which calculates the object pairwise similarity matrix from members, the Co-association matrix (CO) [32] is the most popular method in this approach. The idea of CO is to avoid the label correspondence problem in which the clustering result is obtained through a voting process among the objects. It assumes that similar objects are very likely clustered together by some clustering algorithm, so any objects that co-occur frequently in the same cluster should be regarded as being very similar. Each entry in CO matrix counts the number of times that a given pair of objects is placed in the same cluster among ensemble members.

However, there is a common and tricky issue that appears when roughly half of the members place some object pairs in the same cluster but the other half place them in a different cluster. In this case, we have uncertain agreement between members on how to cluster these pairs and we call them uncertain pairs of objects, and they cause problems in generating reliable consensus clustering results [111, 81]. Recently, researchers such as Wang et al. [107] and Vega-Pons et al. [103] enhanced the CO matrix to extract more information from the members. We believe that when we build a clustering ensemble, there may be some other useful information in the generated members that could be extracted, rather than relying solely on the pairwise relationship between objects. Consequently, we were motivated to design a consensus function based on object pairwise similarity that considers more information than the pair itself to overcome the problem of the uncertain agreement to some extent.

Moreover, one obvious drawback in most similarity-based consensus functions is that they require an ordinary clustering algorithm to be applied over the similarity matrix. This leads to two adverse effects. Firstly, it is difficult to decide which one is to be used, as most of them require a parameter, so there is the question of which is the best value. Therefore, this approach unintentionally suffers from the same difficulties as the single clustering algorithm which the clustering ensemble method aims to solve. Secondly, it takes time to do a further clustering, and this makes the whole clustering ensemble inefficient.

1.2.2 Clustering Ensemble Diversity

Furthermore, it is widely believed that having diverse members in an ensemble is essential for its success. Although many researchers have investigated the effect of diversity on the quality of clustering ensembles, they have not yet arrived at any agreement on the relationship between diversity and ensemble quality. Some researchers have concluded that, through high levels of diversity among members, high levels of ensemble quality can be achieved [25, 20, 51]. By contrast, other researchers suggest that median diversity among members is better in terms of improving the ensemble's quality [39].

Nevertheless, most of these diversity studies either investigated the effect of diversity on one specific consensus function or their own proposed consensus function. Therefore, more studies need to be conducted in order to investigate diversity definitions in their relation with multiple consensus functions.

1.3 Research Questions

The main research question that we would like to answer in this thesis is: *How can we develop an effective clustering ensemble that can improve the quality and consistency of the clustering result ?* In order to answer this question, we believe this research has to consider two essential issues: consensus function and diversity by addressing the following associated questions.

1. How can we design a consensus function that addresses the problem of uncertain pairs of objects?
2. Is there any other information in the ensemble members that we can use to design a new effective consensus function? If so, what is it and how can we design consensus functions?
3. How can we design a similarity-based consensus approach that does not require an additional step of using an ordinary clustering algorithm to produce the final clustering result, which can be implemented in the clustering ensemble framework to generate a reliable and accurate clustering result?
4. How are the existing diversity measures defined in the context of the clustering ensemble?
5. Does the diversity influence ensemble performance?

Questions 1 to 3 are our key questions regarding to the consensus function issue, while, questions 4 and 5 are our key questions regarding to the diversity issue.

1.4 Thesis Organisation

The reminder of this thesis is organised as follows:

Chapter 2: Literature Review This chapter provides a review of clustering analysis, which includes the different clustering techniques and clustering validation index. The clustering ensemble is then introduced in more detail. Work relating to the consensus function is discussed. Finally, this chapter details some of the current clustering ensemble applications.

Chapter 3: Research Methodology In this chapter, our research design is explained, including the adapted clustering ensemble framework, and the strategy used to test the proposed consensus functions. We also describe the implementation and tools used in our research.

Chapter 4: Object-Neighbourhood Clustering Ensemble (ONCE) In this chapter, we present two new consensus functions ONCE and $\mathcal{E}$-ONCE, and discuss the results of testing the effectiveness of ONCE and $\mathcal{E}$-ONCE. We also compare the performance of the proposed methods with a number of clustering ensemble methods. This chapter presents an answer to research question 1.

Chapter 5: Adaptive Clustering Ensemble (ACE) In this chapter, we describe two new consensus functions based on two novel similarity measurements, which are Dual-Similarity Clustering Ensemble (DSCE) and Adaptive Clustering Ensemble (ACE). We conduct some experimental studies to test the effectiveness of DSCE and ACE and compare them to other clustering ensemble methods. We also discuss and analyse the results obtained. This chapter presents answers to research questions 2 and 3 of this thesis.

Chapter 6: The Diversity of the Clustering Ensemble In this chapter, we investigate diversity measurements by looking at their influence on ensemble performance. We analyse and discuss the experimental results obtained. Moreover, we design two experiments to investigate two issues raised from our experimental study, and discuss and analyse the results obtained. This chapter presents answers to research questions 4 and 5.

Chapter 7: Conclusions and Further Work In this chapter, we draw our overall conclusions on the two central points of this research, and we also suggest further work to be done in the future.

Chapter 2

Literature Review

This chapter reviews the literature related to this research, including the background of clustering analysis in Section 2.1, and clustering ensembles in Section 2.2, along with details on their process. In Section 2.4, clustering ensemble applications are reviewed, and finally Section 2.5 includes a summary of this chapter.

2.1 Clustering Methods

Clustering is a task of assigning each object (sometimes called a pattern, observation or data point) in a dataset to a group or cluster in order to identify natural groups within that dataset. Thus, objects in the same cluster are more similar to each other than to the objects in the other clusters [54].

In machine learning, clustering is used to search for groups that reflect hidden structured patterns. This is widely known as unsupervised learning, in contrast to supervised learning, which requires the dataset to be labelled in advance for training purposes. The supervised learning problem is related to predicting categorical and numerical data (i.e., the data classification problem corresponds to categorical data, and the regression problem corresponds to numerical data). However, all of the available data in data clustering problems are unlabelled, so the task is to group the objects based only on the natural relationships among them and the underlying population model [55].

The main problem in clustering is how to define any similarity/dissimilarity between the objects. Generally, similarity between two objects measures the degree to which they are alike on a numerical scale, while the dissimilarity measures the degree to which they are different

A common and important measure is the distance (Dis) between two objects. Several similarity and distance measures exist in the literature; each of them is defined based on the type of measured feature, and more details of these measures can be found in [109]. However, the best-known distance measure is the Euclidean distance. Suppose we have the dataset $X = \{x_1, x_2, \cdots, x_n\} \in \Re^d$, where each object x_i is a set of d features (sometimes called attributes, dimensions or variables). The Euclidean distance (E) can be calculated between two objects x_i and x_j as follows:

$$E(x_i, x_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^2 \right)^{1/2} \tag{2.1}$$

In fact, the Euclidean distance is a special case, $p = 2$, of the Minkowski distance (M), which is defined as follows:

$$M(x_i, x_j) = \left(\sum_{l=1}^{d} |x_{il} - x_{jl}|^p \right)^{1/p} \tag{2.2}$$

Many techniques have been proposed for cluster analysis due to the fact that clustering analysis has been used in a wide variety of applications. However, we may distinguish three main types of clustering techniques: hierarchical, partitional and fuzzy. The main difference between them is that hierarchical and partitional clustering are classified as hard clustering, where each object in the dataset belongs to only one cluster, whereas in fuzzy clustering, which is sometimes called soft clustering, some objects in the dataset can belong to more than one cluster (this kind of clustering is also called overlap clustering). The following sections explain these clustering techniques in more detail.

2.1.1 Hierarchical Clustering

Hierarchical clustering builds clusters in a hierarchy that represents the similarity levels at which the clusters are formed [57]. Compared with partitional clustering, hierarchical clustering is a nested sequence of partitions that are represented as a dendrogram (tree). Hierarchical clustering builds clusters gradually, while partitional clustering is a single partition that learns clusters directly [95].

Hierarchical clustering can be categorised into two different procedures: agglomerative (bottom-up technique) and divisive (top-down technique). The agglomerative technique starts by assigning each object to its own cluster and then gradually merges similar clusters to form larger clusters. This continues until a stopping criterion is achieved. On the other hand, the divisive procedure starts by assigning all objects into one cluster and then splitting this into smaller clusters. This continues until a stopping criterion is achieved [95].

The merge or split procedure is based on the similarity between objects in a cluster and on the dissimilarity between objects in different clusters. An important example of measuring (dis)similarity between two objects is the measure of the distance between them; such measuring is called a linkage metric. There are different linkage methods, such as Single linkage, Complete linkage, Average linkage and Centroid linkage. In the Single linkage method, the distance between two clusters is defined as the minimum distance between a pair of objects drawn from the two clusters (i.e., one object from one cluster, the other from another). This is also called the nearest neighbour method. In contrast, the distance between two clusters in the Complete linkage algorithm is the maximum of all pairwise distances. In the Centroid linkage method, the distances between clusters are determined by the Euclidean distance between centroid objects. The Average linkage method considers the average pairwise distance between all objects in two clusters [95].

Although hierarchical clustering does not require information about the number of clusters, it has many disadvantages. The main disadvantage is high computational complexity, which in most algorithms is $O(n^3)$, where n is the number of objects in the dataset. Thus, they have limited application in large datasets because a distance matrix must be calculated at each step. Moreover, it is sensitive to noise and outliers [109].

2.1.2 Partition Clustering

Partition clustering is "simply a division of the data objects into non-overlapping subsets (clusters)" [95]. It does not have a hierarchical structure, and the partitioning is based on a specific criterion, called the criterion function, such as minimising the sum of the squared distances. It is divided into two main sub-categories: centroid algorithms and medoid algorithms:

Centroid Algorithms These represent each cluster by centre of gravity of the objects. The best-known centroid algorithm is *k-means* [44], which requires the number of clusters k for the dataset to be specified, and then it partitions the data into k clusters. Cluster similarity is measured based on the mean value of the objects in the cluster, which is viewed as the cluster's centre. Thus, all objects in the dataset are assigned to their closest centre [95]. The *k-means* algorithm is the best-known squared error-based clustering algorithm, which is presented below:

1. Set the value of k.
2. Select k random objects as initial centroids, C_j, $j = \{1, \ldots, k\}$
3. For each object x_i in dataset X.
    (a) Compute the distance between x_i and each centroid C_j (for example using the Euclidean distance as in equation 2.1)
    (b) Assign x_i to its nearest centroid.

4. Update the centroid for each cluster by taking the mean of all the objects in that cluster.
5. Repeat steps 3 and 4 until a stable clustering result is reached and/or no change is made to the centroids.

Generally, the main property of the *k-means* algorithm is that it is efficient for large datasets, and it often terminates at a local optimum; the resultant clusters have spherical shapes [109]. However, it is sensitive to noise, as well as outliers in data and initial centroids, and also needs a pre-selected value for k. Each run of *k-means* may generate a different clustering result [95].

Medoid Algorithms In this method, each cluster is represented by one of its elements. The best-known is the *k-medoids* algorithm, also called Partitioning Around Medoids (PAM) [59]. One of its advantages is that it deals with noisy data by setting the mean of a cluster to be the object that is nearest to the 'centre' of the cluster. Moreover, it is efficient for categorical data [109]. The key steps of *k-medoids* are as follows [59]:

1. Randomly select k objects as medoids from dataset X.
2. Assign each object to its closest medoid based on the distance metric.
3. Calculate the sum of distances from all objects to their medoids.
4. Calculate a swapping cost for each pair of non-medoids and medoids. Swapping means using a non-medoid to replace a medoid. If the replacement can decrease the value of the objective function, the swap will be confirmed; otherwise, the medoid will not be replaced by the non-medoid.
5. Repeat steps 2, 3 and 4 until there is no change in the medoids.

One of the disadvantages of this method is that it assumes that each cluster can be well-represented by its medoid, which might not be the case in some datasets where this assumption cannot be applied. Moreover, because the time complexity is $O(k(n-k)^2)$, it is not efficient in dealing with large datasets [59].

2.1.3 Fuzzy Clustering

This allows for an overlap between clusters; it is thus sometimes called soft clustering [109]. The best-known fuzzy clustering algorithm is *c-means*, which was developed by Dunn [22] and improved by Bezdek [8]. *c-means* assigns a degree for each object to express, if it belongs to a cluster. It is similar to k-means in that it minimises the objective function. The key steps in *c-means* are as follows [8]:

1. Choose a value for k clusters.
2. Randomly assign fuzzy coefficients to each object in the clusters.
3. Based on the fuzzy coefficients, compute the centroid for each cluster.
4. Based on the new cluster centres, re-calculate the coefficients of each object.
5. Compare the variance with a predefined sensitivity threshold.
6. Repeat steps 3, 4 and 5 until the variance of the fuzzy coefficients is less than the sensitivity threshold.

c-means is also sensitive to noise and outliers, and like most clustering algorithms, it requires prior knowledge of the number of clusters [109].

2.1.4 Issues with Clustering Algorithms

There are a number of issues related to clustering algorithms. Firstly, several optimal solutions are possible. Different structures for the same dataset can be achieved by a single algorithm (but with different parameters) or by several algorithms. The use of different distance metrics produces different clustering results. This makes the selection of the most appropriate clusters more difficult because the data are unlabelled and the parameters cannot be tuned by using cross-validation [2]. Furthermore, exploring all possible solutions is an expensive computation and, in practice, it is infeasible for large datasets.

Secondly, the correct number of clusters for any given data is often unknown. Current applications involve increasingly complex and large datasets, which may have complex clustering shapes, highly unbalanced clustering sizes, differing densities, and possible overlap clustering; all these issues create several challenges in the selection of a suitable single clustering algorithm for extracting meaningful cluster structures [4]. Therefore, it is logical to combine multiple clustering models to build a clustering ensemble.

2.2 Clustering Ensemble Methods

Ensemble clustering is the process of combining the multiple clustering results of a set of objects into a single improved clustering. It is sometimes referred to as the Consensus solution or Clustering Aggregation. In recent years, various studies have been conducted to develop clustering ensemble methods inspired by the success of the ensemble method in the supervised learning field [94, 26, 97, 27, 98]. However, compared to the research on classification ensemble methods, building a clustering ensemble is not straightforward, and further work is required in this field.

There are several reasons that make the task of building a clustering ensemble more challenging than that of classification. One is that clustering is unsupervised learning in which the data are unlabelled, so there is no prior knowledge with which the algorithm can discover the true cluster structure, and there is no "ground truth" to validate the clustering result. Moreover, no cross-validation technique can be carried out to tune the clustering algorithm's parameters, thus there are no guidelines with which the user can select the most appropriate clustering algorithm for a given dataset. Another challenge is that the number of clusters produced may differ among the generated solutions by different clustering algorithms. In addition, the number of clusters in the final solution is unknown in advance. The final solution is obtained by accessing a set of base solutions, which in fact are cluster labels, and not the original data used.

Ghosh and Acharya [34] pointed out that there are several motivations for using clustering ensembles, and that these are much broader than those for using a classification ensemble, where the main motivation of the latter is to improve the classification accuracy. These reasons include:

- To improve the quality of the clustering results compared to those produced by single clustering algorithms.
- To reuse existing clustering (knowledge reuse): in some applications a variety of partitions may exist, so they can be combined to obtain a final clustering result. This delivers a more consolidated clustering result; several examples are provided in [94].
- To generate robust clustering results across different types of datasets. It is widely known that the popular clustering algorithms often fail to produce a good clustering result when the data do not match with their assumptions.

Among these objectives, the first point is the most widely accepted one. The cluster quality is usually measured with a numerical measurement to assess different aspects of cluster validation [95]. Section 2.2.4 reviews some of these in more detail.

2.2.1 The Process of the Clustering Ensemble Method

Recently, Vega-Pons and Ruiz-Shulcloper [102] summarised the process of clustering ensemble into two main steps: generation and consensus. Figure 2.1 illustrates this process, in which the input is the original dataset and the output is the consensus clustering.

Generation Step This is the first step in the clustering ensemble process, where a number of ensemble members are generated by using particular generation techniques. Vega-Pons and Ruiz-Shulcloper [102] pointed out that greater variance in the set of ensemble members means that more information is available to the consensus function. Moreover, there are no constraints on how the ensemble members must be obtained [102]. Therefore, different strategies could be applied. In the literature,

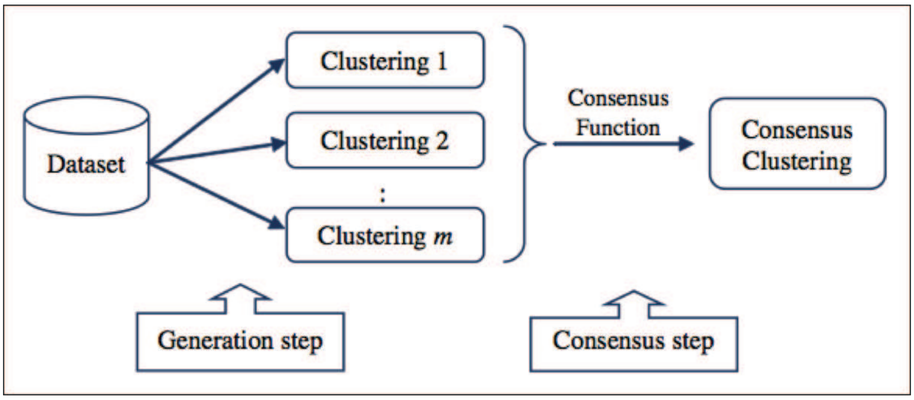


Figure 2.1: The Clustering Ensemble Process [102]

several generation techniques have been used to generate members for building an ensemble; more details on these techniques can be found in section 2.2.2.

Consensus Step The second step is where the generated members are combined using a consensus function to obtain the final clustering result. The success of a clustering ensemble relies on choosing a consensus function that can improve the quality of the final clustering solution [36]. As a result, a number of consensus functions have been proposed in the literature; section 2.2.3 will review some common consensus functions.

2.2.2 Ensemble Generation Techniques

Some researchers have applied techniques based on the types of data or applications that have been used. For high dimensional data, Strehl and Ghosh [94] applied random feature subspaces; members are generated for each of the data subspaces. They also generated members by selecting different subsets of objects for each member. They called this technique object distribution and they applied it to big data. Fern and Brodley [25] generated members based on random projections of objects onto different subspaces, and the Expectation Maximization algorithm (EM) is applied to these subspaces. The resampling method was used by [74, 76, 5], in particular bootstrap, which is a sampling with replacement. Minaei-Bidgoli et al. [74] used the bootstrap technique with a random restart of *k-means* [74], while Monti et al.

[76] used the bootstrap technique with different clustering algorithms, including *k-means*, model-based Bayesian clustering and self-organising map. Moreover, Ayad and Kamel [5] used bootstrap resampling in conjunction with *k-means* to generate the ensemble members.

Others used the most popular clustering algorithm *k-means* to generate the members (with a random initialisation of cluster centres). *k-means* has been widely used due to its simplicity and its low computational complexity [31, 97, 32, 35, 6, 51]. For instance, Fred and Jain [32] used it with random initialisations of cluster centres and a randomly chosen k (number of clusters) from a pre-specified interval for each member. They used a large k value in order to obtain a complex structure within the ensemble members. They also ran *k-means* with a fixed k to compare the two generation techniques and they found that members with a random k are more robust than other members. Dimitriadou et al. [19] and Sevillano et al. [89] applied fuzzy clustering algorithms in particular *c-means* in order to generate soft clustering members, while in Hore et al. [45] they applied fuzzy *k-means.*

Strehl and Ghosh [94] used a graph-clustering algorithm with different distance functions for each member. Topchy et al. [98] used a weak clustering algorithm, which produces a clustering result that is slightly better than a random result in terms of quality by using two different techniques. In the first technique, they used a random projection on one dimension from the original features, whereas in the second technique they split the data into a random number of hyperplanes. The weak algorithm is simple, fast at generating members, and it has been shown that it is able to produce high-quality ensemble results.

Iam-on et al. [50] examined different techniques, including a multiple run of *k-means* with a fixed k for each member and a randomly chosen k from an interval, where the maximum k is equal to $\sqrt{n}$. However, setting k equal to this value appears to be unrealistic for a big dataset. Furthermore, Iam-on et al. [48] applied different generation techniques to categorical data; they ran *k-mode* algorithm with full space and random subspaces with also a fixed k and random k. They found that these two techniques allowed their ensemble method to achieve high performance compared to the *k-mode* clustering algorithm, as well as some other ensemble methods such as those proposed by Strehl and Ghosh [94].

Another popular technique is to use different clustering algorithms for each member [111, 35], where all of the algorithms may complement each other. Yi et al. [111] used the best-known clustering algorithms, such as Hierarchical clustering and *k-means*. Gionis et al. [35] used the Single, Average, Ward and Complete linkage methods and *k-means* to generate ensemble members. Recently, Yu et al. [113] applied the Gaussian mixture model in conjunction with bagging techniques. *k-means* and EM were used to estimate the Gaussian mixture models' parameters.

Iam-on et al. [48] classify the techniques used in the generation step into five categories as shown in Figure 2.2, these are:

- **Homogeneous ensemble:** A single clustering algorithm is used to generate a number of members.
- **Different-k:** Each member is generated with different randomly selected k.
- **Data subspace/subsample:** Each member is generated by a random subsample of the data, or onto different subspaces, or by using a random subset of features.
- **Heterogeneous ensemble:** Each member is generated using a different clustering algorithm.
- **Mixed heuristics:** Any combination of the above techniques can be mixed to generate a number of members.

2.2.3 Review of Consensus Functions

A number of consensus functions have been proposed in the literature; some of them are based on how they represent the clustering ensemble problem, others by

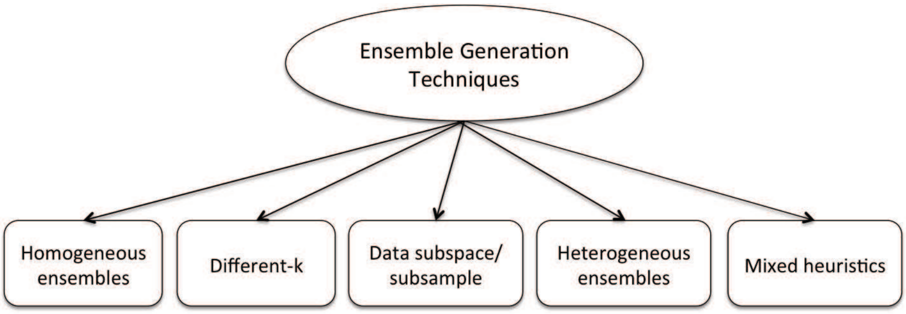


Figure 2.2: Diagram of the five categories of the ensemble generation techniques, as classified by Iam-on et al. [48].

applying well-known mathematical concepts to the problem. As the clustering ensemble is motivated by the preceding work on classification ensembles [64], the voting combination strategy was one of the early developments, where the labelling correspondence problem needs to be solved first. Another representation of the cluster labels is as categorical data [98], where some researchers represent the members as categorical features in which a category-based clustering algorithm is applied. Others transform the members into a binary membership matrix in which the pairwise similarity matrix can be calculated [32] (i.e Co-association matrix (CO)). Other researchers used such a matrix to formulate a graph to which a graph-based clustering method is applied [94].

Recent reviews on clustering ensemble methods can be found in [102, 34], where the authors have been trying to classify these methods according to their techniques. Among them we consider the classification scheme originally proposed by Vega-Pons and Ruiz-Shulcloper [102] due to its simplicity. This facilitates the introduction of the main ensemble methods presented in the literature. Thus, according to them, the consensus function can be classified into two main approaches: Object Co-occurrence-based approaches and Median Partition, which are as follows:

1. The Object Co-occurrence Approach: This first computes the co-occurrence of objects in the members and then determines their cluster labels to produce a consensus result. Basically, it counts the occurrence of an object in one cluster, or the occurrence of a pair of objects in the same cluster, and generates the final clustering result by a voting process among the objects. Examples of such approach are: the Relabelling and Voting method [21, 6, 114], the Co-association matrix [32] and the Graph-based method [94, 26].

2. The Median Partition Approach: This treats the consensus function as the optimisation problem of finding the median partition with respect to the cluster ensemble. The median partition is defined as “the partition that maximises the similarity with all partitions in the clustering ensemble ” [102]. Examples of this approach include the Non-Negative Matrix Factorisation based method [67], the Genetic-based method [112, 70] and the Kernel-based method [101]. More details on these methods can be found in [102].

Vega-Pons and Ruiz-Shulcloper [102] pointed out that consensus functions were primarily studied on a theoretical basis, and as a result many consensus functions based on the median partition approach were proposed in the literature, whereas only a few studies focused on the object co-occurrence approach. The following sub sections review the most common clustering ensemble methods.

2.2.3.1 Graph-based Methods

One of the early methods was proposed by Strehl and Ghosh [94], where they transformed the clustering ensemble problem into a graph problem, and proposed three different consensus functions: the cluster-based similarity partitioning algorithm (CSPA), the hypergraph partitioning algorithm (HGPA) and the meta-clustering algorithm (MCLA). In CSPA, the similarity matrix is used as the adjusted similarity matrix of a fully connected graph, where nodes correspond to objects and edge weights to their similarities. The final result is obtained by using the METIS package[1] in particular PMETIS [58]. This method is similar to the evidence accumulation method described by Fred and Jain [32], where the hierarchal clustering algorithm is applied to obtain the final clustering result.

[1] A set of multilevel graph partition algorithms.

On the other hand, a hypergraph is constructed in HGPA and MCLA, in which each ensemble member is represented as a hyper-edge. In HGPA, the hyper-graph is directly partitioned by cutting a minimal possible number of hyper-edges, where all hyper-edges have the same weight, into k connected nodes of approximately the same size. To do that, the authors used the hypergraph partitioning algorithm HMETIS [58]. In contrast, MCLA first defines the similarity between two clusters in terms of the amount of objects grouped in both, using the Jaccard index. Then a meta-graph is constructed where nodes represent clusters and the edges represent the similarity relations between pairs of clusters. The final partition, which is called meta-clustering, is obtained using PMETIS [58], where the meta-graph is then partitioned into k balanced meta-clusters. The complexity of CSPA, HGPA and MCLA is estimated in [94] as $O(kn^2m), O(knm)$, and $O(k^2nm^2)$, respectively.

Furthermore, Fern et al. [26] proposed the hybrid bipartite graph formulation (HBGF) algorithm by building a bipartite graph. In this type of graph there are only two different types of nodes, and edges exist between nodes of different types. In HBGF, one type of node represents an object, whereas the other type represents clusters, and an edge exists only between the cluster and the object belonging to that cluster. Then, they applied a spectral clustering algorithm to obtain the final partition. Its computational and storage complexity is $O(knm)$, as estimated by Fern et al. [26].

Al-Razgan and Domeniconi [2] proposed two graph-based algorithms: the weighted bipartite partition algorithm (WBPA) and the weighted subspace bipartite partition algorithm (WSBPA). They combine members generated by the local adaptive clustering algorithm (LAC), which designed to work with numerical data and assigns weights to the features in the cluster. PMETIS is also used to obtain the final clustering result. The only difference between these two algorithms is that WSBPA adds a weight vector to each cluster in the final clustering result.

2.2.3.2 Object Pairwise Similarity-based Methods

The most popular pairwise similarity-based method is the Co-association method, which avoids the labelling correspondence problem by mapping the ensemble members onto a new representation in which the similarity matrix is calculated between a pair of objects in terms of how many times a particular pair is clustered together for all ensemble members [32]. The final partition is obtained by applying any similarity-based clustering algorithm to this matrix. This method is Evidence Accumulation (EAC), and each entry in the matrix represents evidence collected from all ensemble members for a pair of objects. EAC calculates the percentage of members in the ensemble in which a given pair of objects is placed in the same cluster as follows:

$$CO(x_i, x_j) = \frac{1}{M}\sum_{m=1}^{M} \delta(P_m(x_i), P_m(x_j)) \tag{2.3}$$

Where x_i and x_j are objects, P_m is a partition, and $\delta(P_m(x_i), P_m(x_j))$ is defined as:

$$\delta = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ are in the same cluster in member m.} \\ 0, & \text{if } x_i \text{ and } x_j \text{ are in different clusters in member m.} \end{cases} \tag{2.4}$$

In Fred and Jain [32], the final partition is obtained by applying Single and Average linkage hierarchical clustering algorithms to the Co-association matrix. Building the hierarchical tree is achieved using the Single linkage edges with a minimum weight, which are cut based on a specific threshold. This threshold is obtained based on the decision of the number of clusters, and they defined this criterion as the range of threshold values needed to obtain k clusters, which they call the k-cluster lifetime. On the other hand, Fred [30] used a fixed threshold equal to 0.5 to obtain the final partition, where objects are joined in the same cluster if they have a similarity value greater than 0.5.

While, the Co-association matrix seems to be an ideal tool for collecting all of the information available in the clustering ensemble, it should be noted that the original Co-association matrix [32] captures only the pairwise relationship between objects in the ensemble members. Recently (in 2009), researchers have realised that more information within the generated members can be obtained to create this matrix. Wang et al. [107] proposed Probability Accumulation (PA) which extends the Co-association method by considering the cluster size and the dimensions of the objects within the data when calculating the Co-association matrix.

In PA, a more informative similarity matrix is obtained from the ensemble members compared with the Co-association method, which means that the chance of obtaining several pairs of objects with the same similarity score is less than that of using Co-association. Vega-Pons et al. [103] proposed a weighted-association matrix that takes three different factors into consideration. These are: the number of elements in the cluster to which a pair of objects belongs; the number of clusters in the ensemble member analysed; and the similarity value between the objects that were obtained by this member. They follow the same philosophy of Co-association by calculating the similarity matrix and then applying a hierarchical clustering algorithm and selecting the one with the highest lifetime criterion. They call this method Weighted Evidence Accumulation (WEA). In their work, they also proposed another algorithm based on the weighted-association matrix, by introducing a new intermediate step, called Information Unification, after the matrix is obtained. This aims to unify the different data representations and (dis)similarity measures into a new data representation, where each object is represented by (dis)similarity values (as new features).

However, we believe that there is more information in the generated members that we should consider when we calculate the similarity matrix, rather than just considering the pairwise relationship between objects.

Recently (in 2012), Yi et al. [111] highlighted an issue that is often overlooked by other methods: how to handle the uncertain data pairs when calculating the similarity matrix. They defined uncertain pairs of objects as the “pairs that have been assigned to the same cluster by approximately half of the partitions in the ensemble, and assigned to different clusters by the other half" [111]. They assumed that if the number of uncertain pairs is large, then this could mislead the consensus function into producing inappropriate final result. They addressed this issue by proposing a new clustering ensemble, based on the matrix completion theory, where they filtered out the uncertain pairs in the Co-association matrix, and then they estimated their value to complete the matrix by applying a matrix completion algorithm, namely the Augmented Lagrangian as proposed by Lin et al. [68]. However, by using a matrix completion process, their approach has the disadvantage that it may cause information loss.

Moreover, a method called weighted-object clustering ensemble (WOEC) was proposed by Ren et al. [81]. It uses the Co-association matrix to define a one-shot weight assignment to objects, where a large object's weight means that it is hard to cluster, whereas a small weight means that it is easy to cluster. In fact, they follow the same idea as the Boosting algorithm [85]. Ren et al. [81] proposed three weighted object versions of the classical clustering ensemble algorithms CSPA, HGPA and MCLA [94] reviewed earlier.

2.2.3.3 Voting-based Methods

In this kind of method, the labelling correspondence problem is first solved, and then a voting process ensues, in which each object should vote for the cluster to which it will belong in the final clustering result. Dudoit and Fridly [21] proposed a consensus function similar to the (Bagging) plurality voting used in classification ensembles, in which they solved the labelling correspondence problem using the Hungarian method [29]. They assumed that all members have the same number of clusters, and they obtained the final clustering result, which also has the same number of cluster as the members, by applying the plurality voting process.

Zhou and Tang [114] proposed a new voting method, where the clusters in the members are aligned by counting their overlapped objects, and the pairs of clusters that have the largest overlap are matched. Then simple voting is used to combine these aligned clusters. They also proposed a weighted voting method where they employed Normalised Mutual Information (NMI) [94] to weight the aligned clusters. Moreover, they proposed two selecting methods based on the NMI weight, where they included in the ensemble just the clusters whose NMI weight was larger than the specified threshold.

On the other hand, three different cumulative voting methods were proposed by Ayad and Kamel [6]; these are Un-normalised fixed-Reference Cumulative Voting (URCV), fixed-Reference Cumulative Voting (RCV) and Adaptive Cumulative Voting (ACV). In these methods, each ensemble member provides a soft or probabilistic vote for each object on which clusters they should belong to in the ensemble result. Then they are thresholded to determine the membership of each object to the ensemble clusters. This process requires a mapping function between the selected reference member and the other members. For this purpose, they used a theoretical information criterion based on the information bottleneck principle [6].

Vega-Pons and Ruiz-Shulcloper [102] argue that the main drawback of these methods is that they restrict the clustering ensemble problem because they require all the members to have the same cluster numbers, as well as the final clustering results produced by the consensus function, and that affects the ensemble quality. Furthermore, these methods require more time to solve the labelling correspondence problem than other consensus functions.

2.2.3.4 Probability-based Methods

The probability model has been used to find the median partition, which is a partition that best summarises the ensemble members. Topchy et al. [97] proposed a method based on a finite mixture model, where each member is modelled as a mixture of multivariate multinomial distributions and the maximum likelihood problem is solved by using the EM algorithm. They applied their method to deal with incomplete members, where some of the cluster labels are missing. Another work by Topchy et al. [98] represented the clustering ensemble as a categorical clustering problem, and the combined partitions were produced based on the median partition. They named their proposed algorithm the Quadratic Mutual Information Algorithm (QMI) [98].

Wang et al. [105] proposed a Bayesian version of the multinomial mixture model; they called it the Bayesian Cluster Ensemble (BCE). They used variational expectation maximisation and Gibbs' sampling to estimate the parameters and the inference. They generalised their algorithm to work when the original features of the data were available. They compared it with BCE and found that the generalised version achieved higher quality.

Louren et al. [69] proposed a probabilistic consensus clustering based on the Co-association matrix, where each entry is regarded as a Binomial random variable, parametrised by the unknown class assignments. They determined the object probability assignments to a cluster by minimising a Bregman divergence between the observed Co-association frequencies and the corresponding co-occurrence probabilities expressed as functions of the unknown assignments. Then to solve the problem under any double-convex Bregman divergence, they proposed an optimisation algorithm. They also adapted their proposed method for large scale datasets.

Recently, Yu et al. [113] proposed a Gaussian Mixture Model Cluster Structure Ensemble method (GMMCE), where as we said they used the Gaussian mixture model to generate the members; each one of them captures the underlying structure from different data sources. The main aim of the ensemble is to identify the most applicable structure of the data. For estimating the parameters of the Gaussian mixture models they used *k-means* and the EM algorithm. Each model is then represented as a new data sampling in which a matrix is constructed representing the relationship between components. They measured the similarly between two components corresponding to their respective Gaussian distributions, measured by a distance function called *Bhattycharya*.

2.2.3.5 Link-based Methods

Iam-on et al. [49, 51] applied link network analysis to clustering ensembles and they proposed a number of methods. First, they proposed two consensus functions based on pairwise similarity, named the Connected-Triple based similarity (CTS) matrix and the SimRank based similarity (SRS) matrix [49]. They then proposed an improved version of SRS, called approximate SimRank-based similarity (ASRS) [52]. Basically, they represented the ensemble members as a link network and then they implemented the well-known link-based similarity measures developed in the classification of web document areas to this member/cluster network, as their names indicate.

In CTS, the members are represented as a cluster network. For example, let us say that we have 3 members, $P_1 = \{a, b\}, P_2 = \{c, d\}$ and $P_3 = \{f, g\}$, and there are two objects, x_1 and x_2, which belong to different clusters a and b respectively in member P_1, whereas they belong to the same cluster c and f in members P_2 and P_3 respectively. According to members P_2 and P_3, the pairs x_1 and x_2 are considered to be similar, but according to P_1 their similarity is equal to zero. Applying the connected triple concept, it is found that clusters a and b are justified as similar as they have 2 connected-triples, which are clusters c and f from the two other members.

The object pairwise similarity matrix for a given pair (x_i, x_j) in CTS is calculated as follows:

$$CTS(x_i, x_j) = \frac{1}{M} \sum_{m=1}^{M} S(x_i, x_j) \tag{2.5}$$

Where $S(x_i, x_j)$ is defined as:

$$S(x_i, x_j) = \begin{cases} 1, & \text{if } x_i \text{ and } x_j \text{ belong to same cluster.} \\ S_{cts}(c(x_i), c(x_j)) \times DC & \text{Otherwise.} \end{cases} \tag{2.6}$$

Where $S_{cts}(c(x_i), c(x_j)) = \frac{T_{ij}}{max\{T\}}$, and $T_{ij} = \sum_{e=1}^{t} \min(w_{i,e}, w_{j,e})$.

The $w_{i,e}$ is the link weight between two clusters i and e, which is calculated as a Jaccard index (equation 2.16). The DC is “the confidence level of accepting two non-identical objects” [49], and it takes value $\in [0, 1]$.

The SimRank (SRS) represents ensemble members as a bipartite graph, which has two types of nodes: clusters and objects, and the link exists only between clusters and objects. It assumes that if two objects have similar neighbours then they are similar as well. The similarity between a given pair of objects is calculated as follows:

$$SRS(x_i, x_j) = \begin{cases} 1, & \text{if } x_i{=}x_j. \\ \frac{DC}{|N_{x_i}||N_{x_j}|} \sum\limits_{a' \in N_{x_i}} \sum\limits_{b' \in N_{x_j}} SRS(a', b') & \text{Otherwise.} \end{cases} \tag{2.7}$$

Where N_{x_i} is the set of cluster nodes connected to object x_i. The similarity SRS matrix can be calculated between a pair of objects, and is defined as the average similarity between clusters to which they belong, which in turn is calculated as the average similarity between their objects. The final SRS similarity matrix is obtained after a number of iterations (t) in order to refine the similarity values to stable values that do not change.

$$\lim_{t \to \infty} SRS_{t+1}(a, b) = \frac{DC}{|N_{x_i}||N_{x_j}|} \sum_{a' \in N_{x_i}} \sum_{b' \in N_{x_j}} SRS_t(a', b') \tag{2.8}$$

The iteration process starts at the outset of: $SRS_0 = 1$ if $x_i = x_j$ and 0 otherwise. In the ASRS, the SRS is improved by eliminating the iteration process to make it more efficient. It is calculated between a given pair of objects as follows:

$$ASRS(x_i, x_j) = \begin{cases} 1, & \text{if } x_i{=}x_j. \\ \frac{1}{|N_{x_i}||N_{x_j}|} \sum\limits_{a' \in N_{x_i}} \sum\limits_{b' \in N_{x_j}} S^c(a', b') & \text{Otherwise.} \end{cases} \tag{2.9}$$

S^c is the similarity between two clusters, which is represented as a subgraph, where the node represents a cluster, the edge connects two clusters together, and the weight of edge $w_{a'b'}$ connecting clusters a' and b' is calculated as the Jaccard index.

$$S^c(a',b') = \frac{wS^c(a',b')}{max\{wS^c\}} \times DC \tag{2.10}$$

$$wS^c(a',b') = \frac{1}{|N'_a||N'_b|} \sum_{y \in N_{a'}} \sum_{z \in N_{b'}} (w_{a'y} \times w_{b'z}) \tag{2.11}$$

However, the final clustering result is obtained by applying a hierarchical clustering algorithm over the obtained similarity matrices. Iam-on et al. [51] also proposed three improved versions of the above algorithms named Weighted Connected Triple (WCT), Weighted Triple-Quality (WTQ), and Combined Similarity Measure (CSM). In these consensus functions, they considered the relationship between and within the members (clusters) in the consensus function. In other words, they considered the similarity within clusters to reflect the similarity between objects in one cluster and the similarity between clusters in different members. In WCT, they extended CTS to represent a weighted network, where nodes represent clusters and edges represent the overlap between them. The concept is very similar to the MCLA method. The similarly between two nodes is measured with respect to their centre of triple as the average of the sum of their minimum edge weight multiplied by DC (decay factor).

The WTQ is inspired by the work in [1], where the quality of the shared triple is taken into account when calculating the similarity between two nodes (clusters). The CSM, on the other hand, combines WCT and WTQ algorithms together. The final clustering result is obtained by applying *k-means*, *k-medoids*, and spectral graph partition to the constructed link similarity matrices. This work shows that mathematical concepts from other disciplines can be applied to the clustering ensemble to represent the members in a way that makes most of their information available to the consensus function. One disadvantage with these methods is that they require a clustering algorithm to be applied to the calculated similarity matrices; which one to use is a question yet to be answered, and may affect the final clustering results just as a common clustering algorithm does in the first place.

2.2.4 Clustering Ensemble Evaluation

Evaluating the quality of the clustering result is called clustering validity assessment. There are three different cluster validation indices: external, internal and relative [82]. The following subsections describe them in more detail.

2.2.4.1 External Validation Index

The external index is the most common validation method used in the clustering ensemble method. It is based on previous knowledge about the data. It measures the similarity of the clustering results to the external information "ground-truth". Hence, any valid similarity measure suitable for partition comparison can be used as an external index [40]. In the literature, most external indices that have been used either to validate the final clustering ensemble result or in diversity measures (Section 2.3.1), are as follows:

Rand Index and Adjusted Rand Index

The Rand index (RI), as well as the Adjusted Rand index (ARI), are classified as counting pair similarity-based measurements. They are the most relevant similarity measures in this type of measurement, which is based on four count situations. Suppose that we have two partitions P_1 and P_2 of the dataset X of n objects, and all pairs of objects are x_i and x_j, where $i \neq j$. There are four possible situations in which those pairs could be accommodated:

- n_{00} - the number of object pairs assigned to different clusters in P_1 and P_2.
- n_{11}- the number of object pairs assigned to the same clusters in both P_1 and P_2.

- n_{10}- the number of object pairs assigned to the same cluster in P_1 and to different clusters in P_2.
- n_{01} - the number of object pairs assigned to different clusters P_1 and to the same cluster in P_2.

The four counts satisfy the following equation:

$$n_{00} + n_{11} + n_{10} + n_{01} = U \tag{2.12}$$

where U is the maximum number of all pairs in the dataset, that is $U = \frac{n(n-1)}{2}$.

The RI was proposed by Rand [80]. Basically, it measures similarity and enables the evaluation of the final clustering result by comparing two partitions, assuming one of them to be the ground-truth partition. It is defined as:

$$RI(P_1, P_2) = \frac{n_{11} + n_{00}}{U} \tag{2.13}$$

It measures the level of similarity within the range $[0, 1]$, where 0 indicates that the two partitions being compared are completely different, and the value 1 indicates that the two partitions being compared are identical. Comparing two random partitions using the Rand index does not give a constant value, which is a problem that has been corrected in its new version, the Adjusted Rand index, as proposed by Hubert and Arabie [47]. It is defined as follows:

$$ARI(P_1, P_2) = \frac{RI(P_1, P_2) - Expected[RI]}{1 - Expected[RI]} \tag{2.14}$$

With simple algebra, the ARI can be simplified to:

$$ARI(P_1, P_2) = \frac{\sum_{i=1}^{k} \sum_{j=1}^{k} \binom{n_{ij}}{2} - [\sum_{i=1}^{k} \binom{n_i}{2} \sum_{j=1}^{k} \binom{n_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_{i=1}^{k} \binom{n_i}{2} + \sum_{j=1}^{k} \binom{n_j}{2}] - [\sum_{i=1}^{k} \binom{n_i}{2} \sum_{j=1}^{k} \binom{n_j}{2}]/\binom{n}{2}} \tag{2.15}$$

where n_{ij} is the number of objects in the intersection of clusters $c_i \in P_1$ and $c_j \in P_2$,

n_i and n_j are the numbers of objects in clusters $c_i \in P_2$ and $c_j \in P_1$ respectively, and $\binom{n}{2}$ is the binomial coefficient $\frac{n!}{2!(n-2)!}$.

The maximum value of ARI is equal to 1, which means that P_1 is identical to P_2, and it has an expected value 0 for independent clusterings. It is not necessary for the number of clusters in P_1 and P_2 to be the same [60].

Jaccard Index

The Jaccard index (J) is also classified as a counting pair similarity-based measurement, and it gives similarity within the range $[0, 1]$ [87]. It is defined as follows:

$$J(P_1, P_2) = \frac{n_{11}}{n_{11} + n_{10} + n_{01}} \tag{2.16}$$

Mutual Information and Normalised Mutual Information

These two measures are classified as information-theoretic similarity-based measurements. They measure how much information is shared by two partitions. Mutual Information treats the compared partition as a random partition. It is defined as follows:

$$MI(P_1, P_2) = \sum\nolimits_{i=1}^{k} \sum\nolimits_{j=1}^{k} n_{i,j} \log \frac{n.n_{i,j}}{n_i.n_j} \tag{2.17}$$

where n_i is the number of objects in cluster $c_i \in P_1$; n_j is the number of objects in cluster $c_j \in P_2$; and $n_{i,j}$ is the number of shared objects between clusters c_i and c_j.

Strehl and Ghosh [94] showed that $MI(P_1, P_2)$ is a metric and that there is no upper bound for $MI(P_1, P_2)$. Thus, they proposed Normalised Mutual Information (NMI), which normalises mutual information to a $[0, 1]$ range; 1 is attained when P_1 is identical to P_2, and 0 is attained when P_1 is completely different from P_2. It is defined as follows:

$$NMI(P_1, P_2) = \frac{MI(P_1, P_2)}{\sqrt{\sum_{i=1}^{k} n_i \log \frac{n_i}{n} \sum_{j=1}^{k} n_j \log \frac{n_j}{n}}} \tag{2.18}$$

2.2.4.2 Internal Validation Index

The internal validation index only relies on information in the data, without any additional information. It is usually employed for the task of class discovery. Most of the internal validation indices are based on two criteria: compactness and separation [95]. The compactness is defined as a measure of how close objects are in a cluster. It is often measured by the variance, and a lower variance indicates better compactness. The separation is defined as a measure of how well-separated a cluster is from other clusters. It is usually measured by the distance between cluster centroids. Such internal validation indices based on these criteria are the Dunn index, the Davies-Bouldin index, the Silhouette index, the SD index and SDbw index. More details on these indices are given below:

Dunn index

The Dunn index is the ratio of the smallest distance between two objects from different clusters to the largest intra-cluster distance [23]. It is calculated as follows:

$$D(P) = \min_{i=1,\cdots,k} \left\{ \min_{j=1,\cdots,k} \left(\frac{\min_{x\in c_i, y\in c_j} Dis(x,y)}{max_{l=1,\cdots,k,c_l\in P} diam(c_l)} \right) \right\} \tag{2.19}$$

diam is the diameter, which is the maximum distance between two objects among all clusters, and the *Dis* is the standard Euclidean distance. The Dunn index has a value between 0 and ∞. A large value of D indicates that the partition P is compact and well-separated. So, this index should be maximised.

Davies-Bouldin Index

The Davies-Bouldin Index (DB) is proposed by Davies and Bouldin [17]. It is calculated as follows:

$$DB(P) = \frac{1}{k} \sum_{i=1}^{k} \max_{j=1,\cdots,k,i\neq j} \left\{ \frac{diam(c_i) + diam(c_j)}{Dis(c_i, c_j)} \right\} \tag{2.20}$$

Where *diam* is the diameter of a cluster calculated as the average Euclidean distance of objects in cluster i to the centroid of cluster i.

Silhouette Index

The silhouette index is a well-known measurement for estimating the number of clusters in a dataset. The silhouette is based on the pairwise difference between the compactness and the separation. The compactness is measured based on the distance within the cluster, which is measured as the average distance between all objects in the same cluster. The separation is measured based on the nearest neighbour distance. The silhouette is calculated as follows:

$$\begin{aligned} Si(P) &= \frac{1}{n}\sum_{c_i \in P}\sum_{x \in c_i}\frac{b(x,c_i)-a(x,c_i)}{max\{a(x,c_i),b(x,c_i)\}} \\ a(x,c_i) &= \frac{1}{|c_i|}\sum_{y \in c_i} Dis(x,y) \\ b(x,c_i) &= \min_{c_j \in c_i}\frac{1}{|c_i|}\sum\nolimits_{y \in c_j} Dis(x,y) \end{aligned} \tag{2.21}$$

SD Index

The SD index was proposed by Halkidi et al. [42], and is based on the average scattering and the total separation of clusters. The compactness *Comp* is measured as the variance of cluster objects, and the separation *Sep* is measured as the total separation between cluster centres C. The value of this index is the summation of these two terms, which is as follows:

$$SD = Sep(c_{max}).Comp(c) + Sep(c) \tag{2.22}$$

$$Sep(c) = \frac{\max_{i,j} Dis(c_i, c_j)}{\min_{i,j} Dis(C_i, C_j)} \sum_{i=1}^{k} \left(\sum_{j=1}^{k} Dis(C_i, C_j) \right)^{-1}$$

$$Comp(c) = \frac{1}{k} \sum_{i=1}^{k} \parallel \sigma(c_i) \parallel / \parallel \sigma(X) \parallel$$

Where σ is the variance vector of cluster. The optimal partition can be obtained by minimising the value of SD.

SDbw Index

Similarly, the $SDbw$ index is the summation of the compactness and the separation [41]. The compactness is measured in the same way as in the SD index, while the separation is measured based on the density of the clusters. It is calculated as follows:

$$SDbw(P) = Comp(c) + Dens\_bw(c) \tag{2.23}$$

$$Dens\_bw = \frac{1}{k(k-1)} \sum_{i=1}^{k} \left(\sum_{j=1, j \neq i}^{k} \frac{\sum_{x \in c_i \cup c_j} f(x, u_{i,j})}{\max(\sum_{x \in c_i} f(x, C_i), \sum_{x \in c_j} f(x, C_j))} \right)$$

where u_{ij} is an object in the middle of the line segment between the centres of clusters c_i and c_j, and $f(x, u_{i,j})$ is equal to 0 when the $Dis(x, u_{i,j})$ is larger than the average standard deviation of clusters, and 1 otherwise. The minimum value of this index indicates optimal partition [41].

2.2.4.3 Relative Validation Index

The concept of the relative validation index is based on comparing the partition to another partitioning resulting from the same algorithm, but under different conditions (e.g. using different parameter values). In other words, it is a measurement of the consistency of the algorithms. Two popular indices are Figure of Merit (FOM) [110] and Stability [65], and they are defined as follows:

Figure of Merit (FOM)

The Figure of Merit is an estimator of the clustering algorithm consistency, which was originally developed for gene expression data, and was proposed by Yeung and Haynor [110]. A gene expression dataset X contains n genes (objects) measured under u experimental conditions (features). Suppose a clustering algorithm is applied to all features in dataset X except feature e to obtain k clusters, $\{c_1^e, c_2^e \cdots c_k^e\}$. The figure of merit for feature e (FOM(e,k)) is calculated as follows:

$$FOM(e,k) = \sqrt{\frac{1}{n} \sum_{j=1}^{k} \sum_{i \in c_j^e} (x_{i,e} - \bar{x}_e^j)^2} \tag{2.24}$$

Where $x_{i,e}$ is the object value i in feature e in dataset X, and $\bar{x}_e^j$ is the average of feature e values only for objects belonging to cluster c_j^e.

Therefore, the FOM is defined as an estimate of the total clustering algorithm consistency over all the features for k clusters as follows:

$$FOM(k) = \sum_{e=1}^{u} FOM(e,k) \tag{2.25}$$

A lower value of FOM indicates a more consistent and better clustering result of the dataset.

Stability

The Stability measure is used to select the number of clusters in the model selection application. It is also used to compare between two partitions. The Stability measure is mainly developed to assess the capability of the clustered dataset to predicate the clustering of another same size dataset sampled from the same source [65, 66]. Assume that we have two datasets X and X' sampled from the same distribution. Applying a clustering algorithm to X and X', we get $P = \{c_1, c_2 \cdots c_k\}$ and $P' = \{c'_1, c'_2 \cdots c'_k\}$ respectively. For $x \in X$ if $x \in c_j$ then $P(x) = j$, where $j = \{1, \cdots, k\}$ and for $x' \in X'$ if $x' \in c'_j$ then $P'(x') = j$. The dataset X and its partition P can be used to train a classifier f to predict a new partition P_1 on X'. Then the consistency between the two sets (X, P) and (X', P') is measured as the dissimilarity between the original labels P' and the predicted labels P_1 using the modified Hamming distance as follows:

$$STB_{\zeta_k}(P', P_1) = \min_{\pi \in \zeta_k} \frac{1}{n} \sum_{i=1}^{n} \delta(P'(x'_i), \pi(P_1(x'_i))) \tag{2.26}$$

Where ζ_k is the set of all the permutations of the k clusters for partition P', and $\delta(P'(x'_i), \pi(P_1(x'_i))) = 0$ if $P'(x'_i) = \pi(P_1(x'_i)))$ and $\delta(P'(x'_i), \pi(P_1(x'_i))) = 1$ if $P'(x'_i) \neq \pi(P_1(x'_i)))$.

Then the stability of the clustering algorithm is computed as the average distance between partitions using the expectation E of the stability for pairs of independent datasets X, X' of size n drawn from the same source as follows:

$$STB(P) = E_{X,X'} STB_{\zeta_k}(P', P_1) \tag{2.27}$$

A smaller value of $STB \in [1, 0]$ indicates a more stable clustering result for the data.

2.3 Clustering Ensemble Diversity

Generally speaking, when somebody wants to form a sports team he/she has to ensure that each member of the team has a different and better skill in a particular aspect, hence each one plays a role and as a whole team they perform better than the individual members and better than a team with players who have identical skills.

The clustering ensemble problem can be seen in a similar way to this example. Intuitively, there is no point in building an ensemble with an infinite set of identical members as they are not going to produce a final clustering result any better than they were at the start. Thus, the ensemble members have to be different enough from each other to provide complementary information and to improve clustering quality over an individual partition when combined, the difference between the members is called diversity.

2.3.1 Related Work on Clustering Ensemble Diversity Measures

In the clustering ensemble, it has been found that diversity is the fundamental and crucial factor for building a successful clustering ensemble because an ensemble of identical members will not outperform the individual members [94, 39, 25]. Accordingly, a number of diversity measures have been proposed [60, 39, 37], most of them based on the matching of labels acquired from the two clustering results.

Two different approaches have been proposed for measuring diversity among members: the pairwise method (p) and the non-pairwise method (np) as seen in Figure 2.3.

2.3.1.1 Pairwise Diversity Measure (p)

In the pairwise method, each ensemble member is compared with the others, and then a common diversity measurement is used to measure the level of disagreement between any two partitions (which is the complement of a similarity measure S), such as $DV(P_i, P_j) = 1 - S(P_i, P_j)$; the Adjusted Rand index can be used as the measure of similarity (S), as defined in 2.15. This pairwise diversity measure (DV_p),

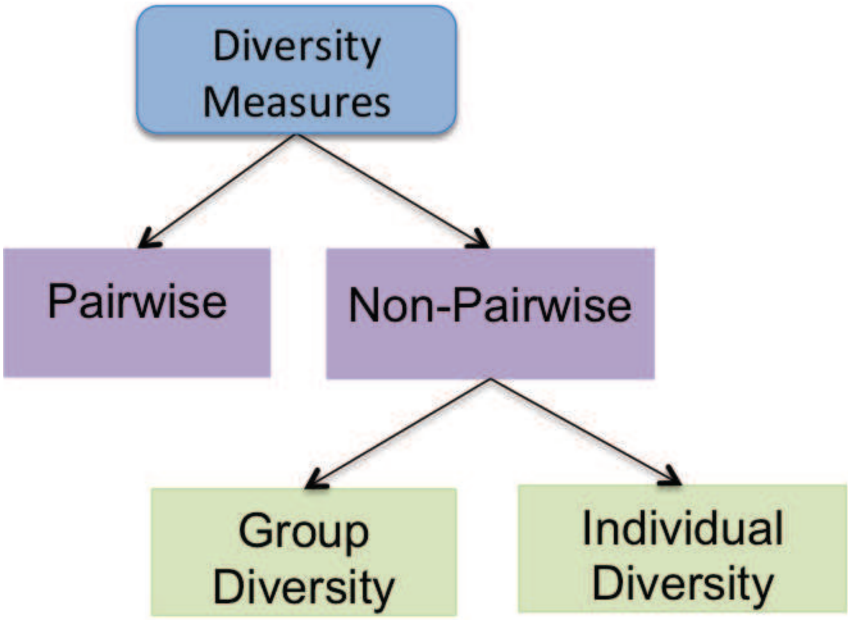


Figure 2.3: The two categories of diversity measures that have been proposed in the literature and the subdivision of the Non-Pairwise measure.

based on the Adjusted Rand index of m members, is defined as follows:

$$DV_p = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (1 - S(P_i, P_j)) \tag{2.28}$$

Where S is the ARI calculated as in 2.15

Fern and Brodley [25] used the same measurement DV_p but with NMI index to measure diversity, as follows:

$$DV_{pNMI} = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} (1 - NMI(P_i, P_j)) \tag{2.29}$$

They use it to analyse the influence of the quality and the diversity of the individual members on the ensemble performance. They found that, based on a number of experiments, there is a strong relationship between improving the ensemble's quality and both the diversity and the quality of its members. They also point out that high diversity leads to better ensemble performance [25].

2.3.1.2 Non-Pairwise Diversity Measures (np)

As discussed in Hadjitodorov et al. [39], the ensemble result is first obtained in the non-pairwise measurement, and then each member is compared with it.

This measurement is divided into: group diversity and individual diversity.

Group Diversity Greene et al. [37] proposed an entropy measurement as a group diversity measure, defined as:

$$Entropy = \frac{2}{n(n-1)} \sum_{i=1}^{n} \sum_{j=i+1}^{n} -(pb_{ij} \log_2 pb_{ij} + (1 - pb_{ij}) \log_2 (1 - pb_{ij})) \quad (2.30)$$

Where pb_{ij} represents the probability of clustering the two objects i and j together; the greater the entropy, the greater the diversity obtained among the members. They highlighted that diversity as well as the selection of the consensus function is important in producing better ensemble results; not diversity alone.

Individual Diversity Another measure, proposed by Hadjitodorov et al. [39], is the average diversity between the members and the ensemble result P^*, which is classified as an individual diversity and it defined as follows:

$$DV_{np1} = \frac{1}{m} \sum_{i=1}^{m} (1 - ARI(P_i, P^*)) \quad (2.31)$$

Moreover, they measured the spread of diversity between ensemble members compared to P^* by measuring the standard deviation as follows:

$$DV_{np2} = \sqrt{\frac{1}{m-1} \sum_{i=1}^{m} (1 - ARI(P_i, P^*) - DV_{np1})^2} \quad (2.32)$$

Using DV_{np2}, they discovered that a larger spread is not strongly related to ensemble quality, and based on this they proposed another measurement, which is as follows:

$$DV_{np3} = \frac{1}{2}(1 - DV_{np1} + DV_{np2}) \tag{2.33}$$

In these measurements, they assume that the ensemble result P^* is close to the ground truth partition of the data, and therefore the quality of each ensemble member is estimated based on how close they are to the ensemble result. They also compared the diverse ensemble members with the non-diverse ones and found that the diverse ones produced more high-quality ensemble results than the non-diverse members, even when the non-diverse members were more accurate than the diverse ones. Furthermore, they constructed another diversity measure as the coefficient of variation as follows:

$$DV_{np4} = \frac{DV_{np2}}{DV_{np1}} \tag{2.34}$$

2.3.2 The Relationship between Diversity and Ensemble

In the clustering ensemble, the above diversity measures have been used to discovered the relationship between the diversity and the clustering ensemble performance. Domeniconi and Al-Razgan [20] compared DV_{pNMI} and DV_{np3}, the latter applied the Adjusted Rand Index. They found that measuring diversity using the Adjusted Rand Index gives more robust and consistent results than NMI. This result is based on using the graph-based consensus function, and is the same as the results found by Hadjitodorov et al. [39], in which they used Co-association method.

However, Table 2.1 summaries the researches that have been done in the literature to discover the relationship between diversity and ensemble performance. Domeniconi and Al-Razgan [20] conclude that high diversity leads to high ensemble quality by using DV_{pNMI} and DV_{np3}, whereas Hadjitodorov et al. [39] discovered that selecting median diverse members leads to better ensemble performance than

Table 2.1: Summary of diversity research proposed in the literature for discovering the relationship between the diversity and the performance of the ensemble, along with the diversity measures proposed/used.

Authors	**Measure Proposed/Used**	**Type of Measure**	**The Recommended Diversity Level**
Fern and Brodley [25]	DV_p	p	High level
Greene et.al. [37]	Entropy	np (Group diversity)	–
Hadjitodorov et.al. [39]	DV_{pARI}, $Entropy$, DV_{np1}, DV_{np2}, DV_{np3} and DV_{np4}	p and np (Individual diversity)	Median level
Domeniconi and Al-Razgan [20]	DV_p and DV_{pNMI}	p	High diversity
Rozmus [83]	DV_p, DV_{np1}, DV_{np2}, DV_{np3} and DV_{np4}	p and np (Individual Diversity)	DV_{np1} discovered low diversity, while other measures discovered high diversity
Iam-On et.al. [51]	DV_{pNMI}	p	High diversity

selecting highly diverse members by using all their proposed measures, including Entropy.

Rozmus [83] studied the relationship between diversity and the performance of the ensemble by using five measurements: $DV_p, DV_{np1}, DV_{np2}, DV_{np3}$ and DV_{np4}, and he applied this using four different similarity indexes, which are the ARI index, the Rand index, the Jaccard index and the Fowlkes and Mallows index. He found that, in detecting a clear relationship between diversity and ensemble quality (using these measurements, in some cases using DV_{np1}), it is observed that lower diversity leads to higher ensemble quality. Whereas in other cases (using the other measurements), it is observed that higher diversity leads to higher ensemble quality. Furthermore, he pointed out that it is hard to distinguish between the indices in delivering a strong correlation between diversity and ensemble quality, but in some cases, using the Jaccard index delivers a more robust result.

It is clear that there is no generally accepted agreement in the literature on how to measure diversity and there is no measurement that is considered particularly effective and popularly accepted. Thus, we think that this is the main reason why there is no agreement on the relationship between diversity and ensemble performance. In addition, there is no single diversity definition specifically defined for clustering ensembles, and most of the proposed diversity measures in the literature are derived from other fields such as clustering validation methods and classification ensembles and not designed specifically in relation to clustering ensembles. In this research, we will investigate the clustering ensemble diversity experimentally using the current measurements to find out whether or not it has an influence on the ensemble performance.

However, diversity has been intensively studied in supervised learning; it is commonly defined as the level of variability between ensemble members. In other words, it is the level of dissimilarity among ensemble members [62]. It has been shown that ensemble learning in the context of classification and regression outperforms single learners both theoretically and empirically [12]. Moreover, there is a wide agreement that there is a trade-off between the accuracy of members and the diversity between them. In other words, it is essential that the ensemble members are highly diverse and sufficiently accurate [14, 62]. Brown et al. [12] reviewed the existing qualitative and quantitative diversity definitions in regression and classification contexts, in terms of how they are defined and how to create diversity in the ensemble. They suggested a taxonomy of methods to create diversity for a classification ensemble, by varying “starting points within the hypothesis space ”, varying the “set of accessible hypotheses” and “traversal of hypothesis space” for each member differently. They concluded that there is no agreed-upon theory to explain why and how the diversity affects the ensemble accuracy, and in classification tasks diversity is still an open question.

Tang et al. [96] studied the relationship between diversity and classification ensembles using 6 diversity measures. In fact, they analysed these diversity measures by relating them to the concept of classifiers' margin. Simply, the concept of margin is defined according to [86] as "the difference between the number of correct votes and the maximum number of votes received by any incorrect label". Tang et al.[96] showed that when they considered the average accuracy of the ensemble members as a constant, they achieved a maximum diversity, which is equivalent to maximising the minimum margin of the ensemble on training sets. They concluded that large diversity may not always correspond to better ensemble performance, so it cannot be explicitly used for selecting the best ensemble members [96].

2.4 Clustering Ensemble Application

The main aim of clustering ensemble methods is to improve the quality of the clustering result compared to the single clustering algorithm. Clustering ensemble methods can be applied to any clustering problems, such as privacy-preserving clustering problems, image segmentation, document clustering, detecting outliers and heterogeneous data clustering problems [35]. Thus, its huge potential has motivated researchers to continuously develop new clustering ensemble methods.

For example, Strehl and Ghosh [94] developed clustering ensemble methods in order to reuse existing clustering results, which they called knowledge reuse. This is where a variety of partitions may already exist, so they are combined to obtain a final clustering result in order to produce an improved clustering result. Clustering ensembles on data from multiple sources, where in some situations, objects in the dataset have multiple views or sources, so the clustering ensemble can be carried out on a restricted view of the dataset [94]. Strehl and Ghosh [94] illustrated two different scenarios for using clustering ensembles: Feature Distributed Clustering, where each member is built by selecting different subsets of features and using all the objects in the dataset, and Object Distribution Clustering, where each member is built by different subsets of objects, using all the features.

Another useful application is to enable clustering over distributed computing, where in some situations, the dataset is distributed and it is not possible to collect it in one place due to privacy issues or data migration costs. Thus, clustering ensembles can be used in these situations where each member has access to a subset of features and/or to a subset of objects [94, 72]. Sevillano et al. [88] used a clustering ensemble and feature diversity in a document clustering application, where they presented several empirical findings on the robustness of their clustering results.

Some researchers have applied clustering ensembles to gene expression data such as Iam-On et al. [50] and Monti et al. [76]. Other researchers applied clustering ensembles to categorical data such as Iam-O et al. [48] and Gionis et al. [35]. Chang et al. [15] applied a clustering ensemble based on Expectation-Maximisation (EM) to a colour image quantisation application.

Recently (in 2013), Saeed et al. [84] applied a graph-based clustering ensemble proposed by Strehl and Ghosh [94] to a chemical structures dataset. Furthermore, clustering ensembles have been developed that have the ability to identify the "correct" number of clusters in the data. Mimaroglu and Aksehirli [73] developed a clustering ensemble method in which the number of clusters can be automatically determined. However, there are still other issues that need to be investigated, such as detecting outliers and heterogeneous data clustering problems.

2.5 Summary

In this chapter, the background of this study was reviewed, including the most well-known clustering techniques; hierarchical, partitional and fuzzy clustering methods. A number of difficulties with clustering algorithms have been reported in the literature, including the fact that different clustering structures can be achieved by single clustering algorithms with different parameters, or by several algorithms. The clustering ensemble method was introduced to overcome the inherent difficulties with single clustering algorithms. It is the process of combining a set of partitions generated from the same data in order to produce a single improved partition of the data. The main process of the clustering ensemble requires two main steps: generation and consensus. In the generation step, a number of ensemble members are generated from the same data, which are then combined using a consensus function in the consensus step.

This review of the related work has indicated that the consensus function is the key component in a clustering ensemble as it determines whether an ensemble is successful or not. Some common consensus functions in the literature were reviewed in this chapter. They are based on how the clustering ensemble problem is represented or on applying well-known mathematical concepts to the problem. A commonly used one is based on the object pairwise similarity (Co-association matrix). Moreover, a number of clustering ensemble applications were reviewed in this chapter, which include the privacy-preserving clustering application, knowledge reuse and multi-view application. In conclusion, through the review in this chapter, some important research issues are identified, which will be investigated in this research.

Chapter 3

Research Methodology

This chapter describes our research methodology in five sections. Section 3.1 introduces the general clustering ensemble framework. Section 3.2 explains the strategy we used throughout the thesis to test the effectiveness of our proposed clustering ensemble methods, which include the experimental study design, the data used and evaluation measures used. Section 3.3 explains the strategy we used to investigate the diversity and its relationship with the ensemble performance. Section 3.4 introduces the implementation of our methods and the tools used in this thesis. Finally, Section 3.5 summarises this chapter.

3.1 The Clustering Ensemble Framework

For a dataset of n objects: $X = \{x_1, x_2, \ldots, x_n\}$, let $P_q = \{c_1^q, c_2^q, \ldots, c_{k_q}^q\}$ be a clustering result of k_q clusters produced by a clustering algorithm as the q^{th} partition, so that $c_i^q \cap c_j^q = \emptyset$ and $\cup_{j=1}^{k_q} c_j^q = X$. A clustering ensemble Φ can then be built with m members, $\Gamma = \{P_1, P_2, P_3, \ldots, P_m\}$ and a consensus function CF is denoted by $\Phi(CF, \Gamma) = CF(P_1, P_2, P_3, \ldots, P_m) = CF(\Gamma)$.

It should be noted that the members may not necessarily have the same number of clusters in their partitions, i.e., k_q may not be equal to a pre-set value k.

The problem of a clustering ensemble is to find a partition P^* of dataset X by combining the ensemble members $\{P_1, P_2, P_3, \ldots, P_m\}$ with CF without accessing the original features, so that P^* is better in terms of consistency and quality than the individual members in the ensemble.

The quality of the ensemble $Q(\Phi)$ can be defined as a non-linear function of a number of factors, which can be denoted by:

$$Q(\Phi) = f(Q(\Gamma), DV, CF, m) \tag{3.1}$$

Where $Q(\Gamma)$ is the quality of the individual members $\{P_1, P_2, P_3, \ldots, P_m\}$ in ensemble Φ, m is the total number of members, and DV is the diversity of the ensemble.

Figure 3.1 shows that the generic clustering ensemble framework consists of three components: ensemble member generation, consensus function and evaluation, which operate in three consecutive phases. As we can see, the input of the clustering ensemble framework is a given dataset to be clustered, and the output is the clustering result of this dataset, which we call the final clustering result P^*.

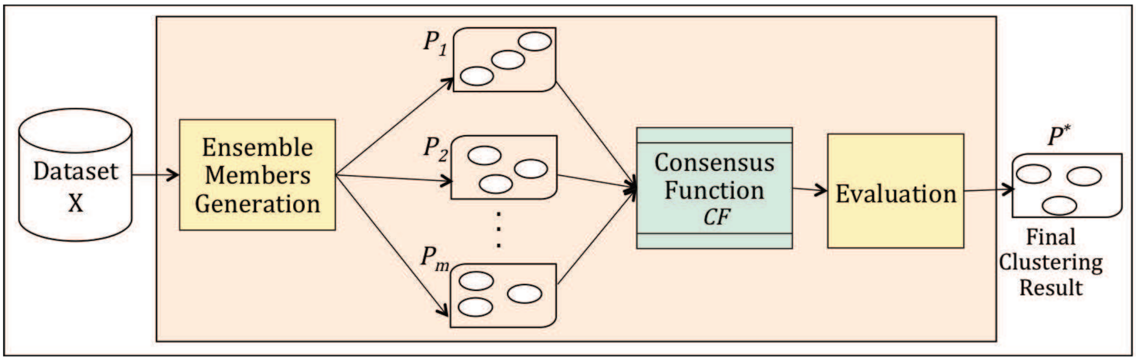


Figure 3.1: A Clustering Ensemble Framework.

Ensemble Member Generation Phase This phase aims to generate m members, using the provided dataset as input. As seen in the previous chapter, there are several techniques that can be used to produce ensemble members. However, in the literature, there is no single clustering algorithm that is universally used and there are no generally agreed criteria for selecting the most suitable algorithms. In this case, it is better to apply the principle of Occam's Razor [9] and choose the one with the greatest simplicity and efficiency, if there is no prior specific knowledge on a given problem. This is why we decided to use two simple widely used generation techniques [98, 32, 94, 111]. Details are given in Section 3.2.2.

Consensus Phase Having obtained the ensemble members, we now need to combine these members using a consensus function in order to produce an improved clustering result. It can determine the quality of the final solution directly, thus it is considered the most important component in an ensemble, and that is why it was chosen as the first focus of this research.

In Chapter 4 we propose a consensus function based on object-neighbourhood similarity in order to solve to some extent the problem of uncertain agreement between members. While, in Chapter 5, we introduce two new consensus functions based on cluster similarity which will not require an ordinary cluster algorithm as final step. More details are given in these chapters.

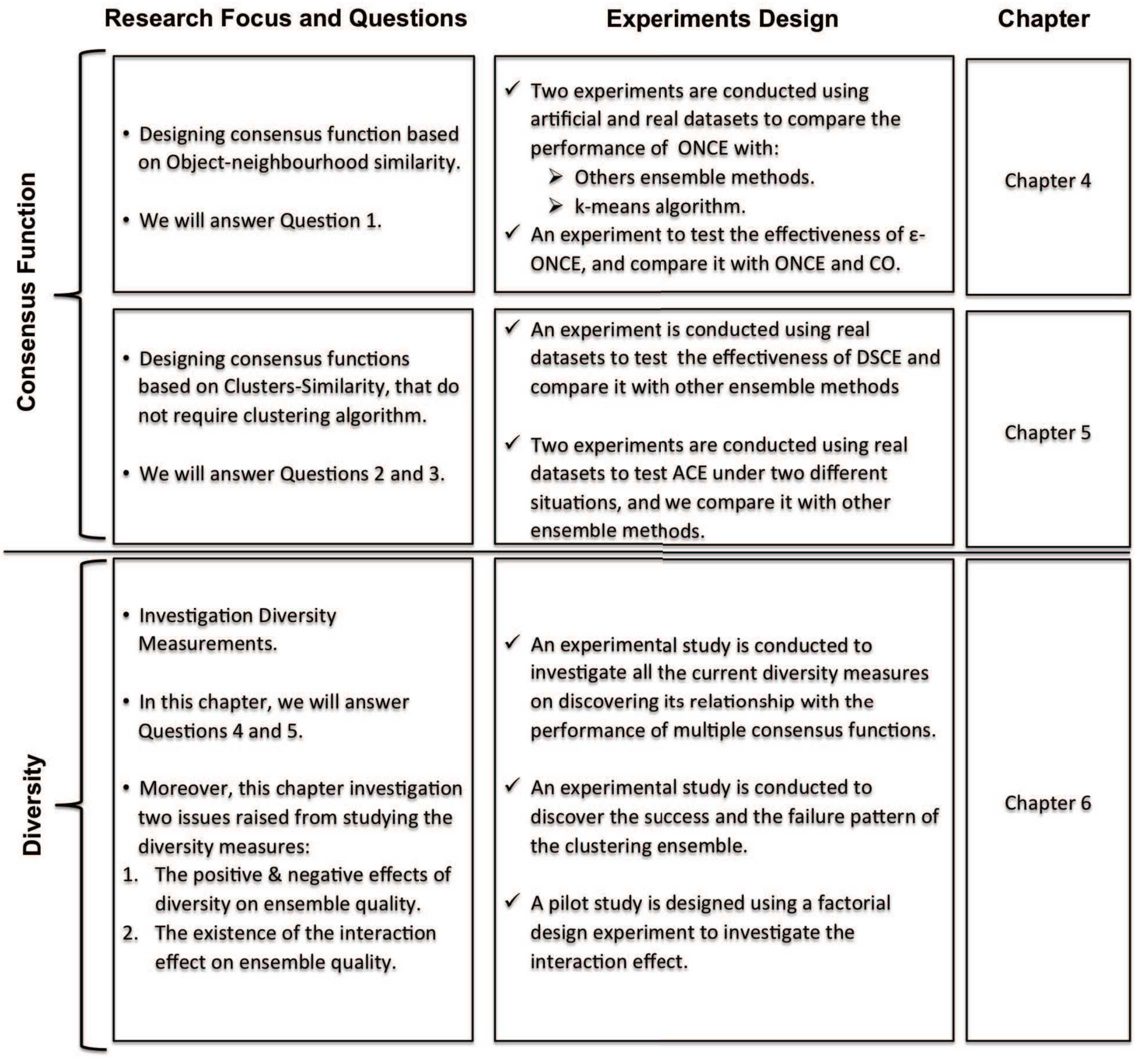
Evaluation Phase In this phase, the aim is to evaluate the final clustering result in terms of quality and consistency. From the clustering point of view, the quality of the clustering result and the consistency of the clustering algorithm can be evaluated either using external information (i.e. a known clustering) if it is available or internal information. In real world applications, it is common not to have any external information. In this case, the quality is defined as how well the clustering result fits the data using internal information such as a measure of cluster cohesion and cluster separation [95]. Whereas, when the external information is available the quality is represented by the degree of similarity between the clustering result and the known clusters of the data (e.g. class labels) [95]. The consistency is defined as the ability that the clustering ensemble method has to produce similar performances on a multiple number of test datasets, and is usually represented by the average of a performance/quality measure and a variance (e.g. standard deviation) [32].

The quality and the consistency of the proposed method will be evaluated using datasets that have a class label and through comparing it with other ensemble methods as well as with single clustering algorithms in order to demonstrate that the clustering ensemble is more reliable and consistent than a single clustering method. Moreover, we will evaluate our proposed consensus function in terms of time complexity.

3.2 Strategies to Test the Effectiveness of the Proposed Consensus Functions

Each of the proposed consensus functions will be tested in a separate experimental study reported in its own chapter. Figure 3.2 summarises the experimental design along with information pertaining to each chapter.

For each experiment, we implement the aforementioned clustering ensemble framework. Then each experiment is repeated 10 times, with different generated members, and the average and the standard deviation of the results of 10 runs are calculated in order to verify the quality and consistency of the ensembles. Moreover, in each experiment we report the average performance for each method across all datasets as well as the standard deviation. More details on the different strategies that are used for each experiment are explained according to each component in the framework as follows: Section 3.2.1, provides detailed information about the datasets that are used in each experiment. Section 3.2.2 reports the ensemble member generation techniques that are used to carry out each experiment. Section 3.2.3 explains our comparison strategy with other clustering ensemble methods. Section 3.2.5 includes details on the used statistical significance test of multiple runs for each experiment.



	Research Focus and Questions	Experiments Design	Chapter
Consensus Function	• Designing consensus function based on Object-neighbourhood similarity. • We will answer Question 1.	✓ Two experiments are conducted using artificial and real datasets to compare the performance of ONCE with: ➢ Others ensemble methods. ➢ k-means algorithm. ✓ An experiment to test the effectiveness of ε-ONCE, and compare it with ONCE and CO.	Chapter 4
	• Designing consensus functions based on Clusters-Similarity, that do not require clustering algorithm. • We will answer Questions 2 and 3.	✓ An experiment is conducted using real datasets to test the effectiveness of DSCE and compare it with other ensemble methods ✓ Two experiments are conducted using real datasets to test ACE under two different situations, and we compare it with other ensemble methods.	Chapter 5
Diversity	• Investigation Diversity Measurements. • In this chapter, we will answer Questions 4 and 5. • Moreover, this chapter investigation two issues raised from studying the diversity measures: 1. The positive & negative effects of diversity on ensemble quality. 2. The existence of the interaction effect on ensemble quality.	✓ An experimental study is conducted to investigate all the current diversity measures on discovering its relationship with the performance of multiple consensus functions. ✓ An experimental study is conducted to discover the success and the failure pattern of the clustering ensemble. ✓ A pilot study is designed using a factorial design experiment to investigate the interaction effect.	Chapter 6

Figure 3.2: Summary of the thesis experimental chapters.

3.2.1 Dataset

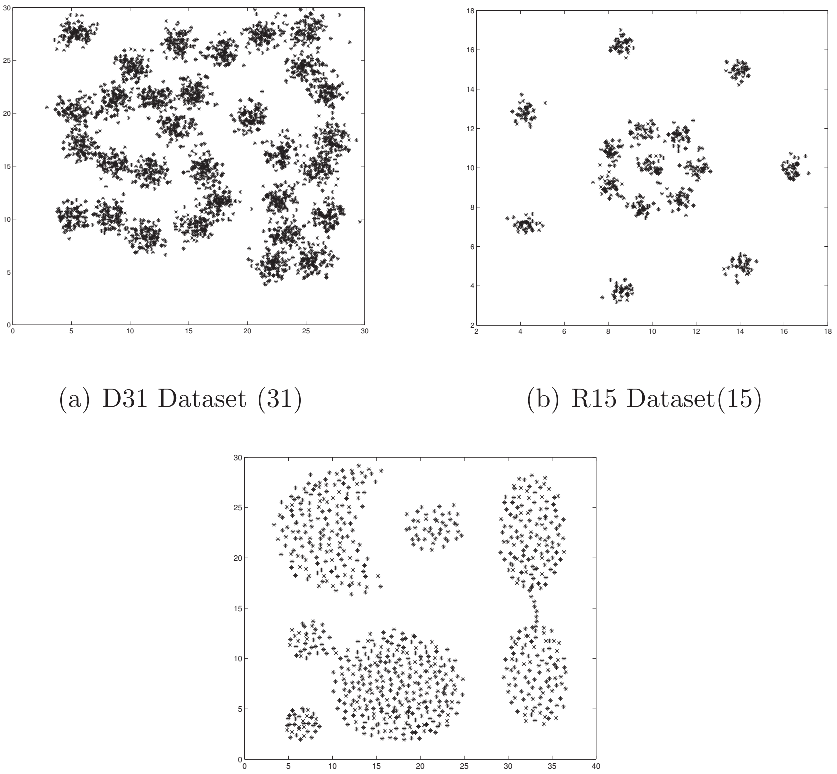
Up to thirteen datasets, as listed in table 3.1, are used to test our proposed consensus functions at different stages of this research. In chapter 4, we use artificial datasets as well as real-world datasets, whereas for the other experimental chapters, we use only real-world datasets. For artificial data, it is easy to obtain the class labels; for real datasets, we use data that already have class labels. We assume that the class labels correspond to clusters in the dataset, which is called ground-truth clustering. Using real benchmark datasets with known class labels has been widely used to evaluate clustering algorithms in the literature. We should mention that these class labels are excluded from the data as the clustering is unsupervised learning and they are used only for the evaluation purpose.

In Chapter 4, the experiments are conducted using 3 artificial datasets, which are shown in Figure 3.3. The D31 and R15 datasets were generated by Veenman et al [100]. D31 contains 31 clusters generated from two-dimensional Gaussian distributions, and R15 has 15 clusters from two-dimensional Gaussian distributions. The aggregation dataset was generated by Gionis et al. [35]; it contains 7 uneven-sized clusters, unequal but with narrow bridges between some clusters. These datasets create difficulties for single clustering algorithms to solve. In the same chapter, the first 8 real datasets are also used to test the proposed consensus functions, which are from the UCI Machine Leaning Repository [77]; these are: Iris, Wine, Thyroid, User modelling (Um), Multiple Features (Mfeatures), Breast Cancer Wisconsin (Bcw), Glass and Contraceptive Method Choice (Cmc). The characteristics of these datasets are given in Table 3.1. In chapters 5, 6 and 6.2, we use only real datasets and as the results in Chapter 4 suggested that the Um and Cmc datasets are not suitable for clustering analysis, we therefore replaced them with the Soybean and Ionosphere datasets in the experiments of other chapters.

As we can see from Table 3.1, three datasets have been modified: Um, Bcw and Ionosphere. Um and Bcw have missing attribute values in some objects which we have removed, and we also removed the second attribute in Ionosphere dataset as only a single value (0) was present for it.

3.2.2 Ensemble Member Generation Techniques Used

For the Ensemble Member Generation Phase, in the experiment in Chapter 4, we use heterogeneous generation techniques, by using different clustering algorithms to generate 7 ensemble members with the pre-defined k value (number of clusters) for each dataset. These are: *k-means*, agglomerative hierarchical clustering using Single



(a) D31 Dataset (31)

(b) R15 Dataset(15)

(c) Aggregation Dataset (7)

Figure 3.3: Three artificial datasets are used in this study. The number of clusters is given in parentheses.

and Average linkage, *k-medoids* and *c-means*[1] as well as *kernel k-means* [91][2] and the Normalised cut algorithm [92][3].

However, in Chapter 5, we implement mixed heuristics generation techniques, precisely the same techniques used by Ren et al. [81] to generate 10 members. Thus, we use *k-means* to generate 5 members with a random sampling of 70% of the data, and we calculate the Euclidean distance between the remaining objects and the cluster centres and assign them to the closest cluster. For each of the remaining members, we use *k-means* on 70% of randomly selected features.

We set k value (number of clusters) equal to the pre-defined cluster (class) value

[1] We use the MATLAB Statistics Toolbox for these algorithms

[2] We use the code available at `http://www.mathworks.co.uk/matlabcentral/fileexchange/26182-kernel-k-means/content/knkmeans.m`

[3] We use the code available at `http://www.cis.upenn.edu/~jshi/software/`

Table 3.1: Details of datasets.

Dataset	# Objects	# Features	# Cluster	Dataset type	Modified
D31	3100	2	31	Artificial	No
R15	600	2	15	Artificial	No
Aggregation	788	2	7	Artificial	No
Iris	150	4	3	Real	No
Wine	178	13	3	Real	No
Thyroid	215	5	3	Real	No
Um	399	5	4	Real	Yes
Mfeatures	2000	2	10	Real	No
Glass	214	9	6	Real	No
Bcw	683	9	2	Real	Yes
Cmc	1473	9	3	Real	No
Soybean	47	35	4	Real	No
Ionosphere	351	34	2	Real	Yes

for each dataset, in all the experiments, except in one experiment in chapter 5, where we set a different k for each member chosen randomly from the interval $[k-2, k+2]$.

3.2.3 Comparison Strategy

We compare our proposed consensus functions with other competitive clustering ensemble methods. In the experiment in Chapter 4, we compare our proposed consensus function with other consensus functions which are also an object pairwise similarity based approach including the Co-assoication (CO) [32] and the recent approaches, which are the connected-Triple based similarity (CTS) matrix, the SimRank-based similarity (SRS) matrix [49] and the approximate SimRank-based similarity (ASRS) matrix [52]. As these consensus functions require a clustering algorithm to be used as a final step, in the first experiment we use three different hierarchical clustering methods: Single (Si), Complete (Cm) and Average (Av) Linkage to compare between them, and for all the following experiments we use the one that achieve better performance. In Chapter 4, we also compare the proposed consensus function with *k-means*, as it is the most well-known clustering algorithm in the literature, and the aim is to find out whether a clustering ensemble is more effective than a single clustering algorithm.

In Chapter 5, we compare the proposed consensus functions with CO [32] (using the Average linkage method), DICLENS [73] and MCLA [94]. Moreover, we compare them with the one proposed in Chapter 4. The CO and MCLA are state-of-the-art clustering ensemble techniques, and they were early successful techniques developed in the clustering ensemble area. CO [32] has around 774 citations, while MCLA [94] has around 2717 citations, according to Google Scholar. DICLENS is the most recent one, and its authors claim that DICLENS outperforms state-of-the-art clustering methods, including CO and MCLA [73].

3.2.4 Evaluation Measures Used

In the evaluation phase, we evaluate the performance of the final clustering results in terms of quality and consistency using the external validation method, and in particular we use the Normalised Mutual Information (NMI) [94] and the Adjust Rand Index (ARI) [47]. When ARI and NMI are applied to evaluate the clustering results, one of the clustering partitions should be the ground "true" partition of the data, which in practice, is normally assumed to be the class labels as there are no other true answers that can be used to verify the quality (accuracy) of the clustering result. The other partition is the clustering result of the ensemble that needs to be evaluated P^*. In Chapter 2, we described how these indices are calculated.

3.2.5 Tests of Statistical Significance

In order to assess the performance of the proposed method in terms of being significantly better or worse than other methods, statistical analysis is necessary. Generally speaking, statistical analysis has been widely used in classification research to assess the performance of different classifiers, but it has not been widely used in clustering analysis research. Recently, researchers have realised the importance of using statistical analysis in clustering analysis research and clustering ensemble, for example, Kuncheva et al.[61], Fern and Lin [28] and Azimi and Fern[7].

According to the recommendations of Demšar [18], we consider the non-parametric testing approach due to the fact that parametric tests, such as t-test, assume that the data are drawn from the normal distribution or homogeneity of variance. Although these tests have been designed for comparing multiple classifiers and have been widely used in supervised learning, we consider the non-parametric approach for comparing the clustering ensemble algorithms as clustering shares a number of key similarities with supervised learning.

To check whether all the results obtained by a number of clustering ensemble algorithms present any equality, we use the Iman-Davenport test proposed by Iman and Davenport [53], who derived a correct measure F of the Friedman test X_F [33], which been shown to be undesirably conservative.

To demonstrate how these tests are implemented, let us run a number of clustering ensemble methods g using t datasets, and the quality of the result is measured using the NMI or ARI indices. So, given a t by g matrix D of quality, the first stage is to rank the competing algorithms for each dataset recorded in the matrix R, where $R_{i,j}$ is the rank of the j^{th} algorithms on the i^{th} dataset. For those algorithms that have equal quality, the average rank is obtained. Then the mean rank for each algorithm is obtained as $R_j = \sum\limits_{i=1}^{t} \frac{R_{i,j}}{t}$. Under the null hypothesis that the mean ranks are equal for all the chosen methods, the Friedman test score X_F is defined as:

$$X_F = \frac{12t}{g(g+1)} \left[\sum_j R_j^2 - \frac{g(g+1)^2}{4} \right] \tag{3.2}$$

And the Iman-Davenport test F is computed by:

$$F = \frac{(t-1)X_F}{t(g-1) - X_F} \tag{3.3}$$

According to the suggestion of Demšar [18], if there are statistically significant differences in the performance of compared clustering ensemble methods, we can proceed with the Nemenyi test as a post hoc test for a pairwise comparison, to discover where the differences lie. If the corresponding average ranks differ by at least the critical difference $CD = q_\alpha\sqrt{\frac{g(g+1)}{6t}}$, where q_α is based on the studentised range statistic, it is said that the performance of two ensembles is significantly different, as we reject the null hypothesis using the Iman-Davenport test.

In summary, to test the significance between multiple clustering ensembles over a number of datasets, we use the Iman-Davenport test with post-hoc Nemenyi test at a significance level of 0.1.

3.3 Strategies Used to Investigate Diversity

As we saw in the first chapter, two questions (4 and 5) are asked on the diversity issue. In Chapter 2, we answered question 4 by reviewing the literature on all existing diversity definitions in the context of the clustering ensemble. In Chapter 6, we answered question 5 by designing an experimental study to investigate the relationship between diversity and ensemble performance, using all of the existing diversity measures and using a number of different consensus functions — CO, MCLA and the one proposed in this research. The experiment set-up and the results are given in Section 6.1.

Furthermore, we investigate two issues raised from our experimental study on diversity (in Section 6.1) by experimentally studying them. These issues are an analysis of the positive and negative effects of diversity on ensemble performance, and studying the interaction between members' qualities and diversity. In the following sections we briefly review them and the strategy we use to investigate them.

3.3.1 Analysis of the Positive and Negative Effects of Diversity on the Ensemble Performance

In the classification ensemble, it has been shown that diversity is not always a beneficial factor for the ensemble performance [63]. In relation to the ensemble accuracy, Kuncheva et al. [63] derived a functional relationship between the ensemble accuracy (given by the majority voting method) and the diversity (measured by pairwise dependence Q-statistics), and accordingly two different effects of diversity were discovered. These were represented by two extreme patterns: the “pattern of success”and the “pattern of failure”, in terms of the voting combinations that the ensemble can have. In the success pattern, the ensemble accuracy (majority voting) is a monotone decreasing function of Q-statistics, while the failure pattern is a monotone increasing function of Q-statistics. They showed that diversity can have a positive effect on the ensemble performance, such as the case in the pattern of success (best pattern), but it can also have a negative effect on the ensemble performance, such as the case in the pattern of failure (worst pattern).

Different effects of diversity are also found in the clustering ensemble context, as some results in Section 6.1 showed that for some datasets there are some “Ups” (positive diversity effects) and “Downs” (negative diversity effects) for the quality of the ensemble. Moreover, these two effects were also reported in the literature, in particular in [83] and [39], where they did not investigate them deeply. Therefore, in Chapter 6, we formally define these two different effects and we conduct an experiment in order to find out the characteristics of these ensemble combination patterns to avoid the negative effect of diversity.

3.3.2 Studying the Interaction between Members’ Qualities and Diversity

The results in Section 6.1, as well as the analysis of the first issue in Section 6.2.1, show that an interaction may exist between the diversity and the members’ qualities.

If such an interaction exists, then the effect on the ensemble performance might be determined jointly by them. This leads us to implement a factorial design experimental study to investigate the interaction between the diversity and the members' quality, and if such an interaction exists we will study the impact of the interaction between them on the ensemble performance. The following section gives a brief introduction to the Factorial Design Experiment, and in Chapter 6 (Section 6.2.2) the factorial experiment study is conducted, and its results are reported, along with the interpretation of the results.

Factorial Design Experiment

Generally, a factorial design experiment is used to study the effects of one or more factors (independent variables) on one or more responses (dependent variables). It is therefore designed to address more complex problems than an experimental study of one variable at time. In a factorial experiment, each factor can be subdivided into different levels, and it is conducted under a set of conditions, where each condition is a combination of two levels from different factors. It is possible to determine the effect of each factor alone (main effect), which is a consistent trend among the different levels of a factor, and the effect of both factors in a combination (interaction). The interaction is defined as how the effect of one of the independent variables differs according to the level of the second independent variable [75]. There are two factor categories, within-subject and between-subject, where subject means the thing that is being experimented on. In our case, the subject is the set of the generated members in different runs in the experiment. In the within-subject factor, the same subjects are used in different conditions in the experiment, while in the between-subject factor, a different set of subjects is used for each condition in the experiment [16].

Analysing a factorial experiment requires a statistical analysis technique, and the most common used is the Analysis of Variance (ANOVA) [71]. ANOVA is a set of statistical methods used to test the general differences among the means of two or more factors, and ANOVA tests the null hypothesis that all the factor means examined are equal. When this null hypothesis is rejected at a chosen significant level, it signifies that at least one mean is different from at least one other mean, but ANOVA does not reveal where the differences occur among the levels of one or more factors. Different experimental designs require different ANOVA approaches; an experiment designed with one factor requires one-way ANOVA, while an experiment with two factors requires two-way ANOVA for the analysis. The latter is the type of ANOVA used in our experiment in Section 6.2.2.

However, ANOVA makes a number of assumptions about the distribution of the dependent variable for each level of the independent variable, and these should be checked to ensure the validity of the ANOVA. The main assumptions are: the normality of the dependent variable distribution, and the homogeneity of variances (the variances of the dependent variable in each combination are the same) [71].

3.4 Research Tools and Implementation

The proposed clustering ensemble framework is implemented using the high-level technical computing language, *MATLAB* (Matrix Laboratory) version $R2012b$ on Apple Macintosh computer 2.3 GHz Intel Core $i5$ with 8 GB memory.

We use as many available clustering algorithms and evaluation methods as possible to build our framework, in order to implement a complete clustering ensemble system. The input of our system is: X the dataset, k the number of clusters, the ensemble generation technique type (which is either heterogeneous or mixed heuristic in our experiments), m the number of members, P^t is the ground-truth partition.

The procedure of our framework is as follow:

1. Select ensemble generation technique type to be used.
2. Generate m members and store them in Γ matrix.
3. Apply the consensus functions ONCE, $\mathcal{E}$-ONCE, DSCE and ACE (more de-

tails on the implementation of each proposed consensus function are given in Chapters 4 and 5).

4. Obtain the final clustering results P^*.
5. Evaluate P^* and compare it with the ground-truth clustering using an external clustering validation measures. We run NMI[4] and ARI[5].

For the second central part of this research, in Chapter 6 we implement all the diversity measures, that we use. In Chapter 6.2, the factorial experiment is carried out using *Minitab* software.

[4] We use the code available at `http://strehl.com`

[5] We use the code available at `http://www.pi-sigma.info`

3.5 Summary

In this chapter, we introduced the clustering ensemble framework, which consists of three phases:

1. Ensemble member generation phase;
2. Consensus phase;
3. Evaluation phase.

In the first phase, the aim is to generate a number of members, which are combined in the consensus phase. The latter phase is commonly known as the consensus function. In the evaluation phase, the aim is to evaluate the quality of the final clustering result of our method in terms of quality as well as consistency and time complexity, relative to our proposed consensus functions and other state-of-the-art clustering ensemble methods.

We described our strategies that are used to test the effectiveness of our proposed consensus functions, including a description about the experimental design along with the details of the datasets that are used and the evaluation measures. Furthermore, we described our strategies to investigate the diversity in terms of its relation with the ensemble performance. Finally, we also explained our research implementation and tools that are used throughout this thesis.

Chapter 4

Object-Neighbourhood Clustering Ensemble

As explained in Chapter 2, the consensus function is the main task in the clustering ensemble framework and its effectiveness determines whether an ensemble is successful or not. In the literature, the most popular method is the Co-association matrix (CO), because it is easy to understand and simple to implement. However, in the situation where there are uncertain agreements between the ensemble members, these could not be resolve by the CO as it only takes into account the object pairwise similarity. We are of the view that the ensemble members have other useful information that can be integrated into calculating the similarity in order to produce improved clustering ensemble results. One such type of information is the object neighbourhood. Thus, in this chapter we investigate how to use the object-neighbourhood information when designing the consensus function, with the intention of resolving the uncertain agreements between ensemble members. This chapter is divided into four main sections.

In Section 4.1, we propose an Object-Neighbourhood Clustering Ensemble (ONCE) method. In Section 4.2, we develop ONCE further by considering the $\mathcal{E}$-neighbourhood region between a pair of objects. In Section 4.3, we compare the performance of ONCE with other consensus functions using a number of real-world datasets. Section 4.4, summarises the whole chapter.

4.1 Object-Neighbourhood Clustering Ensemble

Firstly, before presenting our new algorithm, it is useful to analyse the issue with the Co-association matrix (CO). CO calculates the probability of a given object pair being clustered together by all members (as shown in equation 2.3 listed in Section 2.2.3.2), or in other words, it measures the degree of agreement between ensemble members when clustering a given pair of objects together. $CO \in [0, 1]$ as shown in Figure 4.1. Because of this it is inevitable that some uncertain situation will occur when the values of the CO are around the middle. In the case of $CO(x_i, x_j) \approx 1$, this means that objects x_i and x_j are placed in the same cluster in most of the ensemble members, and therefore the degree of probability is high, which means that the pair is certain, or almost certain, to be clustered together; we call this a certain similar pair. In the case of $CO(x_i, x_j) \approx 0$, this means that x_i and x_j are placed in different clusters by most of the ensemble members, so the degree of probability is low, which means that the pair is almost certain to be clustered in different clusters; we call this a certain dissimilar pair. However, in the case of $CO(x_i, x_j) \approx 0.5$, it means that roughly half of the members placed x_i and x_j in the same cluster, whereas the other half placed them in different clusters, creating the most uncertain agreement between the members; thus, the degree of probability on how to cluster this pair is uncertain, and we call this an uncertain pair.

Generally speaking, in a dataset, these uncertain object pairs are usually objects that are hard to cluster. These hard objects might be located on or around the boundary of clusters, or be overlapping between the clusters in the problem space. In that case, when we calculate the CO matrix, it is highly likely to produce uncertain pairs. These uncertain pairs cause problems with generating reliable consensus clustering results. Therefore, the CO needs to be modified in order to produce consistent and reliable clustering results, and we assume that taking the relationships between the pairs of objects (as well as their neighbourhood relationships) into consideration could, to some extent, overcome the problem of uncertainty.

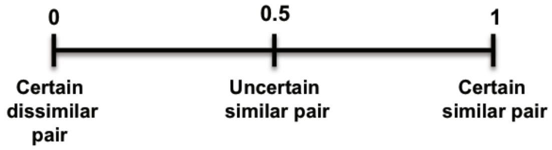


Figure 4.1: The different types of objects pairs and their similarity value.

4.1.1 Definition of Object-Neighbourhood Similarity

In cluster analysis the concept of the neighbourhood space of an object is not new; it has been successfully applied to a number of clustering algorithms such as DBSCAN [24] and the ROCK clustering algorithm [38]. The neighbourhood is the region in the data space covering an object in question. Therefore, objects in the same cluster are all considered to be in the same neighbourhood region, and objects in different clusters are not considered to be in the same neighbourhood region.

The key idea of our similarity definition is derived from Jarvis and Patrick [56], who defined the similarity between object pairs as the number of nearest neighbours that the pair shares, as long as the objects themselves belong to their common neighbourhood. They call it Shared Nearest Neighbour (SNN).

Definition 1. *The common neighbours to a pair of objects are the other objects in the same cluster as the pair itself.*

Thus, the more common neighbours that two objects have, the more similar they are. The difference between our similarity measure and Jarvis and Patrick's [56] measure is that the latter is based on the number of shared nearest neighbours, determined by any similarity/distance measure, whereas we take the similarity score of all the shared "common" neighbours into consideration when we calculate the similarity between pairs of objects.

Assume that x_a is a common neighbour to x_i and x_j, and that Z is the set of all common neighbours between x_i and x_j. For each pair of objects, the average similarity, $B(x_i, x_j)$, of their common neighbours is defined as follows:

$$B(x_i, x_j) = \frac{\sum_{x_a \in Z} \Big(CO(x_a, x_i) + CO(x_a, x_j) \Big)}{2 \mid Z(x_i, x_j) \mid} \tag{4.1}$$

Where $CO(x_a, x_i)$ and $CO(x_a, x_j)$ are the neighbourhood association of x_i and x_j to their common neighbour x_a respectively and can be calculated from the CO matrix (equation 2.3).

$B(x_i, x_j) \in [0, 1]$; when $B(x_i, x_j) = 0$ it means that there are no common neighbours between x_i and x_j, and when $B(x_i, x_j) = 1$ it means that x_i, x_j and their common neighbours are placed in the same cluster by all the ensemble members. When $0 < B(x_i, x_j) < 1$, it means that x_i and x_j have some common neighbours placed in the same cluster by some members in the ensemble.

Then, by adding the average neighbourhood similarity B to CO, which is the similarity between the pair of objects themselves, we obtain the overall similarity W within the range $[0, 2]$. It is defined in the following equation:

$$W(x_i, x_j) = B(x_i, x_j) + CO(x_i, x_j) \tag{4.2}$$

After computing W for all pairs of objects in X and obtaining the full matrix, we scale W by dividing each cell by the maximum value in W, which is W_{max}. This is done in order to scale W to the $[0, 1]$ range. W_{max} takes a value up to 2. However, our similarity definition W has the following properties:

- It is non-negative, and takes a value in the interval $[0, 1]$.
- When $W(x_i, x_j) = 0$ the two objects x_i and x_j are completely different, while when the two objects are identical it takes the value of 1.
- It is symmetric, $W(x_i, x_j) = W(x_j, x_i)$.

W takes the neighbourhood similarity into consideration as well as the object pairwise similarity to enhance and solve to some extent the problem of uncertain object pairs. So, we add the neighbourhood similarity B to CO to obtain the overall similarity W, which always increases compared to CO, or is equal to CO when there are no shared neighbours between a pair of objects, $B = 0$. In some cases, the pairwise relationship does not exist between a pair of objects, and there is no agreement between the ensemble members about this pair. But, they may share a number of neighbours and taking their similarity into consideration should indirectly uncover their similarity, and the W value in this case will be greater than 0.

It is also worth noting, however, that taking the neighbourhood similarity into consideration may also affect the similarity value of certain object pairs, which may decrease after we normalise W. In this case, the certain pair of objects may become uncertain. We will consider this problem in our future work as improvement to our definition.

4.1.2 ONCE Algorithm

Having obtained our neighbourhood similarity matrix W, we then convert it to a distance matrix using the common formula (Distance = 1- Similarity) in order to apply hierarchical clustering algorithms (Single, Complete and Average Linkage) to obtain the final clustering result. We call this algorithm Object-Neighbourhood-based Clustering Ensemble (ONCE); the details of the algorithm are as follows:

4.1.3 Illustrative Example

We generated a simple dataset to illustrate how uncertainty do affect the CO and not affect ONCE. For this purpose, first we identified three parameters which should be controlled when generating an artificial dataset; these are the number of objects in the dataset n, the cluster number k and the degree of separation between clusters. We used a R package called “clusterGeneration”. This package was written by Qiu

Algorithm 2: ONCE Algorithm.

```
Input: $\Gamma = \{P_1, P_2, P_3, \ldots, P_m\}$, $m$ number of clustering members
Output: Partition of Dataset $X = \{x_1, x_2, x_3, \ldots, x_n\}$
for each $i = 1 : n$ do
    for each $j = i + 1 : n$ do
        Calculate $CO$ similarity for pair $(x_i, x_j)$ using equation 2.3

for each $i = 1 : n$ do
    for each $j = i + 1 : n$ do
        Z ← Find common neighbours for pair $(x_i, x_j)$
        if # of element in $Z > 0$ then
            Calculate the average similarity of the common neighbours for pair
            $(x_i, x_j)$ using equation 4.1
            Calculate $W(x_i, x_j)$ using equation 4.2
        else
            $W(x_i, x_j) = CO(x_i, x_j)$
Scale $W$: $W/W_{max}$
Convert the similarity matrix $W$ into distance matrix $W'$
Obtain the final clustering results by applying a hierarchical clustering
algorithm to $W'$
```

and Joe [79], and the key concept of this package is to generate clusters with a specified degree of separation, which is based on the separation index proposed by Qiu and Joe [78]. It measures the separation between the cluster and its nearest cluster, and it takes values within the interval $[-1, 1)$, where the closer a separation value is to 1, the more separated the clusters are. Therefore, we used this package and in particular we used the "genRandomClust" function and we set $n = 200$ and $k = 2$ (100 objects each), and the separation index equal to 0.3. Using these parameters we generated a dataset as shown in Figure 4.2.

Looking closely at the dataset, we can clearly see that there is an overlap around the boundary of the two clusters, so it is very difficult for the clustering algorithm to obtain the true labels and it may be impossible to distinguish these two clusters. Then, we generated 7 members using a heterogeneous ensemble (details are given in Section 3.2.2), and we also ran the experiment using *k-means* as a baseline algorithm. For only the objects placed around the boundary of the two clusters, we modified their clustering results (labels) by making half of the members classify

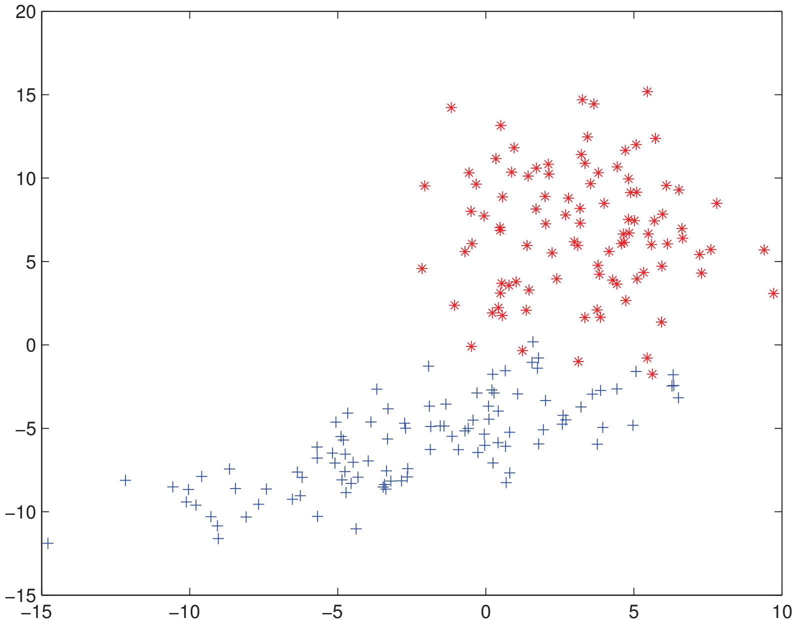


Figure 4.2: The generated artificial dataset consists of 2 clusters and 200 objects.

them correctly, while the other half did not (there are in total 14 objects). This has been done in order to ensure that these objects, which are hard to cluster, will represent the uncertainty when we calculate the pairwise similarity matrices. By controlling these objects we aim to illustrate and prove a situation where ONCE is able to produce a good quality clustering result, while CO is not. After that, we ran ONCE and CO, and then we applied hierarchical clustering algorithms (Single, Complete and Average Linkage) to obtain the final clustering results for each method. Finally, all the clustering results were evaluated using the NMI and ARI indices.

Table 4.1 shows the results of NMI and ARI indices for the CO, ONCE and *k-means* algorithms all by using Single, Complete and Average linkage, and Figure 4.3 shows the clustering label results of the compared methods. It is observed that ONCE-Si, ONCE-Cm and ONCE-Av achieved perfect clustering results (their qualities were equal to the quality of the true label measured by NMI and ARI). On the other hand, CO-Si, CO-Cm and CO-Av achieved lower results, and among them CO-Si achieved the worst performance. It is noteworthy that whatever linkage methods are used with ONCE they have no effect on the clustering result.

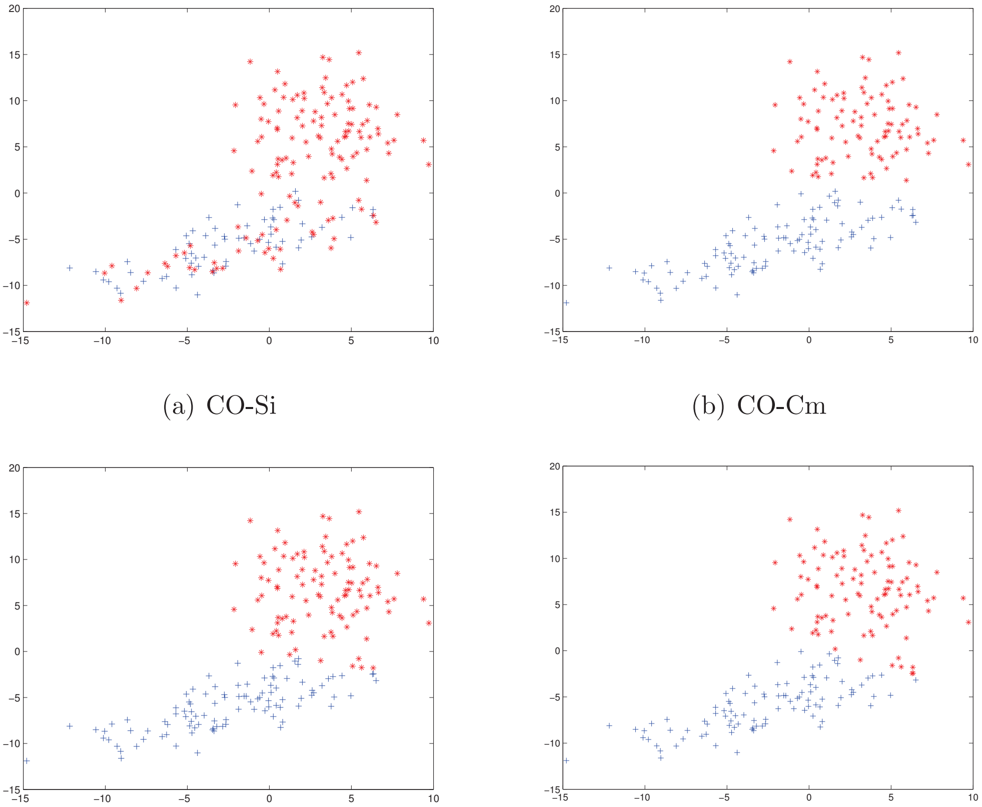
Figure 4.4 shows the heat map of the CO and ONCE matrices which highlight the clustering structure. The colour scheme ranges from strong pink ($CO(x_i, x_j) = 1$) to light green ($CO(x_i, x_j) = 0$), corresponding to the magnitude of similarity between a pair of objects. We see that ONCE discovered some hidden similarity values compared to CO, and it reveals the structure of the two clusters (two blocks in strong pink) with higher similarity values between objects pairs than CO. It is also noticeable that there is an increase in similarity values for certain dissimilar object pairs (colour change from green in CO to blue in ONCE), that is a hidden similarity, and for some uncertain object pairs (colour change from blue in CO to light pink in ONCE) in the ONCE matrix compared to CO.

In summary, the results on this simple dataset confirm that:

1. CO is affected by uncertain agreement between the members on classifying hard objects as there are more blue and green pairs.
2. Relying only on pairwise object information is not enough to generate a reliable clustering result in this situation.

Table 4.1: The quality of the clustering results of CO and ONCE algorithms using Single, Complete and Average Linkage methods as well as the quality of the *k-means* clustering result on the artificial dataset measured by NMI and ARI.

Clustering Ensemble Algorithm	**NMI**	ARI
ONCE-Si	1.00	1.00
CO-Si	0.433	0.37
ONCE-Cm	1.00	1.00
CO-Cm	0.856	0.90
ONCE-Av	1.00	1.00
CO-Av	0.902	0.94
k-means	0.786	0.864



(a) CO-Si

(b) CO-Cm

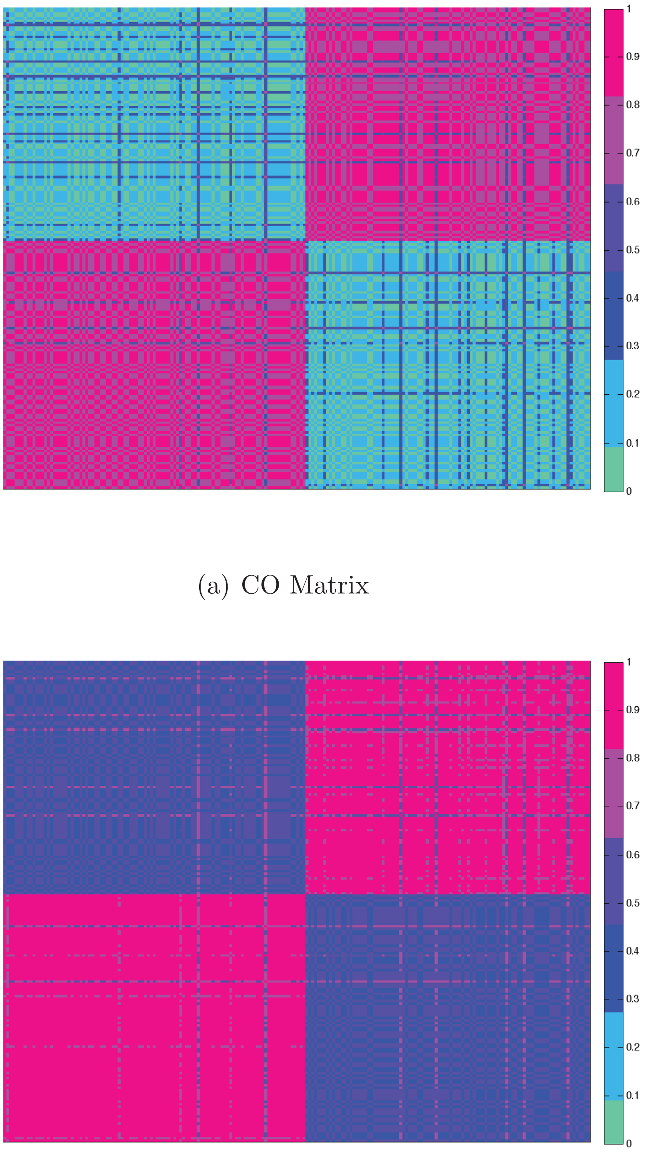
(c) CO-Av

(d) *k-means*

Figure 4.3: The Cluster labels results of CO-Si, CO-Cm, CO-Av, $\mathcal{E}$-ONCE-Cm clustering ensemble methods and *k-means* algorithm.

4.1.4 Experimental Results

As we have mentioned in Section 3.2.1, this experiment was conducted using 3 artificial datasets and 8 real datasets (their details are given in Table 3.1). For each given dataset, the framework described in Chapter 3 was used to carry out the experiments in three phases: the generation phase, the consensus phase and the evaluation phase. In the first phase, as we said, we used the heterogeneous generation techniques to generate 7 members, and in the consensus function, we used the ONCE algorithm to generate the neighbourhood similarity matrix. We also computed the Co-association matrix and the final clustering result was obtained using three different hierarchical clustering algorithms: Single (Si), Complete (Cm) and Average (Av) Linkage over the two matrices. (More details of this experimental



(a) CO Matrix

(b) ONCE Matrix

Figure 4.4: The heat map of the CO and ONCE matrices calculated using the artificial dataset.

design are given in Section 3.2).

Tables 4.2 and 4.4 present the results of the NMI and ARI respectively; each entry in these tables represents the average quality of ten runs, followed by the standard deviation. The results of our method were compared with the CO and in order to make a fair comparison we compared the result of two like-for-like methods.

In other words, the two methods used the same linkage method, so we compared the result of the Single linkage method over the ONCE matrix (ONCE-Si) with the result of the Single linkage method over the Co-association matrix (CO-Si), and then ONCE-Cm with CO-Cm, where the Complete linkage method was used over both matrices, and ONCE-Av with CO-Av, where the Average linkage method was used.

The bold value in each row shows the best result comparing like-for-like methods, and the underlined value represents the highest quality for each dataset. The last two rows show the average quality for each algorithm over all the datasets, and the Wins (W)/Ties (T)/Losses (L) row counts the number of W/T/L (in terms of quality) comparing the two like-for-like methods. Table 4.3 shows W/T/L (in quality) comparing ONCE, CO, Ave-mem and *k-means* with the highest quality achieved for each dataset. This was done in order to compare ONCE with CO, and to compare the ensemble method with the baseline algorithm as well as with the members average. Briefly, in terms of comparison, when we state that algorithm X is better/worse than algorithm Y, it means that X has a better quality cluster than Y, under the same experimental set-up.

Results obtained by NMI Index: As we can see, the quality of ONCE-Si in most of the datasets was improved, relative to CO-Si; in particular, 8 out of 11 datasets in total were improved in terms of quality, whereas for the remaining datasets, the quality was decreased (these are: R15, Bcw, and Thyroid). In the case of the Aggregation dataset, the quality of ONCE-Si, ONCE-Cm and ONCE-Av were increased, relative to CO-Si, CO-Cm and CO-Av, respectively. On the other hand, for the Um dataset, the quality of ONCE-Cm and ONCE-Av were decreased, and the quality of ONCE-Si was slightly improved. However, in general, this dataset also achieved low quality using *k-means* as well as the member average. We noticed that this is also the case in the Cmc dataset, where we obtained low quality with most of the ensemble methods, and we noticed that the quality for the member average is also very low for *k-means*, which indicates that these datasets are not suitable for

Table 4.2: The average performance of 10 runs of each method for each dataset measured by NMI on 11 datasets. The average performance of each method across 11 datasets and the W/T/L for each ensemble method comparing the two like-for-like methods are included. In each row the bold value represents the highest quality comparing two like-for-like methods (e.g. ONCE-Si and CO-Si), whereas the underlined value represents the highest quality comparing all ensemble methods.

Dataset	**ONCE-Si**	**CO-Si**	**ONCE-Cm**	**CO-Cm**	**ONCE-Av**	**CO-Av**	**Ave-mem**	***k-means***
D31	$\mathbf{0.912} \pm \mathbf{0.013}$	0.911 ± 0.018	$\mathbf{0.961} \pm \mathbf{0.004}$	$\mathbf{0.961} \pm 0.005$	0.961 ± 0.005	$\underline{\mathbf{0.965}} \pm \underline{\mathbf{0.002}}$	0.774 ± 0.328	0.916 ± 0.025
R15	0.989 ± 0.009	$\mathbf{0.991} \pm \mathbf{0.007}$	$\underline{\mathbf{0.994}} \pm \underline{\mathbf{0.000}}$	0.989 ± 0.018	$\underline{\mathbf{0.994}} \pm \underline{\mathbf{0.000}}$	$\underline{\mathbf{0.994}} \pm \underline{\mathbf{0.000}}$	0.850 ± 0.272	0.918 ± 0.037
Aggregation	$\mathbf{0.950} \pm \mathbf{0.002}$	0.935 ± 0.022	$\mathbf{0.974} \pm \mathbf{0.010}$	0.941 ± 0.038	$\underline{\mathbf{0.984}} \pm \underline{\mathbf{0.006}}$	0.967 ± 0.029	0.767 ± 0.341	0.851 ± 0.011
Bcw	$0.026 \pm \mathbf{0.008}$	$\mathbf{0.047} \pm 0.044$	0.457 ± 0.250	$\mathbf{0.702} \pm \mathbf{0.154}$	$\mathbf{0.741} \pm 0.003$	$0.736 \pm \mathbf{0.002}$	0.455 ± 0.341	$\underline{\mathbf{0.748}} \pm \underline{0.000}$
Cmc	$\mathbf{0.028} \pm \mathbf{0.005}$	0.012 ± 0.007	$\underline{\mathbf{0.032}} \pm \mathbf{0.001}$	$\underline{\mathbf{0.032}} \pm \mathbf{0.001}$	$\underline{\mathbf{0.032}} \pm \mathbf{0.001}$	$\underline{\mathbf{0.032}} \pm \mathbf{0.001}$	0.025 ± 0.013	$0.032 \pm \underline{0.000}$
Iris	$\mathbf{0.768} \pm \mathbf{0.027}$	0.733 ± 0.034	$0.766 \pm \mathbf{0.017}$	$\underline{\mathbf{0.774}} \pm 0.021$	$\mathbf{0.771} \pm \mathbf{0.021}$	0.763 ± 0.022	0.630 ± 0.282	0.725 ± 0.070
Glass	$\mathbf{0.394} \pm 0.040$	$0.374 \pm \mathbf{0.037}$	$\underline{\mathbf{0.395}} \pm 0.029$	$0.382 \pm \mathbf{0.021}$	$\mathbf{0.394} \pm \underline{\mathbf{0.008}}$	0.383 ± 0.021	0.366 ± 0.133	0.368 ± 0.024
Um	$\mathbf{0.040} \pm 0.003$	0.039 ± 0.003	0.241 ± 0.080	$\mathbf{0.245} \pm 0.090$	0.290 ± 0.133	$\underline{\mathbf{0.359}} \pm \mathbf{0.102}$	0.176 ± 0.150	0.338 ± 0.052
Wine	$\underline{\mathbf{0.435}} \pm \underline{\mathbf{0.000}}$	0.407 ± 0.109	0.422 ± 0.009	$\mathbf{0.424} \pm \mathbf{0.003}$	$\mathbf{0.434} \pm \mathbf{0.011}$	0.429 ± 0.003	0.321 ± 0.187	0.426 ± 0.031
Mfeatures	$\mathbf{0.319} \pm \mathbf{0.087}$	0.142 ± 0.094	$\mathbf{0.472} \pm \mathbf{0.034}$	0.454 ± 0.031	$\underline{\mathbf{0.479}} \pm \underline{\mathbf{0.001}}$	$\underline{\mathbf{0.479}} \pm 0.002$	0.374 ± 0.230	0.478 ± 0.003
Thyroid	$0.127 \pm \underline{\mathbf{0.074}}$	$\mathbf{0.195} \pm 0.108$	$\underline{\mathbf{0.446}} \pm \mathbf{0.075}$	0.368 ± 0.080	$\mathbf{0.403} \pm 0.096$	$0.358 \pm \mathbf{0.080}$	0.228 ± 0.149	0.423 ± 0.071
Ave-P	**0.454**	0.435	0.560	**0.570**	$\underline{\mathbf{0.589}}$	0.588	0.451	0.566
Ave-C	$\underline{\mathbf{0.024}}$	0.044	0.046	**0.042**	0.026	$\underline{\mathbf{0.024}}$	0.221	0.029
W/T/L	8/0/3	3/0/8	5/2/4	4/2/5	6/3/2	2/3/6	0/0/11	1/1/9

Table 4.3: Counts of the W/T/L for each ensemble method as well as average members and *k-means* comparing with the highest quality achieved for each dataset.

	ONCE	**CO**	**Ave-mem**	***k-means***
W/T/L	4/3/4	3/3/5	0/0/11	1/1/9

clustering analysis (or they may need a special distance/similarity measurement).

In Bcw, the cluster qualities of ONCE-Si and ONCE-Cm were reduced, compared with CO-Si and CO-Cm, respectively, whereas ONCE-Av was improved, compared with CO-Av, in which they scored 0.741 and 0.736, respectively. This is almost as good as the highest quality achieved by *k-means*. We believe that improving the uncertain pairs of objects makes both Single and Complete linkage inappropriate for this dataset; in general, Single linkage with the CO and ONCE matrices achieved very low quality compared to other linkage methods.

However, the greatest improvement resulting from our method was in the Glass dataset, which gave the highest NMI score using the Single, Complete and Average linkage methods, comparing them similar methods with Co-association. This indicates that the uncertain pairs of objects affect the Co-association methods. From the results, it is noted that in the cases of Iris and Mfeatures, the quality of ONCE-Si was improved from 0.733 to 0.768 in Iris and from 0.142 to 0.319 in Mfeatures.

As expected, the ensemble method performs better than a single clustering algorithm in this experiment; it performs better than *k-means* except in the case of the Bcw dataset. In general, this confirms the perception that the performance of the ensemble method is much better than that of a single algorithm. When comparing the consistency, we found the ensemble method to be more reliable than the single algorithm, where the latter achieved higher standard deviation in most of our tested datasets except in Bcw and Cmc.

Furthermore, when comparing the average quality across all the datasets, we observed that ONCE-Si and ONCE-Av outperformed CO-Si and CO-Av respectively. On the other hand, on average, using the Complete linkage with the CO-matrix is slightly better than ONCE-Cm, where the averages are equal to 0.57 for CO-Cm and 0.56 for ONCE-Cm. Comparing the three linkage clustering methods used with our method, it can be observed that the Average linkage performed better than the other two linkage methods as it gave a higher average quality using the ONCE and CO matrices.

Looking to Wins/Ties/Losses, we observe that in total our method wins more often than the CO method with respect to comparing the like-for-like methods. Comparing the highest qualities, our method wins four times and ties three times (two of which were highest qualities), and loses four times across the total of the eleven datasets. The CO wins three times and loses five times, and *k-means* wins once. Finally, we observe that the highest quality is achieved by ONCE in six datasets; these are R15, Aggregation, Wine, Thyroid, Mfeatures and Glass.

Results obtained by ARI Index: As we can see from Table 4.4, similar results were obtained using the ARI index to the results described above. It is noticed that in the Wine dataset the performance of CO-Av was slightly better than the ONCE-Av using the ARI index, and accordingly the average quality over all the tested datasets in CO-Av was slightly better than in ONCE-Av.

In conclusion, the results obtained from the two indices suggested that in most of the datasets the performances of the ONCE-Av and CO-Av were very close to each other. We should mention that the differences in performance are so small that they are only seen in the third decimal place of the results. As we did not test the results of this experiment statistically, these results may not show statistical differences.

However, it is interesting to note that the effect of the uncertain pairs of objects varied with the different datasets. This is due to the fact that not every dataset is affected by uncertain pairs of objects, even though these are in fact hard objects to cluster. This is where results with two datasets are improved: the Iris dataset, which has overlapping clusters, and the Aggregation dataset, which has uneven-sized clusters with difficult boundaries, both were improved. Moreover, the results suggested that the Average linkage method is the most appropriate method to use with ONCE and CO.

Table 4.4: The average performance of ten runs of each method for each dataset measured by ARI on 11 datasets. The average performance of each method across 11 datasets and the W/T/L for each ensemble method comparing the two like-for-like methods are included. In each row the bold value represents the highest quality comparing two like-for-like methods (e.g. ONCE-Si and CO-Si), whereas the underlined value represents the highest quality comparing all ensemble methods.

Dataset	ONCE-Si	CO-Si	ONCE-Cm	CO-Cm	ONCE-Av	CO-Av	Ave-mem	*k-means*
D31	**0.693** ± **0.051**	0.679 ± 0.078	**0.929** ± 0.017	0.926 ± 0.021	0.918 ± 0.023	**0.945** ± 0.012	0.635 ± 0.381	0.788 ± 0.083
R15	0.972 ± 0.034	**0.979** ± **0.029**	**0.993** ± 0.000	0.971 ± 0.068	**0.993** ± 0.000	**0.993** ± 0.000	0.742 ± 0.341	0.796 ± 0.089
Aggregation	**0.911** ± **0.002**	0.884 ± 0.042	**0.978** ± 0.010	0.916 ± 0.074	**0.988** ± 0.006	0.954 ± 0.066	0.700 ± 0.337	0.737 ± 0.022
Bcw	0.005 ± **0.003**	**0.019** ± 0.026	0.390 ± 0.393	**0.772** ± 0.241	**0.841** ± 0.003	0.836 ± **0.002**	0.471 ± 0.442	0.846 ± 0.000
Cmc	**0.007** ± 0.003	0.000 ± 0.003	0.025 ± 0.003	**0.026** ± **0.002**	**0.027** ± 0.000	**0.027** ± 0.001	0.020 ± 0.014	**0.027** ±0.000
Iris	**0.670** ± **0.091**	0.597 ± 0.099	0.733 ± **0.006**	**0.736** ± 0.008	**0.736** ± **0.010**	0.731 ± 0.013	0.573 ± 0.271	0.671 ± 0.125
Glass	**0.236** ± 0.016	0.235 ± **0.012**	**0.265** ± **0.011**	0.257 ± 0.022	**0.262** ± **0.006**	0.246 ± 0.029	0.230 ± 0.107	0.228 ± 0.021
Um	0.001 ± 0.001	0.001 ± 0.001	0.111 ± 0.070	**0.134** ± **0.087**	0.157 ± 0.139	**0.255** ± **0.075**	0.106 ± 0.108	0.242 ± 0.048
Wine	**0.301** ± 0.000	0.277 ± 0.097	0.343 ± 0.042	**0.358** ± 0.023	0.353 ± 0.031	**0.371** ± 0.004	0.267 ± 0.189	0.366 ± 0.026
Mfeatures	**0.100** ± 0.054	0.014 ± 0.028	**0.299** ± 0.052	0.278 ± **0.045**	0.313 ± **0.001**	**0.314** ± 0.002	0.234 ± 0.159	0.313 ± 0.002
Thyroid	0.070 ± **0.070**	**0.179** ± 0.156	**0.540** ± **0.055**	0.415 ± 0.151	**0.458** ± **0.145**	0.403 ± 0.147	0.227 ± 0.209	0.517 ± 0.147
Ave-P	**0.361**	0.348	0.510	**0.526**	0.550	**0.552**	0.382	0.503
Ave-C	**0.029**	0.052	**0.060**	0.067	0.033	**0.032**	0.233	0.073
W/T/L	7/1/3	3/1/7	6/0/5	5/0/6	5/3/3	3/3/5		

4.2 $\mathcal{E}$-Object Neighbourhood Clustering Ensemble

In this section, we modified ONCE further by considering only the most similar common neighbours to a pair of objects; this can be done by implementing the concept of $\mathcal{E}$-neighbourhood. Section 4.2.1 gives a definition of the $\mathcal{E}$-Object-Neighbourhood Similarity. Section 4.2.2 includes details of the experiment conducted, along with the analysis of the results.

4.2.1 Definition of $\mathcal{E}$-Object Neighbourhood Similarity ($\mathcal{E}$-ONCE)

The key idea of the $\mathcal{E}$-neighbourhood is to construct just the common neighbours of a pair of objects that have a similarity greater than or equal to a certain threshold $\mathcal{E}$, which takes the value $\in [0, 1]$. Thus, objects that have a similarity with the given object pair greater than or equal to $\mathcal{E}$ are considered to be common neighbours to that pair of objects. The difference between ONCE and $\mathcal{E}$-ONCE is that in ONCE we consider all the objects that are placed in the same cluster as the pair itself to be common neighbours to that pair, while in $\mathcal{E}$-ONCE, we do not consider all of the objects placed in the same cluster as the pair itself — we only consider the ones that have similarities greater than or equal to $\mathcal{E}$ to be common neighbours to that pair.

The details implementation of the $\mathcal{E}$-neighbourhood with the ONCE algorithm are as follows:

Algorithm 3: $\mathcal{E}$-ONCE Algorithm.

```
Input: $\Gamma = \{P_1, P_2, P_3, \dots, P_m\}$, $m$ number of clustering members
Output: Partition of Dataset $X = \{x_1, x_2, x_3, \dots, x_n\}$
for each $i = 1 : n$ do
    for each $j = i + 1 : n$ do
        Calculate the $CO$ similarity for pair $(x_i, x_j)$ using equation 2.3
for each $i = 1 : n$ do
    for each $j = i + 1 : n$ do
        Find the common neighbours list $Z_{x_i,x_j} = \{x_1, x_2, \cdots, x_z\}$, that satisfy the
        following:
        $\forall x_l \in Z_{x_i,x_j}, \exists CO(x_i, x_l) \geq \mathcal{E} \wedge CO(x_j, x_l) \geq \mathcal{E}$
        Calculate the average similarity of the common neighbours for pair $(x_i, x_j)$
        using equation 4.1
        Calculate $W(x_i, x_j)$ using equation 4.2
Scale $W$: $W/W_{max}$
Convert the similarity matrix $W$ into distance matrix $W'$
Obtain the final clustering results by applying a hierarchical clustering
algorithm to $W'$
```

To simplify the calculation time, we adapted our algorithm to work with a sparse CO matrix, in order to calculate the $\mathcal{E}$-ONCE matrix. Then we converted the resulting matrix to a full distance matrix, in order to apply hierarchical clustering to obtain the final clustering result.

4.2.2 Experimental Results

In this experiment, we followed the same experimental procedures as in Section 4.1.4, the only difference being that we replaced datasets Cmc and Um with the Soybean and Ionosphere datasets as we found from the previous experiment (Section 4.1.4) that Cmc and Um are not suitable clustering problems. Therefore, we ran ONCE, CO and $\mathcal{E}$-ONCE, all using the Average linkage method on 11 datasets. Tables 4.5 and 4.6 show the average performance (Ave-P) of ten runs using CO, ONCE and $\mathcal{E}$-ONCE with 4 different values for $\mathcal{E}$ $(0.5, 0.6, 0.7, 0.8)$ on 8 datasets. A value less than 0.5 is too small to consider and a value larger than 0.8 is too narrow to consider, we think that a value $\in [0.5, 0.8]$ is reasonable. They also show the average consistency (Ave-C) for each ensemble method across all of the datasets measured by NMI and ARI respectively. The bold value in each row represents the highest quality for each dataset, while the underlined value in each row represents the best performance in terms of consistency for each dataset.

The results show that $\mathcal{E}$-ONCE did not improve much compared to ONCE in all of the tested datasets, although on the Iris and Ionosphere datasets using $\mathcal{E}$-ONCE slightly improved the quality. Comparing the consistency of the three methods, it is found that the $\mathcal{E}$-ONCE is slightly more consistent than ONCE and CO in 7 and 6 datasets, measured by NMI and ARI respectively. ONCE achieved the highest average performance compared to CO and $\mathcal{E}$-ONCE measured by NMI, while the highest average performance was achieved using ONCE and $\mathcal{E}$-ONCE (when $\mathcal{E}$ is equal to 0.6 and 0.7) measured by ARI. Looking at Wins/Ties/Losses, it is observed that ONCE wins more than CO and $\mathcal{E}$-ONCE, while CO wins on only one dataset (D31), measured by NMI and ARI. $\mathcal{E}$-ONCE (0.5) wins 2 times, while $\mathcal{E}$-ONCE using the other values does not win at all when NMI is used to measure the quality, and $\mathcal{E}$-ONCE (0.5) wins 3 times and $\mathcal{E}$-ONCE (0.7) wins 2 times when the ARI index is used to measure the quality.

In conclusion, this experiment shows that applying the $\mathcal{E}$ neighbourhood concept to the ONCE algorithm did not achieve a further improvement in terms of cluster quality. Considering the additional time required to calculate the $\mathcal{E}$ neighbourhood of each pair of objects, and the lack of improvement in cluster quality in this experimental set-up, we can say that $\mathcal{E}$-ONCE did not achieve its expectations and that ONCE is better than $\mathcal{E}$-ONCE. However, in the next section, we will compare the performance of ONCE with other pairwise similarity-based clustering ensemble methods.

Table 4.5: The average performance and the standard deviation of ten runs of each method for each dataset measured by NMI. The average performance (Ave-P) of each ensemble method across 11 datasets, and the average consistency (Ave-C) are included.

	ONCE-Av	**CO-Av**	**$\mathcal{E}$-ONCE-Av (0.5)**	**$\mathcal{E}$-ONCE-Av (0.6)**	**$\mathcal{E}$-ONCE-Av (0.7)**	**$\mathcal{E}$-ONCE-Av (0.8)**
D31	0.962 ± 0.004	**0.965** ± 0.001	0.963 ± 0.003	0.964 ± 0.002	0.964 ± 0.003	0.963 ± 0.002
R15	**0.994** ± 0.002	0.991 ± 0.009	**0.994** ± 0.000	0.991 ± 0.007	0.991 ± 0.007	0.934 ± 0.034
Agg	**0.971** ± 0.014	0.961 ± 0.032	0.960 ± 0.031	0.961 ± 0.028	0.961 ± 0.029	0.961 ± 0.029
Bcw	0.744 ± 0.006	0.740 ± 0.007	0.742 ± 0.003	**0.746** ± 0.006	**0.746** ± 0.006	0.741 ± 0.003
Iris	0.752 ± 0.025	0.746 ± 0.028	**0.758** ± 0.019	0.754 ± 0.019	0.754 ± 0.019	**0.758** ± 0.006
Glass	**0.402** ± 0.023	0.391 ± 0.018	0.390 ± 0.025	0.400 ± 0.014	0.401 ± 0.025	0.399 ± 0.022
Wine	**0.436** ± 0.091	0.425 ± 0.076	0.426 ± 0.094	0.425 ± 0.074	0.425 ± 0.074	0.329 ± 0.069
Mfeatures	**0.489** ± 0.006	0.480 ± 0.003	0.483 ± 0.006	0.480 ± 0.004	0.484 ± 0.006	0.483 ± 0.005
Thyroid	**0.377** ± 0.091	0.342 ± 0.076	0.352 ± 0.094	0.347 ± 0.074	0.347 ± 0.074	0.332 ± 0.069
Soybean	**0.807** ± 0.058	0.751 ± 0.061	0.751 ± 0.064	0.763 ± 0.061	0.763 ± 0.061	0.763 ± 0.061
Ionosphere	0.132 ± 0.003	0.133 ± 0.002	**0.135** ± 0.000	0.133 ± 0.003	0.133 ± 0.002	0.132 ± 0.003
Ave-P	**0.642**	0.630	0.632	0.633	0.634	0.618
Ave-C	0.117	0.112	0.114	0.110	0.111	0.111
W/T/L	6/1/4	1/ 0/10	2/1/8	0/1/10	0/1/10	0/1/10

Table 4.6: The average performance and the standard deviation of ten runs for each dataset measured by ARI. The average performance (Ave-P) of each ensemble method across 11 datasets, and the average consistency (Ave-C) are included.

	ONCE-Av	**CO-Av**	**$\mathcal{E}$-ONCE-Av (0.5)**	**$\mathcal{E}$-ONCE-Av (0.6)**	**$\mathcal{E}$-ONCE-Av (0.7)**	**$\mathcal{E}$-ONCE-Av (0.8)**
D31	0.937 ± 0.015	**0.948** ± 0.002	0.944 ± 0.011	0.945 ± 0.010	0.942 ± 0.013	0.942 ± 0.013
R15	0.992 ± 0.002	0.983 ± 0.028	**0.993** ± 0.000	0.979 ± 0.029	0.979 ± 0.029	0.750 ± 0.127
Agg	**0.977** ± 0.010	0.948 ± 0.067	0.947 ± 0.065	0.950 ± 0.059	0.949 ± 0.059	0.948 ± 0.063
Bcw	0.843 ± 0.005	0.839 ± 0.006	0.841 ± 0.003	**0.845** ± 0.005	**0.845** ± 0.005	0.841 ± 0.003
Iris	0.729 ± 0.028	0.717 ± 0.026	**0.736** ± 0.025	0.733 ± 0.026	0.733 ± 0.021	0.737 ± 0.025
Glass	0.251 ± 0.009	0.248 ± 0.011	0.251 ± 0.013	0.253 ± 0.011	**0.255** ± 0.010	0.252 ± 0.010
Wine	0.332 ± 0.035	0.367 ± 0.007	0.367 ± 0.005	0.368 ± 0.006	**0.368** ± 0.006	0.206 ± 0.140
Mfeatures	**0.330** ± 0.008	0.317 ± 0.003	0.321 ± 0.007	0.318 ± 0.007	0.322 ± 0.008	0.321 ± 0.007
Thyroid	0.422 ± 0.140	0.387 ± 0.152	0.385 ± 0.156	0.410 ± 0.145	0.410 ± 0.145	0.392 ± 0.100
Soybean	**0.638** ± 0.049	0.584 ± 0.054	0.584 ± 0.054	0.595 ± 0.057	0.596 ± 0.057	0.596 ± 0.057
Ionosphere	0.175 ± 0.003	0.176 ± 0.002	**0.178** ± 0.000	0.176 ± 0.003	0.176 ± 0.002	0.175 ± 0.003
Ave-P	**0.602**	0.592	0.595	0.597	0.598	0.560
Ave-C	0.028	0.033	0.031	0.032	0.032	0.050
W/T/L	3/0/8	1/0/10	3/0/8	0/2/9	2/2/7	0/0/11

4.3 Comparing ONCE with other Consensus Functions

In this section, we compare the performance of ONCE with other pairwise similarity-based consensus functions, in particular with the Connected-Triple based similarity (CTS) matrix, the SimRank-based similarity (SRS) matrix [49] and the Approximate SimRank-based similarity (ASRS) matrix [52].

4.3.1 Experimental Results

We used the same generated members for each dataset in the experiment in Section 4.2.2 to run three link-based methods, which are CTS, SRS and ASRS, and we used the Average Linkage method over their matrices to obtain the final clustering results[1]. As recommended by Iam-on et al. [49, 52], we set the decay factor parameter for CTS, SRS and ASRS to its default value, which is equal to 0.8. We also set the number of iterations for ASRS method to its default value, which is equal to 5.

Tables 4.7 and 4.8 show the results of NMI and ARI on 11 datasets respectively. Please note that the results of ONCE and CO qualities are copied from Tables 4.5 and 4.6 for comparison purposes. The entries in these tables represent the average quality in ten runs along with the standard deviation. In these tables, the best quality for each dataset is indicated by the bold value, and the most consistent algorithm for each dataset is indicated by the underlined value.

The results measured by NMI show that in 4 datasets the ONCE-Av outperformed the performance of other ensemble methods, and in 4 datasets it achieved a very close performance to the highest one in these datasets. In the R15 dataset, the highest quality is achieved using the ONCE-Av, CTS-Av, SRS-Av and ASRS-Av algorithms, and this is the only dataset where the highest quality was achieved by a number of ensemble methods.

[1] We used the LinkCLuE Package available at `https://www.jstatsoft.org/article/view/v036i09`.

The results obtained using ARI indexes show that ONCE-Av outperformed the other ensemble methods in 3 datasets, while in others it achieved a result very close to the highest quality. The performance of SRS on the Mfeatures dataset was very poor using the NMI index, while it was equal to 0 measured by ARI. This result indicates that using SRS as a consensus function can result in an unexpected ensemble clustering quality that is worse than the single clustering algorithm. So building a clustering ensemble using SRS in this case is a failure. Moreover, SRS did not win at all in this experiment measured by NMI, while using ARI it won only once in the Glass dataset, where CTS and ASRS achieved a very close performance. ASRS also won once in the Soybean dataset using both indices, where the quality of ONCE was very close to this winning quality.

However, comparing the three link-based ensemble methods, it is found that CTS performs better than the other two methods. Using the same strategy of comparison used in [49], which is the winning statistic, it is found that CTS wins 3 times, while ONCE wins 4 times and CO wins only 3 times when the results are measured by NMI. Along with the CO algorithm, they both win 3 times when the results are measured by ARI.

On average, CTS improved by 0.006 (measured by NMI), and CO and ONCE improved by the same degree, compared to CTS. Furthermore, it is observed that on average ONCE outperformed CO, CTS, SRS and ASRS using the Average Linkage in this experimental set-up using both indices (NMI and ARI).

Comparing the performance of these methods in terms of the consistency, we found that on average the most consistent algorithms (using the NMI index) in this experiment were SAR and ASRA, equal to 0.019. However, the consistencies of other methods (ONCE, CO and CTS) were very close to this performance, and using the ARI index we found that ONCE is the most consistent algorithm, but that the other compared methods were very close to this performance. Therefore, this experiment indicates that the average performances in terms of the consistency of these pairwise-based clustering ensemble methods are more or less the same, whereas in terms of quality, ONCE-Av outperformed the other compared methods.

Table 4.7: The average performance and the standard deviation of ten runs for each dataset measured by NMI. The average performance (Ave-P) of each ensemble method across 11 datasets, and the average consistency (Ave-C) are included.

	ONCE-Av	**CO-Av**	**CTS-Av**	**SRS-Av**	**ASRS-Av**
D31	0.962 ± 0.004	**0.965** ± 0.001	0.959 ± 0.005	0.963 ± 0.003	0.960 ± 0.007
R15	**0.994** ± 0.002	0.991 ± 0.009	**0.994** ± 0.000	**0.994** ± 0.002	**0.994** ± 0.002
Agg	**0.971** ± 0.014	0.961 ± 0.032	0.969 ± 0.023	0.966 ± 0.026	0.956 ± 0.027
Bcw	**0.744** ± 0.006	0.740 ± 0.007	0.738 ± 0.003	0.740 ± 0.005	0.736 ± 0.000
Iris	0.752 ± 0.025	0.746 ± 0.028	**0.756** ± 0.018	0.755 ± 0.019	0.736 ± 0.010
Glass	0.402 ± 0.023	0.391 ± 0.018	**0.403** ± 0.011	0.397 ± 0.008	0.400 ± 0.010
Wine	**0.436** ± 0.091	0.425 ± 0.076	0.427 ± 0.009	0.424 ± 0.006	0.435 ± 0.010
Mfeatures	**0.489** ± 0.006	0.480 ± 0.003	0.480 ± 0.003	0.035 ± 0.000	0.480 ± 0.004
Thyroid	0.377 ± 0.091	0.342 ± 0.076	**0.453** ± 0.105	0.345 ± 0.085	0.248 ± 0.099
Soybean	0.807 ± 0.058	0.751 ± 0.061	0.792 ± 0.051	0.752 ± 0.054	**0.818** ± 0.036
Ionosph	0.132 ± 0.003	**0.133** ± 0.002	0.026 ± 0.000	0.132 ± 0.003	0.026 ± 0.000
Ave-P	**0.642**	0.630	0.636	0.591	0.617
Ave-C	0.029	0.028	0.021	0.019	0.019
W/T/l	4/1/6	2/0/9	3/1/7	0/1/10	1/1/9

Table 4.8: The average performance and the standard deviation of ten runs for each dataset measured by ARI. The average performance (Ave-P) of each ensemble method across 11 datasets, and the average consistency (Ave-C) are included.

	ONCE-Av	**CO-Av**	**CTS-Av**	**SRS-Av**	**ASRS-Av**
D31	0.937 ± 0.015	**0.948** ± 0.002	0.918 ± 0.024	0.939 ± 0.014	0.924 ± 0.030
R15	0.992 ± 0.002	0.983 ± 0.028	**0.993** ± 0.000	0.992 ± 0.002	0.992 ± 0.003
Agg	**0.977** ± 0.010	0.948 ± 0.067	0.965 ± 0.049	0.962 ± 0.051	0.960 ± 0.036
Bcw	**0.843** ± 0.005	0.839 ± 0.006	0.838 ± 0.003	0.839 ± 0.004	0.835 ± 0.000
Iris	0.729 ± 0.028	0.717 ± 0.026	**0.733** ± 0.026	0.732 ± 0.027	0.648 ± 0.077
Glass	0.251 ± 0.009	0.248 ± 0.011	0.253 ± 0.012	**0.255** ± 0.016	0.253 ± 0.025
Wine	0.332 ± 0.035	**0.367** ± 0.007	0.327 ± 0.039	0.366 ± 0.006	0.331 ± 0.034
Mfeatures	**0.330** ± 0.008	0.317 ± 0.003	0.317 ± 0.003	0.000 ± 0.000	0.317 ± 0.006
Thyroid	0.422 ± 0.140	0.387 ± 0.152	**0.520** ± 0.112	0.378 ± 0.151	0.182 ± 0.101
Soybean	0.638 ± 0.049	0.584 ± 0.054	0.629 ± 0.049	0.584 ± 0.053	**0.651** ± 0.037
Ionosph	0.175 ± 0.003	**0.176** ± 0.002	0.004 ± 0.000	0.175 ± 0.003	0.004 ± 0.000
Ave-P	**0.602**	0.592	0.591	0.566	0.554
Ave-C	0.028	0.033	0.029	0.030	0.032
W/T/L	3/0/8	3/0/8	3/0/8	1/0/10	1/0/10

4.3.2 Comparing ONCE with Individual Members

Tables 4.9 and 4.10 show the quality of individual members of the first run in each dataset from the same experiment run in this main section. In total we generated 7 individual members for each dataset. Generally, we found that the quality of the individual members in each dataset varied from high to very low quality. For example, in D31 the highest quality is equal to 0.952, while the lowest quality is equal to 0.005, measured by ARI as shown in Table 4.10.

We compared the quality of ONCE-Av in tables 4.7 and 4.8 with the maximum individual member quality for each dataset (tables 4.9 and 4.10) measured by NMI and ARI respectively). We found that in most datasets the maximum member quality is higher than the quality of ONCE. Moreover, it also higher than the highest ensemble quality in each dataset as seen in Tables 4.7 and 4.8. In the R15 and Iris datasets, the quality of ONCE is equal to the maximum member quality measured by ARI. Therefore, from these observations we can conclude that the clustering ensemble method does not always outperform the best individual members in terms of quality. On the other hand, in real-word data the best individual member is not always guaranteed to be generated using a single clustering algorithm.

Table 4.9: The performance of the seventh members in the first run of the experiment for each datasets measured by NMI. The bold value represents the maximum quality in each dataset.

	Member 1	**Member 2**	**Member 3**	**Member 4**	**Member 5**	**Member 6**	**Member 7**
D31	0.915	0.893	0.066	0.952	0.952	0.938	**0.967**
R15	0.925	0.889	0.882	0.992	0.271	**0.994**	**0.994**
Agg	0.775	0.794	0.007	**0.991**	0.859	0.826	0.980
Bcw	**0.748**	0.741	0.018	0.677	0.006	0.730	0.191
Iris	0.742	0.631	0.002	0.736	0.722	**0.750**	0.615
Glass	**0.450**	0.381	0.444	0.222	0.393	0.359	0.219
Wine	0.429	0.424	0.091	0.416	0.019	0.417	**0.453**
Mfeatures	0.475	0.476	0.011	0.497	0.471	0.479	**0.598**
Thyroid	0.277	**0.436**	0.084	0.201	0.003	0.339	0.217
Soybean	0.793	0.764	0.058	0.830	**0.848**	0.716	0.370
Ionosph	0.135	0.132	0.010	0.026	0.062	0.130	**0.260**

Table 4.10: The performance of the seventh members for each datasets measured by ARI. The bold value represents the maximum quality in each dataset.

	Member 1	**Member 2**	**Member 3**	**Member 4**	**Member 5**	**Member 6**	**Member 7**
D31	0.770	0.724	0.005	0.907	0.924	0.864	**0.952**
R15	0.819	0.726	0.542	0.989	0.074	**0.993**	**0.993**
Agg	0.616	0.660	0.004	**0.995**	0.766	0.685	0.984
Bcw	**0.846**	0.841	0.003	0.782	0.009	0.830	0.025
Iris	0.716	0.449	0.011	0.564	0.642	**0.729**	0.575
Glass	**0.281**	0.264	0.280	0.143	0.231	0.214	0.139
Wine	0.371	0.352	0.005	0.293	0.013	0.354	**0.478**
Mfeatures	0.314	0.311	0.001	0.333	0.304	0.314	**0.359**
Thyroid	0.211	**0.522**	0.031	0.137	0.002	0.431	0.162
Soybean	0.595	0.659	0.020	0.661	**0.748**	0.545	0.198
Ionosph	0.178	0.173	0.015	0.004	0.099	0.173	**0.209**

4.3.3 Test of Significance

As discussed in Section 3.2.5, to assess the performance of ONCE in terms of being significantly better or worse than the compared methods, statistical analysis is necessary. As recommended by Demšar [18], we applied the Iman-Davenport test in order to test the null hypothesis that all the compared methods in this experiment have an equivalent performance. As suggested by Demšar [18], if there are statistically significant differences, we will proceed with the Nemenyi test as a post-hoc test for a pairwise comparison between them in order to discover where the differences lie.

In the Nemenyi test, we first ranked the competing methods for each dataset. It must be noted that the best performing method receives the rank of 1, the second best is ranked 2 and so on. We averaged the rank for the methods that had the same quality score, and for each method we obtained the mean rank by averaging its ranks across all the datasets. The F-value of the Iman-Davenport test was equal to 3.0224 which results in a p-value (0.028), less than the critical level of 0.1. Thus, we rejected the null hypothesis that these methods are equal in performance, and we conducted the Nemenyi test to find out which methods differed from others. Figure 4.5 shows the critical difference diagram of the Nemenyi test results. As we can see, there are two groups of clustering ensemble methods: the performance of

ONCE-Av is significantly better than that of CO-Av and ASRS-Av, while CTS-Av and SRS-Av belong to both groups. We also observe that ONCE-Av achieves the highest rank under this experiment set-up.

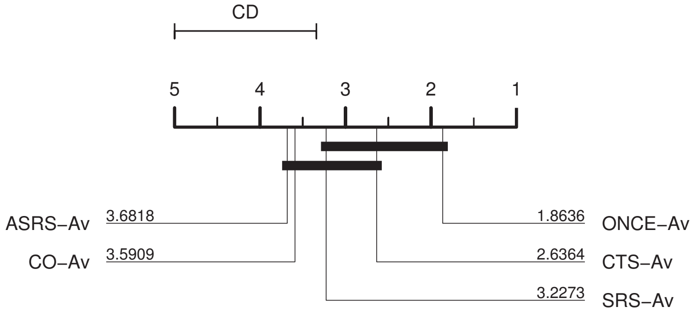


Figure 4.5: The Critical difference diagram of the critical level of 0.1 in which it shows the comparison of five ensemble methods using 11 datasets. The original quality results of these methods are shown in table 4.7.

4.4 Summary

The Co-association matrix [32] is a common clustering ensemble method. We decided to try to improve it by addressing uncertainties among the members in an ensemble. The uncertainty occurs when some objects in the dataset are hard to cluster, which result in them being clustered in different clusters—uncertain agreements between the ensemble members. In this situation, CO could fail to produce a reliable clustering result. One solution that has been suggested by a number of researchers is not just to consider the pairwise object information in the generated members, but rather to enhance the CO matrix by extracting more information from the members [107, 103, 81]. We think that there is other useful information that could be extracted from the ensemble members in order to deal with the uncertain object issue by considering the neighbourhood relationship between pairs of objects.

In this chapter, we presented the Object-Neighbourhood clustering ensemble method (ONCE). The core of ONCE is a new consensus function that addresses the uncertain agreements between members by taking the neighbourhood relationship between object pairs into account in the similarity matrix. We illustrated the problem of the uncertainty using a simple artificial dataset, which includes some hard to cluster objects around the cluster boundary. We ran CO and ONCE on this artificial dataset, and the results showed that CO has been affected by the uncertain agreement between members, while ONCE shows a better performance.

ONCE was tested on 11 datasets (3 artificial and 9 real ones) and compared with the CO using Single, Complete and Average linkage methods. The results show that on average ONCE-Av outperforms the CO-Av method, and the results indicate that the Average linkage is the most appropriate of the linkage methods. Furthermore, the results show that our ensemble method is more consistent and reliable than the single clustering algorithm (*k-means*).

In general, it is interesting to note that the effect of the uncertain pairs of objects varied with the different datasets. This is due to the fact that not every dataset is affected by uncertain pairs of objects, even though these are in fact hard objects to cluster. This is where results for some of the datasets are improved by our method, such as Iris, which has overlapping clusters, and Aggregation, which has uneven-sized clusters with difficult boundaries.

We attempted to extend ONCE further in order to consider only the most similar common neighbours, and proposed $\mathcal{E}$-ONCE. The experiment study however, revealed that there is not much improvement in terms of quality using $\mathcal{E}$-ONCE compared to ONCE, which is preferred as $\mathcal{E}$-ONCE requires more time to be compute.

Finally, we compared ONCE with other object pairwise similarity based consensus functions CTS, SRS and ASRS. In these algorithms, the ensemble members are represented as a network and the well-known link similarity measures have been applied to this network, and have been implemented in the Connected-Triple and SimRank link approaches. The experiment was carried out using 11 datasets, and with all the tested consensus functions we applied the Average Linkage. The results demonstrated that on average ONCE outperforms CO, CTS, SRS and ASRS using the NMI and ARI indices. It was proved statistically by the Nemenyi test that there is a statistical difference between ONCE and ASRS, and between ONCE and CO under this experimental set-up.

Chapter 5

Adaptive Clustering Ensemble

In the previous chapter, we focused on the clustering ensemble methods based on object pairwise similarity, but we found that there are a number of drawbacks for these methods. One of them is that they do not scale very well for a large dataset, as they work at the object level, and most of them do not capture the relationships between clusters or consider the cluster information that is available in the generated members. However, clustering ensemble methods based on cluster similarity, such as MCLA, are much faster than CO and ONCE. Another point is that most of the clustering ensemble approaches (including CO, ONCE, CTS and MCLA) transform the initial clusters produced by the members into a new representation, and then produce the final clustering result by clustering this new representation with an ordinary clustering algorithm. When applying the same representation to a different clustering algorithm, their performance can vary considerably and it can be difficult to decide in advance which clustering algorithm is the best one to use. Therefore, considering the simplicity of the similarity-based consensus functions, there is a need for a new consensus function that is able to construct as much information from the members as possible to produce a reliable clustering result, without requiring an ordinary clustering algorithm to be applied over a similarity matrix.

In this chapter, we propose two clustering ensemble methods to address these drawbacks. First we develop a new consensus function named Dual-Similarity Clustering Ensemble (DSCE) that measures the similarity between initial clusters in members and accordingly derives the similarity between the object and candidate clusters. We use the certainty of agreement between members to reduce the calculation needed in the consensus function. Moreover, we develop DSCE further to become an Adaptive Clustering Ensemble method (ACE) that takes into account the neighbourhood similarity for uncertain objects and overcomes some of the limitations of DSCE.

This chapter is organised into two main sections. The first Section 5.1 describes the proposed clustering ensemble method DSCE. The second Section 5.2 describes ACE method. Finally, Section 5.3 gives a summary of the chapter.

5.1 Dual-Similarity Clustering Ensemble (DSCE)

The main idea of the proposed consensus function is that, instead of calculating the similarity between a pair of objects (the object pairwise similarity) as in the CO method, we calculate the similarity between pairs of clusters generated by the members and then we derive the similarity between newly formed clusters and objects. The rationale is that we have already generated clusters in the first phase of the ensemble process, so it is obviously more efficient and possibly more effective to consider just the similarity between the initial clusters instead of object similarity. We can then extend the concept of common neighbour information from the object level to the cluster level. Therefore, two clusters are considered to be well-associated if their objects resemble one another to a certain degree. If two clusters have a high proportion of objects in common as determined by the ensemble members, they should be merged, whereas if two clusters have a smaller proportion of objects in common, they should be kept separated.

Nevertheless, instead of following some of the single clustering algorithm procedures in building the consensus function, we use the generated members as initial clusters of the dataset and the final clustering is generated by performing three stages, as shown in Figure 5.1, these are:

- **Stage 1: Transformation Stage.** The clustering members are transformed into a binary representation to form our initial clusters.
- **Stage 2: Generating Consensus Clusters.** Firstly, the similarity between initial clusters in terms of how many objects they have in common is measured, and then we merge the most similar ones to form a new consensus clusters.
- **Stage 3: Assigning Object to Only One Cluster.** We identify the candidate clusters, which contain only certain classified objects, and we calculate their certainties. We produce the final clustering result by an iterative process assigning the remaining objects to a cluster that has a minimum effect on its certainty.

These stages help to determine if an object should be placed in a particular cluster or not as classified by the ensemble members, and to find the most suitable cluster for it. Section 5.1.1 presents the definitions of the similarity measures and terminologies that are used with DSCE and ACE, and Section 5.1.2 explain in detail how DSCE works in three stages. Section 5.1.3 illustrates how the DSCE work using a simple example. Sections 5.1.4 and 5.1.5 include the experiment design and analysis of the experiment results respectively.

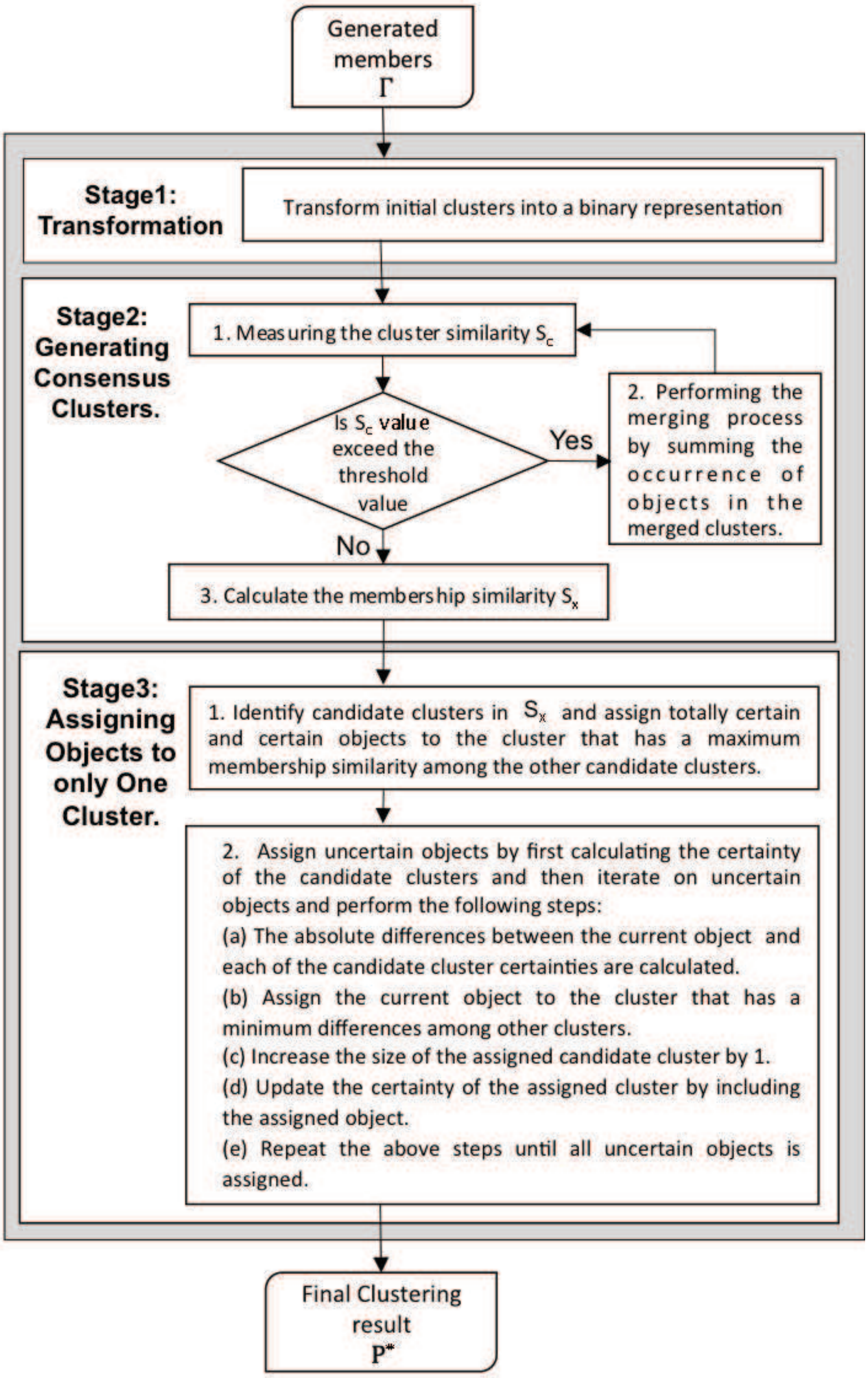

Generated members
Γ
Stage1: Transformation
Transform initial clusters into a binary representation
Stage2: Generating Consensus Clusters.
1. Measuring the cluster similarity S_c
Is S_c value exceed the threshold value
Yes
2. Performing the merging process by summing the occurrence of objects in the merged clusters.
No
3. Calculate the membership similarity S_x
Stage3: Assigning Objects to only One Cluster.
1. Identify candidate clusters in S_x and assign totally certain and certain objects to the cluster that has a maximum membership similarity among the other candidate clusters.
2. Assign uncertain objects by first calculating the certainty of the candidate clusters and then iterate on uncertain objects and perform the following steps:
(a) The absolute differences between the current object and each of the candidate cluster certainties are calculated.
(b) Assign the current object to the cluster that has a minimum differences among other clusters.
(c) Increase the size of the assigned candidate cluster by 1.
(d) Update the certainty of the assigned cluster by including the assigned object.
(e) Repeat the above steps until all uncertain objects is assigned.
Final Clustering result
P*

Figure 5.1: The DSCE flow chart.

5.1.1 Definitions and Notations

Given an ensemble Φ that is built with m clustering partitions $\Gamma = \{P_1, P_2, P_3, \ldots, P_m\}$ for a dataset $X = \{x_1, x_2, \ldots, x_n\}$, where the qth member, $P_q = \{c_1^q, c_2^q, \ldots, c_{k_q}^q\}$ is a clustering result of k_q clusters, we defined two similarity measures:. These were similarity between clusters from different members and membership similarity between objects and clusters. The latter is measured by the degree of membership by which an object belongs to a cluster, hence it is called membership similarity. Before defining these similarity measures, we briefly define the main notations that we use throughout this chapter as follows:

- S_c: The cluster similarity measure between two clusters.
- S_x: The membership similarity measure.
- θ_1: The membership matrix, where the columns of this matrix correspond to clusters and the rows correspond to objects.
- δ: A binary membership value of an object to a particular cluster, $\delta \in \{1, 0\}$.
- α_1: A cluster merging threshold, the value of which is chosen from S_c.
- α_2: A certainty threshold of classifying objects in a cluster, the value of which is chosen from S_x.
- λ: Number of clusters in θ_1.
- $\overleftarrow{C}$: The set of all the newly formed clusters after the merging process has concluded.
- p_{c_g}: Cluster certainty, only calculated for each newly formed cluster $\in \overleftarrow{C}$.

Definition 1. Cluster similarity S_c *is a measure of similarity between two clusters from different members/partitions regarding how much overlap there is between them.*

Any binary-based similarity measurements can be used as a cluster similarity. Section 5.1.2 gives more details on the cluster similarity measurement that we used in DSCE and ACE.

Definition 2. Membership similarity S_x *is a measure of similarity between an object and a cluster which estimate the degree of membership of an object to a cluster, hence it is called membership similarity. The threshold value* α_2 *of this measure is used to determine how strong this membership similarity is between an object and a cluster.*

In general, membership similarity is similar to the concept of membership degree in soft clustering (where an object x may be placed in more than one cluster). It uses a degree for each object in order to express whether it belongs to a cluster. The membership similarity is formed after the merging process has concluded. Section 5.1.2 includes more details on forming the membership similarity S_x. Generally, the value of S_x is bounded between $[0, 1]$, and a higher value means a stronger membership or a higher degree of certainty that an object belongs to a cluster. Therefore, objects with different values of this measure can be classified as certain, uncertain, totally certain or totally uncertain for a given threshold value α_2, as defined below:

Definition 3. Certain object*: An object,* x_i*, is defined as a certain object if its maximum membership similarity* S_x *is greater than* α_2*, i.e.*

$$\max_{\overleftarrow{C}}((S_x(x_i, \overleftarrow{C})) > \alpha_2. \tag{5.1}$$

That means more than $(\alpha_2 * 100)\%$ of ensemble members agree to assign this object to the same cluster, so we are certain about classifying this object.

Definition 4. Uncertain object*: An object is defined to be an uncertain object if its maximum membership similarity* S_x *is less than or equal to* α_2*.*

$$\max_{\overleftarrow{C}}(S_x(x_i, \overleftarrow{C})) <= \alpha_2. \tag{5.2}$$

That means less than or equal to $(\alpha_2 * 100)\%$ of ensemble members agree to assign this object to the same cluster, so we are uncertain about classifying this object.

Definition 5. Totally certain object*: An object is defined as a totally certain object if its maximum membership similarity S_x for a particular cluster is* 1*.*

That means we are totally certain that all the ensemble members agree to assign this object to a specific cluster.

Definition 6. Totally uncertain object*: An object is defined as a totally uncertain object if its membership similarity S_x for a particular cluster is* 0*.*

That means we are totally uncertain that all the ensemble members agree to assign this object to a specific cluster. .

Based on the objects that are assigned to each cluster, we can calculate a cluster certainty for each newly formed cluster $\in \overleftarrow{C}$ as follows:

Definition 7. Cluster certainty*: The cluster certainty, ρ_{c_g}, is defined as the mean of membership similarity of all objects belonging to that particular cluster $\overleftarrow{c}_g$.*

$$\rho_{c_g} = \frac{1}{|\overleftarrow{c}_g|} \sum_{i=1}^{|\overleftarrow{c}_g|} S_x(x_i, \overleftarrow{c}_g). \tag{5.3}$$

5.1.2 The DSCE Algorithm

Stage 1: Transformation

Having generated m members, which represent unmatched clusters of objects, this stage transforms them into a new representation. In order to avoid solving the relabelling problem between clusters, we transform each cluster (c) to a column binary characteristic vector where a value of 1 indicates that the corresponding object belongs to that cluster, and 0 indicates that the object does not belong to that cluster.

In general, for cluster c_j in clustering member q, its corresponding vector is represented as $c_j^q = [\delta(x_1, c_j^q), \ldots, \delta(x_n, c_j^q)]^T$, where $\delta(x_i, c_j^q)$ is the binary membership and takes the following value:

$$\delta(x_i, c_j) = \begin{cases} 1, & \text{if } x_i \in c_j \ , \forall\, i = 1, \ldots, n. \\ 0, & \text{if } x_i \notin c_j \end{cases} \tag{5.4}$$

Where i is the index of data objects; $j = 1, \ldots, k_q$ is the index of clusters in each member; $q = 1, \ldots, m$ is the index of members in an ensemble.

There will be k_m vectors that are combined to form the initial value of the membership matrix $\theta_1 = [c_1^1, c_2^1, \ldots, \ldots, c_{k_m}^q]$, where $k_m = m.k_q$ and $k_q = k$, $\forall q = 1, \cdots, m$.

Stage 2: Generating Consensus Clusters

In this stage the following three steps are required:

1. Measuring the cluster similarity S_c.
2. Performing the merging process.
3. Calculating the membership similarity S_x between objects and the newly formed (consensus) clusters.

1. Measuring the cluster similarity (S_c). Starting with k_m initial clusters, we measure the cluster similarity by employing the ‘set correlation’ as a cluster similarity measurement, which measures the overlap between two clusters and takes their size into account. It has been developed in the Relevance-Set Correlation (RSC) [46] model, as this measure is an equivalent of the Pearson correlation in clustering analysis. After some simplification and derivation, it can be represented as follows:

$$\begin{aligned} S_c(c^q_{j_q}, c^\ell_{j_\ell}) &= \frac{|c^q_{j_q} \cap c^\ell_{j_\ell}| - \frac{|c^q_{j_q}||c^\ell_{j_\ell}|}{n}}{\sqrt{|c^q_{j_q}||c^\ell_{j_\ell}|(1 - \frac{|c^q_{j_q}|}{n})(1 - \frac{|c^\ell_{j_\ell}|}{n})}} \\ &= \frac{n.CM(c^q_{j_q}, c^\ell_{j_\ell}) - \sqrt{|c^q_{j_q}||c^\ell_{j_\ell}|}}{\sqrt{(n - |c^q_{j_q}|)(n - |c^\ell_{j_\ell}|)}} \end{aligned} \tag{5.5}$$

Where q and ℓ are two members, $q \neq \ell$, and j_q, j_ℓ are the cluster index in q and ℓ respectively. CM is the Cosine similarity measurement [43]:

$$CM(c^q_{j_q}, c^\ell_{j_\ell}) = \frac{|c^q_{j_q} \cap c^\ell_{j_\ell}|}{\sqrt{|c^q_{j_q}||c^\ell_{j_\ell}|}} \tag{5.6}$$

S_c is symmetric, i.e. $S_c(c_i, c_j) = S_c(c_j, c_i)$ and its value is bounded in [-1, 1]. A value of 1 indicates that the two clusters “are identical”, and a value of -1 indicates that the two clusters are “a complement of each other” [104].

2. Performing the merging process. At the beginning of this process, we have three inputs (θ_1, S_c and α_1):

1. θ_1 is the membership matrix resulting from the transformation stage (it contains the initial clusters from the members).
2. S_c is the cluster similarity matrix, which is calculated between the initial clusters in θ_1 in the previous step.
3. α_1 is the merging threshold, which is determined in advance; it can take a value in the interval $[-1, 1]$ (as S_c) (we will discuss and analyse the best value for α_1 in Section 5.1.7).

The merged process is based on the following criterion:

$$\text{if } \; S_c(c^q_{j_q}, c^\ell_{j_\ell}) >= \alpha_1 \quad \Rightarrow \quad c^q_{j_q} \; \text{ and } \; c^\ell_{j_\ell}\text{are similar, hence merged.} \tag{5.7}$$

$$\text{if } \; S_c(c^q_{j_q}, c^\ell_{j_\ell}) < \alpha_1 \quad \Rightarrow \quad c^q_{j_q} \; \text{ and } \; c^\ell_{j_\ell}\text{are dissimilar, not merged.} \tag{5.8}$$

From S_c, any clusters that satisfy a criterion given in equation 5.7 will be merged to replace them in θ_1 with a new cluster $\overleftarrow{c}_j$. This has the result of summing the object memberships of the merged clusters. So, θ_1 is updated as follows:

$$\theta_1(x_i, \overleftarrow{c}_g) = \sum_{u=1}^{r} \delta(x_i, c_u), \qquad \forall\, i = 1, \ldots, n.$$

where r is the set of all merged clusters that formed $\overleftarrow{c}_g = \{c_i + c_j + \cdots + c_r\}$

Then we go back to step 1 in this stage to recalculate the S_c for the updated θ_1 and then iterate until a termination criterion 5.8 is reached for all the similarities between clusters in the updated S_c.

3. Calculate the membership similarity (S_x). S_x is specifically used to refer to the measure of similarity between objects $x_i \in X$ in a newly formed cluster after the merging process is carried out in the previous step as follows:

$$S_x(x_i, \overleftarrow{c}_g) = \frac{1}{\max\{\theta_1(x_i, \overleftarrow{C})\}} \theta_1(x_i, \overleftarrow{c}_g) \qquad \forall\, i = 1, \ldots, n. \tag{5.9}$$

where, $\overleftarrow{C}$ is the set of all the newly formed clusters, $\overleftarrow{C} = \{\overleftarrow{c}_1, \ldots, \overleftarrow{c}_g, ...\}$.

Stage 3: Assigning Objects to Only One Cluster.

In this stage, the aim is to ensure that each object is only assigned to one cluster and to eliminate inappropriate clusters. The inputs of this stage are S_x and α_2. α_2 is the certainty threshold of classifying objects in a cluster and it is determined in advance by the user (Section 5.1.7 will discuss how to specify this threshold).

A number of steps are required in this stage:

1. Identify candidate clusters in S_x and assign totally certain and certain objects.

Based on α_2 we identify clusters in S_x that contain at least one *totally certain* object (definition 5) or *certain* object (definition 3) as a candidate cluster. As these objects have a higher certainty value than α_2, we assign them to the candidate cluster that has a maximum membership similarity among the other candidates. The assigning step for the totally certain and certain objects is done by keeping their maximum membership similarity with the candidate cluster in S_x and setting their membership similarities with other clusters in S_x to be equal to 0. They therefore have only one value of S_x larger than 0 with a particular cluster, which means that the object belongs to that cluster only. In case of a tie between which candidate clusters are assigned to a given totally certain/certain object, we arbitrarily break the tie in favour of the $\overleftarrow{c}_g$ with smallest g. If we put the candidate clusters in a list, this would be the candidate cluster that comes first in the list.

2. Assign uncertain object to only one cluster.

This is for other unassigned objects in S_x that we classified as uncertain objects (definition 4) or totally uncertain objects (definition 6). We should mention that this step is only required when there are any uncertain objects in S_x.

So, firstly we calculate the cluster certainty for each candidate cluster considering only their assigned objects using equation 5.3

At the beginning, as the *totally certain* and *certain* objects are the only ones that are assigned to candidate cluster $\overleftarrow{c}$, we iterate on uncertain objects and in each iteration the following steps are performed:

(a) We set CC as the set of all the candidate clusters in S_x, and for each candidate cluster we calculate the absolute difference between the current object (i.e.x_i)

membership similarity with the identified cluster and its certainty as follows:

$$D_{CC} = \sum_{g=1}^{|CC|} (|S_x(x_i, \overleftarrow{C_g}) - \rho_{\overleftarrow{C_g}}|)$$

(**b**) Assign the current object to the candidate cluster that has a minimum difference among other clusters in CC, that is:

$$\min(D_{CC})$$

(**c**) Increase the size of the assigned candidate cluster by 1.

(**d**) Update the certainty of the assigned cluster using equation 5.3 and this time include the current object.

(**e**) Repeat the above steps until all uncertain objects are assigned.

By assigning uncertain objects to the cluster that has a minimum difference, we maintain the original certainty of the candidate clusters as high as possible. At the beginning of this stage, the only objects that are assigned to candidate clusters are certain objects (either a totally certain or a certain objects) and by definition they have membership similarity larger than α_2, so we expect the certainty of the candidate clusters to be high. Other clusters that do not contain any certain objects are not considered to be good candidate clusters and they are eliminated.

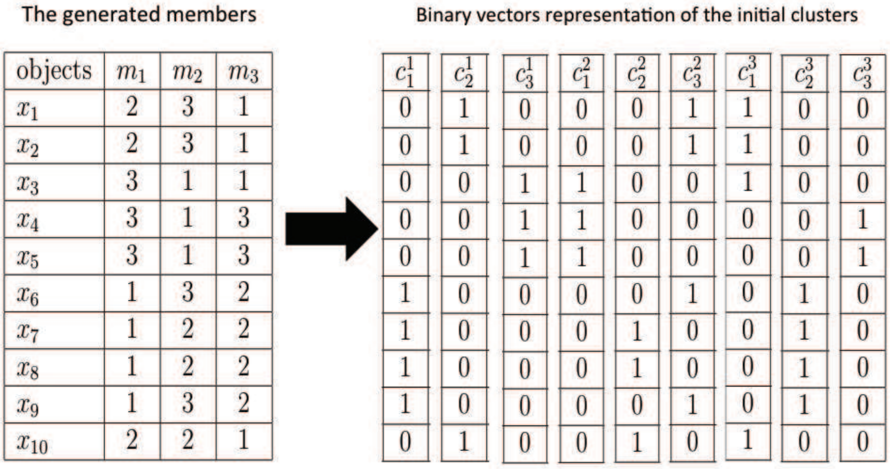
However, at the end of this stage all objects are assigned to only one cluster so the output of the algorithm is the final clustering results P^* of the dataset.

5.1.3 An Illustrative Example

We illustrate how DSCA works with a simple example. Suppose we have a dataset X that contains 10 objects, $X = \{x_1, x_2, \ldots, x_{10}\}$ and that we have generated 3 members ($m = 3$), each of which has 3 clusters ($k = 3$). We run the DSCE algorithm in three stages as follows:

Transformation Stage:

We transform the members into a binary vector representation as shown in Figure 5.2, in which each cluster in the generated member is represented by a binary vector with 9 binary vectors in total. For example, vector c_3^2 is the third cluster in the second member m_2. Four objects x_1, x_2, x_6, x_9 were assigned to cluster c_3^2, so we set their value equal to 1, whereas for other objects in c_3^2 we set a value of 0. These vectors are the input of the second stage.



The generated members

objects	m_1	m_2	m_3
x_1	2	3	1
x_2	2	3	1
x_3	3	1	1
x_4	3	1	3
x_5	3	1	3
x_6	1	3	2
x_7	1	2	2
x_8	1	2	2
x_9	1	3	2
x_{10}	2	2	1

Binary vectors representation of the initial clusters

c_1^1	c_2^1	c_3^1	c_1^2	c_2^2	c_3^2	c_1^3	c_2^3	c_3^3
0	1	0	0	0	1	1	0	0
0	1	0	0	0	1	1	0	0
0	0	1	1	0	0	1	0	0
0	0	1	1	0	0	0	0	1
0	0	1	1	0	0	0	0	1
1	0	0	0	0	1	0	1	0
1	0	0	0	1	0	0	1	0
1	0	0	0	1	0	0	1	0
1	0	0	0	0	1	0	1	0
0	1	0	0	1	0	1	0	0

Figure 5.2: An illustrative example of three clustering members for dataset X of 10 objects, and the transformation from members into a binary vectors representation.

Generating Consensus Cluster Stage:

In this stage, we first measure the similarity between the initial clusters, and generate the similarity matrix S_c as shown in Table 5.1. Then we perform the merging process as follows:

Firstly, we set α_1 equal to 0.8. Looking at S_c, we find that c_1^1 and c_2^3 are identical and have a similarity greater than α_1 with c_2^2, so we merge them by replacing them with $\overleftarrow{c_1}$, which contains the summation of their object membership. In addition, c_2^1 have a similarity greater than α_1 with c_3^2 and c_1^3, so we merge them too as $\overleftarrow{c_2}$. We also merge c_3^1 and c_1^2 as $\overleftarrow{c_3}$. As a result, we gain four clusters, $\overleftarrow{c}_1$, $\overleftarrow{c}_2$, $\overleftarrow{c}_3$ and $\overleftarrow{c}_4$ in the updated θ_1, as shown in Table 5.2. Then we recalculate the similarity measures S_c for the updated θ_1 as shown in Table 5.3.

Table 5.1: The Similarity Matrix S_c, which is the result of measuring the similarity between initials cluster vectors in our illustrative example (Figure 5.2) using S_c measure. $---$ cells indicates that this similarity is not calculated as they are placed in the same member.

	c_1^1	c_2^1	c_3^1	c_1^2	c_2^2	c_3^2	c_1^3	c_2^3	c_3^3
c_1^1	—	—	—	-0.535	0.802	-0.250	-0.667	1	-0.408
c_2^1	—	—	—	-0.429	-0.429	0.802	0.802	-0.535	-0.327
c_3^1	—	—	—	1	-0.429	-0.535	-0.089	-0.535	0.764
c_1^2	-0.535	-0.429	1	—	—	—	-0.089	-0.535	0.764
c_2^2	0.802	-0.429	-0.429	—	—	—	-0.535	0.802	-0.327
c_3^2	-0.250	0.802	-0.535	—	—	—	0.583	-0.250	-0.408
c_1^3	-0.667	0.802	-0.089	-0.089	-0.535	0.583	—	—	—
c_2^3	1	-0.535	-0.535	-0.535	0.802	-0.250	—	—	—
c_3^3	-0.408	-0.327	0.764	0.764	-0.327	-0.408	—	—	—

Table 5.2: The result of θ_1 after we merge the most similar clusters, which are $\overleftarrow{c}_1 = \{c_1^1 + c_2^2 + c_2^3\}$, $\overleftarrow{c}_2 = \{c_2^1 + c_3^2 + c_1^3\}$, $\overleftarrow{c}_3 = \{c_3^1 + c_1^2\}$ and $\overleftarrow{c}_4 = \{c_3^3\}$

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$	$\overleftarrow{c}_4$
x_1	0	3	0	0
x_2	0	3	0	0
x_3	0	1	2	0
x_4	0	0	2	1
x_5	0	0	2	1
x_6	2	1	0	0
x_7	3	0	0	0
x_8	3	0	0	0
x_9	3	0	0	0
x_{10}	0	3	0	0

Based on α_1, we find that there are no more similar clusters to be merged in the updated similarity matrix S_c. Thus, we calculate the membership similarity S_x as shown in Table 5.4 and it becomes the input for the next stage.

Assigning Objects to only One Cluster.

In this stage, firstly we set $\alpha_2 = 0.5$ and we identify the candidate clusters in S_x that had at least one totally certain or certain objects, and we find that based on

Table 5.3: The updated Similarity Matrix S_c after the first step of the merging process is performed, which is the result of measuring the similarity between four clusters in θ_1 (in Table 5.2)

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$	$\overleftarrow{c}_4$
$\overleftarrow{c}_1$	—	-0.408	-0.535	-0.408
$\overleftarrow{c}_2$	-0.408	—	-0.218	-0.500
$\overleftarrow{c}_3$	-0.535	-0.218	—	0.764
$\overleftarrow{c}_4$	-0.408	-0.500	0.764	—

Table 5.4: The result of S_x after we perform the second stage.

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$	$\overleftarrow{c}_4$
x_1	0	1	0	0
x_2	0	1	0	0
x_3	0	0.3	0.6	0
x_4	0	0	0.6	0.3
x_5	0	0	0.6	0.3
x_6	0.6	0.3	0	0
x_7	1	0	0	0
x_8	1	0	0	0
x_9	1	0	0	0
x_{10}	0	1	0	0

α_2 we have three candidate clusters: $\overleftarrow{c}_1$, $\overleftarrow{c}_2$ and $\overleftarrow{c}_3$. As $\overleftarrow{c}_4$ does not contain at least one totally certain or certain object, we eliminated it. After that we assign totally certain and certain objects to the candidate cluster that have a maximum membership similarity among the other candidates by keeping this maximum similarity and modifying the other values to be equal to 0. The updated membership similarity matrix S_x so far is shown in Table 5.5.

In the last step, we check whether S_x (Table 5.5) contains any uncertain objects and in this example based on the value of α_2 we do not have any uncertain objects and the final clustering result for objects in $X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}\}$ is produced $P^* = \{2, 2, 3, 3, 3, 1, 1, 1, 1, 2\}$.

However, if we set α_2 to a higher value equal to 0.9, and re-run the last stage, we find that based on α_2 we only identify $\overleftarrow{c_1}$ and $\overleftarrow{c_2}$ as candidate clusters and this time

Table 5.5: The updated membership similarity matrix S_x after identifying candidate clusters, eliminating non-candidate cluster and assigning totally certain and certain objects.

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$
x_1	0	1	0
x_2	0	1	0
x_3	0	0	0.6
x_4	0	0	0.6
x_5	0	0	0.6
x_6	0.6	0	0
x_7	1	0	0
x_8	1	0	0
x_9	1	0	0
x_{10}	0	1	0

we eliminate $\overleftarrow{c_3}$ and $\overleftarrow{c_4}$. We also identify x_1, x_2, x_7 and x_8 as totally certain objects, while we identify the other objects (x_3, x_4, x_5, x_6, x_9 and x_{10}) as uncertain objects. In this case, we perform step 2 in stage 3, firstly by calculating the certainty for each candidate cluster using equation 5.3, which is for $\overleftarrow{c}_1$ and $\overleftarrow{c}_2$ is equal to 1.

Then, we iterate on uncertain objects and each iteration steps (a to d) were executed as described in Section 5.1.2. For example, the results of these steps of the first iteration (for object x_3) are as follows:

(**a**) We set CC as the set of all candidate clusters, and we calculate the absolute difference as follows:
$CC = \{\overleftarrow{c_1}, \overleftarrow{c_2}\}$, and $D_{CC} = \{|0-1|, |0.3-1|\} = \{1, 0.7\}$.

(**b**) We assign the current objects as follows:
$\min(D_{CC}) = 0.7$, then $x_3 \in \overleftarrow{c_2}$.

(**c**) We increase the size of the assigned candidate cluster by 1: $|\overleftarrow{c_2}| = |\overleftarrow{c_2}| + 1$.

(**d**) We update the certainty of the assigned cluster as follows: $\rho_{c_2} = 0.77$

After all uncertain objects are assigned the final clustering results for dataset X are produced as $P^* = \{2, 2, 2, 2, 2, 2, 1, 1, 2, 2\}$.

5.1.4 Experimental Design

To empirically evaluate the performance of DSCE, we used the same datasets and quality validation indices as in previous chapters. As described in Section 3.2, we followed the clustering ensemble framework (shown in Figure 3.1), and in the generation phase, we used a mixed heuristic technique to generate ten members. We compared the performance of DSCE with a number of clustering ensemble methods including CO [32] (using the average linkage method), DICLENS [73], MCLA [94], and with our previously proposed consensus function ONCE, also using the average linkage method. We set $\alpha 1 = 0.8$, $\alpha_2 = 0.7$. More details of the experiment procedure are given in Section 3.2 in Chapter 3.

5.1.5 Experimental Results

Tables 5.6 and 5.7 present the results of ARI and NMI respectively; each entry in each table represents the average quality of ten runs of the experiment, followed by the standard deviation. The bold value in each row represents the highest quality for each dataset, while the underlined value in each row represents the best performance in terms of consistency. The last column represents the average performance of the generated members, and the last row shows the average quality for each algorithm over all the datasets, as well as the average consistency.

Results obtained by ARI Index: As shown in Table 5.6, there are several interesting observations. First, DSCE achieved the best performance on most tested datasets with respect to average ARI values of ten runs. On the Mfeatures dataset, all of the compared algorithms achieved a quality very close with the highest quality achieved by the DSCE and ONCE algorithms. On the Bcw dataset, DSCE achieved 0.849, as well as CO and MCLA with an equal standard deviation. On the Wine dataset, DSCE achieved the highest quality followed by MCLA, while CO, ONCE and DICLENS performed equally, achieving an average quality of 0.369, and they also achieved a similar performance in terms of consistency, which was better than

DSCE.

Second, the DICLENS algorithm did not perform very well on some datasets and the standard deviation indicates that in some datasets it is not consistent, including Glass and Mfeatures. It achieved the highest quality only in one dataset which was Soybean. Moreover, it achieved a lower quality compared to the average members in the Iris, Glass and Mfeatures datasets.

Third, comparing the average quality across all the datasets, we observed that DSCE outperformed other algorithms, whereas DICLENS achieved the lowest quality with a high average consistency, indicating that this method is the least consistent algorithm when compared with the others. In contrast, CO is the most consistent algorithm, as well as MCLA, followed closely by ONCE and then DSCE. Looking at the average members, we found that our proposed algorithm outperformed the average members in all datasets.

Results obtained by NMI Index: Table 5.7 shows similar results to those obtained by ARI index. We note that DSCE wins on 4 datasets and on 3 datasets achieved a very close performance to the wining method. On average DSCE outperformed other compared methods.

We believe that the main reasons for the better performance of DSCE compared to its competitors are as follows: first DSCE captures the relationships among clusters in the ensemble members, as it deals with them as initial clusters for the final results in the first stage; second, it identifies the object's certainty of being classified in the initial clusters and in the second stage it focuses on the cluster certainty and classified objects based on the lowest affected cluster's certainty; third, this strategy allows for the number of clusters to be converged from the generated members and the overall procedure requires less memory compared with ensemble methods based on object similarity. This means that it will scale very well with big datasets.

Table 5.6: The average performance and the standard deviation of ten runs for each dataset measured by ARI. The average performance (Ave-P) of each ensemble method across 8 datasets, and the average consistency (Ave-C) are included.

	CO-Av	**ONEC-Av**	**DSCE**	**DICLENS**	**MCLA**	**Ave-mem**
Iris	0.725 ± 0.012	0.726 ± 0.009	**0.732** ± 0.021	0.680 ± 0.077	0.723 ± 0.012	0.702 ± 0.038
Wine	0.369 ± 0.005	0.369 ± 0.005	**0.377** ± 0.025	0.369 ± 0.005	0.372 ± 0.002	0.366 ± 0.004
Thyroid	0.559 ± 0.024	0.584 ± 0.044	**0.609** ± 0.032	0.582 ± 0.044	0.563 ± 0.025	0.473 ± 0.036
Mfeatures	0.315 ± 0.006	**0.316** ± 0.005	**0.316** ± 0.004	0.290 ± 0.069	0.308 ± 0.021	0.293 ± 0.029
Glass	0.509 ± 0.029	0.526 ± 0.030	0.528 ± 0.027	0.392 ± 0.123	**0.534**±0.020	0.501 ± 0.009
Bcw	**0.849** ± 0.004	0.847 ± 0.003	**0.849** ± 0.004	0.842 ± 0.005	**0.849** ±0.004	0.830 ± 0.021
Soybean	0.547 ± 0.006	0.550 ± 0.015	0.578 ± 0.052	**0.632** ± 0.046	0.548 ± 0.006	0.566 ± 0.025
Ionosphere	0.163 ± 0.014	0.166 ± 0.008	**0.169** ± 0.005	0.161 ± 0.009	0.166 ± 0.006	0.149 ± 0.007
Ave-P	0.505	0.511	**0.520**	0.493	0.508	0.485
Av-C	0.012	0.015	0.017	0.048	0.012	0.031

Table 5.7: The average performance and the standard deviation of ten runs for each dataset measured by NMI Index. The average performance (Ave-P) of each ensemble method across 8 datasets, and the average consistency (Ave-C) are included.

	CO-Av	**ONEC-Av**	**DSCE**	**DICLENS**	**MCLA**	**Ave-mem**
Iris	0.751 ± 0.015	0.752 ± 0.012	**0.763** ± 0.024	0.757 ± 0.008	0.749 ± 0.015	0737 ± 0.025
Wine	0.428 ± 0.003	0.428 ± 0.003	**0.432** ± 0.014	0.427 ± 0.004	0.429 ± 0.001	0.428 ± 0.003
Thyroid	0.434 ± 0.047	0.473 ± 0.062	0.480 ± 0.056	**0.501** ± 0.053	0.418 ± 0.033	0.403 ± 0.026
Mfeatuers	**0.479** ± 0.002	**0.479** ± 0.003	**0.479** ± 0.002	0.468 ± 0.026	0.475 ± 0.009	0.460 ± 0.027
Glass	0.712 ± 0.027	0.725 ± 0.029	0.725 ± 0.021	0.617 ± 0.107	**0.728** ± 0.017	0.704 ± 0.007
Bcw	0.750 ± 0.005	0.749 ± 0.004	0.750 ± 0.005	0.742 ± 0.006	**0.751** ± 0.005	0.731 ± 0.023
Soybean	0.717 ± 0.002	0.723 ± 0.024	0.756 ± 0.064	**0.822** ± 0.056	0.717 ± 0.002	0.736 ± 0.019
Ionosphere	0.122 ± 0.014	0.124 ± 0.009	**0.128** ± 0.005	0.119 ± 0.009	0.124 ± 0.006	0.108 ± 0.006
Ave-P	0.549	0.557	**0.564**	0.557	0.549	0.491
Ave-C	0.015	0.018	0.024	0.034	0.011	0.017

Identifying the true number of clusters in DICLENS

In our experiment, DICLENS produces the number of clusters automatically, while for CO and ONCE the number of clusters is provided in advance as input. Therefore, we compared the number of clusters produced by DSCE (as shown in figure 5.3) with the number of clusters produced by DICLENS (as shown in figure 5.4). We observed that the DSCE algorithm determined the true number of clusters in four datasets out of eight in all runs: these include Iris, Thyroid, Bcw and Ionosphere. The DICLENS algorithm also found the true number of clusters in four datasets including Wine, Bcw, Thyroid and Ionosphere, while in the Glass dataset, 3 clusters were discovered instead of 6 (true clusters) in six runs out of ten by DICLENS. In the Iris dataset, 2 clusters were discovered instead of 3 in three runs out of ten. In the Mfeatures dataset, 11 clusters were discovered in run number 3, while in run 5, 3 clusters were discovered instead of 10 clusters by DICLENS. In percentages, in 88.7% of the total number of runs in all datasets DSCE determined the true number of clusters, whereas 76.2% were discovered by the DICLENS algorithm. The results indicate that DSCE is more accurate in determining the number of clusters from the generated members.

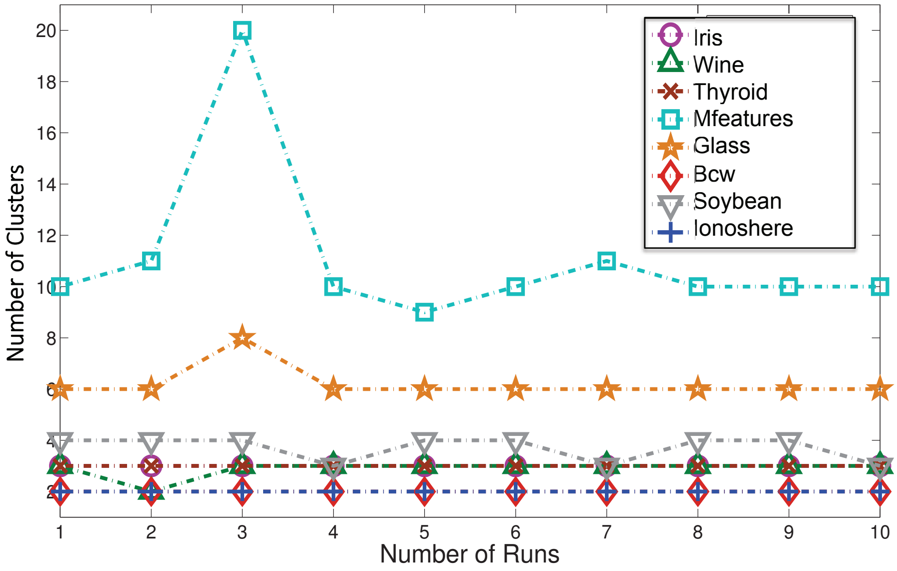


Figure 5.3: Number of clusters produced by DSCE algorithm for each dataset in ten runs. The true number of clusters for {Iris, Wine, Thyroid} = 3, Mfeatuers = 10, Glass = 6, Bcw = 2, Soybean = 4, Ionosphere = 2.

5.1.6 Test of Significance

We applied the Iman-Davenport test [53] to assess our method and other compared methods under the null hypothesis that the mean ranks are equal for all methods. In the Iman-Davenport test, we can reject the null hypothesis of the mean rank being equal for all methods (the result of the Iman-Davenport test was equal to 6.9780, which gives a negligible p-value equal to 4.9845e-04). As suggested by Demšar [18],

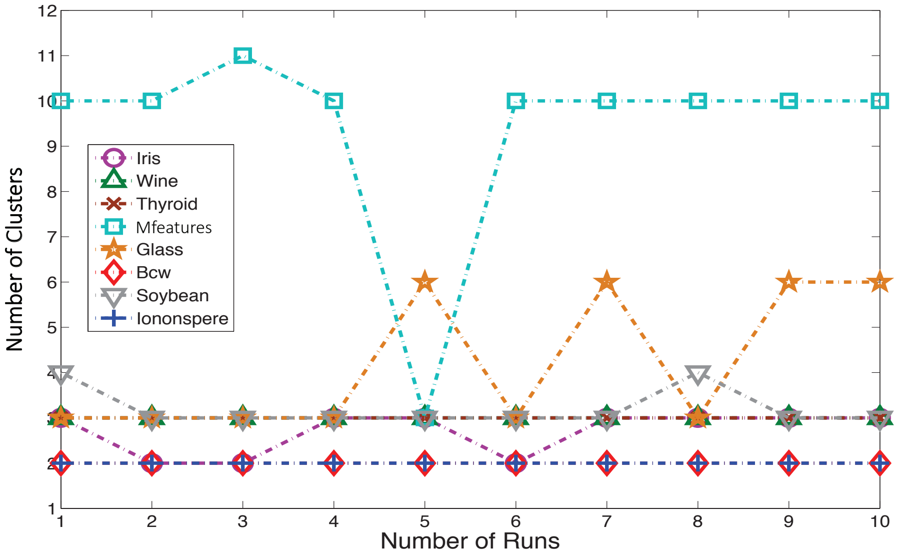


Figure 5.4: Number of clusters produced by DICLENS algorithm for each dataset in ten runs. The true number of clusters for {Iris, Wine, Thyroid} = 3, Mfeatuers = 10, Glass = 6, Bcw = 2, Soybean = 4, Ionosphere = 2.

we used the Nemenyi test as a post-hoc test for a pairwise comparison, to discover where the differences lie. Figure 5.5 shows the result of the post-hoc Nemenyi test in the critical differences diagram at the critical level of 0.1. This diagram shows the mean rank order of each method on a linear scale. The solid bars in these diagrams show a group of algorithms in cliques, indicating that there are no significant differences in rank from one to another, whereas there are significant differences in rank between algorithms in different groups.

The critical difference (CD) is equal to 1.9448. We can identify two groups of algorithms; the first group includes DSCE, MCLA and ONCE, and the second group includes MCLA, ONCE, CO and DICLENS, which indicate that there is not a statistically significant difference between methods in one group. The results suggest that our clustering ensemble algorithm DSCE is significantly better than the CO and DICLENS, but not better than MCLA and ONCE under this experimental set-up.

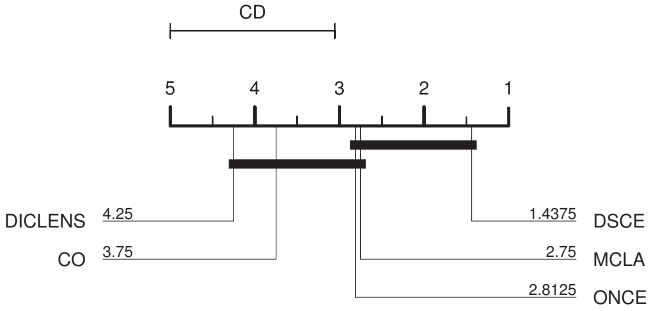


Figure 5.5: The critical difference diagram at the critical level of 0.1. It shows the comparison of four ensemble methods using 8 datasets.

5.1.7 Analysis of Parameters and Time Complexity

In DSCE, we have two parameters, α_1 and α_2, that need to be specified. The first parameter is the minimum allowed similarity between initial clusters to be merged. The value of α_1 can be chosen from the interval [-1,1], but as it is the minimum allowed similarity we limit its value to be one of the following $\alpha_1 = 0.5 \sim 0.9$. The second parameter α_2, is the certainty threshold of classifying objects in a cluster.

To test how sensitive DSCE is to different values of α_1 and α_2 and to what extent they affect the quality of the final clustering result, we used the Wine, Mfeatures and Glass datasets. We ran our proposed algorithm with different values of α_1, and each with all possible values of α_2, which is $\alpha_2 = 0.3 \sim 0.9$, ten times. In each run, we generated ten members by using *k-means* with a random initialisation and we set k to the number of pre-defined clusters for the dataset, for each dataset in all the generated members. Figure 5.6 illustrates the relationship between the average performance of DSCE measured by the ARI index for ten runs and the different value of α_1 for all values of α_2 in the three datasets.

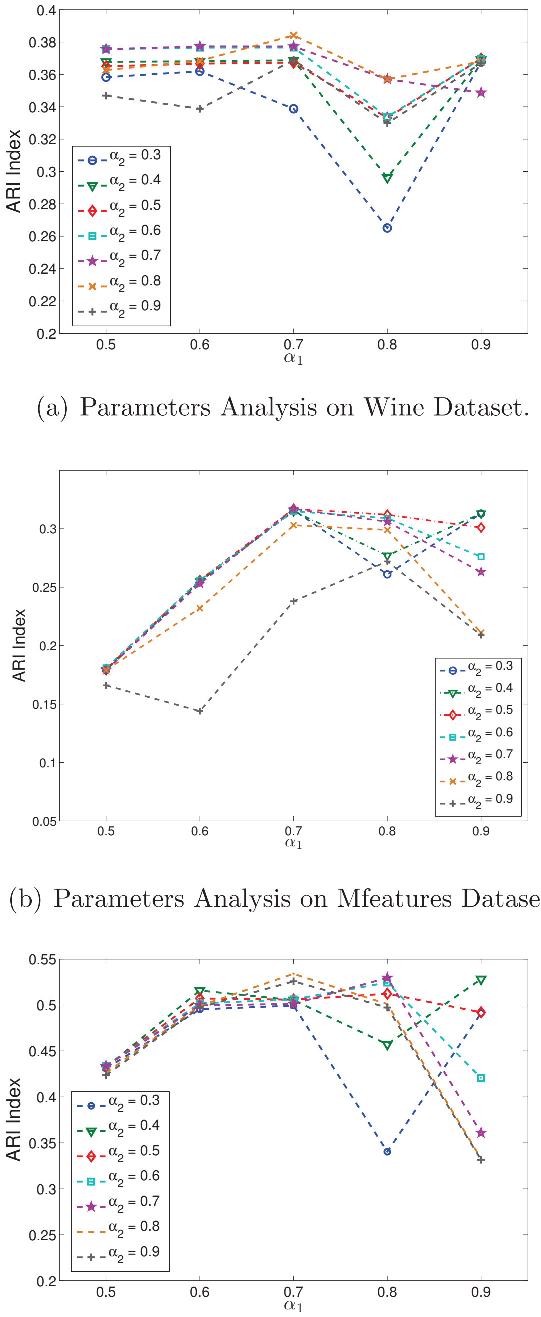
The performance of DSCE was more sensitive when α_2 equals 0.3 and 0.9 compared to other values; this is the case with all values for α_1 between $[0.5, 0.9]$. The best performance of DSCE was when α_1 was equal to 0.7 in the Mfeatures and Glass datasets for most values for α_2, and in particular the highest performance of DSCE in the Mfeatures dataset was when $\alpha_2 = \{0.5, 0.6, 0.7\}$ and in the Glass dataset was when $\alpha_2 = 0.8$. Generally, the performance of DSCE was almost the same when

$\alpha_1 = 0.7$ and when α_2 is between 0.5 and 0.7.

In conclusion, the general guideline for choosing α_1 when all members have a fixed number of clusters is that α_1 should be set to a high value between 0.7 and 0.8. Furthermore, we should avoid a too small value for α_2 as well as a too large value. A value between 0.6 and 0.8 is a reasonable value for α_2.

The time complexity for the worst-case scenario of DSCE is $O(k^2m^2n_u)$, where k is the number of pre-defined clusters for the dataset, m is the number of ensemble members and n_u is the number of uncertain objects in θ_1. In CO the time complexity is equal to $O(n^2)$ and in ONCE is equal to $O(n^4)$, plus the time required by the average linkage method, which is equal to $O(n^3)$, where n is the number of objects in the dataset. While, for MCLA it is equal to $O(km^2n)$.

However, in DSCE the most expensive term is $(km)^2$. For a small size dataset, it may have a number of cluster between $k = 2$ to 10 and a minimum number of ensemble members that can be generated as $m = 3$, so $(km)^2$ becomes more expensive than CO and ONCE. But as the size of the data nowadays is rapidly increasing and as in reality, $n_u < n$, $k \ll n$ and $m \ll n$ hold, then $(km)^2 < n$ and we can say that DSCE is efficient compared to other methods.



(a) Parameters Analysis on Wine Dataset.

(b) Parameters Analysis on Mfeatures Dataset.

(c) Parameters analysis on Glass Dataset.

Figure 5.6: The Average ARI index of ten runs for analysing the two parameters α_1 and α_2.

5.2 The Adaptive Clustering Ensemble (ACE)

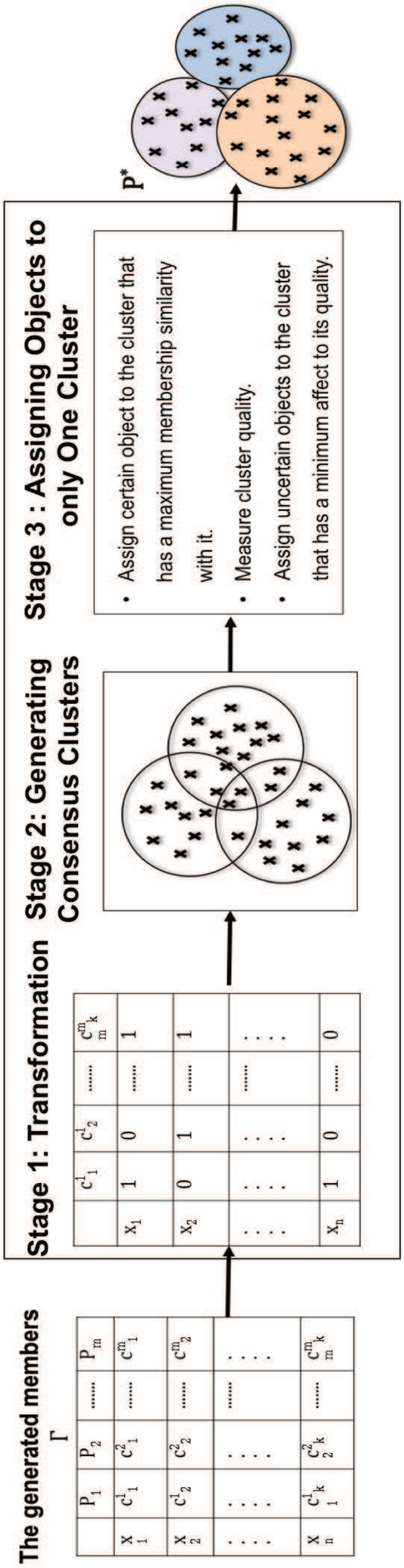
The DSCE algorithm has been modified for three reasons. Firstly, to improve the stability of the DSCE in producing the final clustering result with pre-defined k, even when the members have a different number of clusters. Secondly, to reduce the effect of the two thresholds (α_1 and α_2) on the quality of the final result by applying an adaptive strategy for the value for these thresholds. Finally, to take into account the object neighbourhood similarity for the totally uncertain objects in order not to lose any information when we eliminate an inappropriate cluster.

The adaptive version of the DSCE is composed of the three main stages as we can see in Figure 5.7: Transformation, Generating Consensus Clusters and Resolving Uncertainty. The first stage is to transfer the members into a binary vector representation. The second is to generate the consensus clusters, where the similarity between initial clusters is measured and the pre-defined k clusters are produced. The third stage is to solve uncertain objects, where a certain object is first assigned to the cluster that has a higher membership value and then the uncertain objects are classified to the cluster in a way that has a minimum effect on the cluster quality. The following subsection explains in detail how the algorithm works.

5.2.1 The ACE Algorithm

Stage 1: Transformation

In this stage, the initial clusters in the generated members are transformed into a column binary vector as described in Section 5.1.2. The only difference here is that there is no constraint on the number of clusters that the generated members can have.



The generated members
Γ
Stage 1: Transformation
Stage 2: Generating Consensus Clusters
Stage 3 : Assigning Objects to only One Cluster
P*

	P_1	P_2	......	P_m
x_1	C^1_1	C^2_1	......	C^m_1
x_2	C^1_2	C^2_2	......	C^m_2
....	....	....	......	....
x_n	$C^1_{k_1}$	$C^2_{k_2}$	......	$C^m_{k_m}$

	C^1_1	C^1_2	......	$C^m_{k_m}$
x_1	1	0	......	1
x_2	0	1	......	1
....	....	....	......	....
x_n	1	0	......	0

- Assign certain object to the cluster that has a maximum membership similarity with it.
- Measure cluster quality.
- Assign uncertain objects to the cluster that has a minimum affect to its quality.

Figure 5.7: The diagram of the ACE algorithm

Stage 2: Generating Consensus Clusters

In this stage, three steps are required as described in Algorithm 4. These are:

1. Measuring the cluster similarity S_c.
2. Performing the merging process.
3. Producing k consensus clusters.

1. Measuring the cluster similarity S_c.

In this step, we measure the similarity between initial clusters S_c (equation 5.5), as we did in Stage 2 of the DSCE algorithm (Section 5.1.2).

2. Performing the merging process.

In this step, we perform the merging process as described in Stage 2 in the DSCE algorithm (Section 5.1.2). We should mention that the parameter α_1, which is a threshold for the merging process as shown in equations 5.7 and 5.8, is determined in ACE adaptively based on the similarity values in the cluster similarity matrix S_c. Its influence and sensitivity on the quality of the final clustering result are studied and the details are given later in Section 5.2.6. Our empirical study indicates that it can usually start with a relatively high value, e.g. 0.8, and then adapt its value in accordance with the similarity values in the current similarity matrix. After the most similar initial clusters are merged, we have the updated θ_1, which represents newly formed clusters and perhaps some remaining non-merged initial clusters with their corresponding cluster similarity S_c matrix; then we move onto the next step.

3. Producing k consensus clusters.

In this step, we check whether the number of clusters in θ_1 is exactly equal to k clusters, which will be taken as the final candidate clusters.

For convenience, let λ be the number of clusters in θ_1. There are three possible scenarios: (a) $\lambda = k$, (b) $\lambda > k$, and (c) $\lambda < k$, when checking the number of clusters in θ_1.

(a) When $\lambda = k$, i.e. the number of clusters in θ_1 is equal to the pre-defined k, we then take the clusters in θ_1 as the candidate clusters and adapt α_2 to a value based on S_x so that it can represent a specific percentage of the membership certainty. Then we move onto Stage 3.

(b) When $\lambda > k$, i.e. the number of clusters in θ_1 is greater than the pre-defined k, which is the most likely scenario in practice, there are two options: (A) to terminate the process or (B) proceed with brutal merging or eliminating.

(**Option A**) Coming to this point, the clusters in θ_1 are more dissimilar from each other than the given threshold α_1. If the value of α_1 has reached the minimum acceptable similarity (α_{1min}), it indicates that the clusters in θ_1 for the given dataset are too dissimilar from each other to be merged to obtain the intended k number of clusters. We then conclude that the pre-set value for k is unreasonable and unachievable, and output the generated clusters.

(**Option B**) However, as there is no gold-standard for setting up the minimum acceptable similarity threshold (α_{1min}), it is then also reasonable to go ahead with the process by adapting the threshold value α_1 to reflect the similarity distribution in the current similarity matrix S_c. Then the clusters are merged as described in step 1, or when no more merging process is needed we calculate the membership similarity (S_x) as described in equation 5.9. If the number of clusters in S_x (nb_{cls}), which is identified based on α_2, is still larger than k, then we perform the elimination process as follows:

i. The certainty of each cluster in θ_1 is calculated using equation 5.3.

ii. The certainty values of the clusters are ranked in a descending order.

iii. If each of the top k clusters contains at least one certain object based on the current value of α_2, then these clusters are taken as the final candidate clusters. The non-candidate clusters (eliminated clusters) will be moved from θ_1 to a new membership matrix θ_2, and we move onto stage 3.

iv. Otherwise, we adapt α_2 to be the maximum membership similarity to the kth cluster and consider the first k clusters as the final candidate clusters and we move onto stage 3.

(c) When $\lambda < k$, i.e. the number of clusters in θ_1, is less than the pre-chosen k, then we consider whether any clusters in θ_1 can be divided by adapting the value of α_1. In this case, it is possible that α_1 is unreasonably low and should be adapted incrementally to an appropriate value. In that case we should go back to the beginning of this step (step 3) until the number of the clusters in θ_1 reaches k and we move onto stage 3.

Stage 3: Assigning Objects to only One Cluster.

The aim here is to ensure that each object is assigned to only one cluster. So, the inputs of this stage are: S_X, which is the membership similarity matrix; θ_2, which contains the membership similarity of the eliminated clusters if we performed the elimination process in the previous step; and α_2, which is the adaptive certainty threshold. Two main steps are required here:

1. **Identify totally certain (definition 5) and certain objects (definition 3) in S_x.**

   As certain objects have a higher similarity value than α_2, we assign them to the cluster that has a maximum membership similarity among other clusters in S_x.

2. **Resolving uncertain objects if they exist.**

   This step is only required when there are any uncertain objects. As defined earlier, for an object, if its maximum membership value $S_x(x_i, c_j) <= \alpha_2$ ($\forall j = 1, \ldots, k$), it is considered to be an *Uncertain object* (definition 4), and a *Totally uncertain object* if its maximum membership value is zero (definition 6.). We resolve each one of them differently as follows:

   For totally uncertain objects. There is a possibility that the previous stage may have resulted in totally uncertain objects in S_x. This is of particular concern during the elimination process, as this may have caused information to be lost for some objects, so we verify that each object in S_x has a membership value associated with at least one cluster.

   If S_x contains some totally uncertain objects, we calculate their neighbourhood similarity with clusters in θ_2. We are in fact modifying our early definition of neighbourhood similarity (in Chapter 4) [3], by calculating the average occurrence of their objects' neighbours and the other objects placed in the candidate clusters. In other words, we calculate the similarity between the totally uncertain object and the candidate clusters in S_x as the average of how many times they are classified in the same cluster in θ_2 with other objects that are already placed in the candidate clusters in S_x. Then based on their updated membership similarity S_x, we identify each one of them again as either certain or uncertain objects. For certain objects we go back to step 1 to assign them, whereas for uncertain objects we move onto the next step to resolve them.

   For uncertain objects. Firstly, we measure the quality of each candidate cluster in S_x. In principle, any cluster quality measure can be used, so in this study we measure the compactness of the certain objects in a cluster as the quality metric, and here we call it the original quality of each cluster.

   The compactness of a cluster is usually measured by the variance (Var), which is the average of the squared differences from the mean, as follows:

$$Var(c) = \frac{1}{|\overleftarrow{c}|} \sum_{i=1}^{|\overleftarrow{c}|} (S_x(x_i, \overleftarrow{c}) - p_{\overleftarrow{c}})^2 \tag{5.10}$$

It is essentially the absolute value of the difference between the membership similarity value of object x_i in cluster $\overleftarrow{c}$, and the mean of the objects similarity in cluster $\overleftarrow{c}$ (cluster certainty $p_{\overleftarrow{c}}$ calculated by equation 5.3).

At the beginning, the size of each candidate cluster equals the total number of classified objects, and these objects are the only ones that we can assign to a candidate cluster with certainty, as they have the maximum membership similarity with the classified candidate clusters. For each uncertain object the following steps are performed:

(a) For each candidate cluster in S_x, we recalculate its quality using the equation 5.10 by including the current object membership similarity with the identified cluster.

(b) Compare the original quality and the current quality for each candidate cluster.

(c) Assign the current object to the cluster that has a minimum effect on its original quality.

(d) Increase the size of the assigned cluster by 1.

(e) Update the original quality of the assigned cluster to be equal to the current quality.

(f) Repeat the above steps until all the uncertain objects are assigned.

Generally, we assign uncertain objects to a cluster in such a way that this will have a minimum effect on its quality. By doing so, we aim to ensure that the original quality of the cluster has not been affected too much, as it is widely known that a small value for cluster quality indicates a compact cluster result.

Therefore, by assigning each object to only one cluster we obtain the final clustering result P^* of the dataset X.

Algorithm 4: The Pseudocode of the ACE Algorithm.

```
Input   : $\Gamma = \{P_1, P_2, P_3, \ldots, P_m\}$, $\alpha_1$, $\alpha_2$, $\alpha_{1min}$, $\Delta\alpha$, and $k$
Output: $P^*$
$\theta_1 \leftarrow$ Transform $m$ members into binary vectors as initial clusters;
$S_c \leftarrow$ Compute clusters similarity for clusters in $\theta_1$ with equation 5.5;
while true do
    $\theta_1 \leftarrow$ MergeCls(initial clusters, $S_c$, $\alpha_1$);
    if # clusters in $\theta_1, \lambda >= k$ then
        break;
    else
        Adapt $\alpha_1 = \alpha_1 + \Delta\alpha$;
$\lambda \leftarrow$ find # of clusters in $\theta_1$;
while $\lambda > k$ do
    Update $S_c$ with equation 5.5;
    Adapt $\alpha_1 \leftarrow$ maximum similarity value in $S_c$;
    if $\alpha_1 < \alpha_{1min}$ then
        break;
    else
        $new\theta_1 \leftarrow$ MergeCls($\theta_1$, $S_c$, $\alpha_1$);
    if # clusters in $new\theta_1 < k$ then
        break;
    else
        $\theta_1 \leftarrow new\theta_1$
Compute similarity measure $S_x$ with equation 5.9;
$nb_{cls} \leftarrow$ find # of clusters in $S_x$ that contain at least one certain object
specified by $\alpha_2$;
if $nb_{cls} == k$ then
    Consider these clusters as candidate clusters in $P^*$;
    $\theta_2 \leftarrow$ non-candidate clusters;
else
    Compute cluster certainty in $S_x$ with equation 5.3;
    Sort the cluster certainties in descend order;
    Adapt $\alpha_2 \leftarrow S_x max\{k\}$;
    Keep the top $k$ clusters in $S_x$ as the candidate clusters;
    Remove the remaining clusters in $S_x$ to $\theta_2$;
$P^* \leftarrow$ AssignObjectToOnlyOneCluster($S_x$, $\theta_2, \alpha_2$);
```

5.2.2 An illustrative Example

In this section, we use the same simple example that we used with DSCE in Section 5.1.3 to demonstrate how ACE works. We set $\alpha_1 = 0.8$, $\alpha_2 = 0.5$, and $k = 3$, and we run stage 1 as described in Section 5.1.3. For stage 2, the first two steps are done exactly the same way and we obtain θ_1, as shown in Table 5.2.

For the third step in stage 2, we first check the number of clusters (λ) in θ_1, and we find that $\lambda = 4$, which is larger than k. Then we apply Option B by measuring the cluster similarity S_c for clusters in θ_1 as shown in Table 5.1 and we adapte α_2 to the maximum similarity in S_c, which is equal to 0.764. We merge $\overleftarrow{c}_3$ and $\overleftarrow{c}_4$ and we updated θ_1 as shown in Table 5.8. As a result we obtain $\lambda = k = 3$. Then we calculate the membership similarity S_x as shown in Table 5.9.

Table 5.8: The result of updating θ_1 after we merge $\overleftarrow{c}_3$ and $\overleftarrow{c}_4$ by summing their objects membership similarity and result in $\overleftarrow{c}_3$

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$
x_1	0	3	0
x_2	0	3	0
x_3	0	1	2
x_4	0	0	3
x_5	0	0	3
x_6	2	1	0
x_7	3	0	0
x_8	3	0	0
x_9	3	0	0
x_{10}	0	3	0

Then we move to stage 3, by first identifying totally certain and certain objects. So, based on α_2, we identify $x_1, x_2, x_4, x_5, x_7, x_8, x_9$ and x_{10} as totally certain objects, while we identify x_3 and x_6 as certain objects. Then we assign them to the candidate cluster that has a maximum membership similarity among other candidates, and S_x is updated as follows:

Then we check whether S_x contains any uncertain objects and it does not, so we

Table 5.9: The results of S_x after no more merging step is needed.

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$
x_1	0	1	0
x_2	0	1	0
x_3	0	0.3	0.6
x_4	0	0	1
x_5	0	0	1
x_6	0.6	0.3	0
x_7	1	0	0
x_8	1	0	0
x_9	1	0	0
x_{10}	0	1	0

Table 5.10: The results of assigning totally certain and certain objects to the candidate cluster.

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$
x_1	0	1	0
x_2	0	1	0
x_3	0	0	0.6
x_4	0	0	1
x_5	0	0	1
x_6	0.6	0	0
x_7	1	0	0
x_8	1	0	0
x_9	1	0	0
x_{10}	0	1	0

produce the final clustering result $P^* = \{2, 2, 3, 3, 3, 1, 1, 1, 1, 2\}$.

Assume that we set $\alpha_2 = 0.9$, which is a high value. The number of clusters (nb_{cls}) in S_x that contain at least one certain object is equal to 2. As there is no further merging process to be done, we calculate S_x, which is shown in Table 5.10. Then we implement the elimination process that is described in Option B (steps *i* to *iv*). So, for each cluster in S_x, we calculate their certainties (using equation 5.3), and we obtain $\rho_{\overleftarrow{c_1}} = 0.9$, $\rho_{\overleftarrow{c_2}} = 0.85$, $\rho_{\overleftarrow{c_3}} = 0.6$, $\rho_{\overleftarrow{c_4}} = 0.3$. We rank these certainties in descending order and we obtain $\{0.9, 0.72, 0.6, 0.3\}$. Then we adapt

Table 5.11: The result of S_x after we perform the second stage.

	$\overleftarrow{c}_1$	$\overleftarrow{c}_2$	$\overleftarrow{c}_3$
x_1	0	1	0
x_2	0	1	0
x_3	0	0.3	0.6
x_4	0	0	0.6
x_5	0	0	0.6
x_6	0.6	0.3	0
x_7	1	0	0
x_8	1	0	0
x_9	1	0	0
x_{10}	0	1	0

α_2 to the maximum certainties of the kth clusters in this ranked list, which is equal 0.6. As result, we identify $\overleftarrow{c_1}$, $\overleftarrow{c_2}$ and $\overleftarrow{c_3}$ as candidate clusters and we eliminate $\overleftarrow{c_4}$. We update S_x accordingly as shown in Table 5.11 :

Then we move onto stage 3, and based on α_1 we identify x_1, x_2, x_7, x_8, x_9 and x_{10} as totally certain objects, and we identify other objects as uncertain objects. We measure the quality of the candidate clusters using equation 5.10 as follows:
$Var(\overleftarrow{c_1}) = \frac{1}{3}((1-0.9)^2 + (1-0.9)^2 + (1-0.9)^2) = 0.01$
$Var(\overleftarrow{c_2}) = \frac{1}{3}((1-0.72)^2 + (1-0.72)^2 + (1-0.72)^2) = 0.0784$
$Var(\overleftarrow{c_3}) = 0$

Then, we iterate on uncertain objects, and we proceed with steps (a) to (e). The detailed results of these steps for object x_3 are as follows:

(a) For each candidate cluster we recalculate its quality by including this time x_3:
$Var(\overleftarrow{c_1}) = \frac{1}{4}((1-0.9)^2 + (1-0.9)^2 + (1-0.9)^2 + (0-0.9)^2) = 0.21$
$Var(\overleftarrow{c_2}) = \frac{1}{4}((1-0.72)^2 + (1-0.72)^2 + (1-0.72)^2 + (0.3-0.72)^2) = 0.1029$
$Var(\overleftarrow{c_3}) = \frac{1}{1}((0.6-0.6)^2) = 0$

(b) We compare for each cluster the original quality and the current quality: $Var(\overleftarrow{c_1}) = 0.21 - 0.01 = 0.2$, $Var(\overleftarrow{c_2}) = 0.1029 - 0.0784 = 0.0245$,

$Var(\overleftarrow{c_3}) = 0 - 0 = 0$

(c) We assign x_3 to the cluster that has a minimum effect on its quality, that is done as follows: $min\{0.2, 0.0245, 0\} = 0$
So, we assign x_3 to cluster $\overleftarrow{c_3}$.

(d) We increase the size of $\overleftarrow{c_3}$ by 1.

(e) We update the original quality of $\overleftarrow{c_3}$ to be equal to the current quality.

After all the uncertain objects are assigned, we produce the final clustering result, which is : $P^* = \{2, 2, 3, 3, 3, 1, 1, 1, 1, 2\}$.

5.2.3 Experimental Design

Two experiments were conducted to test ACE. In the first experiment, we ran the same the experiment as we did to test DSCE (Section 5.1.4).

In the second experiment, we tested ACE under the situation where each member has a different number of clusters k chosen randomly from the interval $[k-2, k+2]$. We chose this interval because we already know the number of clusters in the tested datasets so the minimum of this interval is set to less than k by 2 and the maximum set to a value larger than k by 2.

The main aim of these experiments is to test the performance of ACE, and also to see how effective it is compared to other competitive clustering ensemble methods. Therefore, we ran the algorithm ten times, and each time the performance was measured by ARI and NMI, and at the end of these run we calculated the average performance and the standard deviation for each ensemble clustering method.

In both experiments, we set $\alpha 1 = 0.8$, $\alpha_2 = 0.7$, $\alpha_{1min} = 0.6$, and $\Delta\alpha = 0.1$. The following section includes the results and analysis of the two experiments.

5.2.4 Experimental Results

5.2.4.1 Results of Ensembles Built with Fixed k

Tables 5.12 and 5.13 show the average value of ten runs of the compared algorithms measured by ARI and NMI respectively, along with their corresponding standard deviations. The bold value in each row shows the best performance in each dataset in terms of the quality of the clustering result and the underlined number shows the best value in terms of consistency. The last column of Table 5.12 and 5.13 represents the average performance of the generated members, and the last two rows show the average quality for each ensemble method over all datasets, as well as the average consistency of each method.

Results obtained by ARI Index: There are a number of interesting observations. Firstly, the performance of ACE is better than CO-Av and ONCE-Av in five datasets, whereas it performed very closely to them on other datasets. In particular, in the Iris, Thyroid and Glass datasets, ACE produced the highest results: 0.734, 0.611 and 0.534 respectively. Secondly, ACE achieved the same performance as CO, DSCE and MCLA algorithms in the Bcw dataset, and that is the most accurate result for this dataset. Thirdly, ACE outperformed DICLENS in all datasets except in the Soybeans dataset, and we will explain later this particular situation for DICLENS. On average the DSCE algorithm achieved the best performance compared with other algorithms, followed closely by the ACE algorithm.

In terms of consistency measured by the standard deviation, ACE was the most consistent algorithm in the thyroid dataset compared with the others, and it achieved a very close value to the most consistent algorithm in the most examined datasets such as the Bcw, Mfeatures and Wine datasets. The worst performance for the ACE algorithm was on the Soybean dataset, where it achieved a value equal to 0.081 compared with other algorithms, but this is still a small value.

Looking at the average performance of the generated members, we found that all the ensemble methods outperformed the average of members in all the datasets, except DICLENS which performed lower than the average members in the Glass and Mfeatures datasets as well as ACE in the Soybeans dataset.

However, the ACE algorithm performed second best on average compared with the others, and it is close to the best performing algorithm measured by the ARI index, which is DSCE under these experimental settings.

Results obtained by NMI Index: In summary, these results are very similar to the results represented by ARI explained in the previous paragraph. The only difference is that on average the ACE achieved the best performance, along with the DSCE algorithm, measured by NMI.

Under this experimental set-up, i.e. with a fixed value for k for each dataset, ACE does not show a superiority to its predecessor DSCE, although it does in comparison to the other methods. However, it is worth noting that its predecessor DSCE has an obvious weakness, which is that it can only work with fixed k values, which limits its application on real-world problems when the true number of clusters, k, is not known in advance. That is why we extended DSCE to ACE to cope with variable numbers of clusters generated by the members. The next experiment is designed to demonstrate and compare their capability.

Table 5.12: Results of the first experiment listed in Table 5.6 updated by adding the average performance of ACE and the standard deviation of ten runs for each dataset measured by ARI Index.

	CO-Av	**ONCE-Av**	**DSCE**	**ACE**	**DICLENS**	**MCLA**	**Ave-mem**
Iris	0.725 ± 0.012	0.726 ± $\underline{0.009}$	0.732 ± 0.021	**0.734** ± 0.023	0.680 ± 0.077	0.723 ± 0.012	0.702 ± 0.038
Wine	0.369 ± 0.005	0.369 ± 0.005	**0.377** ± 0.025	0.371 ± 0.008	0.369 ± 0.005	0.372 ± $\underline{0.002}$	0.366 ± 0.004
Thyroid	0.559 ± 0.024	0.584 ± 0.044	0.609 ± 0.032	**0.613** ± $\underline{0.023}$	0.582 ± 0.044	0.563 ± 0.025	0.473 ± 0.036
Mfeatures	0.315 ± 0.006	**0.316** ± 0.005	**0.316** ± $\underline{0.004}$	0.314 ± 0.008	0.290 ± 0.069	0.308 ± 0.021	0.293 ± 0.029
Glass	0.509 ± 0.029	0.526 ± 0.030	0.528 ± 0.027	**0.535** ± 0.029	0.392 ± 0.123	0.534±$\underline{0.020}$	0.501 ± 0.009
Bcw	**0.849** ± 0.004	0.847 ± $\underline{0.003}$	**0.849** ± 0.004	**0.849** ± 0.004	0.842 ± 0.005	**0.849** ±0.004	0.830 ± 0.021
Soybean	0.547 ± $\underline{0.006}$	0.550 ± 0.015	0.578 ± 0.052	0.532 ± 0.081	**0.632** ± 0.046	0.548 ± $\underline{0.006}$	0.566 ± 0.025
Ionosphere	0.163 ± 0.014	0.166 ± 0.008	**0.169** ± $\underline{0.005}$	0.165 ± 0.008	0.161 ± 0.009	0.166 ± 0.006	0.149 ± 0.007
Ave-P	0.505	0.511	**0.520**	0.514	0.493	0.508	0.443
Ave-C	$\underline{0.012}$	0.015	0.017	0.023	0.048	$\underline{0.012}$	0.031

Table 5.13: Results of the first experiment listed in Table 5.7 updated by adding the average performance of ACE and the standard deviation of ten runs for each dataset measured by NMI Index.

	CO-Av	**ONCE-Av**	**DSCE**	**ACE**	**DICLENS**	**MCLA**	**Ave-mem**
Iris	0.751 ± 0.015	0.752 ± 0.012	0.763 ± 0.024	**0.766** ± 0.028	0.757 ± 0.008	0.749 ± 0.015	0.737 ± 0.025
Wine	0.428 ± 0.003	0.428 ± 0.003	**0.432** ± 0.014	0.429 ± 0.006	0.427 ± 0.004	0.429 ± 0.001	0.428 ± 0.003
Thyroid	0.434 ± 0.047	0.473 ± 0.062	0.480 ± 0.056	**0.531** ± 0.042	0.501 ± 0.053	0.418 ± 0.033	0.403 ± 0.026
Mfeatures	**0.479** ± 0.002	**0.479** ± 0.003	**0.479** ± 0.002	0.478 ± 0.007	0.468 ± 0.026	0.475 ± 0.009	0.460 ± 0.027
Glass	0.712 ± 0.027	0.725 ± 0.029	0.725 ± 0.021	0.726 ± 0.022	0.617 ± 0.107	**0.728** ± 0.017	0.704 ± 0.007
Bcw	0.750 ± 0.005	0.749 ± 0.004	0.750 ± 0.005	**0.751** ± 0.005	0.742 ± 0.006	**0.751** ± 0.005	0.731 ± 0.023
Soybean	0.717 ± 0.002	0.723 ± 0.024	0.756 ± 0.064	0.712 ± 0.076	**0.822** ± 0.056	0.717 ± 0.002	0.736 ± 0.019
Ionosphere	0.122 ± 0.014	0.124 ± 0.009	**0.128** ± 0.005	0.123 ± 0.008	0.119 ± 0.009	0.124 ± 0.006	0.108 ± 0.006
Ave-P	0.549	0.557	**0.564**	**0.564**	0.557	0.549	0.491
Ave-C	0.015	0.018	0.024	0.024	0.034	0.011	0.017

5.2.4.2 Results of Ensembles Built with Random Variable k

We did not run the DSCE algorithm in this experimental set-up as it is not capable of dealing with variable numbers of clusters generated by the members in an ensemble. All the other methods were run for comparison.

Results obtained by ARI Index: Table 5.14 shows the average performance measured by the ARI index along with the standard deviation in each dataset, and the average performance of the generated members. The results indicate that the ACE algorithm usually performs better than the other clustering ensemble algorithms. This is particularly true for five datasets, which are Wine, Glass, Bcw, Soybean and Ionosphere, whereas in Iris, Thyroid and Mfeatures it achieved a result close to the highest performance in these datasets, which was achieved by ONCE in Mfeatures and MCLA in the other two datasets. However, the result on the Mfeatures dataset indicates that ACE is applicable to a large dataset.

ACE also enhances the performance of the generated members in all investigated datasets except the Ionosphere dataset, which is slightly better than the clustering ensemble algorithms; this may be due to random k in these members.

In terms of consistency, ACE was more consistent in two datasets, which are Glass and Bcw, while in the Iris, Wine and Ionosphere datasets it was the second most consistent algorithm compared with other algorithms. On average, three algorithms achieved very close results in terms of consistency; these are MCLA, ONCE and ACE, which are equal to 0.035, 0.037 and 0.038 respectively.

Results obtained by NMI Index: Similar experimental results are also observed using NMI index shown in Table 5.15, where ACE achieved the highest performance on three datasets Iris, Bcw, and Ionosphere. However, with Wine, Mfeatures and Glass it achieved results very close to the highest performance. In the Soybean dataset the highest performance was achieved by the DICLENS algorithm, which also performed very well with the Wine and Mfeatures datasets. These results were only achieved by the NMI index and not the ARI index, which leads us to investigate further the number of clusters discovered by DICLENS, as it has the ability to discover k automatically. This is in contrast to other clustering ensemble algorithms examined, in which k is provided by the user in advance.

Identifying the true number of clusters in DICLENS

Figure 5.8 shows the number of clusters discovered by the DICLENS algorithm in all tested datasets over ten runs for the results of the second experiment (in Section 5.2.4.2). It is observed that the number of clusters in most datasets is unstable and changeable over the ten runs. This has an effect on the NMI index, which is an information theory based index that measures the shared information between two clustering results. Most of the DICLENS results in the majority of datasets had fewer number of clusters than the actual number of clusters (the ground-truth labels) in the data. It is clear that one cluster produced by DICLENS can share a number of objects with more than two true clusters and that can lead the NMI result to be increased.

For example, it was highlighted for the Wine dataset over the ten runs that the discovered k was equal to 2 which is less than the number of the true labels, 3. Therefore, the NMI measure, as it is based on how much information the compared clustering results share, unfairly indicates that this result is more accurate than

ACE. Moreover, in the Soybean dataset the discovered k was equal to 2 in three runs, 3 in four runs and 4 in the remaining three runs, whereas the number of the true labels is equal to 4. It is obvious that fewer clusters shared more objects with more true clusters in this case, and the NMI scored higher than ARI compared with other clustering results obtained by other algorithms. It is observed that when the number of clusters in the compared results is less than the number of true labels of the data, the NMI measure inappropriately indicates that this result is more accurate than others that have produced exactly the number of the true clusters.

Table 5.14: Second experiment results: the average performance and the standard deviation of ten runs for each dataset measured by ARI. Includes the average performance of each ensemble method across 8 datasets.

	CO-Av	**ONCE-Av**	**ACE**	**DICLENS**	**MCLA**	**Ave-mem**
Iris	0.669 ± 0.065	0.674 ± 0.057	0.696 ± 0.038	0.565 ± 0.009	$\mathbf{0.722} \pm 0.043$	0.605 ± 0.029
Wine	0.324 ± 0.045	0.344 ± 0.060	$\mathbf{0.403} \pm 0.014$	0.367 ± 0.024	0.393 ± 0.008	0.326 ± 0.011
Thyroid	0.252 ± 0.175	0.189 ± 0.121	0.303 ± 0.032	0.308 ± 0.118	$\mathbf{0.448} \pm 0.119$	0.285 ± 0.053
Mfeatures	0.325 ± 0.002	$\mathbf{0.326} \pm 0.001$	0.325 ± 0.005	0.324 ± 0.006	0.277 ± 0.013	0.321 ± 0.005
Glass	0.265 ± 0.006	0.259 ± 0.008	$\mathbf{0.269} \pm 0.004$	0.200 ± 0.048	0.152 ± 0.022	0.258 ± 0.005
Bcw	0.866 ± 0.018	0.860 ± 0.016	$\mathbf{0.869} \pm 0.014$	0.853 ± 0.031	0.864 ± 0.014	0.773 ± 0.037
Soybean	0.534 ± 0.000	0.534 ± 0.000	$\mathbf{0.578} \pm 0.160$	0.575 ± 0.070	0.547 ± 0.039	0.547 ± 0.036
Ionosphere	0.076 ± 0.047	0.037 ± 0.035	$\mathbf{0.084} \pm 0.034$	0.076 ± 0.039	0.061 ± 0.019	0.117 ± 0.014
Ave-P	0.414	0.403	**0.441**	0.409	0.433	0.404
Ave-C	0.045	0.037	0.038	0.043	**0.035**	0.024

Table 5.15: Second experiment results: the average performance and the standard deviation of ten runs for each dataset measured by NMI. Including the average performance of each ensemble method across 8 datasets.

	CO-Av	**ONCE-Av**	**ACE**	**DICLENS**	**MCLA**	**Ave-mem**
Iris	0.753 ± 0.017	0.749 ± 0.027	$\mathbf{0.766} \pm 0.032$	0.753 ± 0.026	0.755 ± 0.037	0.706 ± 0.012
Wine	0.406 ± 0.010	0.415 ± 0.022	0.421 ± 0.014	$\mathbf{0.435} \pm 0.018$	0.415 ± 0.005	0.410 ± 0.010
Thyroid	0.293 ± 0.077	0.250 ± 0.066	0.308 ± 0.050	0.331 ± 0.040	$\mathbf{0.356} \pm 0.048$	0.302 ± 0.035
Mfeatures	0.486 ± 0.002	0.487 ± 0.002	0.490 ± 0.008	$\mathbf{0.493} \pm 0.005$	0.464 ± 0.007	0.484 ± 0.005
Glass	0.441 ± 0.018	$\mathbf{0.449} \pm 0.016$	0.430 ± 0.016	0.389 ± 0.032	0.307 ± 0.032	0.423 ± 0.011
Bcw	0.773 ± 0.024	0.765 ± 0.021	$\mathbf{0.776} \pm 0.019$	0.759 ± 0.032	0.770 ± 0.019	0.687 ± 0.028
Soybeans	0.710 ± 0.000	0.710 ± 0.000	0.722 ± 0.127	$\mathbf{0.767} \pm 0.070$	0.716 ± 0.018	0.734 ± 0.020
Ionosphere	0.043 ± 0.035	0.023 ± 0.012	$\mathbf{0.048} \pm 0.026$	0.043 ± 0.029	0.030 ± 0.013	0.099 ± 0.016
Ave-P	0.488	0.481	0.495	**0.496**	0.477	0.480
Ave-C	0.023	**0.021**	0.036	0.032	0.022	0.017

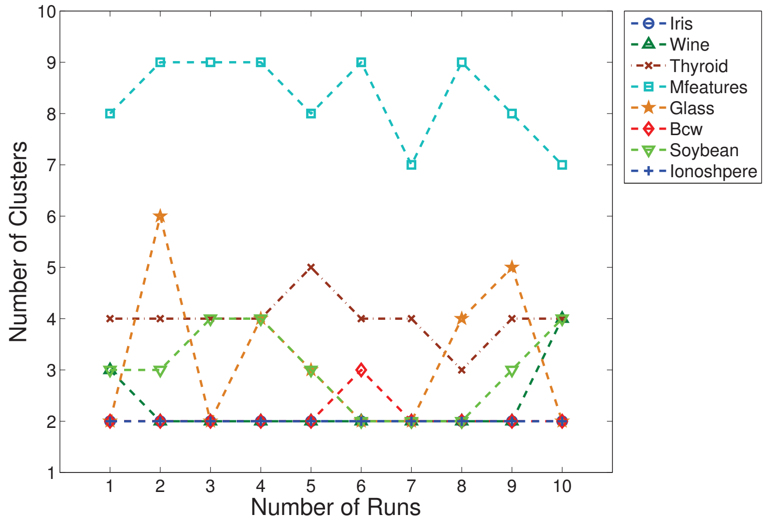


Figure 5.8: Number of clusters produced by DICLENS algorithm for each dataset in ten runs for the result in the second experiment. The true number of clusters for {Iris, Wine, Thyroid} = 3, Mfeatures= 10, Glass = 6, Bcw = 2, Soybean = 4, Ionosphere = 2.

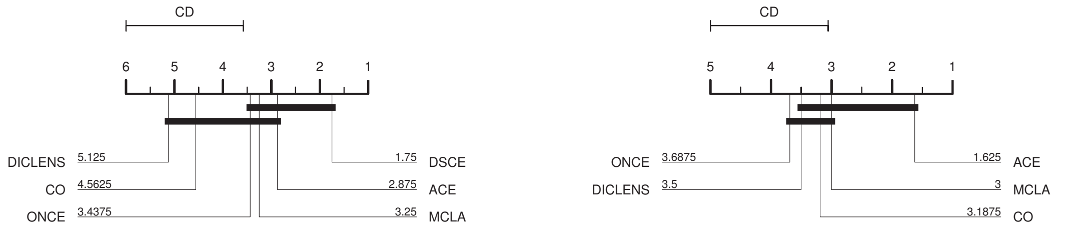
5.2.5 Test of Significance

We tested the statistical significance of the results of the two experiments that we performed in Sections 5.2.4.1 and 5.2.4.2 on the two types of ensemble.

We applied the Iman-Davenport test [53] to the results in Tables 5.12 and 5.14 under the null hypothesis that the mean ranks are equal for all the examined algorithms. The significant level is set to 0.1 by default. For the first experiment, we can reject the null hypothesis of the mean rank of the performance being equal for all algorithms (the Iman-Davenport test result is equal to 4.4051 which gives a small p-value equal to 0.0032, which indicates that there is a significant difference). For the second experiment in Table 5.14, the Iman-Davenport test result is equal to 2.5434, which gave a small p-value equal to 0.0617, indicating that there is a significant difference.

Therefore, we proceeded with the Nemenyi test as a post-hoc test for a pairwise comparison to discover where the differences lie. Figure 5.9(a) shows the critical difference diagram at the critical level of 0.1 for the results presented in Table 5.12, and the critical difference, CD, is equal to 2.4218. As we can see from the diagram, we have two solid bars which show two groups of algorithms in cliques, indicating that there is no statistically significant difference between algorithms in the same group, whereas there is a significant difference between algorithms in the different groups. We observed that, based on the average ranks, DSCE was first followed by ACE and then MCLA. Moreover, DICLENS was last in this average ranking. This demonstrated that the performance of DSCE is significantly better than CO and DICLENE based on this experimental set-up.

Figure 5.9(b) shows the critical difference diagram of the results presented in Table 5.14. As we can see, there are two groups of algorithms in two cliques. The first group includes ACE, MCLA, CO and DICLENS, whereas the second group includes MCLA, CO, DICLENS and ONCE. The results indicate that there is a significant difference between algorithms placed in different groups, and in this case between the ACE and ONCE algorithms, in this experimental set-up, although ACE is ranked the first with a considerable distance from the second algorithm, MCLA.



(a) The critical difference diagram of the first experiment.

(b) The critical difference diagram of the second experiment.

Figure 5.9: The Critical difference diagram of the critical level of 0.1 in which it shows the comparison of six ensemble methods using eight datasets.

5.2.6 Analysis of Parameters and Time Complexity

There are two parameters in ACE, which are α_1 and α_2. α_1, as stated previously, is the minimum similarity allowed between initial clusters, whereas α_2 is the certainty threshold of classifying objects in a cluster.

To find out how these parameters can affect the quality of the final clustering result of ACE, we analyse them with the two types of ensembles using Wine, Mfeatures and Glass datasets as an illustration. For the second type of ensemble, we allow for α_1 to take more values than its values in the first experiment, due to the fact that when the members have different k from one another they are more dissimilar than when they have fixed k. Therefore α_1 can take a value between 0.5 and 0.9 in the first experiment, whereas in the second experiment it takes a value between 0.3 and 0.9.

In the first experiment, we ran ACE for ten times with a different initial values of α_1, and each one of them with all the possible values for α_2. We firstly ran the *k-means* algorithm to generate ten members all with the fixed k equal to the true number of classes for each dataset. Figure 5.10 illustrates the effect of different values of α_1 and α_2 on the average ARI performance of the ensembles built by members with a fixed k, over ten runs. We note that on the Wine dataset the average performance of ACE is the same for all values of α_1 and α_2; this indicates that the ACE is not sensitive to its parameters. In the Mfeatures dataset, the average performance of ACE is slightly improved when α_1 is equal to 0.8 and 0.9. We note that all the values of α_2 have the same performance with all the values of α_1. The average performance of ACE in the Glass dataset is the same when α_1 is equal to 0.5 and 0.6, which is slightly improved when α_1 is equal to 0.7 and 0.9; when it is equal to 0.8 it reaches its highest performance.

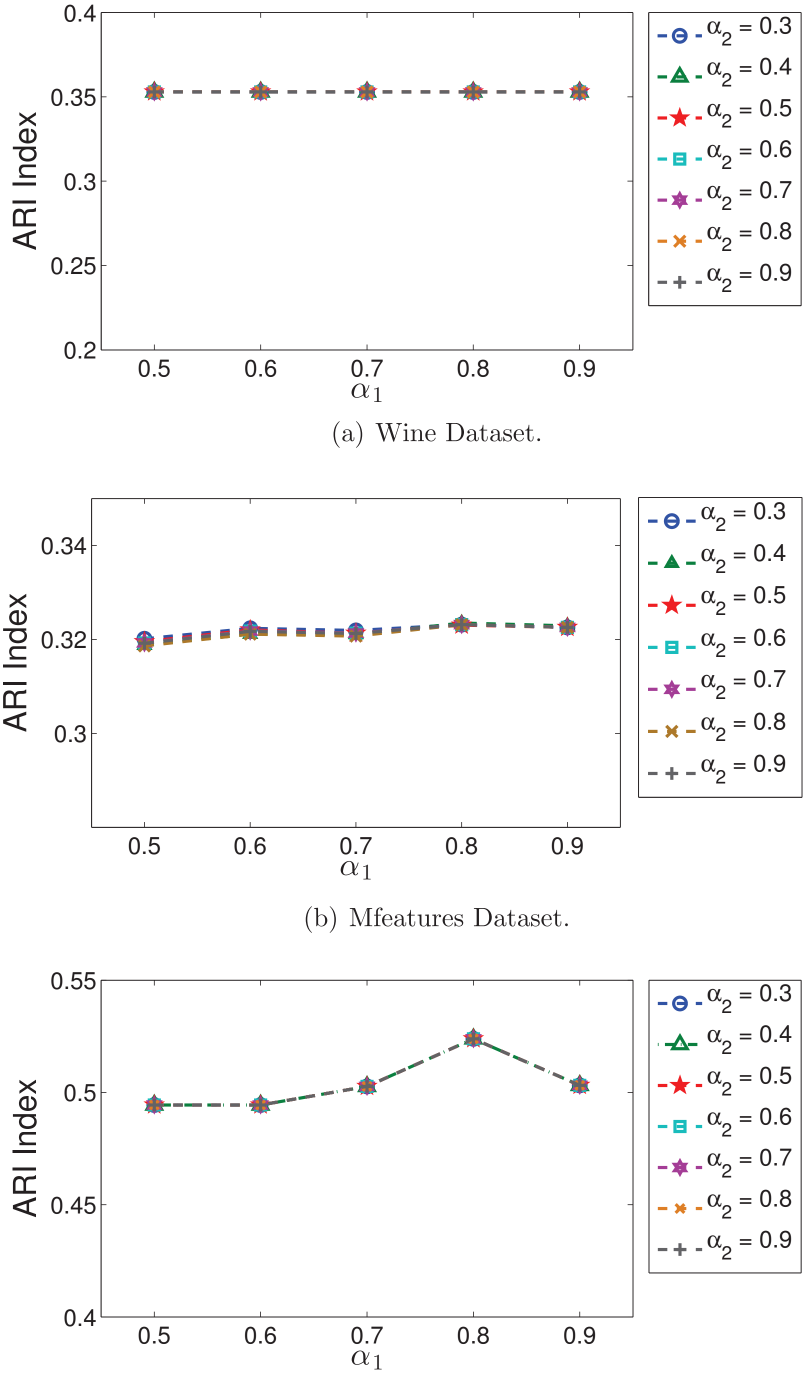
We note that all values of α_2 achieved the same performance with all values of α_1 in all the examined datasets, this indicates that the different values of α_2 have no effect on the performance of the ACE when it is built with members that have a fixed k.

On the other hand, Figure 5.11 illustrates the effect of the different values of two parameters on the average performance of the ACE ensemble built with members having a random variable k. We can see that in the Wine dataset the ACE performance is decreased a little when α_1 is equal to 0.7 in which case the performance remains stable with 0.8 and 0.9 in all possible values of α_2. In the Mfeatures dataset, the ACE performance is slightly improved when α_1 is less than 0.7. However, in the Glass dataset the ACE performance fluctuates with a slight increase to reach a value of 0.6 and then a slight drop when α_1 is equal to 0.7 after a stable performance. We note that with all the possible values of α_2 that the average performance of ACE remains the same in almost all cases for α_1.

Therefore, the results suggested that α_2 has no effect on the performance of ACE, and α_1 has a slight effect on ACE performance. A value between 0.6 and 0.8 is better for an ensemble built with fixed k, whereas a value between 0.3 and 0.5 is better for an ensemble built with different k and when α_2 is between 0.5 to 0.9, as these values have no effect on the ACE performance.

The time complexity for the worst-case scenario of ACE algorithm is estimated to be $O(k_m^2(k_m + n_u))$, where k_m is the total number of clusters in all the generated members, and n_u is the number of uncertain objects which is in the worst case scenario equal to $(n_u = n - k)$, n is the number of objects in the dataset and k is the number of pre-defined clusters for the dataset. As can be seen, this time complexity is better than that of DSCE (i.e. $O(k^2m^2n_u)$). We observed that the actual running time for Mfeatures dataset (which is the biggest size dataset we had in our experiment) to produce the result by DSCE = 0.713, CO = 2.419, ONCE = 4.961 and ACE = 0.159 measured in seconds [1]. As we can see ACE is faster than other methods. Hence, for big real-world datasets ACE holds some promise.

[1] We ran our experiment using Apple Macintosh computer 2.3 GHz Intel Core $i5$ with 8 GB installed RAM

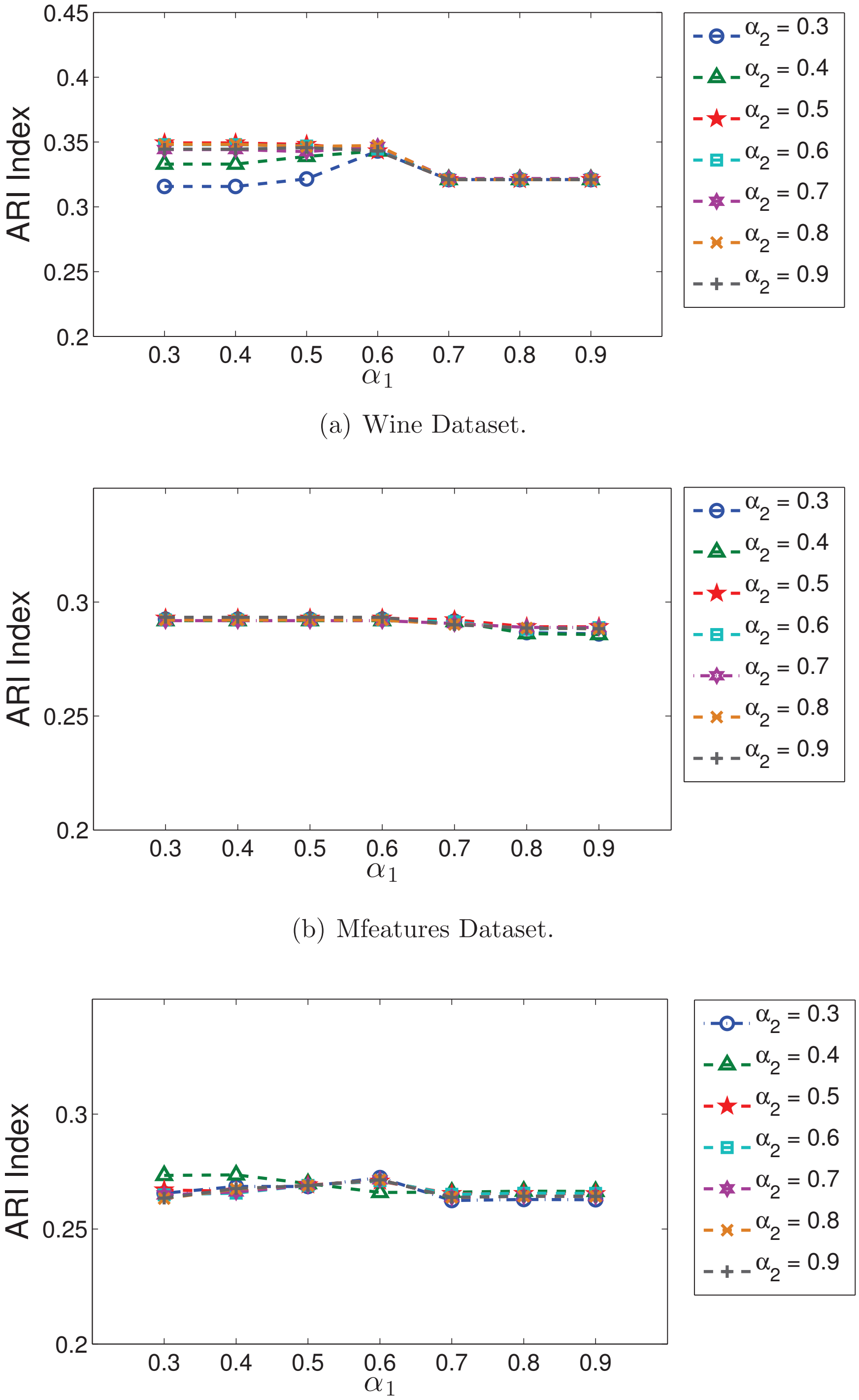


(a) Wine Dataset.

(b) Mfeatures Dataset.

(c) Glass Dataset.

Figure 5.10: The Average of ARI index of ten runs for analysing the two parameters α_1 and α_2 using members with fixed k.



(a) Wine Dataset.

(b) Mfeatures Dataset.

(c) Glass Dataset.

Figure 5.11: The Average of ARI index of ten runs for analysing the two parameters α_1 and α_2 using members with random k.

5.3 Summary

In this chapter, the aim was to propose a consensus function that incorporate the similarity from two different levels, at an object level and cluster level, and it does not require an ordinary clustering algorithm as a final step to produce the final clustering label. As a result, two consensus functions were proposed, named the Dual-Similarity Clustering Ensemble (DSCE) and the Adaptive Clustering Ensemble (ACE).

There are a number of advantages to these two new clustering ensemble methods:

1. DSCE and ACE avoid cluster relabelling problems when aggregating the ensemble members.
2. DSCE and ACE utilise the information on the similarity between clusters and the membership of objects to clusters in order to generate consensus clusters.
3. DSCE does not restrict the produced clustering solution to having a specific number of clusters k, and it converges k to a stable value from the generated member.
4. ACE is able to deal with any generated ensemble members, even when they have different numbers of clusters, as ACE converts them exactly or very closely to the true number of clusters in the final clustering result.
5. ACE resolves the objects' uncertainty by considering their object neighbourhood similarity in order to not lose any information when an inappropriate cluster is eliminated.
6. DSCE and ACE are more efficient. Instead of calculating the similarity between objects like the others do, they calculate the similarity between the initial clusters of the ensemble members, which is much smaller than the number of objects, and they do not require a single clustering algorithm to be applied over the similarity matrix to produce the final clustering results. Hence, DSEC and ACE have potential to be applied in big data clustering problems.

7. ACE is more stable due to the different values of the two parameters (α_1 and α_2). The experimental analysis revealed that α_2 has no effect on the ACE performance, and α_1 has a slight effect on ACE performance.

DSCE and ACE were tested using 8 real-world datasets. The first experiment was designed to test DSCE, and the results demonstrated that on average DSCE outperforms the state-of-the-art cluster ensemble algorithms, which the MCLA, CO, DICLENS algorithms, and our early method ONCE. It has been proven that DSCE is statistically different from the CO and DICLENS clustering ensemble methods.

However, the same experiment was conducted to test ACE, and the results showed that on average ACE outperforms the other clustering ensemble methods, and comparing ACE with its predecessor DSCE, it achieved a very close performance. Moreover, we tested ACE in the situation where the generated members had different numbers of clusters, and the results showed that on average ACE is better than the other clustering ensemble methods.

Chapter 6

The Diversity of the Clustering Ensemble.

In this chapter, we focus on the second central part of this thesis by trying to answer the following question: Does diversity influence the ensemble performance? To do that, in Section 6.1, we conduct an experimental study to investigate the influence of diversity on the ensemble quality using a number of consensus functions. The results of this experimental study raised two issues. Firstly, the results showed that diversity can have a positive or negative effect on the ensemble performance. Secondly, the results revealed there may be an interaction between diversity and the members' quality. Thus, in Section 6.2 we investigate these two raised issues. In Section 6.3, we discuss our investigation of ensemble diversity and the results of our analysis on the two issues raised regarding diversity. Finally, in Section 6.4, we summarise the main findings of this chapter.

6.1 Experimental Studies on Clustering Ensemble Diversity

The main aim of this experiment is to investigate whether or not the diversity has an influence on the ensemble performance using the current diversity measures described in Section 2.3.1.

6.1.1 Experimental Design and Procedure

In our experiment, we used 8 real datasets, including Wine, Iris, Glass, Bcw, Mfeatures, Soybean and Ionosphere datasets; Table 3.1 in Chapter 3 shows the details of these datasets. The experiment was performed as follows:

1. For each dataset, we generated 5 sets of members. The first four sets were generated with the Homogeneous generation strategy, whereas the last set was generated using the Heterogeneous generation strategy. For each one of the four sets, we generated 5 members, and for the final set, 7 members were generated. Thus, in total, we generated 27 members. These were:

   (a) Using *k-means* with random initialisation for the initial centroids with the predefined k value (number of clusters) for each dataset for all members (Homogeneous Ensemble).

   (b) Using *k-means* with random k for each member chosen from the interval $[k-2, k+2]$ (Homogeneous Ensemble).

   (c) Using *k-means* with random k for each member chosen from the interval $[2, \sqrt{n}]$ (Homogeneous Ensemble).

   (d) Using *k-means* with random features (Homogeneous Ensemble).

   (e) For the heterogeneous generation methods, we used different algorithms with a predefined k value; these are: agglomerative hierarchical clustering using single, complete and average linkage, *k-medoids*, *c-means*, kernel *k-means* [91], and the normalised cut algorithm [92].

2. We combined the generated members using the 4 consensus functions CO-Av and ONCE-Av, MCLA and ACE. So for each dataset, we constructed 4 final clustering results, P^*_{CO}, P^*_{ONCE}, P^*_{MCLA} and P^*_{ACE}.
3. We calculated the $Q(\Phi)$ for each consensus function's results, and $Q(\Gamma)$ both using the ARI index.
4. We measured the diversity of the generated members using 7 definitions: DV_{pARI}, DV_{pNMI}, $Entropy$, DV_{np1}, DV_{np2}, DV_{np3} and DV_{np4}.
5. We repeated the above steps (1-4) 100 times.

In each run for one dataset we have 4 ensemble results, and for 8 datasets, we have 32 ensemble results. Thus, 32 × 100 clustering ensemble results have been generated in total.

6.1.2 Experimental Results

The results of each dataset are stored in a Table, for a total of 8 tables. These tables are too large to be presented here, since each table has 100 rows; each row represents one run of the experiments and includes 24 columns, where 19 columns represent the values for the diversity measures plus 4 values of the consensus function quality results measured by ARI plus the average quality of the members also measured by ARI. Thus, in total, each table has 100 rows and 24 columns.

The statistical summary of the ensemble quality results for each dataset, the qualities of their generated members measured by the ARI and the diversity measure results are presented in appendix A. Then we plot the correlation between all diversity measures and the ensemble method qualities as well as the correlation between all the diversity measures and the average quality of the members in 100 runs for each dataset and for each ensemble method used. To assess how the diversity actually correlates with the ensemble quality, we carried out a correlation coefficient test for the 8 datasets, as presented in Section 6.1.3. However, in this section we only highlight and summaries the most remarkable results, whereas the full details of this experiment results are shown in Appendix A.

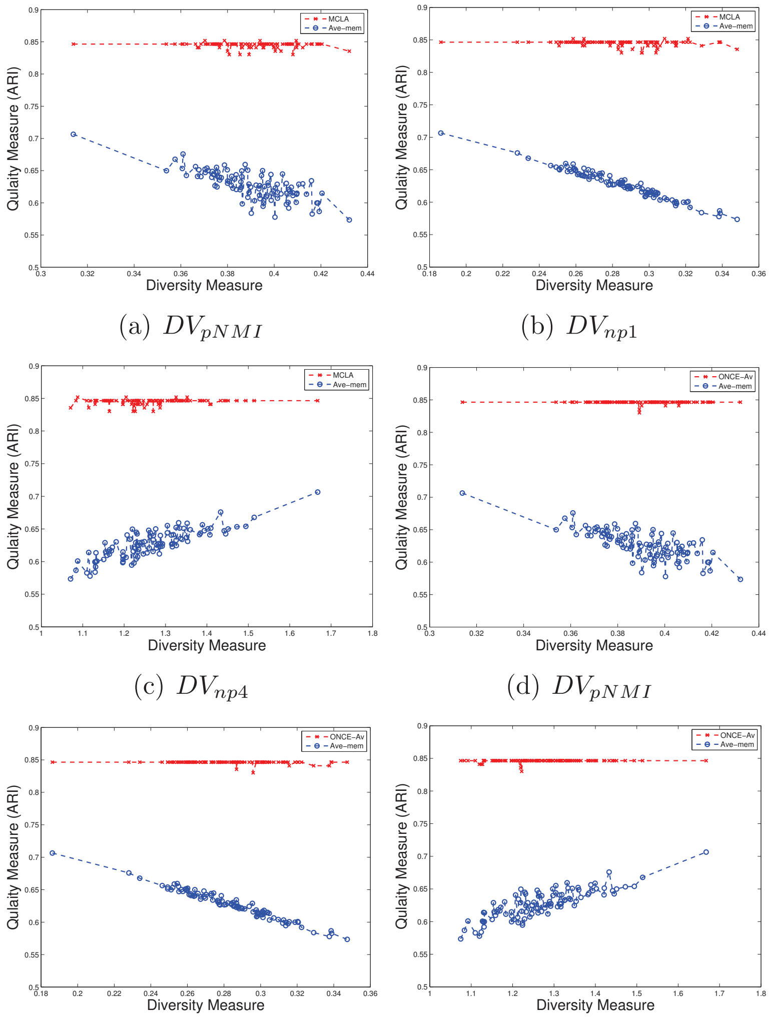
In the figures below, we sort the results in ascending order with respect to a specific diversity measure, and for each of the consensus function results, we plot its ensemble quality in the y axis against the sorted diversity measure in the x axis (represented by the symbol x and a red dashed line), and we also plot the average quality of the members (represented by the symbol o and a blue dashed line) against the diversity measure.

In the relation between the ensemble quality and the diversity, we found that there are two types of patterns in the results, these are:

1. The first pattern shows the diversity has no effect on the ensemble quality, as the ensemble qualities remain slightly stable along all diversity scores and even when the average member quality decreases. This pattern was discovered in the Bcw and Ionosp datasets in all the tested ensemble methods and in the Iris and Soybeans datasets using the CO and MCLA methods.
2. The second pattern shows that the ensemble qualities are fluctuating over the diversity, where there is no consistent trend that can be visually be identified from them. In this type, we have Glass, Wine, Mfeatures and Thyroid in all used ensemble methods and Soybean using only the ONCE and ACE ensemble methods.

In the first pattern for example, we have Figures 6.1 and 6.2. We think the reason behind this is that generating more diverse members caused them to be poor in quality. As the diversity in these datasets was not very high, these poor quality members were very few, which is why they did not affect the quality of the ensembles. We checked the first 10 runs in Bcw dataset we found that the number of poor members (their qualities below 0.3) are between 5 to 7 out of 27 members in one run.

The first pattern is also discovered in the Iris dataset (Figure 6.3), and it is



(a) DV_{pNMI} (b) DV_{np1}

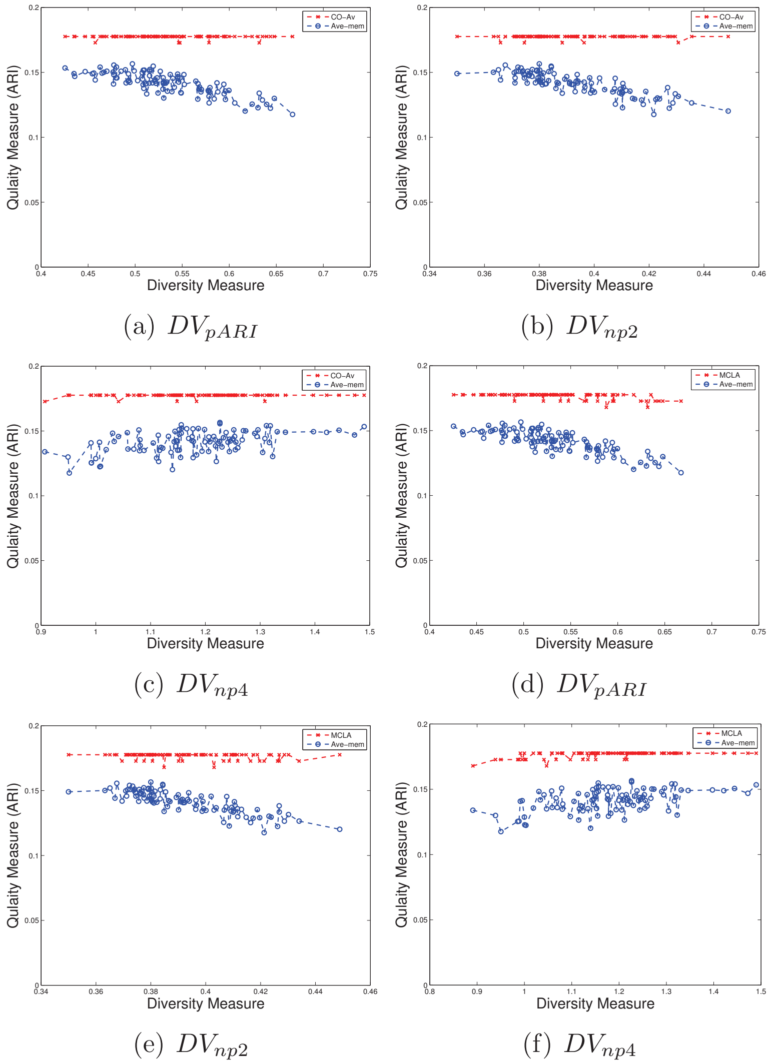
(c) DV_{np4} (d) DV_{pNMI}

(e) DV_{np1} (f) DV_{np4}

Figure 6.1: The DV_{pNMI}, DV_{np1} and DV_{np4} measures from Bcw dataset. (a), (b) and (c) using MCLA, while (d), (e) and (f) using ONCE-Av.

noticed that for a few ensemble cases, when the average members is improved the ensemble quality is also improved, which means that the ensemble improvement may be contributed by the high average member quality and not by the diversity. On the other hand, there are some cases where the ensemble improvements are affected by the diversity, as their average member quality are very low compare to others.

With the Soybeans dataset, most of the ensemble qualities of CO (Figures 6.4(a)) and 6.4(b)) are somewhat stable along the diversity in all the used diversity measures, where in most cases, the average member quality is slightly better than or close to the ensemble quality. This indicates that in some datasets, some ensemble



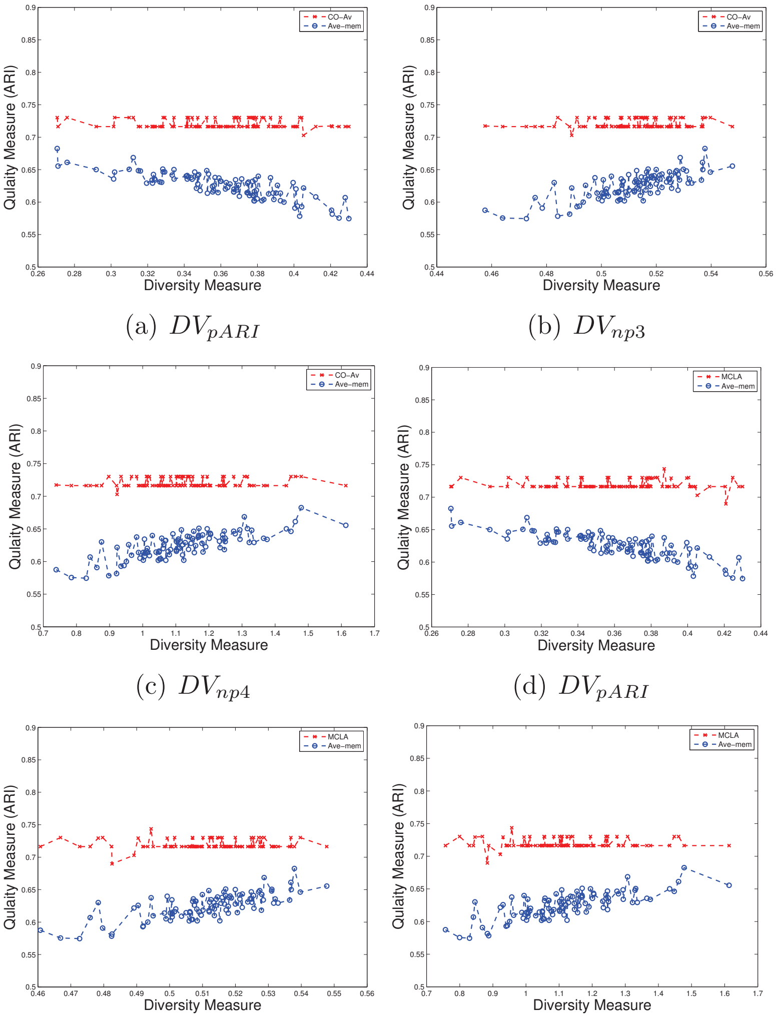
(a) DV_{pARI} (b) DV_{np2}

(c) DV_{np4} (d) DV_{pARI}

(e) DV_{np2} (f) DV_{np4}

Figure 6.2: The DV_{pARI}, DV_{np2} and DV_{np4} measures from Ionosphere dataset (a),(b) and (c) using CO-Av., while (d), (e) and (f) using MCLA.

methods can perform as equal as the performance of the individual members or even worse than them.

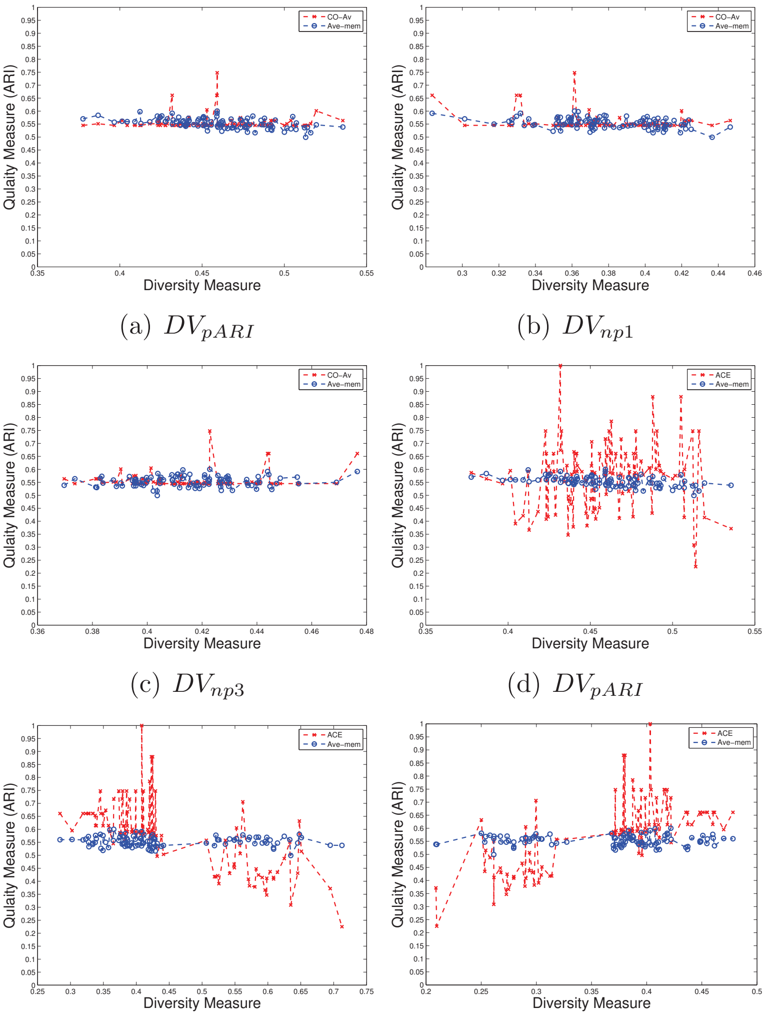


(a) DV_{pARI} (b) DV_{np3}

(c) DV_{np4} (d) DV_{pARI}

(e) DV_{np3} (f) DV_{np4}

Figure 6.3: The DV_{pARI}, DV_{np3} and DV_{np4} measures from Iris dataset, (a), (b) and (c) using CO-Av, while (d), (e) and (f) using MCLA.



(a) DV_{pARI}

(b) DV_{np1}

(c) DV_{np3}

(d) DV_{pARI}

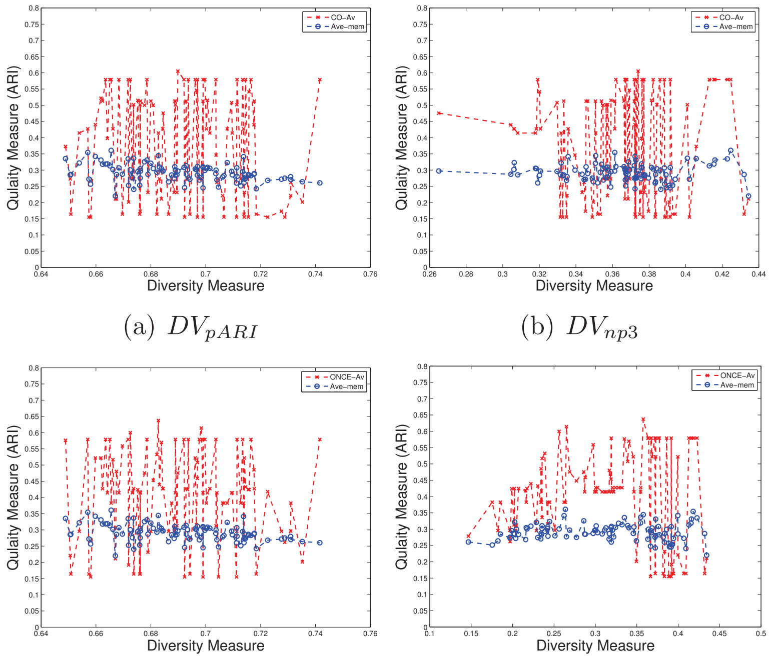
(e) DV_{np1}

(f) DV_{np3}

Figure 6.4: The DV_{pARI}, DV_{np1} and DV_{np3} measures from Soybean dataset , (a), (b) and (c) using CO-Av, while (d), (e) and (f) using ACE.

In the second pattern, we noticed that the diversity is slightly higher than the first pattern, measured by the pairwise diversity measure, and we think the high diversity might cause the fluctuation in this pattern.

With the Thyroid dataset (Figures 6.5 and 6.6), the pattern fluctuates most compared to all the other datasets. Most ensemble cases have a high quality over the diversity measured by DV_{pARI}, DV_{pNMI} and $Entropy$, while in other ensemble cases, the quality of the CO and ACE ensembles is lower than the average quality of the members compared to a few cases in the MCLA and ONCE ensemble methods. But generally, there are no members which have a diversity lower than 0.5 measured by DV_{pARI}, DV_{pNMI}, $Entropy$ and DV_{np1}. Thus, we think that the fluctuation in ensemble diversity is caused roughly by a high level of diversity compared to other datasets. The highest quality in this dataset was achieved by MCLA, accompanied by a high diversity measured by DV_{pARI}, DV_{pNMI} and $Entropy$.



(a) DV_{pARI}

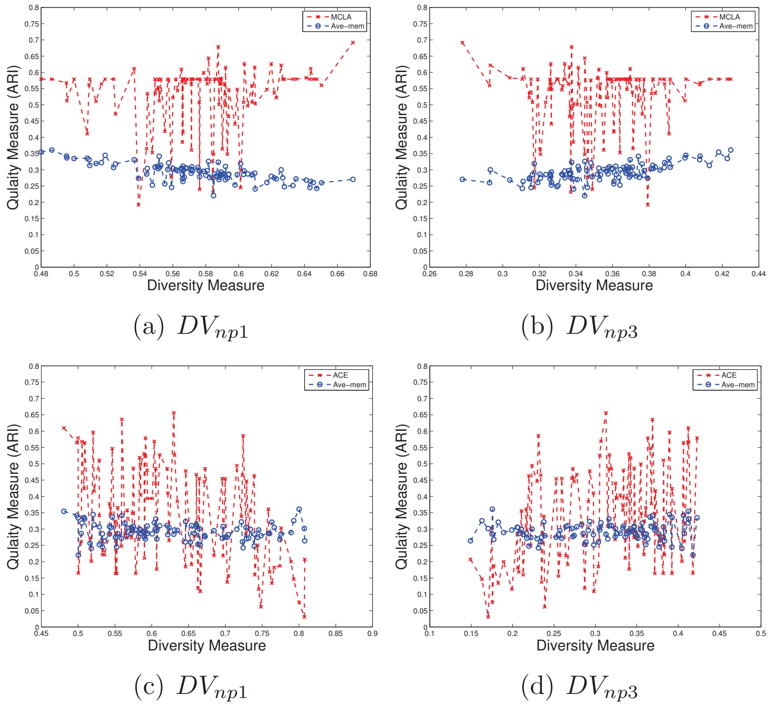
(b) DV_{np3}

(c) DV_{pARI}

(d) DV_{np3}

Figure 6.5: The DV_{pARI} and DV_{np3} measures from Thyroid dataset, (a) and (b) using CO-Ave, while (c) and (d) using ONCE-Av.

With the Wine dataset, the pattern discovered by using the ACE methods (Figure 6.7) reaches a peak value where the quality of the ensembles is equal to (0.706),



(a) DV_{np1}

(b) DV_{np3}

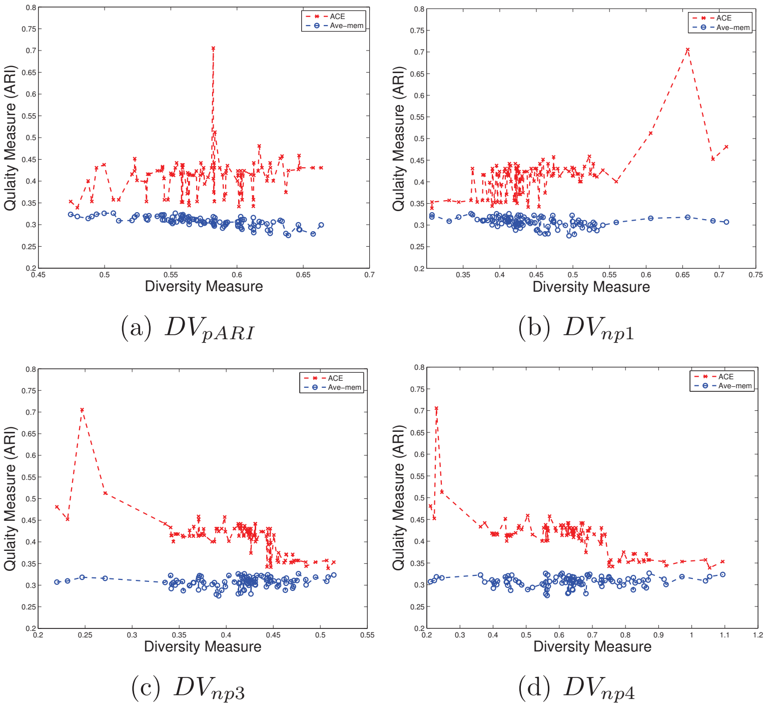
(c) DV_{np1}

(d) DV_{np3}

Figure 6.6: The DV_{np1} and DV_{np3} measures from Thyroid dataset, (a) and (b) using MCLA, while (c) and (d) using ACE.

accompanied by a low level of diversity measured by DV_{np2}, DV_{np3} and DV_{np4} while accompanied by a moderate level of diversity measured by the rest of the measurements. Looking to the highest maximum member quality that occurs in the Wine dataset, it reaches a value only equal to 0.601, indicating that this peak case is not caused by the high quality members, and in fact, is affected by their diversity. This phenomenon might have happened because each member had made a different error from one another in terms of the cluster structure in the dataset and with moderate diversity. This makes integrating them using ACE more accurate. The other ensemble cases using the ACE method show a fluctuating pattern along the diversity scores.

With the Glass dataset, the ensemble qualities are also distributed evenly along the diversity measured by DV_{pARI}, DV_{pNMI}, and $Entropy$ in all ensemble methods, and the pattern has a slight fluctuation in all ensemble methods except MCLA. Figure 6.8 shows the ensemble qualities of CO-Ave and MCLA, and the diversity measured by $Entropy$, DV_{np1} and DV_{np3} in Glass dataset. It is noticed that the



(a) DV_{pARI} (b) DV_{np1}

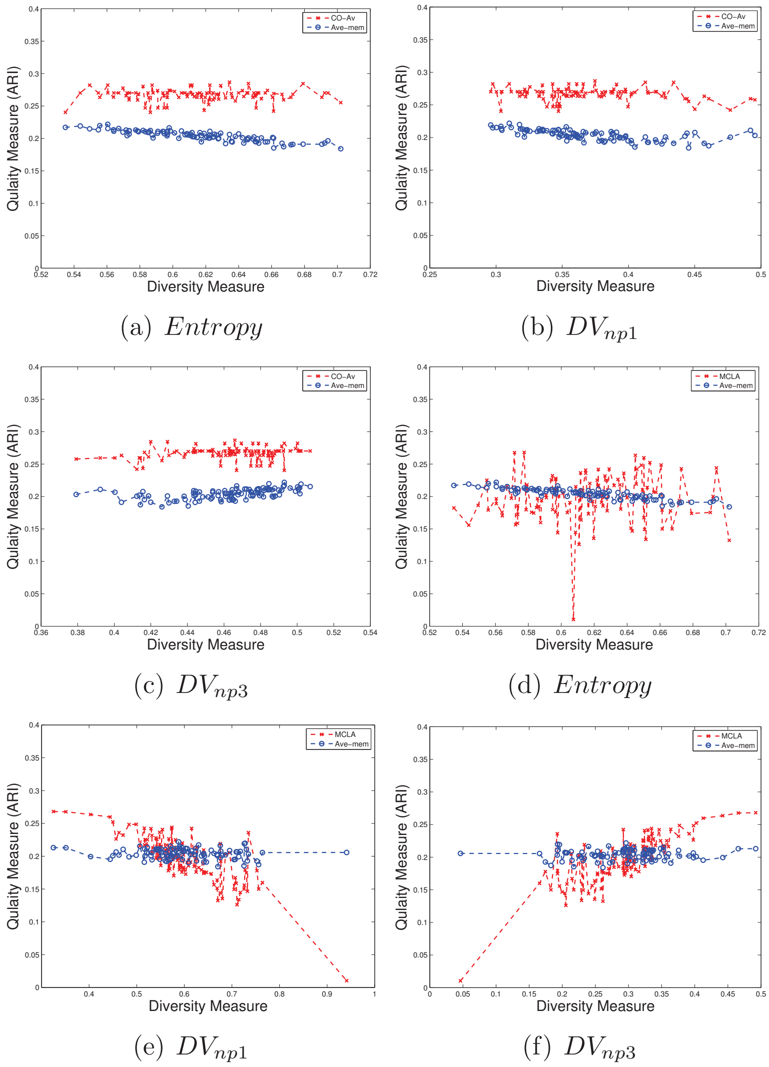
(c) DV_{np3} (d) DV_{np4}

Figure 6.7: The DV_{pARI}, DV_{np1}, DV_{np3} and DV_{np4} measures from Wine dataset using ACE.

MCLA ensemble (Figures 6.8(d), 6.8(e) and 6.8(f)) performs badly in most cases compared to the average quality of the members. Also, it shows a negative correlation with the diversity measured by DV_{np1} (Figure 6.8(e)), whereas it shows positive correlation with the diversity measured by DV_{np3} Figure 6.8(f). The positive correlation is shown with diversity measured by DV_{np2} and DV_{np4} (Figure A.14 in Appendix A).

With the Mfeatures dataset, it is noticed in Figure 6.9 that as the diversity reaches a high level, the quality of the ensemble using the MCLA method drops to a low value compared to the average member quality measured by DV_{np1}, in contrast to DV_{np3}, in which a poor ensemble quality results when the diversity is low. This shows that the poor quality of the MCLA ensemble is affected by a lower diversity even when the average quality slightly decreases compared to other ensemble cases.

With Soybeans dataset, we noticed that in Figures 6.4(d) and 6.4(e) there is a perfect solution for the problem discovered by using ACE. Looking closely to this



(a) *Entropy*

(b) DV_{np1}

(c) DV_{np3}

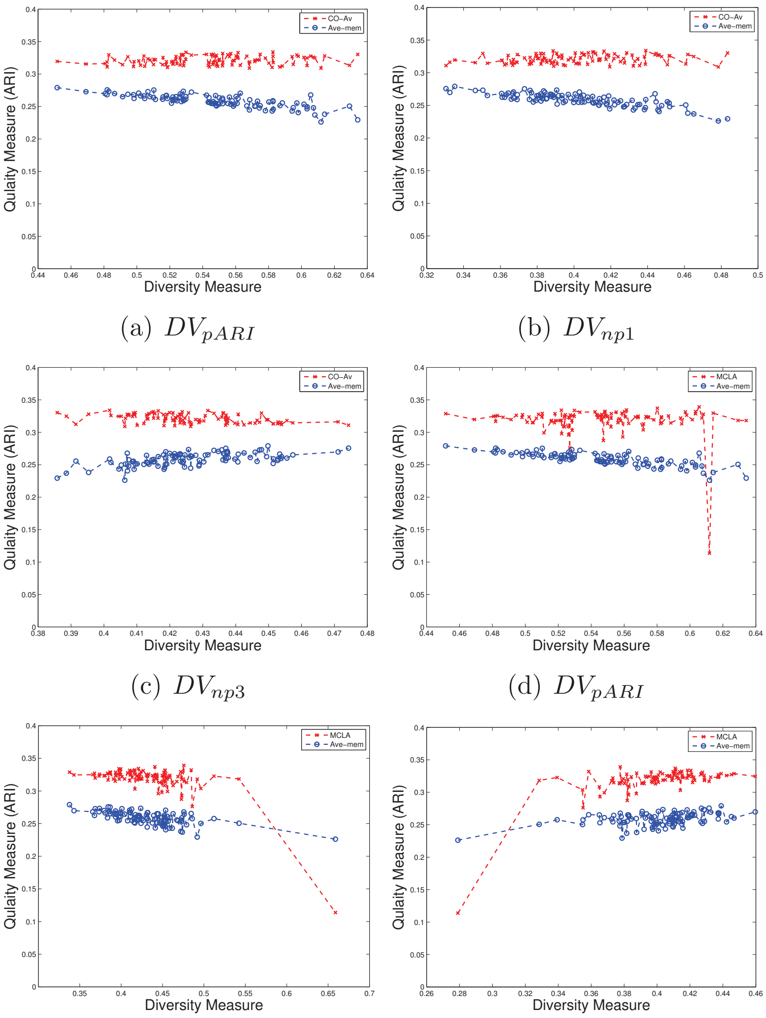
(d) *Entropy*

(e) DV_{np1}

(f) DV_{np3}

Figure 6.8: The *Entropy*, DV_{np1} and DV_{np3} measures from Glass dataset,(a), (b) and (c) using CO-Ave, while (d), (e) and (f) using MCLA.

particular case, we found that there were three members that had also the same solution (perfect) and none of the other ensemble methods had that result, where the performance of CO, ONCE and MCLA were equal to 0.661, 0.661 and 0.545 respectively. That is because ACE is based on computing the similarity between clusters and membership similarity between objects and clusters and it does not implement any ordinary clustering algorithm such as a graph based clustering algorithm that applies in MCLA or the Hierarchical algorithm that applies in the CO and ONCE.



(a) DV_{pARI}

(b) DV_{np1}

(c) DV_{np3}

(d) DV_{pARI}

(e) DV_{np1}

(f) DV_{np3}

Figure 6.9: The DV_{pARI}, DV_{np1} and DV_{np3} measures from Mfeatures dataset (a), (b) and (c) using CO-Ave, while (d), (e) and (f) using MCLA.

In the relation between the diversity and the average quality of the members, we discovered in the Iris dataset a negative linear correlation between them measured by DV_{pARI} (Figure 6.3) as well as DV_{pNMI} (Figure A.4(b)) and *Entropy* (Figures A.4(c)). This shows that as the diversity increases, the average quality of the members decreases accordingly, while all four ensemble qualities remain stable in most cases, even when the average member quality decreases. The reason behind this phenomenon is that generating more diverse members would result in many incorrect clustering structures with less accurate members. This is especially the case with a dataset that has $k \ll n$ (n the number of objects and k the number of clusters), such as Bcw or Ionosphere.

However, with Iris dataset for the DV_{np3} (Figure 6.3(e)) and DV_{np4} (Figure 6.3(f)) measurements, the pattern changes to a positive correlation between the ensemble quality and the diversity. These linear correlations are also found with Bcw (Figure 6.1) and Ionosphere (Figure 6.2), measured by DV_{np3} and DV_{np4}.

With the Glass (Figure 6.8), Mfeatures (Figure 6.9) and Wine datasets (Figure 6.7), we discovered that the average quality of the members has a slightly negative relationship with the diversity measured by DV_{pARI}, DV_{pNMI} and *Entropy*. While the average quality of the members fluctuates slightly over the diversity scores measured by DV_{np1}, DV_{np2}, DV_{np3} and DV_{np4}, where in most cases DV_{np1} and DV_{np3} have moderate diversity.

With the Thyroid dataset, the average quality of the members fluctuates over the diversity scores measured by all examined diversity measures, and most of the cases have a high level of diversity measured by DV_{pARI}, DV_{pNMI}, *Entropy* and DV_{np1}, whereas most cases have a moderate diversity level measured by DV_{np2} and DV_{np3}. Finally, with the Soybean dataset (Figure 6.4), the average quality of the members shows a slight linear correlation with the diversity in all the tested measurements.

6.1.3 Studying the Correlation between the Ensemble Performance and the Diversity.

We carried out the correlation coefficient (cc) test to measure the strength and direction of a linear relationship between ensemble quality and the values of a diversity measure. We also carried out the significance test of whether, based upon these samples, there is any evidence to suggest that the linear correlation is present in the population. The value of $cc \in [-1, 1]$ and of the p-value $\in [0, 1]$. When $cc = 1$, it indicates a perfect positive linear relationship between the diversity and the ensemble quality, and when $cc = -1$, it indicates that there is a perfect negative linear relationship between them. When $cc = 0$, this indicates that there is no linear relationship between them. The p-value indicates that the degree of that relationship is statistically significant at a 95% confidence interval. In other words, we tested the null hypothesis that there is no correlation in the population against the alternative hypothesis that there is linear correlation present.

Table 6.1 shows the correlation coefficient between each diversity measurement and the performance of each ensemble method, and bold value of correlation coefficient indicates that we reject the null hypotheses.

Generally, the correlation coefficient results indicate that the relationship between the ensemble quality and the diversity varies from one dataset to another and from one ensemble method to another. The results suggest that the pairwise diversity measures have similar cc values, which indicates that for some cases, the ensemble quality has a weak positive relationship (14 cases in DV_{pARI} and 17 cases in DV_{pNMI}), while for other cases, there is a weak negative linear relationship with the diversity (17 cases in DV_{pARI} and 15 cases in DV_{pNMI}). In addition, there is only one case of no linear relationship between them discovered by DV_{pARI} in Bcw dataset.

However, for the non-pairwise measures, we noticed that DV_{np1} always has the opposite linear relationship to that discovered by DV_{np3}. For example, in the Glass dataset, using the MCLA ensemble method, the *cc* value of -0.813 for DV_{np1} indicates a strong negative relationship, whereas the *cc* of 0.806 for DV_{np3} indicates a strong positive linear relationship. The range of *cc* values in each of the non-pairwise diversity measures varies from a strong positive linear relationship to a strong negative relationship. Thus, obviously there is no agreement between them across the 8 datasets and in the use of the different ensemble methods. But in most cases, what are discovered for DV_{np1} and DV_{np2} are negative weak linear relationships (14 and 15 cases respectively), while DV_{np3} and DV_{np4}, in most cases, fall between a positive weak relationship and a positive moderate relationship (in DV_{np3}, 14 weak cases and 6 moderate cases out of 32, and in DV_{np4}, 12 weak cases and 7 moderate cases out of 32).

For the statistical significance test, we found that for most of the correlation cases using the pairwise diversity measure and *Entropy*, we cannot reject the null hypotheses for most of the ensemble methods used. But for Iris using ONCE, Wine using CO, Glass using ACE and Ionosphere using MCLA, we reject the null hypotheses for these diversity measurements. In contrast, for the non-pairwise diversity measure, in most cases for the ensemble methods used, we reject the null hypotheses. The only exception to this rule is the Bcw dataset, where we cannot reject the null hypotheses for all the diversity measures tested and all ensemble methods used.

6.2 Investigation of Issues Raised

The results in Section 6.1 showed that diversity can have a positive or negative effect on the ensemble performance. Thus, in Section 6.2.1 we are motivated to find out under which conditions diversity can have a positive or a negative effect on the ensemble performance. Furthermore, the results revealed there may be an interaction existing between diversity and the members' quality, and in Section 6.2.2 we are motivated to find out if this interaction exists and if so what is the effect of the interaction on the ensemble performance.

Table 6.1: Correlation coefficient between each diversity measure and ensemble result for each tested dataset. A bold values represent a rejection of the null hypotheses which is there is no correlation between the ensemble quality and the diversity measure.

Datasets	**CF**	DV_{pARI}	DV_{pNMI}	$Entropy$	DV_{np1}	DV_{np2}	DV_{np3}	DV_{np4}
Iris	**CO**	-0.124	-0.154	-0.144	-0.139	-0.080	0.130	0.123
	ONCE	**-0.335**	**-0.389**	**-0.354**	**-0.546**	**0.488**	**0.688**	**0.541**
	ACE	0.151	0.153	0.142	**0.251**	**-0.318**	**-0.305**	**-0.339**
	MCLA	-0.064	-0.062	-0.058	-0.053	-0.038	0.046	0.048
Wine	**CO**	**0.288**	**0.232**	**0.278**	**0.456**	**-0.923**	**-0.848**	**-0.860**
	ONCE	0.130	0.061	0.110	**0.345**	**-0.920**	**-0.793**	**-0.807**
	ACE	**0.211**	0.192	0.197	**0.594**	**-0.729**	**-0.705**	**-0.710**
	MCLA	0.009	0.019	0.043	0.181	**-0.606**	**-0.426**	**-0.491**
Thyroid	**CO**	-0.082	0.056	-0.083	-0.049	-0.154	-0.048	-0.077
	ONCE	-0.012	0.077	-0.011	-0.085	-0.048	0.030	-0.037
	ACE	-0.092	-0.033	-0.065	**-0.374**	**0.320**	**0.360**	**0.327**
	MCLA	0.190	**0.272**	0.182	0.139	0.247	0.028	0.119
Mfeatures	**CO**	0.153	0.185	0.138	0.222	**-0.461**	**-0.425**	**-0.483**
	ONCE	0.035	0.050	0.053	0.110	**-0.395**	**-0.291**	**-0.366**
	ACE	-0.070	-0.079	-0.146	**-0.583**	**0.636**	**0.650**	**0.600**
	MCLA	-0.117	-0.188	**-0.278**	**-0.605**	**0.282**	**0.594**	**0.456**
Glass	**CO**	-0.000	-0.046	0.001	-0.189	0.058	0.173	0.151
	ONCE	0.136	0.053	0.137	-0.129	0.199	0.161	0.138
	ACE	**-0.359**	**-0.299**	**-0.371**	**-0.548**	0.236	**0.487**	**0.423**
	MCLA	0.073	0.016	0.077	**-0.813**	**0.758**	**0.806**	**0.745**
Bcw	**CO**	0.000	-0.000	-0.000	-0.000	-0.000	0.000	0.000
	ONCE	-0.120	-0.033	-0.097	-0.169	-0.031	0.171	0.169
	ACE	-0.140	-0.081	-0.122	-0.167	-0.013	0.179	0.166
	MCLA	-0.167	-0.103	-0.160	-0.189	-0.113	0.141	0.160
Soybean	**CO**	0.012	0.014	-0.072	-0.234	-0.092	0.120	0.090
	ONCE	-0.106	0.007	-0.099	**-0.550**	0.320	**0.530**	**0.521**
	ACE	0.048	0.113	0.144	**-0.703**	**0.544**	**0.701**	**0.614**
	MCLA	0.032	0.079	-0.021	0.220	-0.221	**-0.284**	**-0.265**
Ionosphere	**CO**	-0.080	-0.070	-0.079	-0.103	0.025	0.148	0.127
	ONCE	-0.054	-0.047	-0.033	-0.077	0.038	0.123	0.098
	ACE	0.062	0.056	0.089	0.107	-0.068	-0.179	-0.161
	MCLA	**-0.494**	**-0.486**	**-0.484**	**-0.530**	-0.201	**0.560**	**0.510**

6.2.1 Analysis of the Positive and Negative Effects of Diversity on the Ensemble Performance

The formal definitions of these two different effects of diversity on clustering ensemble are as follows:

Definition 8. Positive effect: *For a given consensus function, if the diversity has a positive effect on the ensemble performance, then the ensemble quality $Q(\Phi)$ is higher than the average member quality $Q(\Gamma)$, and we say that the ensemble has a good combination of the individual members.*

$$DV^{+} \Rightarrow Q(\Phi) > Q(\Gamma)$$

In other words, the diversity brings an improvement to the clustering ensemble performance, and in this case the diversity is constructive to the clustering ensemble.

Definition 9. Negative effect: *For a given consensus function, if the diversity has a negative effect on the ensemble performance, then the ensemble quality $Q(\Phi)$ is less than the average member quality $Q(\Gamma)$, and we say that the ensemble has a bad combination of the individual members.*

$$DV^{-} \Rightarrow Q(\Phi) < Q(\Gamma)$$

In this case, the diversity harms the performance of the clustering ensemble, and it is destructive to the clustering ensemble.

In the next section, we will look to the relationship between diversity and the ensemble performance from a different angle. We will make a comparison between a pair of ensembles consisting of one positive case and one negative case, where both pairs are built under same or similar conditions in terms of the average member quality and the level of diversity. The difference between them therefore is that the first ensemble represents a combination pattern when the ensemble succeeds (best ensemble performance due to best combination pattern), whereas the second one represents a combination pattern when it fails (worst ensemble performance due to worst combination pattern).

6.2.1.1 Experimental Design

In Section 6.1, we run the experiment on 8 datasets using 4 consensus functions, and for each consensus function we run the experiment 100 times. According to the above definition of the negative effect, we found that the negative effect occurred on the Thyroid, Soybean, Wine and Glass datasets. Among these the Thyroid dataset clearly had the highest number of negative cases on all the consensus functions used (CO, ONCE, ACE and MCLA) compared to other datasets (as shown in Figures 6.5, 6.6 and in Appendix A). Thus, here we focus our study on the results obtained from the Thyroid dataset using all 4 consensus functions. In this dataset, for each consensus function the number of negative cases in the 100 runs was equal to 40, 21, 42 and 4 in CO, ONCE, ACE and MCLA results respectively.

In this analysis, the idea is to find a pair of ensemble cases (Φ_a, Φ_b), where Φ_a has a negative effect $(Q(\Phi_a) < Q(\Gamma_a))$, and Φ_b has a positive effect $(Q(\Phi_b) > Q(\Gamma_b))$, and they share the following conditions:

- $DV(\Phi_a) \approx DV(\Phi_b)$ for most of the DV measures (DV_{pARI}, DV_{pNMI}, $Entropy$, DV_{np1}, DV_{np2}, DV_{np3}, and DV_{np4}).
- $Q(\Gamma_a) \approx Q(\Gamma_b)$
- $Std(Q(P_q \in \Gamma_a)) \approx Std(Q(P_q \in \Gamma_b))$

We are interested in investigating the negative effect associated with most of the used consensus functions. As we used 4 consensus functions, we found that there is not a case that is negative in all the 4 consensus functions. So, we identified a case in two situations: first it is negative in only two consensus functions and second it is negative in at least three of them. In total, we identified 22 pairs of cases as shown in Table 6.2, which lists the ensemble qualities of these pairs and the average member quality (all measured by ARI) for the first and second situations.

In each pair of ensembles, the first ensemble is an odd number, representing the negative ensemble case, and the second is the next even number, representing the positive ensemble case. For example, pair number one consists of case 1 (positive) and case 2 (negative), and pair number two consists of case 3 and case 4 and so on.

For each of these ensemble pairs, we analysed the quality of their individual members. We used a simple counting approach of the poor-, good- and medium-quality members to compare between a pair of ensemble cases, one representing the negative case, the other representing the positive case.

Table 6.2: The quality of ensembles using CO, ONCE, ACE, MCLA consensus functions and the average member quality (Ave-mem) in 22 cases. Cases with bold font indicate that these are negative cases, which are case 1, 3, 5, 7, 9 ,11, 13, 15, 17, 18 ,19 , and 21. These cases are all taken from the results in section 6.1.2 for Thyroid dataset.

Pair #	Case #	Q(CO)	Q(ONCE)	Q(ACE)	Q(MCLA)	Ave-mem
1	**Case 1**	0.579	0.579	**0.211**	**0.231**	0.275
	Case 2	0.513	0.414	0.303	0.508	0.285
2	**Case 3**	**0.231**	0.297	0.324	**0.192**	0.274
	Case 4	0.513	0.336	0.192	0.548	0.310
3	**Case 5**	**0.164**	0.296	**0.221**	0.417	0.264
	Case 6	0.440	0.485	0.394	0.497	0.287
4	**Case 7**	**0.155**	**0.155**	**0.164**	0.579	0.245
	Case 8	0.579	0.579	0.530	0.579	0.289
5	**Case 9**	**0.211**	**0.211**	**0.165**	0.347	0.220
	Case 10	0.579	0.414	0.511	0.579	0.307
6	**Case 11**	**0.164**	**0.164**	**0.119**	0.441	0.253
	Case 12	0.502	0.414	0.265	0.535	0.304
7	**Case 13**	**0.173**	**0.164**	**0.164**	0.579	0.276
	Case 14	0.513	0.414	0.526	0.579	0.297
8	**Case 15**	**0.273**	**0.221**	**0.222**	0.579	0.285
	Case 16	0.579	0.579	0.446	0.579	0.280
9	**Case 17**	**0.221**	0.383	**0.273**	**0.240**	0.279
	Case 18	0.546	0.316	0.374	0.570	0.301
10	**Case 19**	**0.164**	**0.164**	**0.262**	0.582	0.286
	Case 20	0.515	0.414	0.486	0.535	0.308
11	**Case 21**	**0.155**	**0.155**	**0.252**	0.560	0.260
	Case 22	0.579	0.579	0.446	0.579	0.280

6.2.1.2 Summary of Results

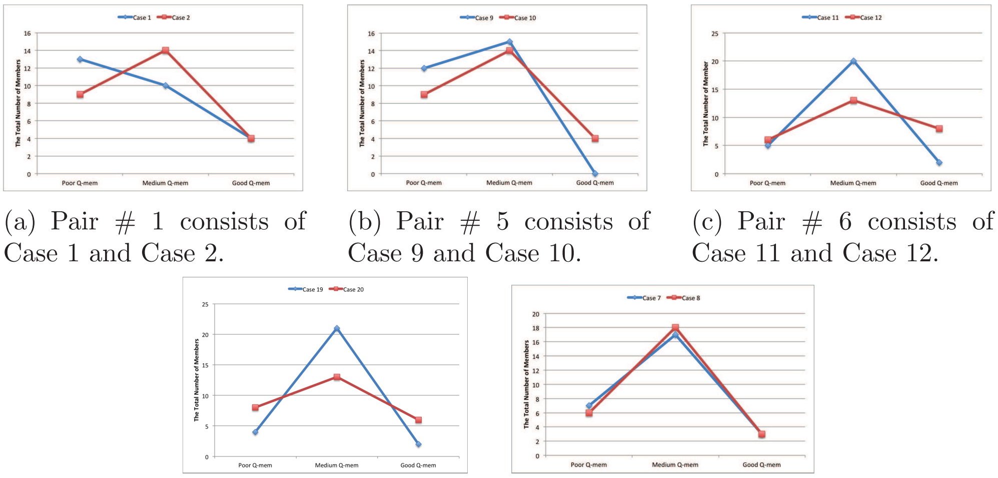
The full results of this experiment are given in Appendix B. In summary, we observed the following:

1. In most negative cases for some consensus functions such as CO and ONCE, the high number of poor-quality members had indeed affected the ensemble performance, while others such as MCLA had not been affected by the same members. An example is Pair # 1 and 5 as shown in Figures 6.10(a) and 6.10(b) respectively.
2. For other negative cases the pattern of the number of poor-, medium-, and good-quality members show an inverted V shaped pattern, where the number of members with a medium-quality was higher than the other two categories and with a high level of diversity measured by independent measures. This

contributes to limit the ensemble quality to being lower than the average quality of its members. An example is Pair # 6 and 10 as shown in Figure 6.10(c) and 6.10(d) respectively.

3. In most positive cases, the distribution of the individual members qualities was higher than the comparative negative cases.

4. It was also observed that two cases had an equal number of poor-quality and good-quality members, but one case represented a pattern of success for the ensemble, while the other represented a pattern of failure (see Pair # 4 in Figure 6.10(e)). But generally, the distributions of the individual members quality in these cases indicate that in the pattern of success the members had a higher quality than in the pattern of failure, and these high-quality members with a high diversity level (measured by independent measures) contribute to the production of a high-quality ensemble.



(a) Pair # 1 consists of Case 1 and Case 2.

(b) Pair # 5 consists of Case 9 and Case 10.

(c) Pair # 6 consists of Case 11 and Case 12.

(d) Pair # 10 consists of Case 19 and Case 20.

(e) Pair # 4 consists of Case 7 and Case 8.

Figure 6.10: The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in each case.

Therefore, we concluded that this approach did not give us a clear indication of the reason behind the negative performance of the ensemble. But an extended experiment was designed in the next section to investigate the effect of removing the poor-quality member on the performance of different consensus functions in the negative and the positive cases.

6.2.1.3 The Experiment of Eliminating Poor Members

In this section, we analyse and study how the ensembles perform using the different consensus functions when we gradually remove one member at a time, based on its quality in the negative combined pattern, as well as in the positive combined pattern.

Therefore, for each of the identified cases in table 6.2, we saved its members in pool O, of which there are 27 members, and the following steps were implemented:

1. The quality of the individual members in pool O is measured using ARI.
2. Then the members in O are sorted in an ascending order based on their quality (from the lowest to the highest quality member).
3. The first member in the current O is removed, which represents the poorest quality member in the current O.
4. The remaining members in O are combined using the CO, ONCE, ACE and MCLA consensus functions.
5. The following values are calculated: ensemble quality for each of the consensus functions $Q(CO)$, $Q(ONCE)$, $Q(ACE)$ and $Q(MCLA)$, the average quality of members (avg-mem), and the diversity of the ensemble using DV_{pARI} and DV_{np3}, where the latter is measured for each of the consensus functions.
6. Steps 3 to 5 are repeated until only 3 members are combined, which represents the last run.

Thus, in each run the size of the ensemble decreased by 1 until run 25, where the size was equal to 3.

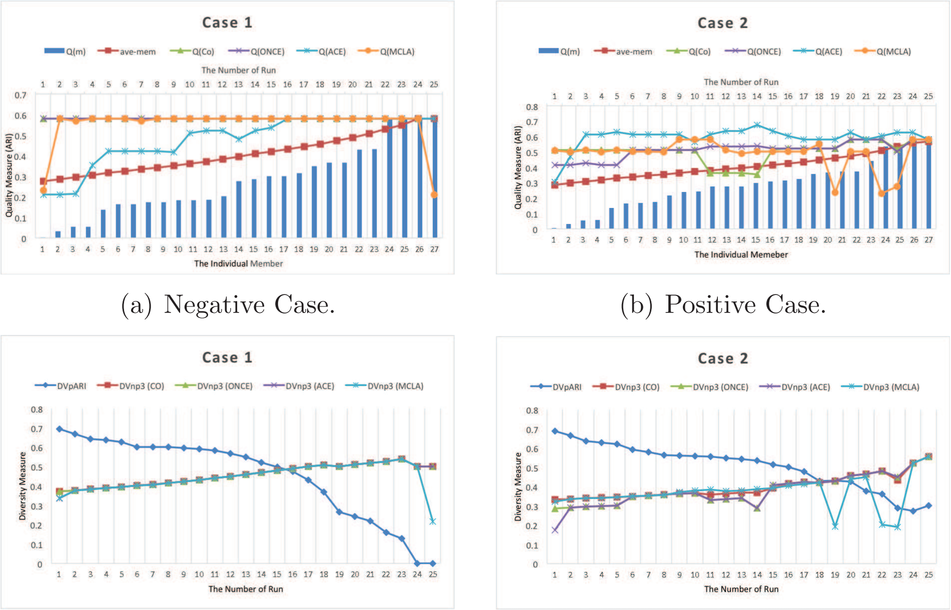
Experimental Results

The results of each identified case in our experiment are analysed in more detail in Appendix B, and here we analysis the typical ones.

Clearly, the results show that as a consequence of removing the poorest quality member in each run, the average members quality increased and accordingly the diversity decreased (measured by DV_{pARI}). This indicates that in this experiment there is an inverse relationship between the average member quality and the DV_{pARI} measure.

It is also noticed that in 6 out of 22 cases (Figures B.27, B.30, B.33, B.34 and B.33), the three highest quality members had the same quality clustering results, and their ensemble diversity value of DV_{pARI} was equal to 0. This also confirms the idea that diversity as a factor is highly associated with member quality, because when we fix the members' qualities to a constant value their diversity is most likely to be zero. In these cases (cases 1, 7, 8, 13, 15 and 17) , combining these members using CO, ONCE and ACE achieved ensemble results of the same quality as the members. But, using MCLA we achieved a very poor performance, which indicates that MCLA is not a good choice when diversity does not exist among the members. Looking at the diversity measured by DV_{np3}, it is noticed that it is sensitive to the ensemble performance because when the performance is low in value, the DV_{np3} is also low in value. These results confirm those of Section 6.1, that the dependent diversity measures are sensitive to the ensemble performance.

The lowest average member quality (equal to 0.220) was in case 9 (Figure B.31) with a high DV_{pARI} (equal to 0.667) and all the consensus functions produced low quality performance, with three of them being below the average member quality (CO= 0.211, ONCE= 0.211, ACE= 0.164 and MCLA = 0.347). It is therefore clear that sometimes the individual members are not good enough to be combined in terms of their quality and the diversity among them. In summary, the results show that removing the poor-quality members did to some extent improve the performance of all the identified consensus functions in all the negative cases, in addition



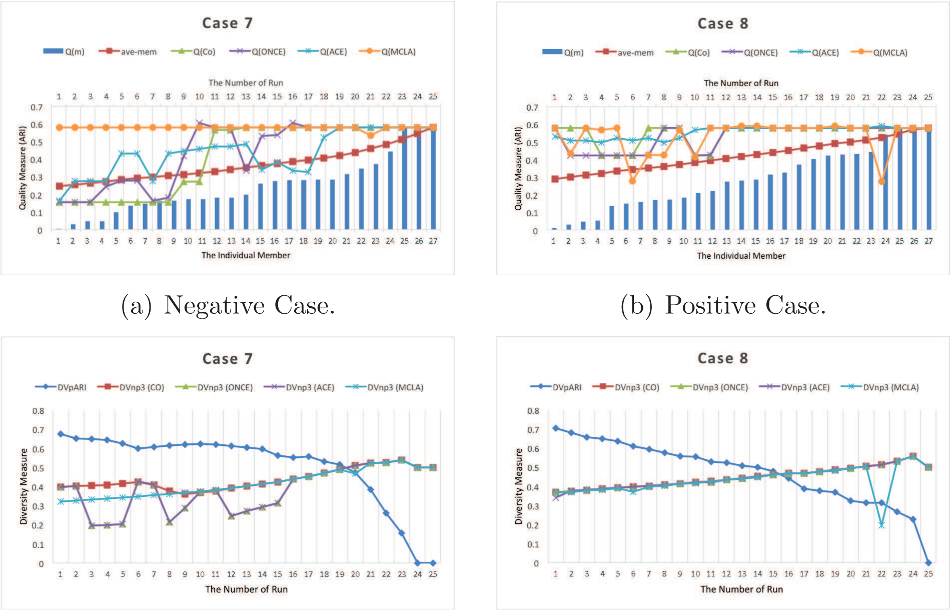
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure 6.11: 25 ensemble runs for case 1 & 2, in each run one member is removed.



(a) Negative Case.

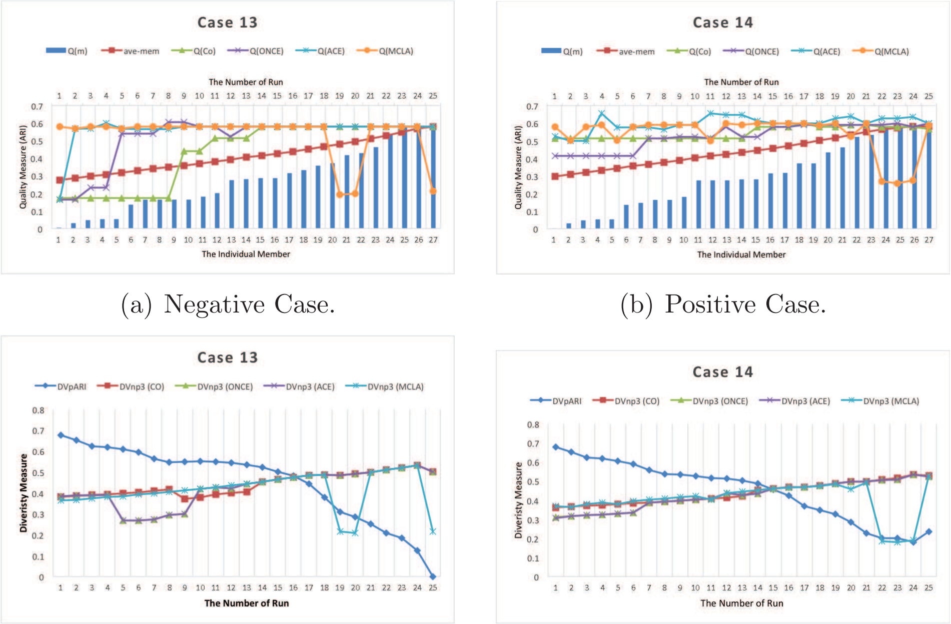
(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure 6.12: 25 ensemble runs for case 7 & 8, in each run one member is removed.

to improving the performance of some consensus functions in the positive cases, which are ONCE and ACE. However, each consensus function had different reactions to removing poor-quality members, due to the difference in their implemented



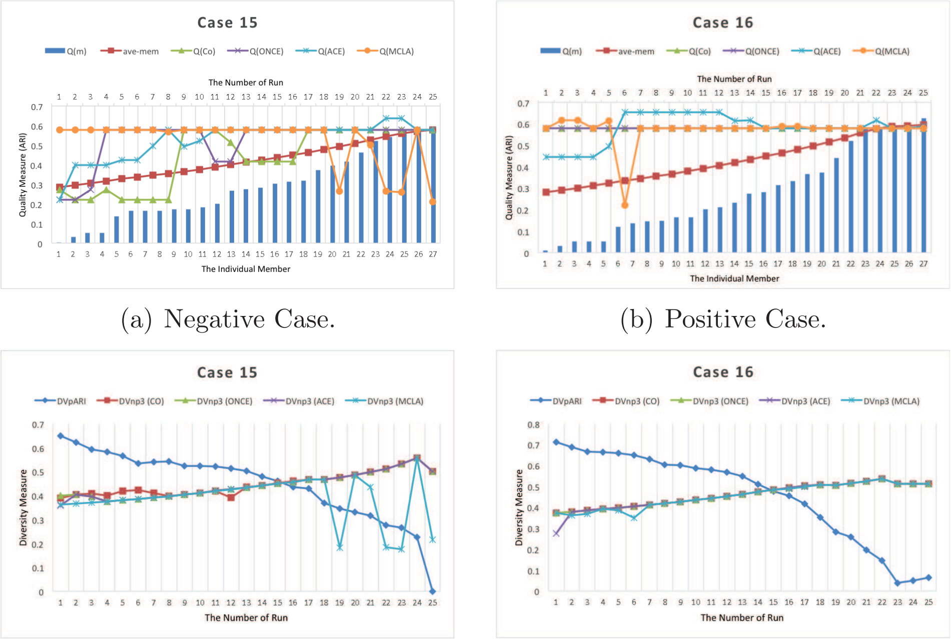
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure 6.13: 25 ensemble runs for case 13 & 14, in each run one member is removed.



(a) Negative Case.

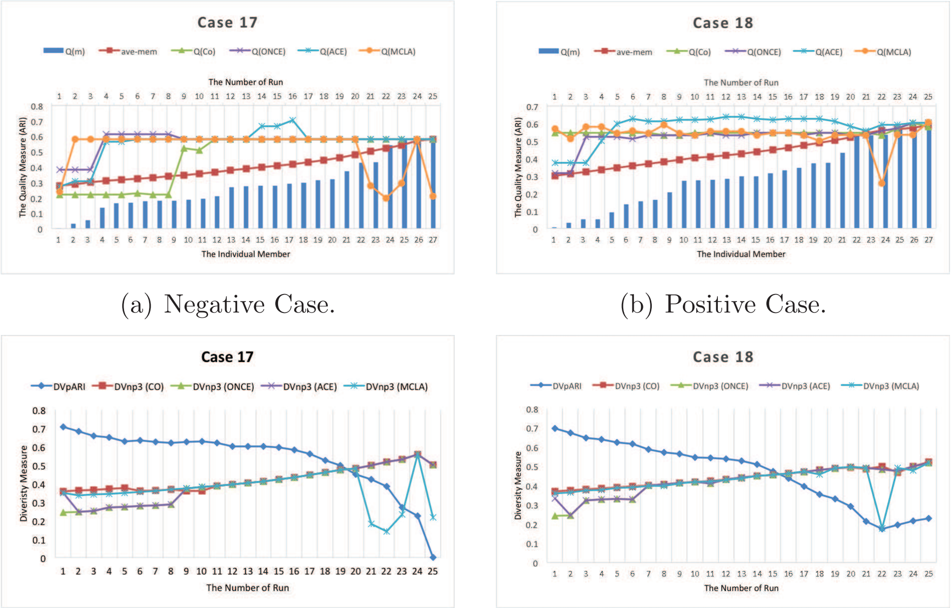
(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure 6.14: 25 ensemble runs for case 15 &16, in each run one member is removed.

techniques. Furthermore, the results in this investigation showed that diversity is highly associated with the members qualities, therefore there is a great need to study



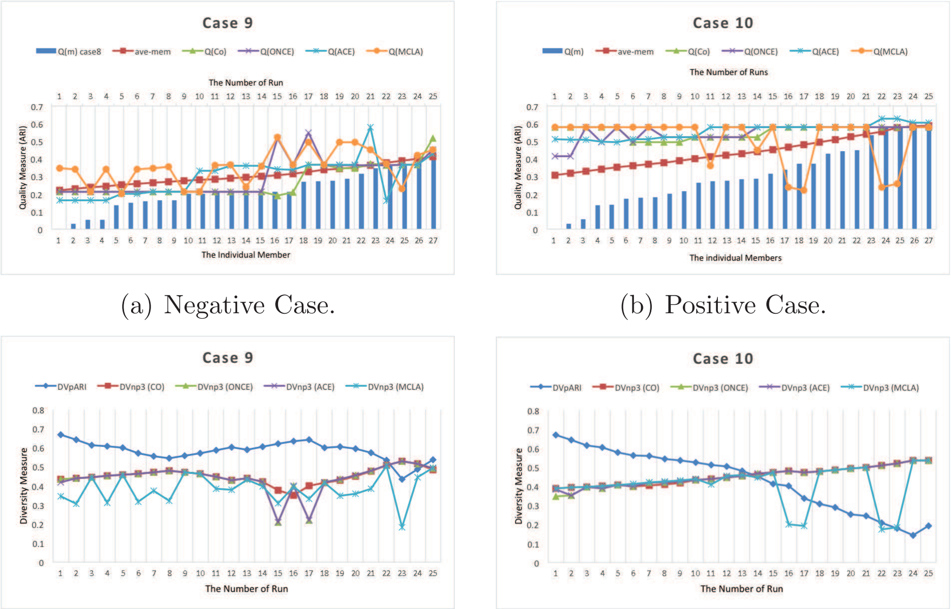
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure 6.15: 25 ensemble runs for case 17 & 18, in each run one member is removed.



(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure 6.16: 25 ensemble runs for case 9 & 10, in each run one member is removed.

the effect of the interaction between them on the ensemble performance.

However, the following characteristics have been noticed in our experiment for the pattern of success and failure; although they should not be generalised to all consensus functions as each has its own characteristics.

Pattern of Success

In the pattern of success, based on our studies the following conclusion are drawn:

1. It is preferable to have a medium level of average member quality accompanied by a medium level of DV_{pARI}. The results support the argument that the quality of the individual members alone is not enough to improve the ensemble performance compared to the individual members' quality, but it is also necessary for them to be diverse.
2. CO and ONCE preferred the members to have between medium and high quality with a medium diversity among them. If the members quality is poor, they must have the "right"diversity among them, meaning that the certainty of the correctly classified pairs of objects is maintained higher than that of the wrongly classified ones. If among the members there is one with a very high-quality (unexpected clustering results), then the other members must support this one. The results showed that the performances of CO and ONCE are equal to the quality of identical members, when there is no diversity among them.
3. ACE also preferred a medium-level average member quality accompanied with a medium DV_{pARI}.
4. Using MCLA as a consensus function, the members must have some sort of diversity, otherwise MCLA will perform poorly compared to the ensemble members' quality.

Pattern of Failure

In this pattern, the lower average member quality, accompanied with a high level of DV_{pARI} and low level of DV_{np3}, is responsible for the failure of the ensemble. However, this pattern can be changed to a successful one by increasing the average member quality, decreasing DV_{pARI}, and increasing DV_{np3}. The reaction to this gradual changing in the pattern differs from one consensus functions to another, but CO is the last one to be improved — it improved by removing most poor-quality members.

6.2.2 The Experimental Study of the Interaction between Members' Qualities and Diversity

In this section, we investigate whether or not there is an interaction effect between the diversity and the members' qualities on the ensemble performance. A factorial design experiment is implemented to investigate this interaction, where the independent variables (factors that we are interested in studying) are diversity (DV) and members' qualities (Mem-Quality) and the dependent variable is the ensemble performance. Our hypothesis is that there is an interaction between the ensemble diversity and the members' qualities, and that this may affect the performance of the ensemble.

For the diversity we used the DV_{pARI} measurements, while for the quality of the members we used the ARI index. We can subdivide the range of diversity values and member quality into three different levels: High, Medium and Low, as shown in Figure 6.17. However, we considered the high level in member quality as an extreme case, which is rarely achieved by a single clustering algorithm for the real datasets used in Section 6.1.

Therefore, we implemented a 3×2 factorial experimental design with 6 conditions (combinations of the two factors) to investigate the simultaneous interaction between the diversity and member qualities on ensemble quality. Because, our interest is in the interaction we had to also include their individual main effects. We considered the diversity and the members' qualities as between-subject factors, as we generated a new set of members for each condition.

For each of the datasets used in the previous experiment in Section 6.1 we col-

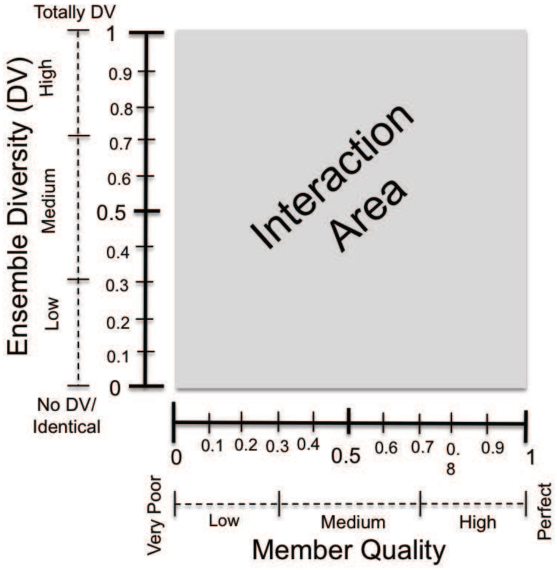


Figure 6.17: The range values of the member quality and the ensemble diversity, including the interaction area between them.

lected all the members generated by 100 runs in pool O (2700 members in total) and we classified them based on the 6 conditions. In 8 datasets examined, we had only two datasets where a number of runs could be built on all of the 6 identified conditions; these were the Thyroid and Wine datasets. With the other datasets, there were one or more conditions that could not be built and therefore we could not use them. In addition, we could not design the experiment as incomplete, as the main aim of this experiment was to study the interaction effect, and this cannot be done with an incomplete factorial design. Thus, we carried out our experiment on only these two datasets as a pilot study. For each condition, we built the ensemble 30 times, but it was noticed that under two conditions we could not generate 30 sets of members, and that was when the conditions were at low levels in both factors (3 sets in the Thyroid datasets and 2 sets in the Wine dataset), and with a medium level in the member quality and a high level in diversity (2 sets in the Thyroid dataset and 7 sets in the Wine dataset). This is due to the fact that controlling/limiting the quality of the members to a certain range makes restricted room for diversity among them. So, the experiment is designed as unbalanced, where the number of runs for each condition are not equal.

However, we run the experiment by building the ensemble under the identified conditions using four consensus functions (CO, ONCE, ACE and MCLA), and the quality of the ensembles clustering results were measured (using an ARI index) and recorded for each condition. For each dataset, we had four factorial experiments (one for each consensus function).

We first investigated the main effects of each factor independently, and then we investigated the interaction effect between them. Generally, the main effect of one factor represents the overall means of the ensemble performance on the different levels of that factor over the levels of the other factor. It is, in fact, the overall effect of one factor while ignoring the effect of the other factor [90]. On the other hand, the interaction effect is represented by the mean of ensemble performance in each combination between two factors.

Both of these two effects can be visualised in a line chart plot. In the main effect plot, the non horizontal line between the levels of one factor indicates that there is a significant effect of that factor on the response, and the steeper the slope of the line the greater the size of the main factor effect. Whereas, the non parallel lines in the interaction plot indicate a sign of interaction between two factors. The greater the difference in slope between lines, the high the degree of interaction between two factors [16]. We should mention that as our design experiment was unbalanced so the ensemble performance means for a factor in these plots were calculated as the unweighted mean, which controls for the effect of other factors, so the confounding caused by unequal sample size is eliminated.

6.2.2.1 Experimental Results

The Main Effects Results:

Figures 6.18 and 6.19 show the plots of the main effect of the diversity and members' qualities for the Thyroid and Wine datasets, respectively. The dashed horizontal lines in these plots show the overall mean of ensemble performance (the performance mean in the whole sample data). In both of the datasets in all the used consensus functions, the main effect of diversity and the members' qualities was shown by non-horizontal lines between the factors' levels (as shown in figures 6.18 and 6.19), which indicate that the different levels of these factors affect the ensemble performance differently.

In the Thyroid dataset, the ensemble performance mean with high and medium diversity levels was lower than the overall mean of the ensemble, whereas with a low diversity level it was higher than the overall means for all consensus functions. In the Wine dataset, the main effect of diversity is different from that in the Thyroid dataset. The ensemble performance mean in the low diversity level was lower than the overall mean of the ensemble for all consensus functions. However, it was higher than the overall ensemble mean in the high and medium diversity levels for all consensus functions except ACE, where its performance mean was lower in the high level and higher in the medium level compared with the overall mean.

However, in both datasets it is clear that the slope of the line between the two levels of member quality is steeper than that of the lines between the levels of diversity. This indicates that the effect size of the members' qualities on the ensemble performance is greater than the one related to diversity. Intuitively, as the quality of the individual member increases, the ensemble quality naturally increases too. Moreover, it was observed that when the members' qualities were at a medium level, then the ensemble performance always had a higher mean. In addition, ensembles generated using a low value for member quality always had a lower performance mean, and that is true for all the consensus functions used on the two datasets under this experimental set-up.

The Interaction Effect Results:

Figures 6.20 and 6.21 illustrate the ensemble performance mean of combinations of levels from the two factors in order to show the trend of the interaction between them for the Thyroid and Wine dataset, respectively. In the right upper corner of these figures, we can see clearly that the lines are not parallel and they are crossing

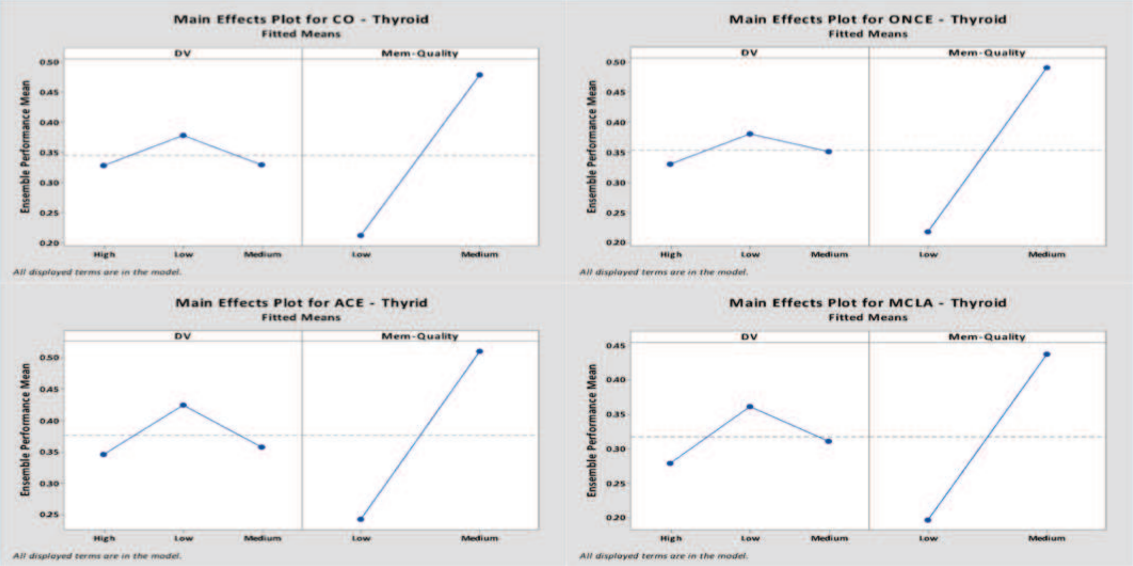


Figure 6.18: The main effects of the diversity and member quality on the response variables, which are CO, ONCE, ACE and MCLA, for Thyroid dataset.

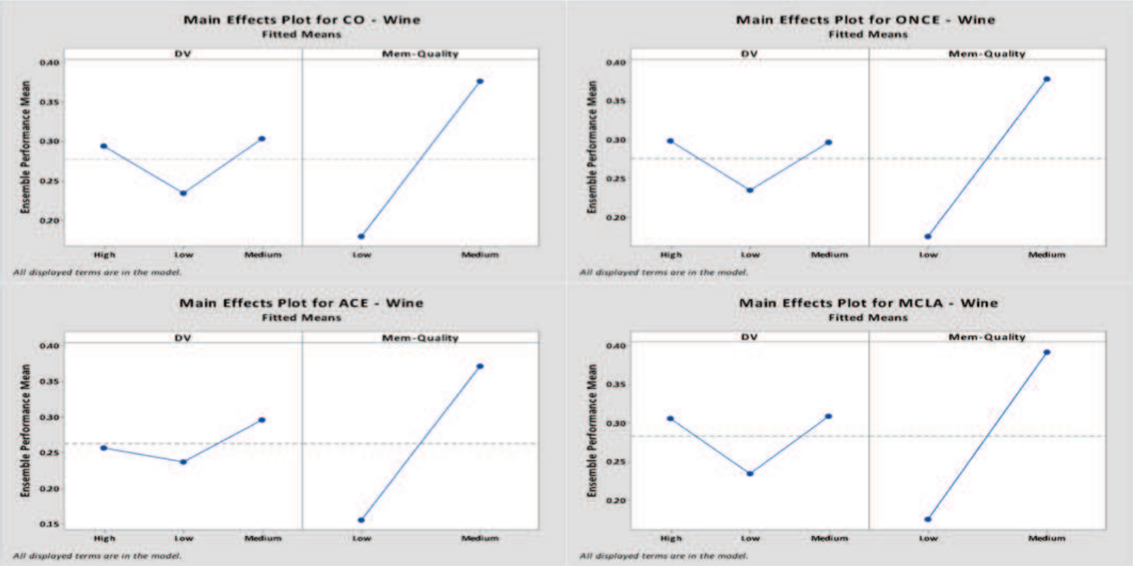


Figure 6.19: The main effects of the diversity and member quality on the response variables, which are CO, ONCE, ACE and MCLA, for Wine dataset.

at some point. This indicates that there is an interaction between the diversity and the members' qualities, and the degree of interaction differs between the different consensus functions.

With the Thyroid dataset, it was observed that when the members had a low quality, whatever the diversity among them, the ensemble performance mean was low, using CO and ONCE. It was observed that the ensemble achieved a higher performance mean when the combining members had a medium quality and a low diversity among them, with all used consensus functions. With ACE and MCLA, it was noticed that the lines of medium and low diversity levels were close to parallel with a very slight tendency towards the low-quality members, indicating that there was very little interaction between these diversity levels and members' qualities using these consensus functions.

With the Wine dataset, it was noticed that the high and medium levels of diversity crossed in the middle, meaning that diversity (at high and medium levels) had the opposite effect on the ensemble performance mean for low-quality members to that of medium-quality members.The lines of the high and low diversity levels were close to parallel with a very slight tendency towards the low-quality members' level. This was observed with CO, ONCE and MCLA. However, in ACE, there was more or less the same performance mean in these two levels of diversity for the low-quality members.

However, the results at this stage suggested that there are main effects of diversity and members' qualities on the ensemble performance, as well as an interaction between them; the degree of interaction is different between the consensus functions. Thus, a statistical test is needed to determine whether it is justifiable to conclude that these effects exist in the population. The following section presents the results of ANOVA tests of these factorial experiments.

6.2.2.2 Result of ANOVA

Before we ran the ANOVA test we checked its assumptions, which is the normality and the homogeneity of variances. Appendix C shows the full details of checking these assumptions, which show that the data meets the ANOVA assumptions. Therefore,

we applied the two-way ANOVA using type III sums of squares for F-statistics on the transformed data sample. On the other hand, we also applied the same

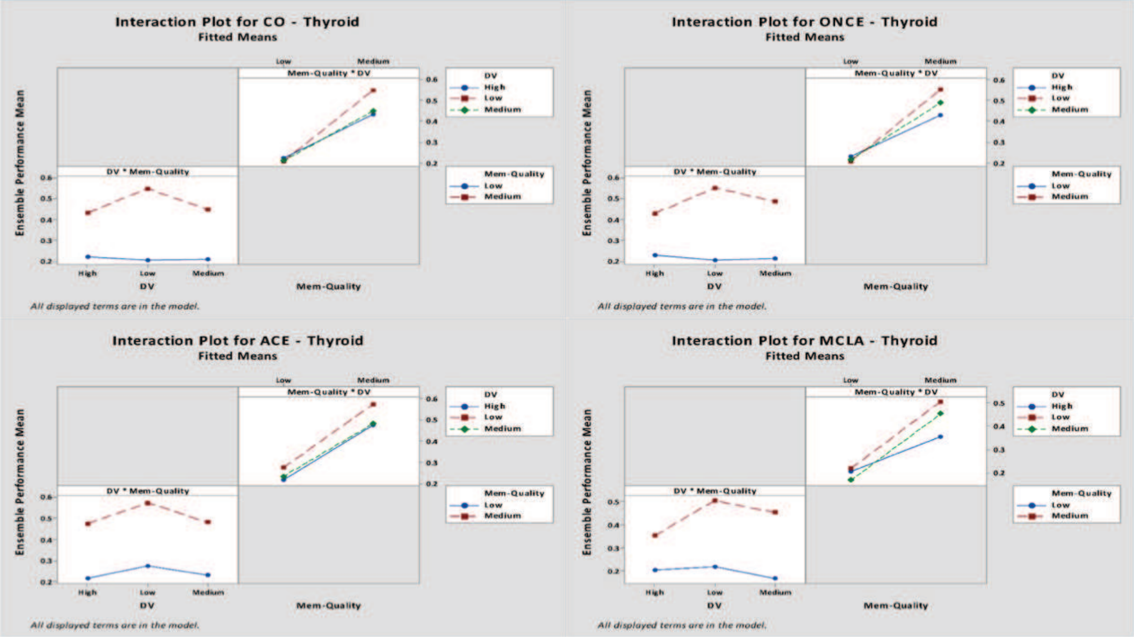


Figure 6.20: The interaction effects of the diversity and member quality on the response variables, which are CO, ONCE, ACE and MCLA, for Thyroid dataset.

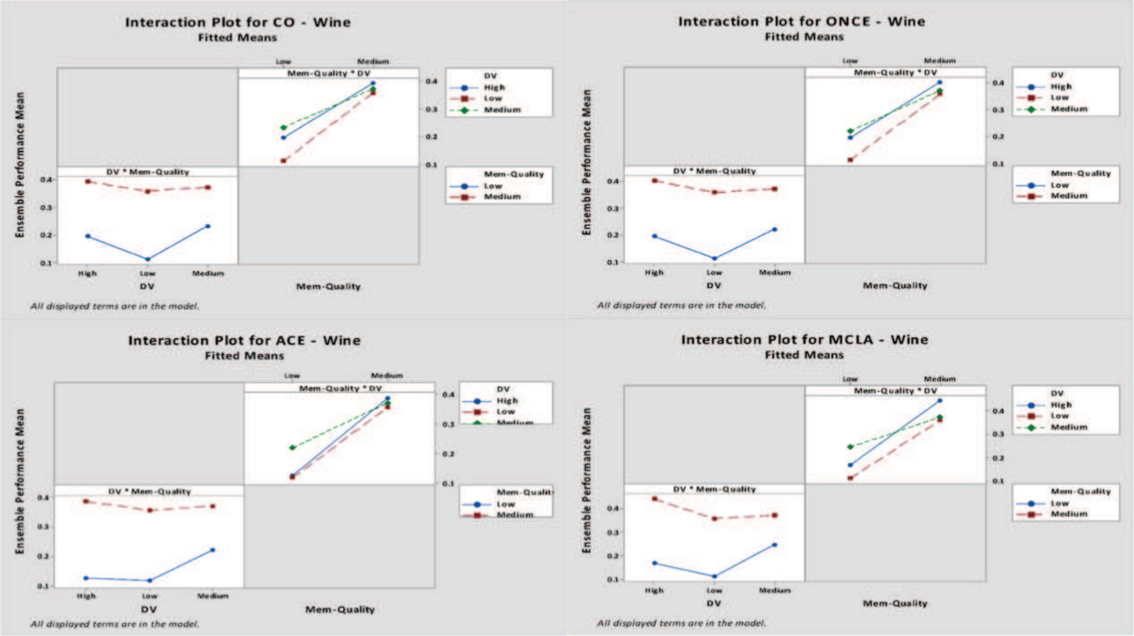


Figure 6.21: The interaction effects of the diversity and member quality on the response variables, which are CO, ONCE, ACE and MCLA, for Wine dataset.

test to the rest of the samples, for which Box-Cox suggested no transformations are needed, because ANOVA is still robust for small and even moderate departures from normality and homogeneity of variance. A rule of thumb is that the ratio of the largest to the smallest group variances should be 3 to 1 or less [10], and in our sample this ratio is too small.

Table 6.3 shows the results of the ANOVA, which include the type III sum of squares (SS), the degrees of freedom (DF), the mean squares (MS), and the F test statistics (F). Interactions between diversity (DV) and members' qualities are represented by DV * Mem-Q. P-values less than 0.05 represent rejection of the null hypothesis that the mean of the ensemble performance is statistically equal at all levels of the corresponding factors.

We observed that on both datasets and on all four consensus functions the member quality is statistically significant, whereas the diversity is not statistically significant in most cases except for the Wine dataset, where using the ACE consensus function it is significant (p-value$<$ 0.05). The interaction between members' qualities and diversity is not statistically significant in most cases, where we cannot reject the null hypotheses, but a small p-value on the Wine dataset using ACE and MCLA justifies rejection of the null hypothesis. So, in these particular cases the interaction is statistically significant.

However, the factor with most influence on the ensemble performance is the member quality (more so than the diversity among them). This observation is true for all the consensus functions used based on this experimental set-up. There is an interaction between the members' qualities and the diversity, but in most cases it is not statistically significant.

As the only significant interaction we had was in the Wine dataset using ACE and MCLA, we tested all their pairwise mean comparisons using a Tukey test [99] to find out where the significance was coming from. Table 6.4 shows the results of the Tukey test on the Wine dataset for ACE and MCLA. We can see that ten pairwise groups are significant out of 15 (in total). It is clear that most of the significant differences in these pairwise comparisons groups are coming from changes in levels of the members' qualities from low to medium. The only significant case (in both consensus functions) that the difference come from changes in the diversity levels

Table 6.3: The results of ANOVA Tests on two datasets using four consensus functions (CO, ONCE, ACE and MCLA), the bold value in the P-value column represents a statistical significant, which less than 0.05

Dataset	Consensus Function	Factor/Interaction	DF	SS	MS	F	P-value
Thyroid	CO	DV	2	0.016	0.008	2.10	0.128
		Mem-Quality	1	0.244	0.244	62.39	**0.000**
		DV * Mem-Quality	2	0.015	0.007	1.89	0.156
	ONCE	DV	2	0.014	0.007	2.48	0.089
		Mem-Quality	1	0.240	0.240	82.08	**0.000**
		DV * Mem-Quality	2	0.010	0.005	1.87	0.160
	ACE	DV	2	0.044	0.022	1.60	0.207
		Mem-Quality	1	0.672	0.672	48.72	**0.000**
		DV * Mem-Quality	2	0.006	0.003	0.22	0.802
	MCLA	DV	2	0.034	0.017	2.00	0.140
		Mem-Quality	1	0.539	0.539	62.76	**0.000**
		DV * Mem-Quality	2	0.031	0.015	1.85	0.162
Wine	CO	DV	2	0.029	0.014	1.86	0.160
		Mem-Quality	1	0.424	0.424	54.34	**0.000**
		DV * Mem-Quality	2	0.027	0.013	1.74	0.181
	ONCE	DV	2	0.025	0.012	1.68	0.191
		Mem-Quality	1	0.454	0.454	61.00	**0.000**
		DV * Mem-Quality	2	0.023	0.011	1.56	0.215
	ACE	DV	2	0.040	0.020	2.61	0.078
		Mem-Quality	1	0.520	0.520	67.43	**0.000**
		DV * Mem-Quality	2	0.055	0.027	3.59	**0.031**
	MCLA	DV	2	0.034	0.017	2.22	0.113
		Mem-Quality	1	0.517	0.517	66.57	**0.000**
		DV * Mem-Quality	2	0.098	0.049	6.33	**0.003**

and not from the members' qualities is (Medium Low) and (High Low).

However, these interaction effects tell us that the effect of diversity is conditioned by the members' qualities, and this is for all the three levels of diversity, on the other hand the effect of only the low level members' qualities is conditioned by the diversity, but not for the medium level of members' qualities. These results only occurred when using ACE and MCLA; and not with other consensus functions.

Table 6.4: The results of the Tukey test with Wine dataset using ACE and MCLA consensus functions, the bold value in the P-value column represents a statistically significant difference between the two groups compared, which is less than 0.05

Difference of DV*Mem-Quality Levels	the P-value of the ACE sample	the P-value of the MCLA sample
(High Medium) - (High Low)	**0.000**	**0.000**
(Low Low) - (High Low)	1.000	0.944
(Low Medium) - (High Low)	**0.000**	**0.000**
(Medium Low) - (High Low)	0.001	**0.015**
(Medium Medium) - (High Low)	**0.000**	**0.000**
(Low Low) - (High Medium)	**0.003**	**0.000**
(Low Medium) - (High Medium)	0.975	0.314
(Medium Low) - (High Medium)	**0.000**	**0.000**
(Medium Medium) - (High Medium)	0.998	0.403
(Low Medium) - (Low Low)	**0.007**	**0.004**
(Medium Low) - (Low Low)	0.608	0.306
(Medium Medium) - (Low Low)	**0.002**	**0.001**
(Medium Low) - (Low Medium)	**0.000**	**0.002**
(Medium Medium) - (Low Medium)	0.995	0.997
(Medium Medium) - (Medium Low)	**0.000**	**0.000**

6.2.2.3 Summary of Results

The experiment was motivated by the results in Sections 6.1 and 6.2.1, which suggested that there may be an interaction between the members' qualities and diversity. The main findings are as follow:

1. The results of the ANOVA showed that the main effect of the members' qualities is statistically significant, while the main effect of the diversity is not statistically significant measured by DV_{pARI}.
2. The result confirms that the quality of the members has more influence on the ensemble performance than diversity (measured by DV_{pARI}).

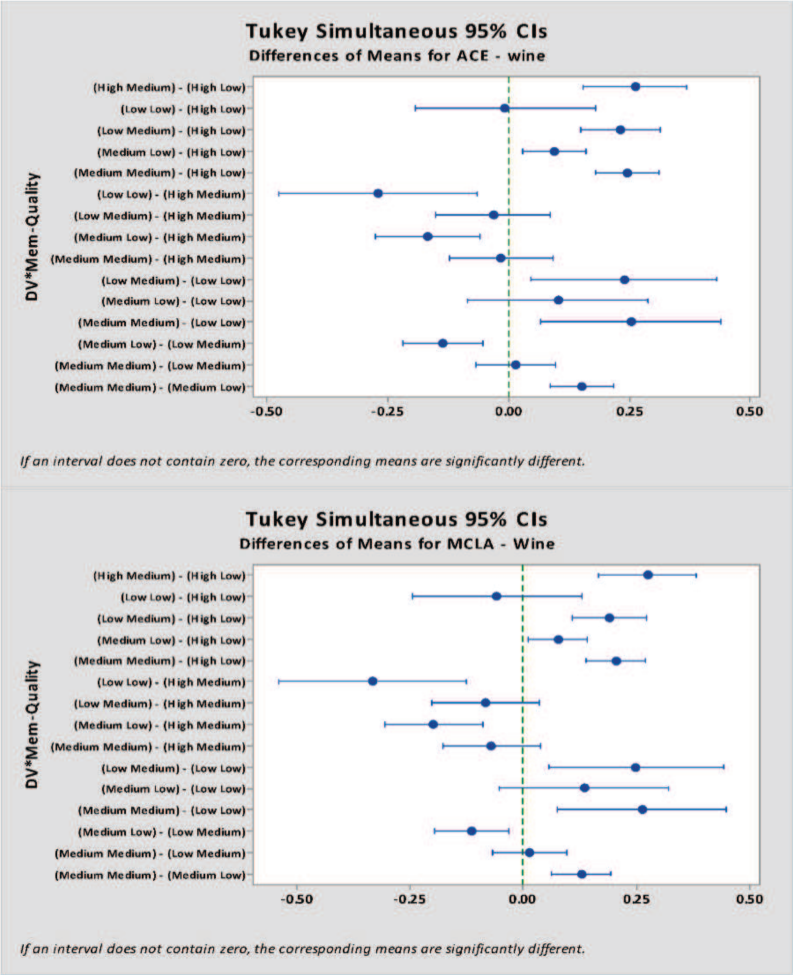


Figure 6.22: The Tukey results with Wine dataset using the ACE and MCLA. The Tukey test used to determine specifically which means are statistically significant different of the interaction effects using these consensus functions.

3. It is observed that graphically there was a small degree of interaction effect between the members' qualities and the diversity on the ensemble performance, but it was not statistically significant for Thyroid dataset

However, we cannot generalise these findings, due to two reasons:

1. The experiment was conducted using only two datasets.
2. The experiment was conducted using only DV_{pARI} diversity measure.

A further study might be undertaken, to design a suitable experiment, using artificial data for example, to study the interaction effect and to redesign it as a balanced design because it is easier in terms of the analysis and the interpretation of the results. In addition, further studies could use a large number of datasets and a large sample size so that the result could be generalised.

6.3 Discussion

The second central point of this research is to investigate diversity in the context of clustering ensembles, which was done in this chapter. Our primary investigation was carried out in order to discover the relationship between diversity and the ensemble quality.

To do that, firstly we reviewed the literature on the definition of diversity in the context of the clustering ensemble in Chapter 2. We found that there are two different types of definitions for diversity, and we named and defined them as follows:

1. **Ensemble Output Dependent (EOD):** the ensemble diversity is defined as the level of variation between its members and its final clustering result in terms of their matching labels. This kind of definition, includes DV_{np1}, DV_{np2}, DV_{np3} and DV_{np4} [39].
2. **Ensemble Output Independent (EOI):** the ensemble diversity is defined as the level of variation among the members themselves in terms of their matching labels. This kind of definition includes DV_{pARI}, DV_{pNMI} [25], and $Entropy$ [37].

Secondly, the experimental study was carried out in order to investigate the current existing diversity measures, for the EOI definition we have DV_{pARI}, DV_{pNMI} [25], $Entropy$, and for the EOD definition we have [37], DV_{np1}, DV_{np2}, DV_{np3} and DV_{np4} [39]. To the best of our knowledge, these are the only diversity measures that exist in the clustering ensemble literature. Although there have been some attempts to modify these measurements to be used with other external clustering validation indices, we did not consider them in this experiment, because they did not show superior results compared to the original ones [83].

The main finding of these experimental results was that these diversity measures are not capable of discovering a clear relationship between the ensemble quality and its diversity. This is in spite of the fact that a number of researchers have established a statement about this relationship, as we saw in Chapter 2 (Table 2.1 summarises them). Here we discussed our findings with the most recent of these statements, and as quoted from their authors, these are as follows:

1. In 2006, Hadjitodorov et al. [39] compared the existing diversity measures and said that: “The results favoured DV_{np3} as the one most related to the ensemble accuracy. Two typical patterns of diversity-accuracy relationship were found”. “One is almost monotonic-the larger the measure value, the higher the accuracy, while the other is shaded as a parabola with a maximum at about the middle of the diversity range”.
2. In 2009, Domeniconi and Al-Razgan [20] said that: “Our results reveal that a diversity measure based on ARI is more robust and consistent, and that high diversity signifies large accuracy”.
3. In 2010, Rozmus [83] said that: “From the experiments carried out it is rather difficult to find a strict and clear relationship between ensemble accuracy and the used measures of diversity, but in some cases it can be observed that for the pairwise and first non-pairwise measure lower diversity went together with higher accuracy whereas for the rest of non-pairwise measures higher diversity went together with higher accuracy”.
4. In 2011, Iam-On et al. [51] said that: “This result suggests that a high level of ensemble diversity is recommended for an accurate outcome”.

We argue that these statements are not really convincing and cannot be generalised, due to weaknesses identified in them. We noticed that in [39], the identified patterns are only related to their proposed measures DV_{np3}, in their experimental study they only used one consensus function which is the CO, and they have not tested that using different consensus functions, or at least using a commonly used consensus function such as MCLA. Moreover, a number of weaknesses are noticed with their proposed measures, which will be listed later in the section.

A similar weakness was noticed in [83], where they only used one consensus function which is based on an optimisation process, and again their result is only applied to this consensus function. In their statement, "the first non-pairwise measure" is the DV_{np1}, clearly what they found is that DV_{np1} revealed an opposite relationship between diversity and ensemble quality compared to the one discovered by other non-pairwise measures. This is exactly what we discovered about DV_{np1} that although it is by definition a diversity measure, its behaviour shows that it is in fact a measure of similarity between the ensemble and the members. This result was also noticed by Hadjitodorov et al. [39]. However, the main aim in [83] is to test the modified diversity measures in discovering a clear relationship, where they apply different clustering validating indices instead of ARI in the original diversity measure, including the Rand Index, the Jaccard and the Fowlkes and Mallows index; but they did not find a clear relationship between diversity and quality using these measurements.

The experiment in [51] only used one diversity measure, which is DV_{pNMI}, and it had a number of weaknesses. Firstly, it was conducted using only their proposed link-based methods and on 5 datasets. Secondly, looking closely to the results, we found that clearly the results from 3 out of 5 datasets do not support their statement. Thirdly, in the other 2 datasets, in particular in Glass dataset, the improvement in ensemble accuracy was not very high when the diversity was high compared with when it is in median level, while in Diabetes dataset the accuracy of the ensemble was around 0.1 measured by NMI, which is too low to be meaningful. Finally, we also noticed that they used a mixed heuristic generating technique to generate 60 members in each run, among them 20 generated using random k values chosen from the interval $[2, \sqrt{n}]$, and they used NMI as a measure of a member's accuracy and diversity measure, although previous work by Domeniconi and Al-Razgan [20] suggests that “a measure of accuracy/diversity based on ARI might be more robust and consistent than a measure based on NMI”. In this research, we found (in Chapter 4) that the NMI is sensitive when the two compared partitions have different numbers of clusters. Therefore, based on these arguments we do not think their results are valid.

However, in [20], we noticed that they investigated the diversity also using their proposed consensus functions and they used only two diversity measures (DV_{pNMI} and DV_{np3}). They concluded that DV_{np3} is highly related to the ensemble quality. They also made the following statement: “We finally note that ‘universal’ rules for choosing the preferred level of diversity should be used with caution, as the ‘optimal’ level clearly depends on the consensus function and on the dataset”. This corroborates our conclusions.

Furthermore, the experimental study in Section 6.1 also revealed other remarkable findings. It was observed that there is difference in behaviour between the EOI and EOD diversity measures. This difference is explained by the fact that the latter involved the ensemble results in their equations, whereas the other does not.

We also highlighted that the EOD diversity measures have a number of disadvantages:

1. To use them we have to combine the members and get the output of the ensemble, so they are useless without the ensemble results.
2. For this reason, they are sensitive to the type of the consensus function that is used to produce the ensemble output.

Therefore, until a new suitable diversity measure is defined, we recommend the EOI diversity measures. In particular, we recommended DV_{pARI} for the following reasons:

1. It is not dependent on the ensemble output; one can use it without running the full ensemble framework.

2. It is based on the matching labels between members, and it follows the common diversity definition, which is the level of variability among the ensemble members.
3. It is the most common diversity measure that has been used in the literature so far, some other researchers recommend to use it as well, due to its reliability and consistency.
4. It is very easy to compute and interpret.

Nevertheless, the results in Section 6.1 highlighted two issues that are related to the diversity and its relation to the quality of the clustering ensemble, the following subsections will discuss these issues in more detail.

6.3.1 Discussion on the Issues Raised in our Diversity Studies

In this Section, we discuss the work done in Section 6.2, which consider two issues raised in our studies of clustering ensemble diversity. To the best of out knowledge these two issues have not been studied in the literature, so we are unable to compare our results with others, and we hope our research will highlight them. The following sections discuss them separately.

Analysis of the Positive and Negative Effects of Diversity on the Ensemble Performance

Two different effects of diversity on the clustering ensemble appeared in the results in Section 6.1, which we defined as positive and negative effects because the diversity makes an ensemble better or worse than the average quality of the members. These two effects were observed also in Hadjitodorov et al. study's [39] and Rozmus study's [83], although they did not explain why these effects had occurred, so to the best of our knowledge these different effects of diversity on the clustering ensemble have not been looked at previously. There could be a number of reasons for this. Firstly, studying these effects is not an easy task at all in the context of cluster analysis, as it is unsupervised learning. Secondly, the absence of useful diversity measures for detecting a clear relationship makes the task more difficult as we saw in Section 6.1 in that none of the existing diversity measures are able to discover a clear relationship between the ensemble performance and diversity. Thirdly, the diversity and the other factors $(Q(\Gamma), CF, m)$ have a chain of interactions among themselves and these interactions make analysis of them extremely difficult.

Wang [106] suggested that a common and simple methodology in the classification ensemble is to study these factors one at a time, where only one factor is changed at any one time, while the impact of other factors are reduced to a minimum or kept fixed. In a clustering ensemble, factors such as the consensus function CF and the number of members m can be easily fixed to a known constant, but as we saw from the experiment in Section 6.1, diversity is very difficult to separate from the individual member quality. As a result of exploring the diversity of the members, their individual qualities are affected. In fact, we empirically tested this by selecting a number of members generated in the experiment (Section 6.1) that have same quality in the Thyroid dataset, and we found that their diversity (measured by DV_{pARI}) is equal to zero. This means that based on the generation techniques used in this experiment, whenever we have members with a fixed quality they are most likely to be identical or highly similar to each other in terms of the cluster structure.

However, in order to analyse these two different diversity effects, firstly we formally defined them and then we designed an experiment guided by the results in Section 6.1, which involved comparing a pair of two ensemble cases, where the first case represented the negative, while the second case represented the positive case. The average members qualities and ensemble diversities of these two ensemble cases told us that there was no difference between them, but the performance of the first case was lower than the average member quality (negative), while the performance of the second one was higher than the average member quality (positive). So, we looked at the individual members' qualities to compare them. In other words, we used a simple method that counts how many members have a poor-, good- and medium-quality compared to a particular pair.

The results of this experiment did not give us a solid conclusion on how to avoid the negative impact of diversity, and in fact it supports what we actually found in the previous chapters, that the consensus function is a very important factor for the ensemble performance.

A further analysis was carried out to gradually eliminate poor-quality members in these cases until we had only three high quality members, and we used CO, ONCE, ACE, and MCLA as the consensus function in turn. The results of this analysis showed that this elimination improved the average quality of members and decrease the diversity, and as a result improved the performance of the consensus functions in negative cases.

The Interaction between Members' Qualities and Diversity

We define the interaction between the members' qualities and diversity as how the diversity effect on the ensemble performance varies with the members' qualities and vice versa. We designed an experimental study in order to explore this interaction effect, and we implemented a 3 × 2 factorial experimental design study on two real-world datasets, which was Thyroid and Wine for the purpose of demonstrating the concept. Due to the time limit, we only used the DV_{pARI} measures in this experiment, and the members' qualities were measured by using an ARI index. For the diversity, the high, medium and low levels were considered, while for the members' qualities, only the medium and low levels were considered, the reason why we did not consider the high members' qualities level is that it is very difficult to generate this high level with the datasets tested using a single clustering algorithm. We ran the experiments on 4 consensus functions under 6 combinations of the two factor levels.

The results of the two-way ANOVA revealed that the main effect of the members' qualities is statistically significant, but it is unsurprisingly not statistically significant for the diversity factor (measured by DV_{pARI}). Conceptually, both of the factors are very important to the ensemble performance, and diversity is widely accepted as a crucial factor for building a successful ensemble; there is no need to build an ensemble with identical members [94, 39, 25]. Thus, we can say that the non-significant difference of diversity's main effect is down to the choice of the DV_{pARI} measure.

On the other hand, for the interaction effect, the result shows that graphically there is a small degree of interaction effect on the ensemble performance on the two datasets. The only interaction cases that were statistically significant were in the Wine dataset using ACE and MCLA. Then we used a Tukey test [99] to ascertain where the difference came from. The results provided evidence for the fact that the effect of the diversity varies depending on the level of the members' qualities, whereas the effect of only the low-quality members varies depending on the level of diversity with this dataset.

In conclusion, it is not clear whether the results of this experiment can be generalised or not, mainly because this experiment was carried out on only two datasets and the choice of the diversity measures. These results, in fact, only confirm the results that we obtained in Section 6.1. This specific diversity definition is not helpful in detecting the main effect of diversity on the ensemble performance. It also confirms that the quality of the members has more influence on the ensemble performance than diversity (measured by DV_{pARI}).

Secondly, this experiment was conducted as a result of an issue raised in Section 6.1, and as studying the interaction effect was not the focus of this research or one of its main objectives and also due to the time limited available, we could not spend much time on this experiment. The complexity of the nature of these factors and the non-existence of a useful definition of diversity make this experiment very hard to design. Moreover, as the interaction effect between diversity and the members' qualities has to be studied under their different combinations of levels, and as generating different sets of members for these combinations is data-dependent, so the data used needs to allow us to generate all the kinds of members' sets under these combinations. This would be best achieved using artificial data. To investigate this further would be beyond the scope of this thesis, but would be an interesting further study.

6.3.2 General Discussion on Diversity

In this section, we discuss our investigation on the diversity as a whole. In general, the results highlighted that diversity is an important factor to all the consensus functions used, probably after the quality of individual members, in terms of improving the ensemble quality, but using the existing diversity measures we were not able to discover a clear relationship between diversity and ensemble quality.

This finding leads us back to the original question of whether diversity is really important factor to the clustering ensemble performance. We have seen in the review in Section 2.3.1, that there is a general agreed perception upon the conceptual utility of diversity and there is no point in building an ensemble of identical members. This means that the members have to somehow be different from each other in order to gain the benefit of their combination.

Thus, in principle, diversity should be a useful factor in constructing a clustering ensemble, although all the existing definitions of diversity do not show clear evidence to support this principle in reality. One of the possible reasons, we think is that no diversity measure has been directly associated with the consensus function, which as we know, determined the output of the ensemble.

To the best of our knowledge there has been no attempt to use any of the current diversity measures in guiding the consensus functions when combining the members. We think the reason for this is that until now there has not been a universally accepted diversity definition, and the effectiveness of diversity in the context of the clustering ensemble is still questioned. Most previous work on diversity used diversity measurements to measure the diversity in the generated members and then select those with the desired level. Hadjitodorov et al. [39], for example, used the EOD diversity measures to select the better performance ensembles by varying the diversity from a lower level to a higher level. They recommended selecting an ensemble with a moderate level rather than a high level of diversity. To run a number of ensembles and select the one with the desired level of diversity is a time-consuming task that leads us to the same problem of the single clustering algorithm, which the clustering ensemble is meant to overcome.

On one hand, there is agreement on the importance of diversity, but on the other hand how to measure it and how to use this measure in designing an effective clustering ensemble is still an open question in this field. Therefore, after investigating the diversity and highlighting its related issues, in this study we suggest that it is essential to develop a new diversity measure in the context of the clustering ensemble and a way to use this measure in conjunction with the member combining process.

6.4 Summary

This chapter investigates the diversity of the clustering ensemble and its relation with the ensemble performance. To do that, we designed an experimental study to test the validity of the existing diversity measures using 4 consensus functions including CO, ONCE, ACE and MCLA. The main finding of this experimental study is that although all the current diversity measures are designed to measure diversity among members, they are not doing their job properly in terms of measuring the actual members' diversity, and helping in discovering a clear relationship between diversity and ensemble performance. Furthermore, the results raised two issues, these are: (1) Diversity can have a positive or negative effect on the ensemble performance. (2) There may be an interaction existing between diversity and the members' quality.

In regard to the first issue, we had two sets of ensemble members that all DV measures, their average member qualities and the standard deviations of their member qualities tell us that there are no significant differences between the two sets of ensemble members. However, they produced two significant clustering results: one with a good ensemble performance (a successful combination pattern as the diversity has a positive effect on the ensemble), while the other one had a poor ensemble performance (a failure combination pattern as the diversity had a negative effect). We used a simple method of looking at the quality of the combined individual members and count how many of them are as poor as the negative ensemble case, as good as the positive ensemble case and had a medium quality, which is between good and poor quality. The results showed that this simple method clearly did not explained how to avoid the negative effect of the diversity, but some characteristics of the pattern of success and failure for each of the four consensus functions have been reported.

In regard to the second issue, we investigated if there is an interaction effect between diversity and members' qualities on the ensemble performance. We implemented a 3×2 factorial experimental design study using two real-world datasets (the Thyroid and Wine datasets). The results revealed that there was small degree of interaction between the diversity and members' quality, and in one case this interaction proved to be statistically significant on the ensemble performance when only ACE or MCLA was used as a consensus function. Moreover, this experiment demonstrated that there was a statistical significance for the main effect of the members' qualities on the ensemble performance, but not for diversity's main effect.

However, the answer to the question being asked in this Chapter (Does the diversity influence the ensemble performance?) is that: conceptually, yes, as there is a wide agreement in the literature on the importance of diversity with regard to ensemble performance. Practically, the correlation between the ensemble quality and diversity, as measured by most of the current definitions, indicates that there is a weak relationship between them, although there are a few cases where a strong relationship is observed. This was only discovered by dependent measures, and we noticed that these measures are inconsistent in their results. For the same dataset, with the same generated members and using different consensus functions, the discovered relationship can be changed from a strong to a weak relationship from one consensus function to another. However, at the moment, as a result of the absence of a useful diversity measure, we are unable to fully answer this question. A useful measure is viewed as one it would allow us to measure the true diversity in the ensemble members, that can be used by the consensus function to combine the members to produce high-quality clustering results.

Chapter 7

Conclusions and Further Work

7.1 Conclusions

Upon completing this research on two main issues: consensus function and diversity, the following conclusion can be drawn.

7.1.1 On the Consensus Function

The first focus of this thesis is the consensus function. Firstly, we proposed the Object-Neighbourhood Clustering Ensemble (ONCE) to address the problem of uncertain agreements between members. We studied the effectiveness of ONCE and we compared it with CO using Single, Average and Complete linkage, and with three link-based method named CTS, SRS and ASRS. Also, we compared ONCE with the well-known clustering algorithm *k-means* and the experimental results showed that:

1. The most appropriate linkage method is the average linkage method.
2. On average, ONCE outperforms CO, CTS, SRS and ASRS.
3. There is a statistical difference between ONCE and ASRS, and between ONCE and CO under our experimental set-up.

4. We tried to develop ONCE further by considering only the most common neighbours to objects pair results in a new algorithm called $\mathcal{E}$-ONCE. The experimental results show that using $\mathcal{E}$-ONCE does not improve the quality of the ensemble much further compared to ONCE, which is preferred.

Secondly, we proposed two new consensus functions named the Dual-Similarity Clustering Ensemble (DSCE) and the Adaptive Clustering Ensemble (ACE). The novelties of DSCE and ACE are as follows:

1. They are based on two similarity definitions; the similarity between the initial clusters themselves, and the membership of objects to clusters.
2. They produce the final clustering result without requiring the application of an ordinary clustering algorithm, unlike most of the existing clustering ensemble methods including CO, CTS, SRS , ASRS and ONCE.
3. They are efficient, because they only calculate the pairwise similarity between initial clusters and not objects, and the number of these clusters is much smaller than the number of objects in the dataset.

ACE is an improved version of the DSCE algorithm in three main aspects. Firstly, the stability of the DSCE has been improved by producing the final clustering result with the pre-defined k. Secondly, the effect of its two parameters (α_1 and α_2) on the quality of the final result has been reduced by applying an adaptive strategy for the value for these parameters. Finally, the object neighbourhood similarity for the uncertain objects has been taken into account, in order not to lose any information when we eliminate inappropriate clusters. ACE works in three stages, which are:

1. Transformation stage: the initial clusters are transformed into binary vector representations.
2. Generating Consensus Clusters: this calculates the similarity between initial clusters and captures the relationship between clusters. It merges the most similar clusters to produce the intended k consensus clusters.

3. Resolving Uncertainty: identifies the object's certainty of being assigned in the initial clusters. It focuses on the cluster quality and resolves the uncertain objects by assigning them to a cluster in a way that has a minimum effect on its quality.

We tested DSCE and ACE methods on 8 real-world benchmark datasets. The experimental results showed that:

1. On average DSCE outperforms the other clustering ensemble methods including MCLA, CO, ONCE and DICLENS.
2. DSCE is statistically significantly better than the CO and DICLENS methods, but not the ACE, ONCE and MCLA methods.
3. ACE does not outperform its predecessor DSCE under our experimental setup, although it outperforms the other methods. But, ACE has the ability to combine members without any conditions about the number of clusters they have.

7.1.2 On Diversity

Diversity in the context of the clustering ensemble has two different types of definition: the Ensemble Output Dependent (EOD), where the ensemble diversity is defined as the level of variation between its members and its final clustering result in terms of their matching labels, and the Ensemble Output Independent (EOI), where the ensemble diversity is defined as the level of variation among the members themselves in terms of their matching labels.

The second focus of this thesis was to investigate ensemble diversity. Our investigation in Chapter 6 revealed the following:

1. The existing measures (EOI and EOD) are unable to determine a clear relationship between diversity and the ensemble performance.

2. Most diversity measures only revealed a weak correlation between diversity and ensemble performance using most consensus functions (CO, ONCE, ACE and MCLA).
3. The EOD diversity measures require the final clustering results to be available, otherwise the measures cannot be used. To use them in selecting ensemble members with the desired level of diversity is time-consuming.
4. The EOD diversity measures are sensitive to the type of consensus function used, and the discovered relationship can vary from a strong to a weak relationship, from one consensus function to another.
5. We observed that the EOI diversity measures behave in similar way to each other, but different from the EOD diversity measures.
6. Among the EOD diversity measures, the DV_{np1} is not a valid diversity measure as it always gives an opposite pattern compared to other measurements.
7. The experimental study on the diversity measurements raised two issues that required investigation:

   (a) Diversity can have a positive and negative effects on the ensemble performance. The issue was that all the existing DV measures and the average member qualities told us that there was no difference between two ensemble patterns, but the ensemble performance of the first pattern was lower than the average member quality (negative), while the ensemble performance of the second pattern was higher than the average member quality (positive).

   (b) There may be an interaction existing between the diversity and the members' qualities, then the effect on the ensemble performance might be determined jointly by them.

The two issues noted in point 7 above, were investigated and the following was achieved:

1. We established the formal definitions of the positive effect and the negative effect of diversity on the ensemble performance.
2. In the negative cases, we found that removing the poor-quality members contributed to improving the performance of CO, ONCE, ACE, and MCLA. In addition, the performance of ONCE and ACE improved further by removing the poor-quality members in the positive cases.
3. The effect of diversity differs from one consensus function to another, but the main characteristics of the pattern of success and the pattern of failure are as follows:
    - In the pattern of success, the ensemble members appeared to have a medium level of average quality accompanied by a medium level of diversity among them (measured by DV_{pARI}). Precisely, in order to use CO and ONCE as the consensus function, the members should have between medium and high average quality (measured by ARI) with a medium diversity among them. In order to use ACE, the combined members should have a medium level of average quality, accompanied by a medium level of diversity DV_{pARI}. MCLA prefers the combined members to have some sort of diversity, otherwise it will perform poorly, even when the members have high-quality clusters.
    - In the pattern of failure, an ensemble with a low average member quality, accompanied with a high level of DV_{pARI} and a low level of DV_{np3}, would result in a poor ensemble performance. A gradual increasing of the average member quality, along with decreasing DV_{pARI} and increasing DV_{np3} by removing the poor-quality members, improves the ensemble quality.
4. We ran a pilot study by implementing a factorial design experiment to investigate the interaction effect between the diversity and members' quality.
    - We found that the main effect of diversity on the ensemble performance was not statistically significant (diversity measured by DV_pARI), whereas

the members' quality effect was statistically significant.

- We showed that graphically there was a small degree of interaction effect on the two datasets used, but only on the Wine dataset, using ACE and MCLA, was this interaction effect statistically significant.

For diversity research, it is widely accepted that diversity is an important factor when building a clustering ensemble, as there is no need to build an ensemble with identical members. However, we conclude that how to measure diversity in the context of a clustering ensemble, and how to use it, is still an open question.

7.1.3 Contributions

The contributions made in this thesis are as follows:

- A new consensus function has proposed based on Object Neighbourhood Similarity, named an Object Neighbourhood-based Clustering Ensemble (ONCE).
- Two new consensus functions based on Dual-Similarity Measurements have been proposed (DSCE and ACE), where the similarity between initial clusters is measured, followed by membership similarity between candidate clusters and objects.
- A better understanding has been gained of the existing clustering ensemble diversity definitions in terms of their ability to discover the relationship between diversity and ensemble quality. Also, two diversity issues have been highlighted, which are:
  - The positive and the negative effects of diversity on the ensemble quality.
  - The possibility that an interaction exists between diversity and member quality.

7.2 Suggestions for Further Work

This research has highlighted a number of areas that could be explored further in the future; these are:

- The definition of W in the ONCE algorithm intends to solve the problem of uncertain objects by taking into account the similarity of their neighbours. However, by doing that, we may affect the similarity of some certain objects to make them become uncertain, so in our future work we will look further into this issue.
- Testing DSCE and ACE on big datasets.
- In this research we used the 'set correlation' as a cluster similarity measurement to measure the similarity between clusters; a further development of ACE would be to use other binary similarity measurements instead of S_c, or a combination of more than two similarity measures.
- In ACE, the quality of the cluster is measured as compactness; other measurements of cluster quality could also be investigated.
- A further development of ONCE and ACE would be to integrate the elimination mechanism of poor-quality members, which we introduced in Chapter 6.2, in the process of combining the members.
- A new diversity measure should be developed, and researchers should investigate in depth how ensemble members can be different from each other in terms of clusters. In the clustering analysis field, one should ask in which aspects two clustering results can be different/dissimilar from each other. This kind of comparison has been studied in clustering validation methods, and maybe using or modifying one of the internal validation indexes, for use as a measure of diversity, would be useful.
- Researchers should investigate how we can use diversity to guide the consensus function in combining the members, and to generate more members if needed.

- An experimental study should be designed using an artificial dataset, to investigate the interaction effect between diversity and the members' quality on the ensemble performance.

Appendix A

In this appendix, we give the complete results obtained for the experiments conducting in Chapter 6 in particular in Section 6.1. Firstly, the statistical summary of the ensemble quality results for each dataset as well as the qualities of their generated members is presented in Table A.1, Section A.1. Secondly, a statistical summary of all the diversity measure results is plotted in a boxplot in Section A.2. Finally, Section A.3 demonstrates the Experiment Results of the Diversity in Line Charts.

A.1 The Statistical Summary of the Results

The statistical summary of the ensemble performance for each of the 8 datasets is shown in Table A.1, which includes the maximum, minimum and average values as well as the standard deviation of 100 runs. The Table also includes the highest maximum value of the members' performance. For clarity, the bold value in each column represents the best ensemble performance in terms of the quality for the specified dataset compared to other ensemble methods.

The ACE method achieved the highest maximum quality in 7 datasets and also achieved the best average quality in 4 datasets, including Iris, Wine, Glass and Bcw, compared with other ensemble methods. We note that on average, the CO method achieved the best performance for only one dataset (the Bcw dataset). The CO method achieved a performance very close to the best performance in the Mfeatures and Glass datasets.

Table A.1: Statistical summary of the ensemble qualites and the generated members (Mem) in all tested datasets.

		Iris	**Wine**	**Thyroid**	**Mfeatures**	**Glass**	**Bcw**	**Soybean**	**Ionosphere**
CO	**Max , Min**	0.730 , 0.703	0.431 , 0.333	0.605 , 0.155	0.334 , 0.309	0.287, 0.240	0.846 , 0.846	0.748 , 0.545	0.178 , 0.173
	Ave-Std	0.721 ± 0.007	0.381 ± 0.039	0.376 ± 0.163	0.321 ± 0.007	0.267 ± 0.010	**0.846**±0.000	0.556 ± 0.029	0.177 ± 0.001
ONCE	**Max , Min**	0.730 , 0.56	0.438 , 0.333	0.637, 0.155	0.334 , 0.308	0.270 , 0.219	0.846 , 0.830	0.661 , 0.545	0.178 , 0.173
	Ave-std	0.716 ± 0.032	0.371 ± 0.039	0.406 ± 0.138	0.322±0.007	0.249 ± 0.011	0.846±0.002	0.585±0.047	0.177 ± 0.001
ACE	**Max , Min**	**0.834** , 0.633	**0.706** , 0.339	0.656 , 0.031	**0.343** , 0.192	**0.303** , 0.224	**0.857** ,0.839	**1.000** , 0.225	**0.183** , 0.178
	Ave-Std	0.732±0.023	0.411±0.045	0.343 ± 0.153	0.315 ± 0.027	0.269±0.016	0.846 ±0.004	0.577 ± 0.129	0.178 ± 0.000
MCLA	**Max , Min**	0.744 , 0.690	0.445 , 0.315	**0.692** , 0.192	0.339 , 0.114	0.268 , 0.010	0.852 , 0.830	0.875 , 0.545	0.178 , 0.168
	Ave-Std	0.719 ± 0.007	0.377 ± 0.025	0.531±0.097	0.319 ± 0.023	0.196 ± 0.038	0.845 ± 0.004	0.553 ± 0.035	0.177 ± 0.002
Mem	**Max , Min**	**0.868** , 0.012	0.601 , 0.011	0.687 , 0.012	**0.503** , 0.000	**0.305** , 0.011	**0.868** , 0.052	**1.000** , 0.048	**0.299** , 0.005
	Ave-Std	0.625 ± 0.007	0.307 ± 0.008	0.292 ± 0.017	0.259 ± 0.004	0.204 ± 0.004	0.627 ± 0.013	0.554 ± 0.023	0.141 ± 0.006

We found that CO was the most consistent method, as it had a small standard deviation in 5 datasets; although the diversity varied between 0.2 to 0.65 in these datasets, which means that the CO method was not affected by it. It is also noticed that standard deviations of CO in Bcw and of ACE in Ionosphere dataset are equal to 0 which means that their performance in 100 runs are identical, although the diversity varies between 0.33 to 0.54 in Bcw and between 0.43 to 0.66 in Ionosphere (measured by DV_{pARI} as seen in Figure A.1, A.2 and A.3). Thus, the performance of CO on Bcw and ACE on Ionosphere were unaffected and remained constant by the generated diversity. We will investigate these findings further in terms of how much diversity was generated in the next two sections, using boxplot and line charts.

A.2 Demonstrating Results in Boxplots

Figure A.1, A.2 and A.3 show 5 boxplots for the results from measuring the diversity achieved with the different methods on the 8 datasets. These boxplots show the range of diversity values for 3 pairwise (Figure A.1) and 4 non-pairwise diversity measures (Figure A.2(a), A.2(b), A.3(a) and A.3(b)). For clarity, these boxplots present six statistics: the minimum, the lower quartile, the median, the upper quartile, the maximum and the mean (represented by a star) in a visual display. The larger height of the box means that the diversity values in 100 runs for a particular dataset and using a particular measure are wider, while a box of small height means that the diversity values are very close to each other. From these plots, the aim is to discover the distribution of the diversity results from 7 different measurements in the tested datasets.

Generally speaking, all the diversity measures have a value range of $[0, 1]$ except DV_{np4}, which has an open value range of $[0, \infty]$. It is clear that the range of diversity values varies from one type of measurement to another type. The range of diversity values measured by pairwise measurements DV_{pARI} and DV_{pNMI} is more or less the same in most datasets, which indicates that these measurements display similar behaviour in measuring/estimating the diversity. The maximum level of diversity with all datasets reached just below 0.8, using the *Entropy* measurement in the Thyroid dataset, while the minimum level of diversity was equal to 0.2 with the Iris dataset.

On the other hand, the range of diversity in non-pairwise measurements is not too wide, especially using DV_{np1}, DV_{np2} and DV_{np3}. We noticed that involving the ensemble result in calculating the diversity in these measurements is highly associated with the ensemble quality. For example, when using the MCLA method on the Soybean dataset, the group diversity measures gave different results from those using the other ensemble methods. This indicates that the MCLA results are more diverse from the members than the other three methods in this dataset.

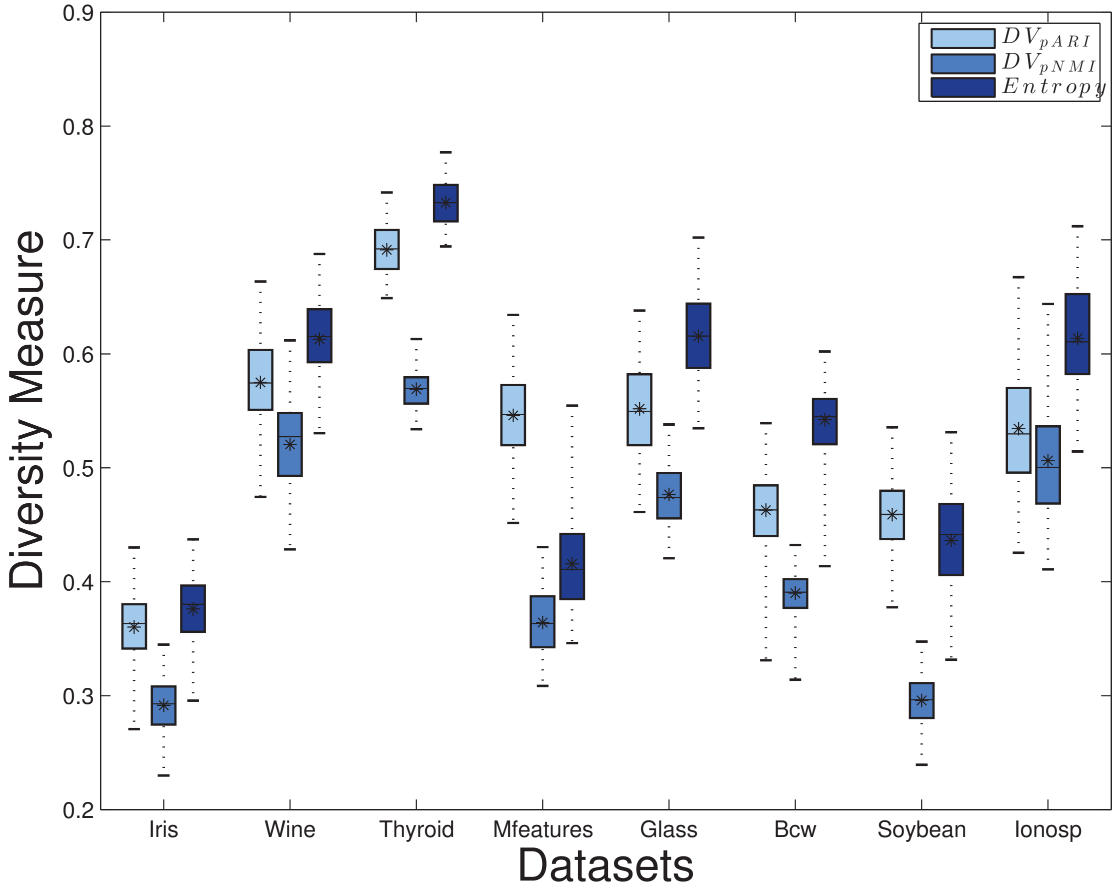
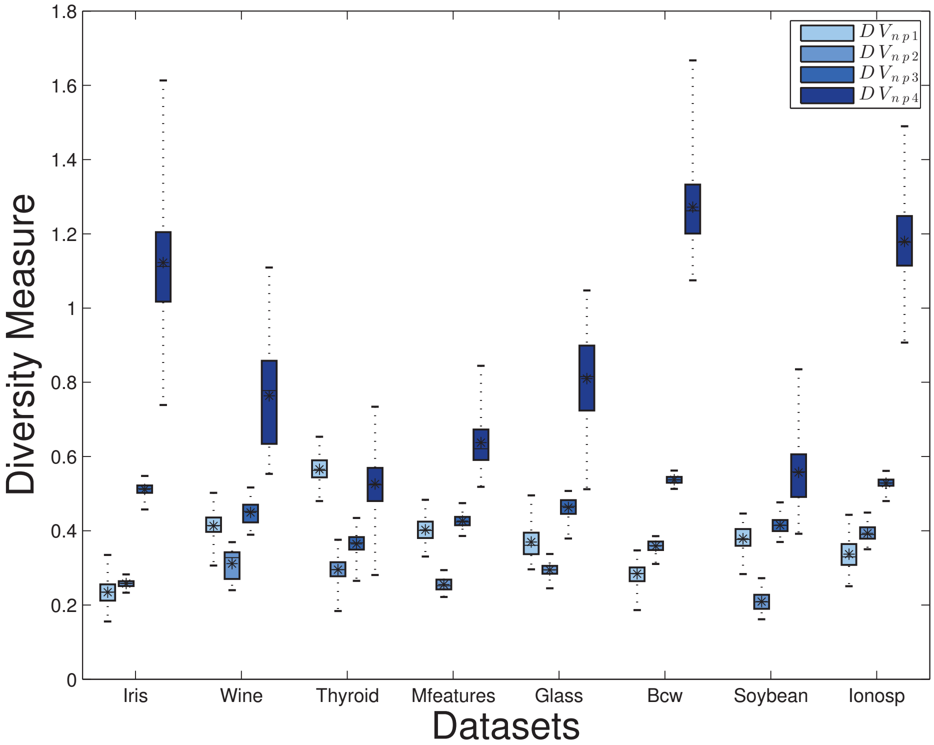
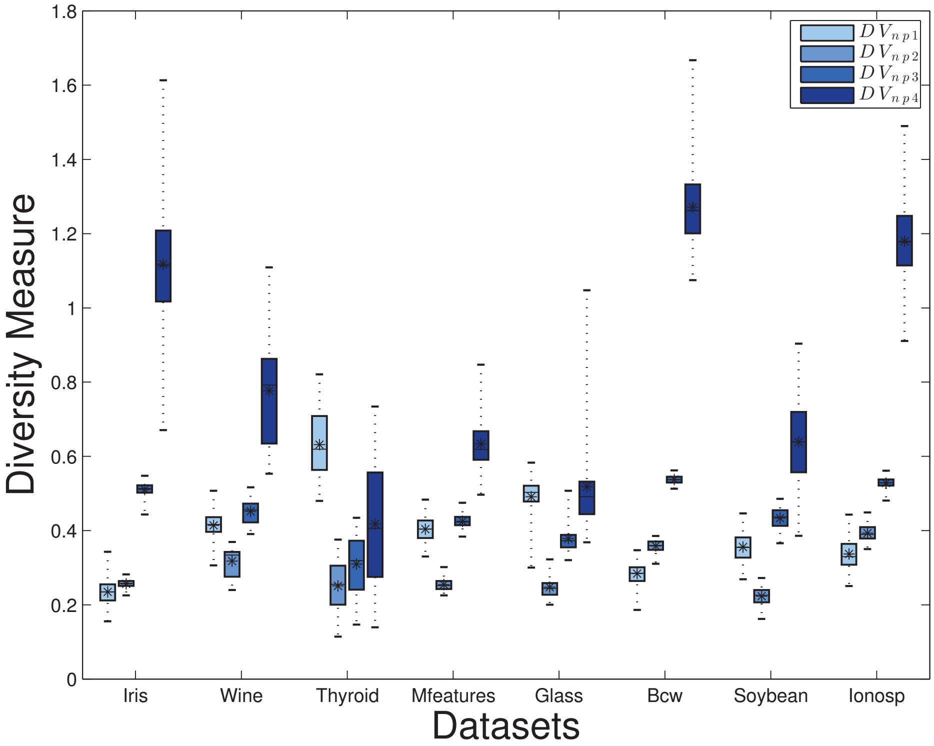


Figure A.1: The boxplot of the diversity results measured by the pairwise diversity measures: it shows the distribution of the diversity values of generated members in 100 runs in the 8 tested datasets. The line in each box represent the median value of the diversity and the star represents the mean value of 100 runs.

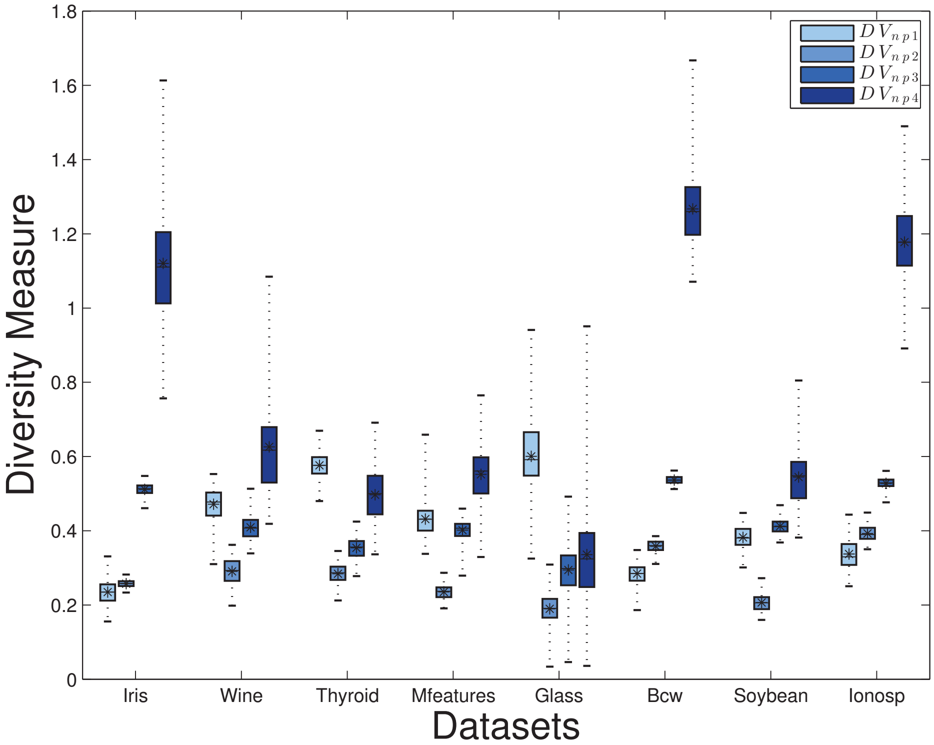


(a) The non-pairwise diversity measures using CO method

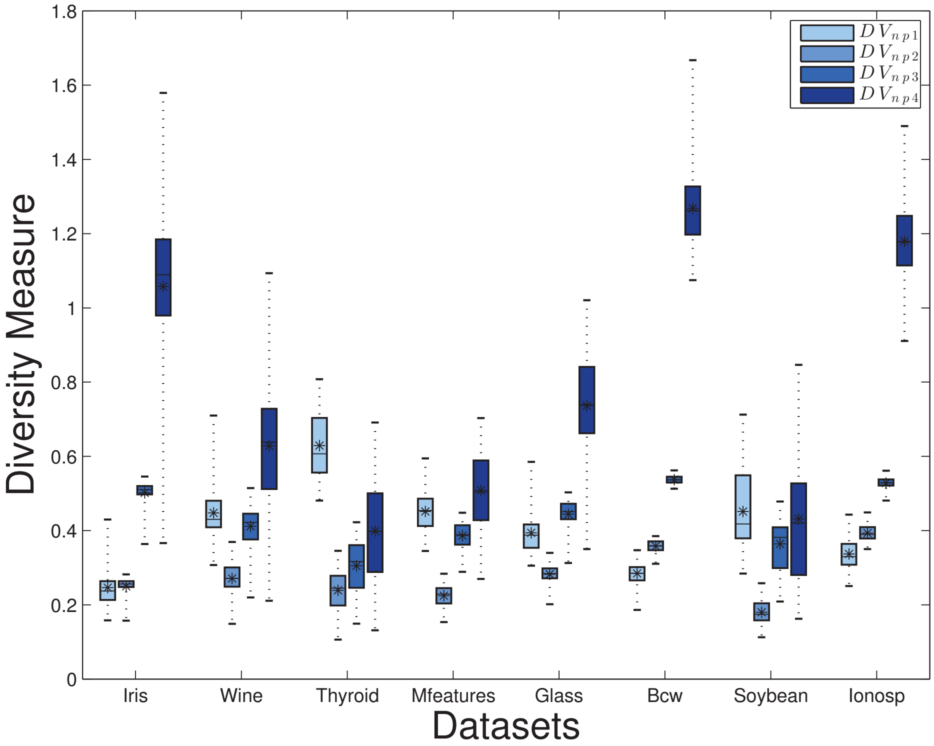


(b) The non-pairwise diversity measures using ONCE method

Figure A.2: The boxplot of the diversity results measured by the non-pairwise diversity measures using CO and ONCE methods.



(a) The non-pairwise diversity measures using ACE method



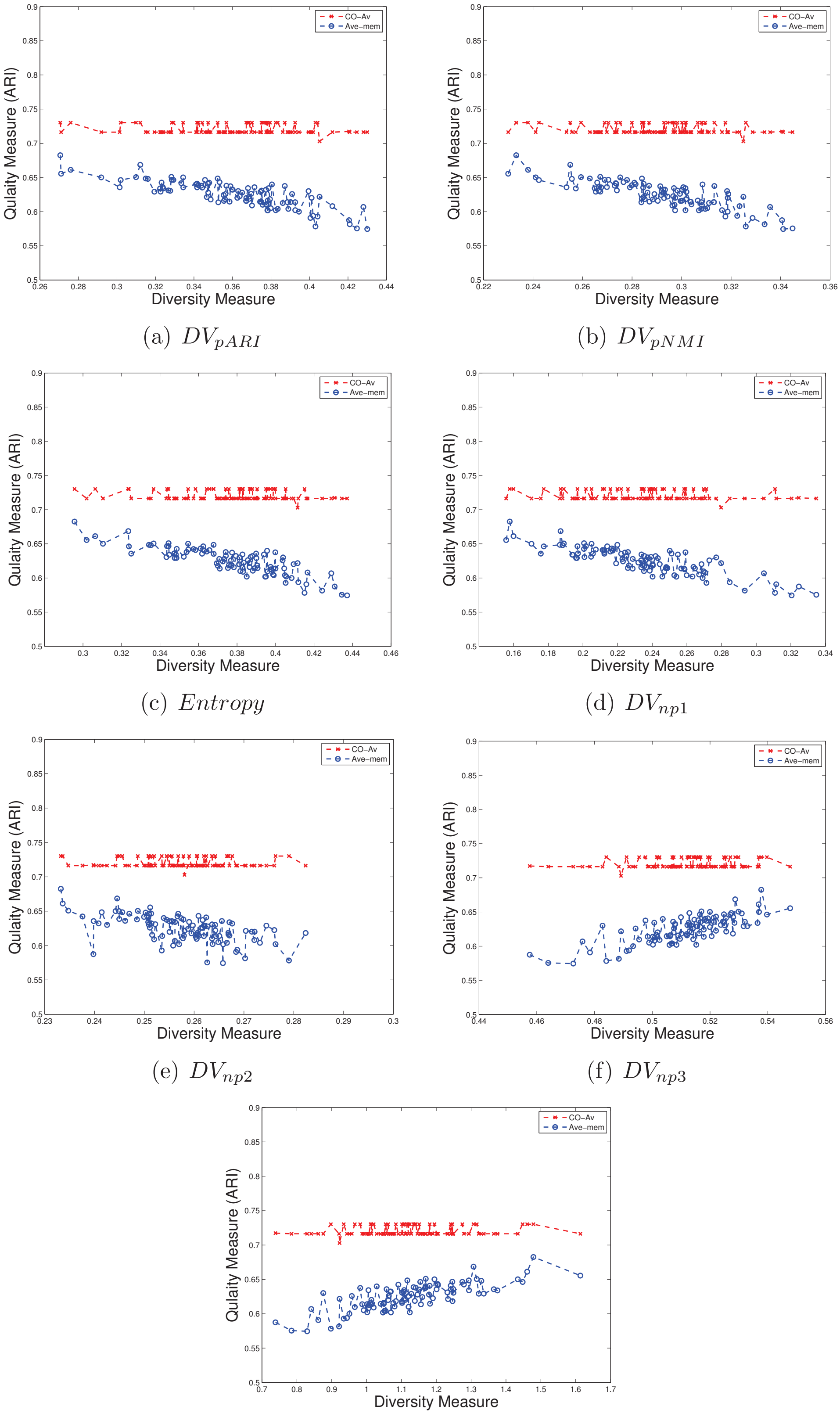
(b) The non-pairwise diversity measures using MCLA method

Figure A.3: The boxplot of the diversity results measured by the non-pairwise diversity measures using ACE and MCLA methods.

A.3 Demonstrating Results in Line Charts

For each dataset, we have 4 ensemble results; thus we plot 32 Figures, as shown from Figure A.4 to A.35. In each Figure, we have 7 subfigures, each one of which represents a different diversity measure. So in total, we have 224 subfigures.

In Figures A.4 $\sim$ A.35, a high value in DV_{pARI}, DV_{pNMI} and *Entropy* means that the members of the ensemble are very different from each other, while a lower value in these measures means that members are very similar to each other. We should mention that these interpretations are different from other diversity measurements as a high value in the non-pairwise individual diversity measures means that members are different from the ensemble results and a lower value means that they are similar to the ensemble results. A high value of the quality measure means that the ensemble quality and the average member quality are more accurate to the truth label of the dataset, and a lower value means that they are inaccurate.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

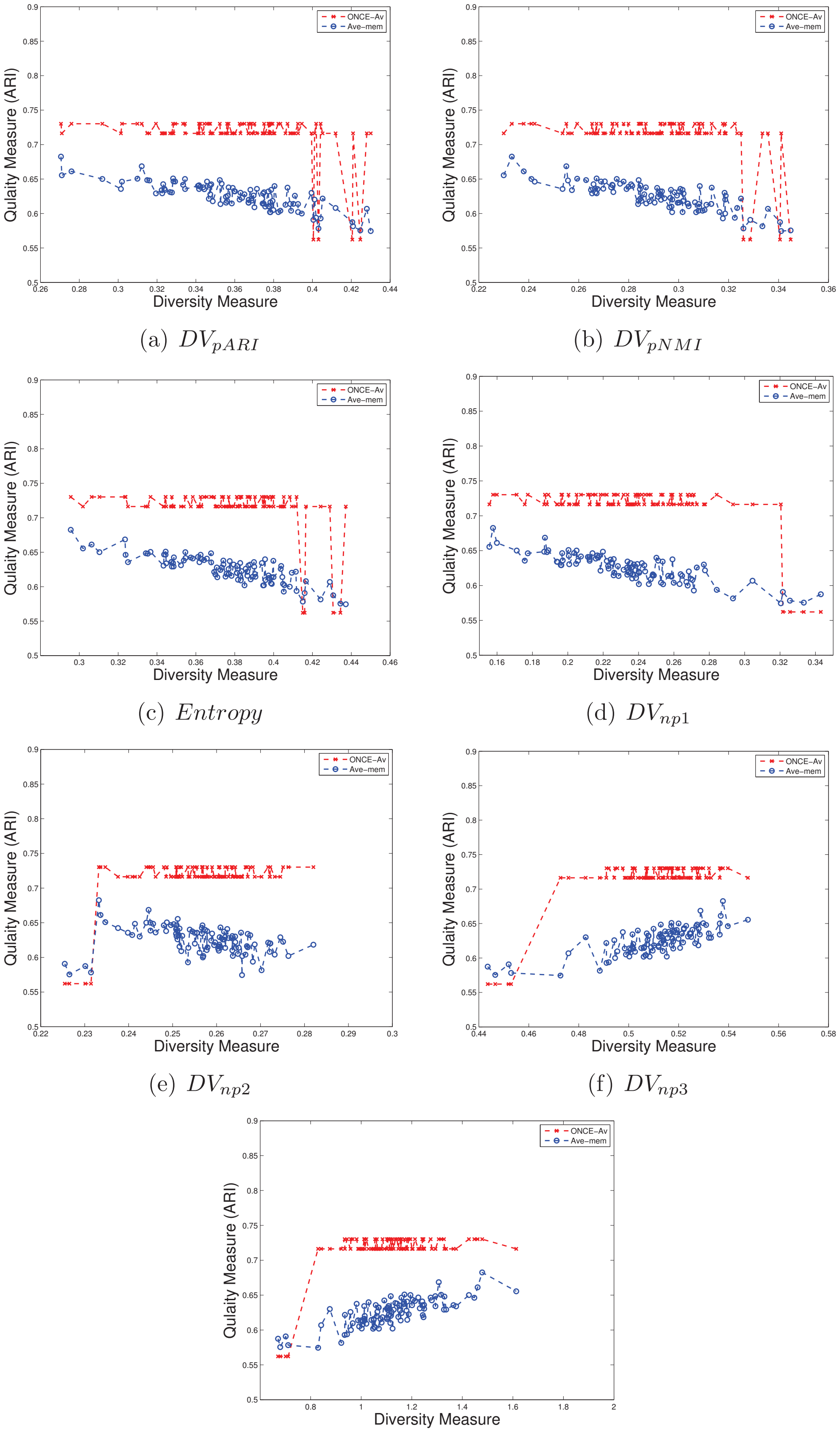
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.4: The seven diversity measures from Iris dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

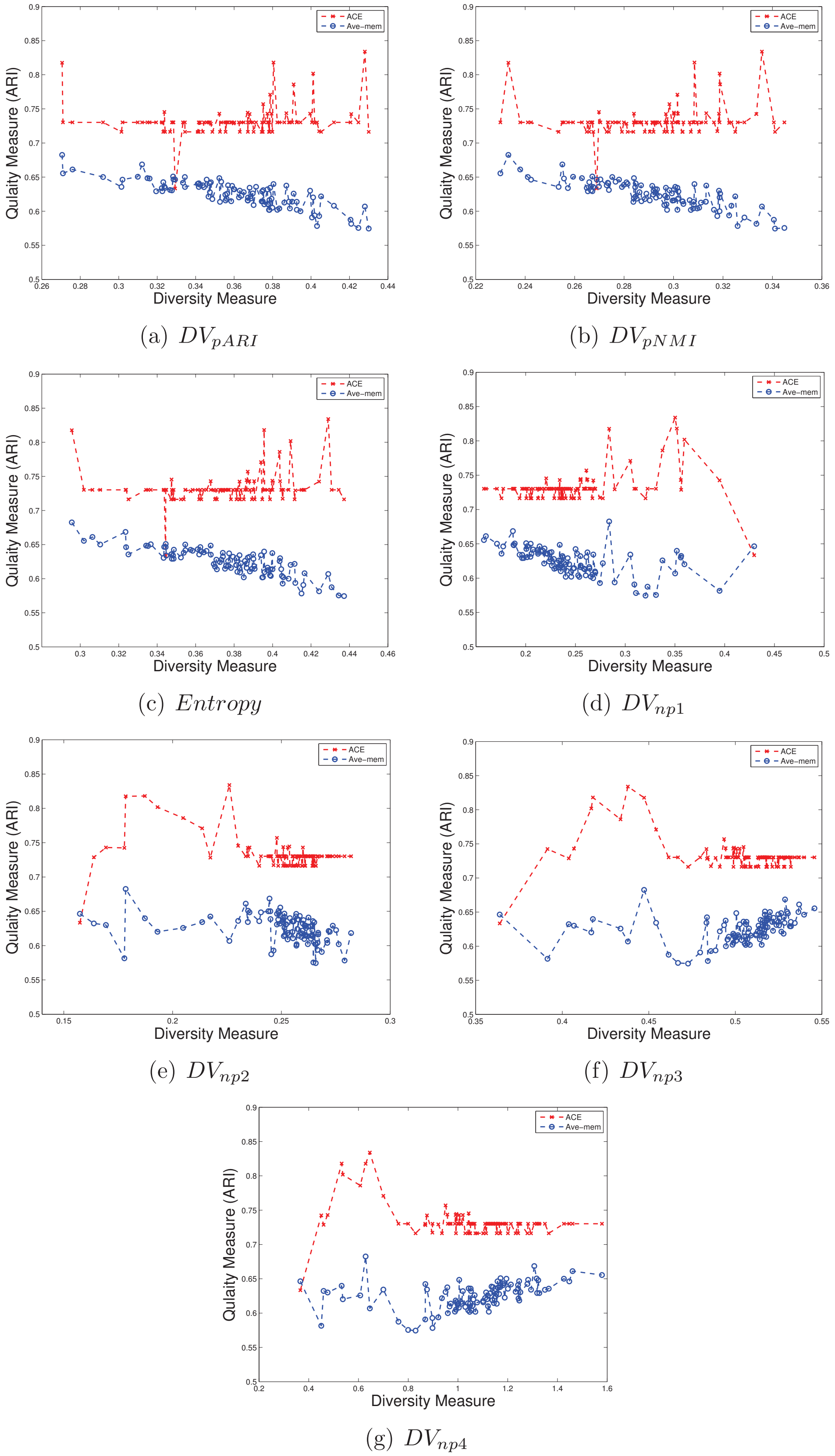
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.5: The seven diversity measures from Iris dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

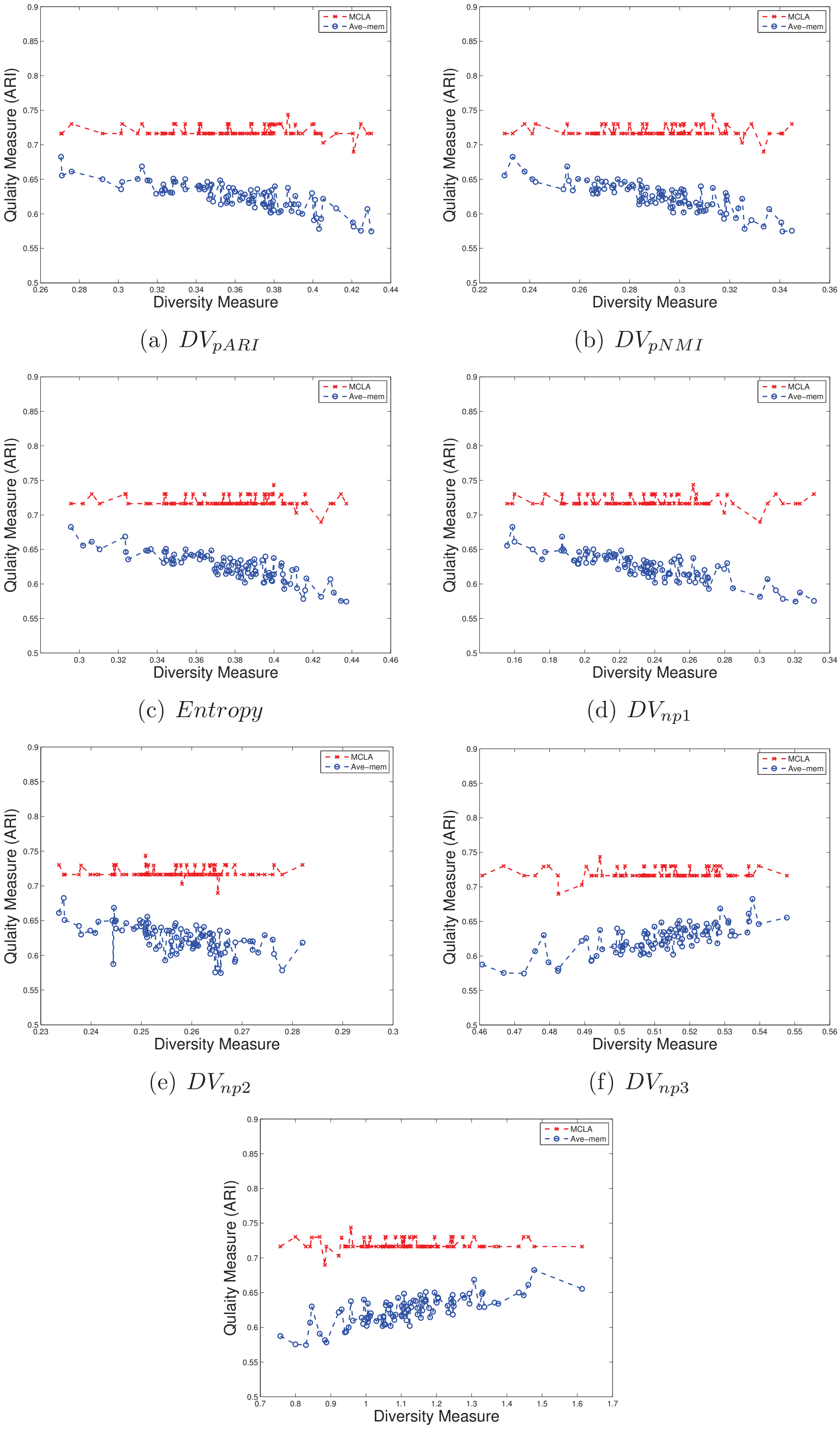
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.6: The seven diversity measures from Iris dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

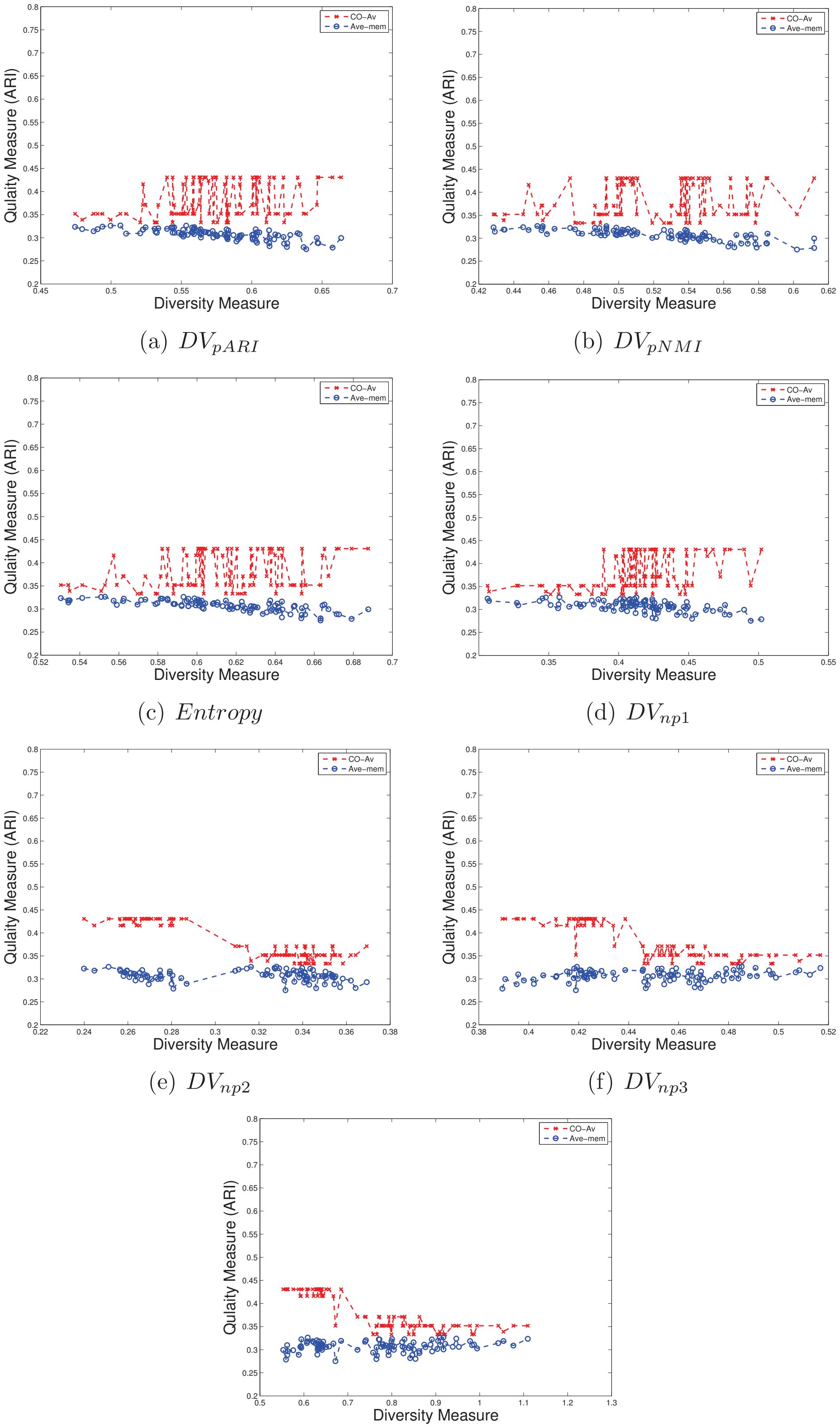
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.7: The seven diversity measures from Iris dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

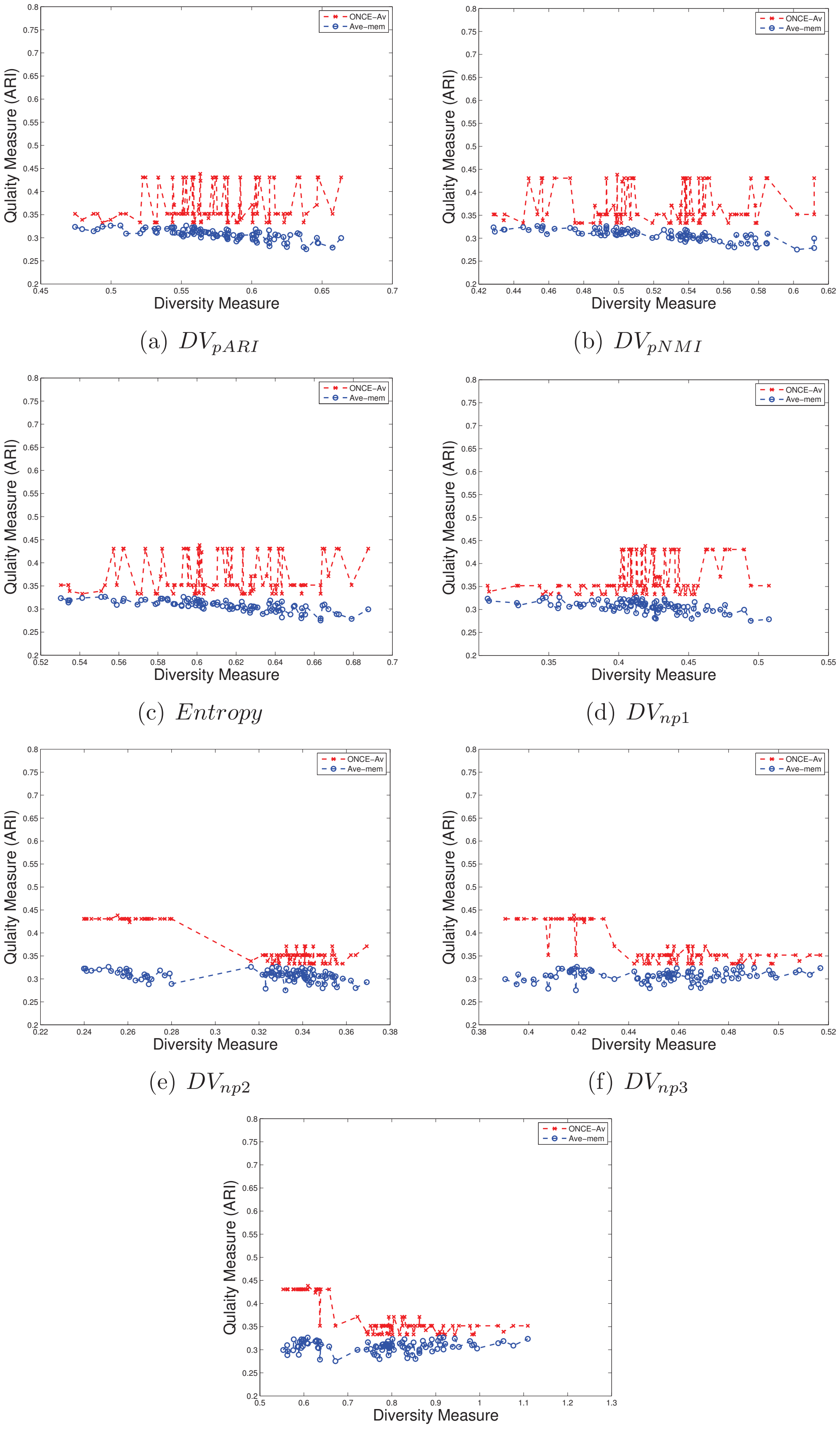
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.8: The seven diversity measures from Wine dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

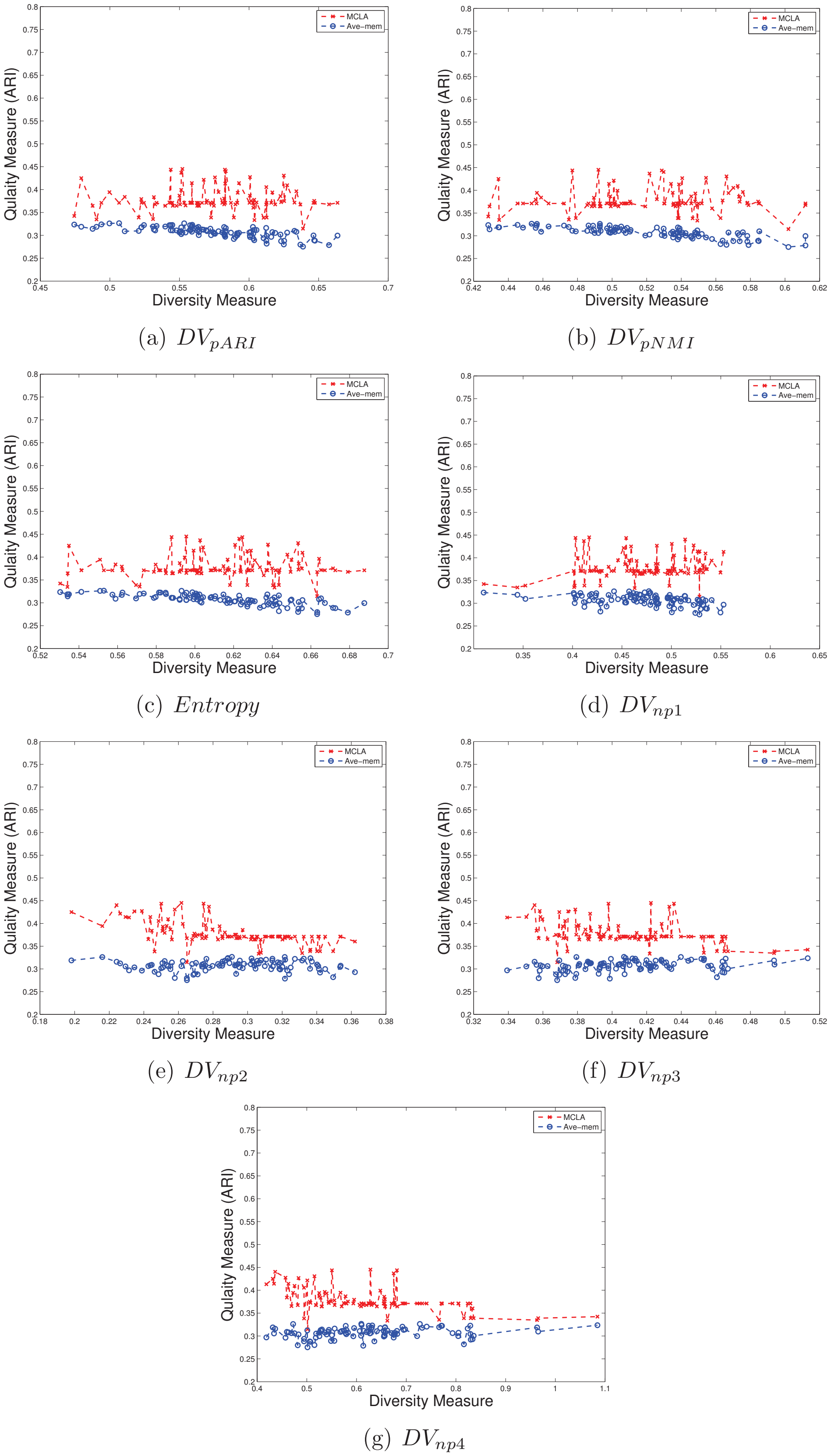
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.9: The seven diversity measures from Wine dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

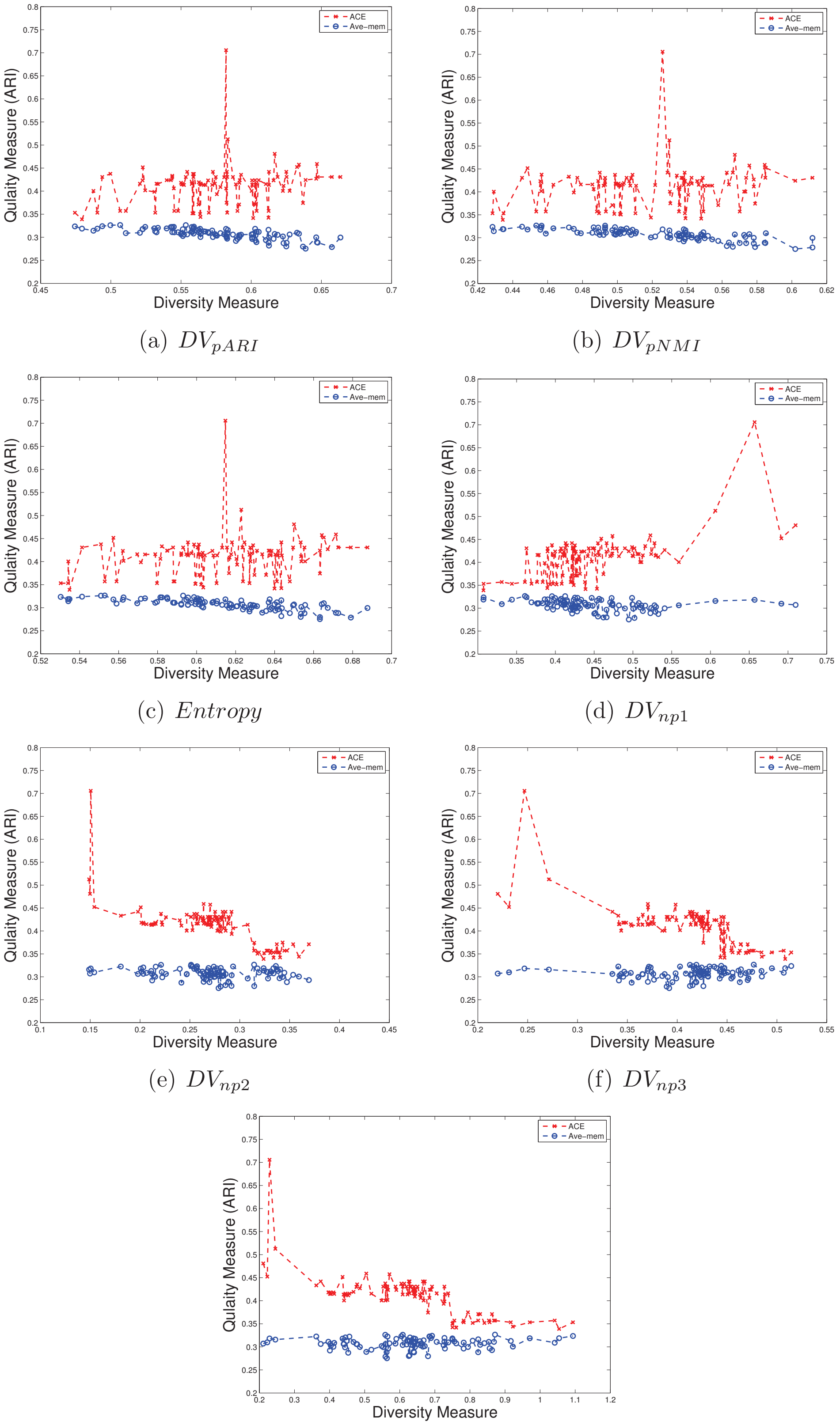
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.10: The seven diversity measures from Wine dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

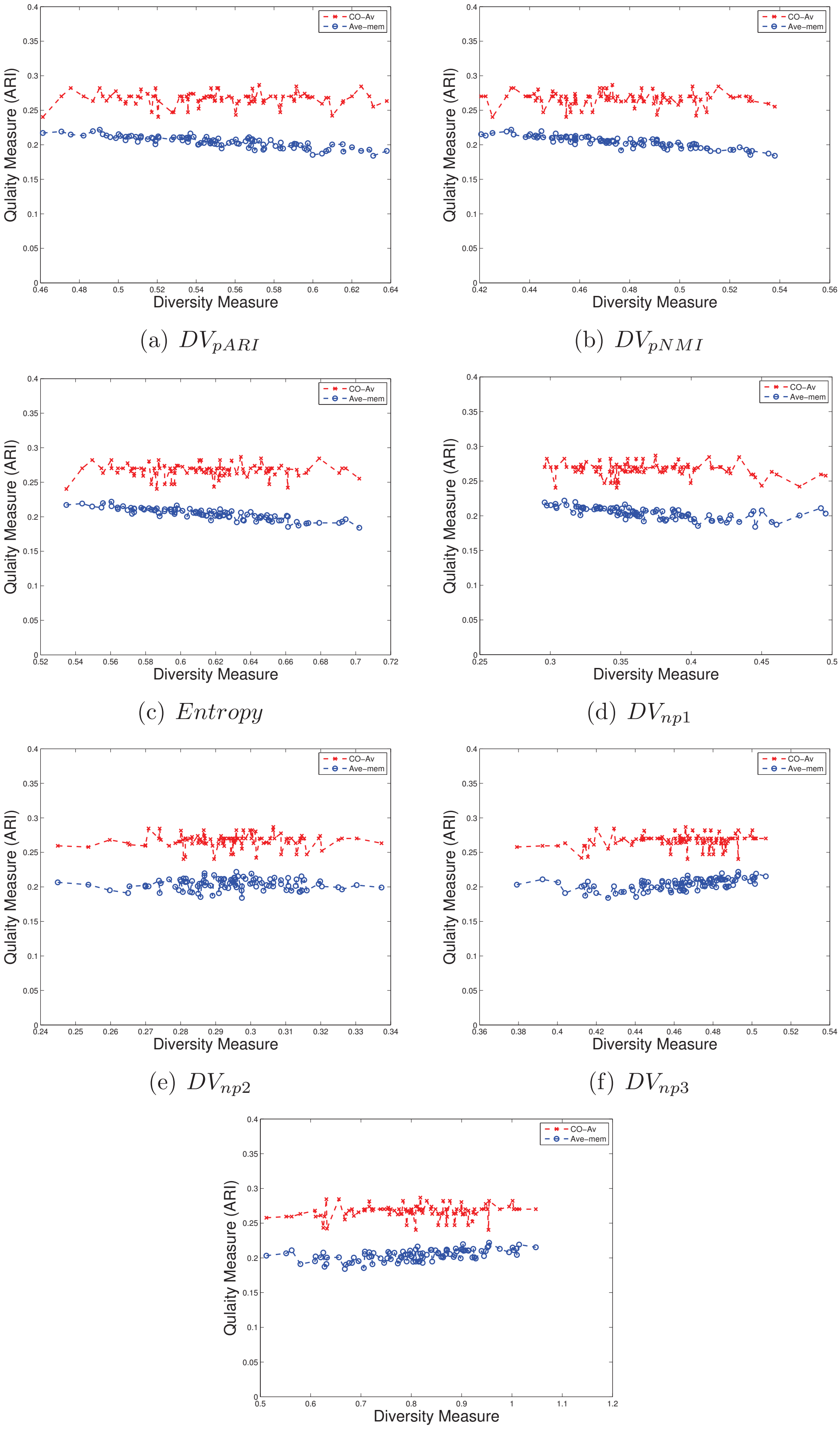
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.11: The seven diversity measures from Wine dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

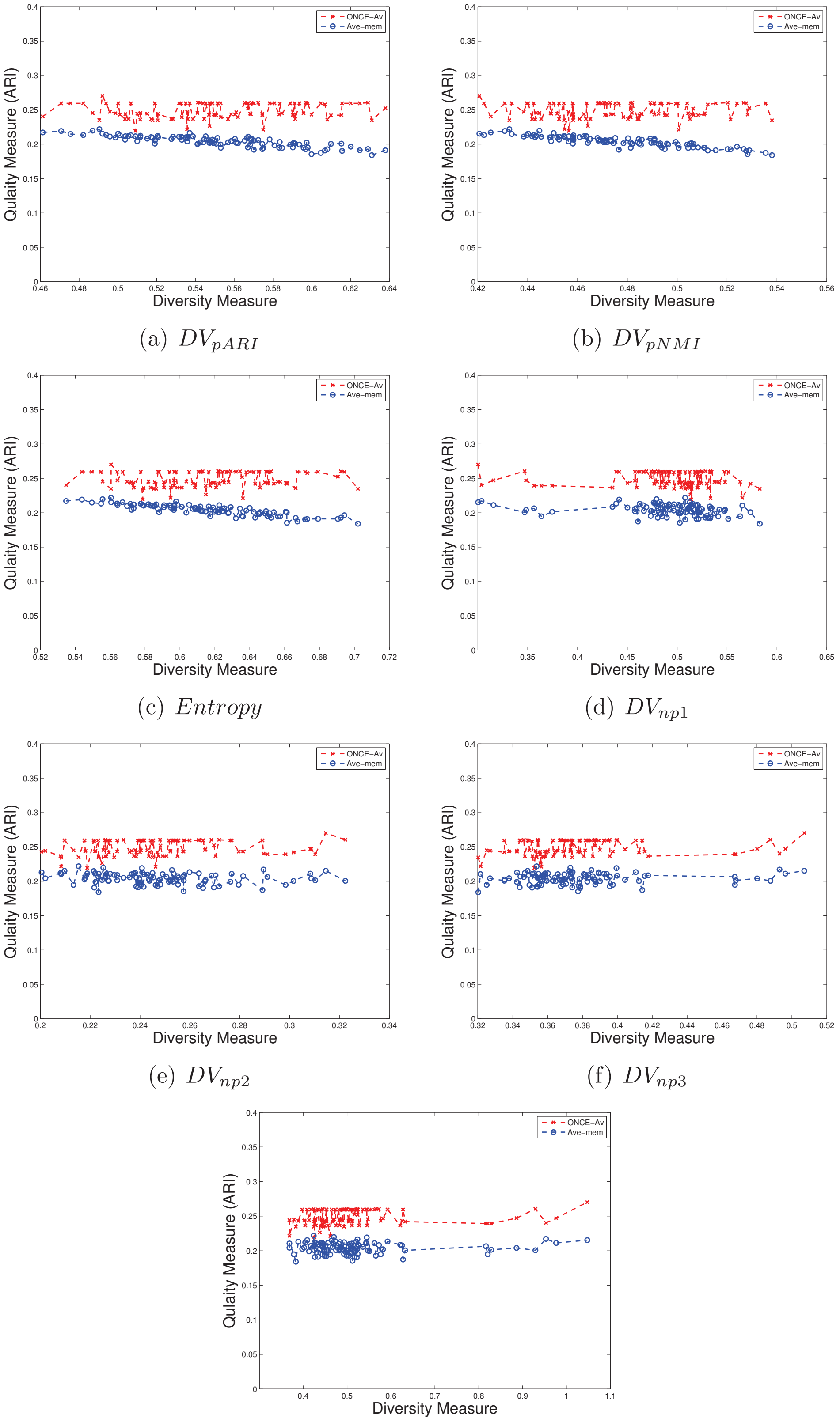
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.12: The seven diversity measures from Glass dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

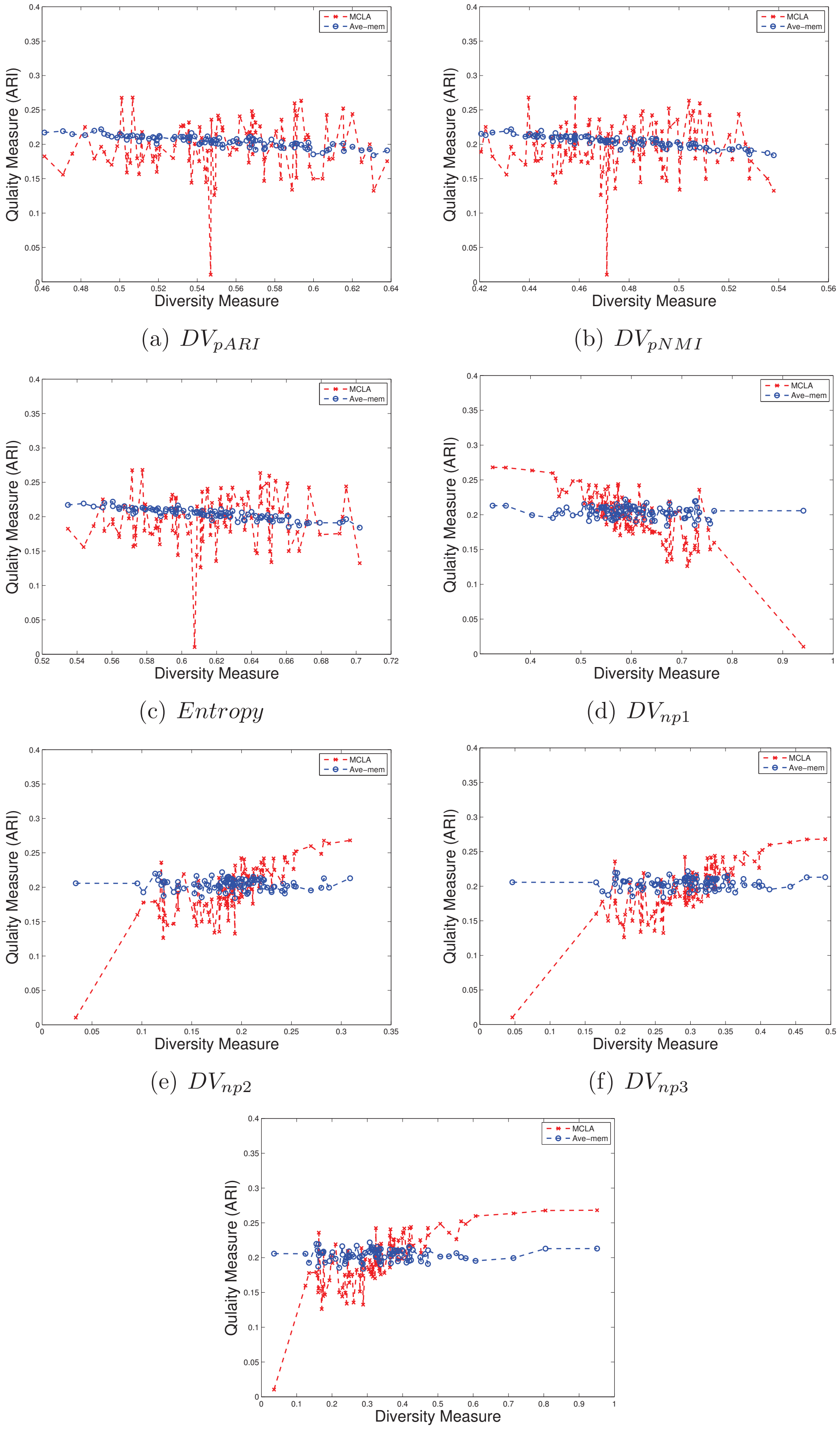
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.13: The seven diversity measures from Glass dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

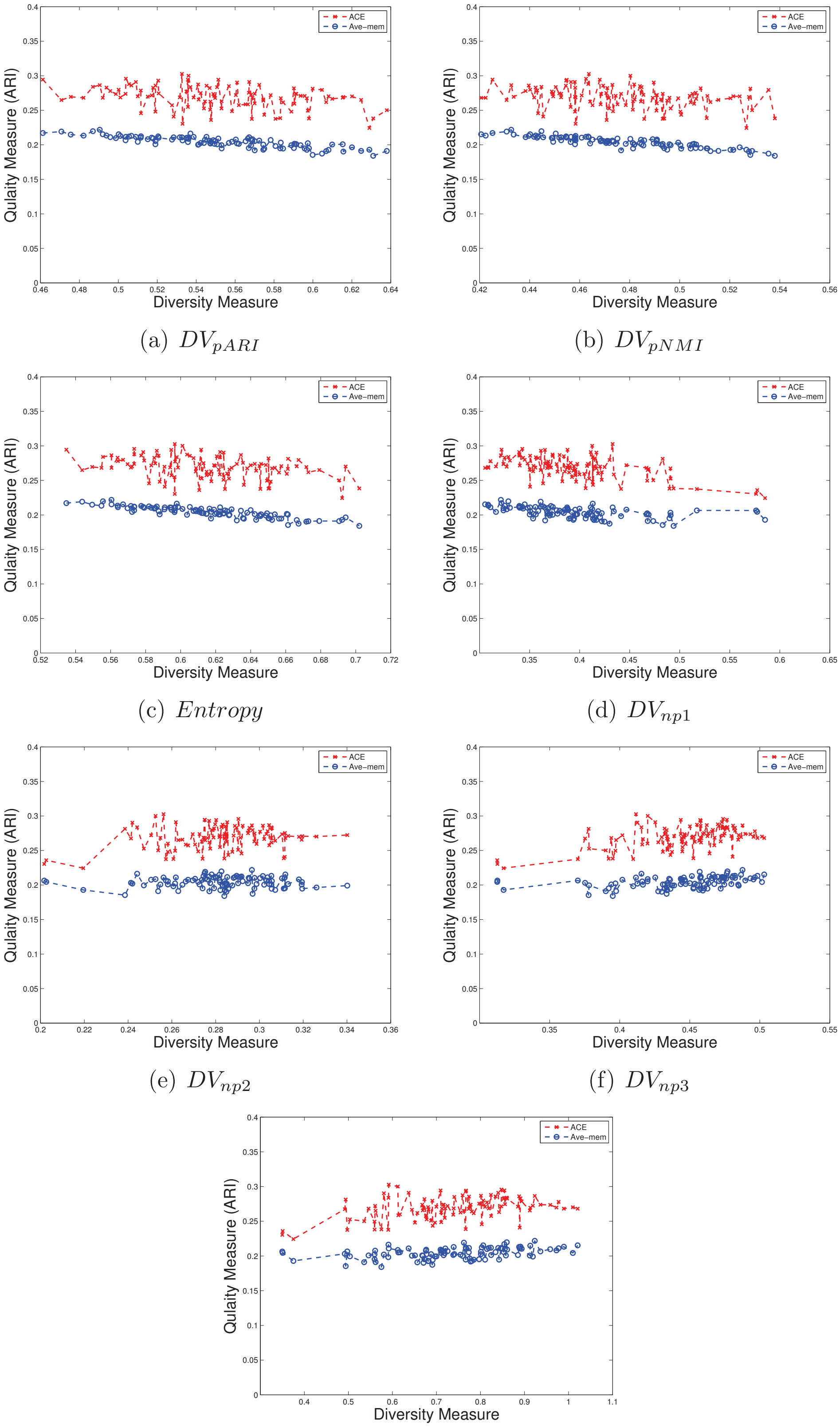
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.14: The seven diversity measures from Glass dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

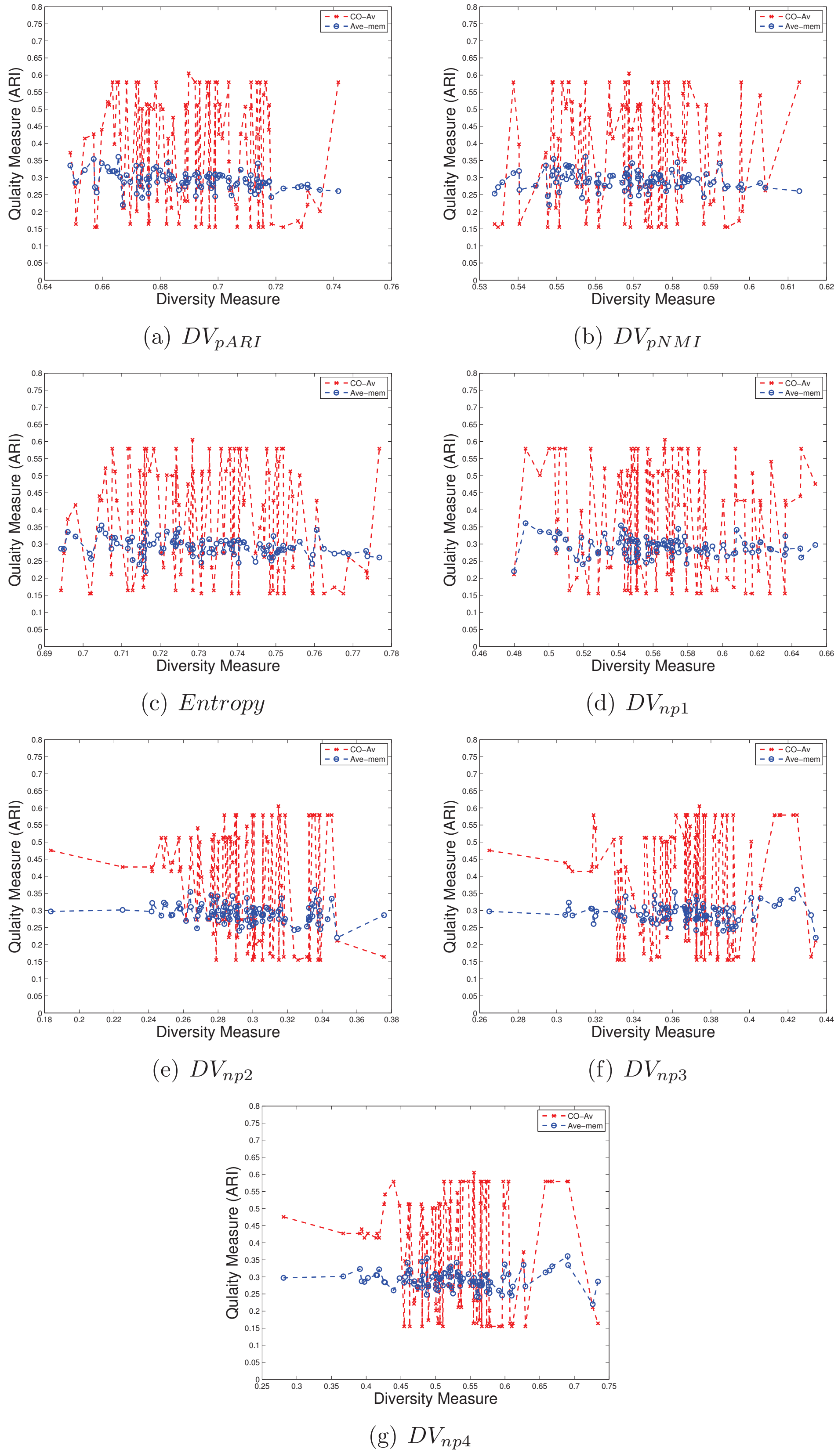
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.15: The seven diversity measures from Glass dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

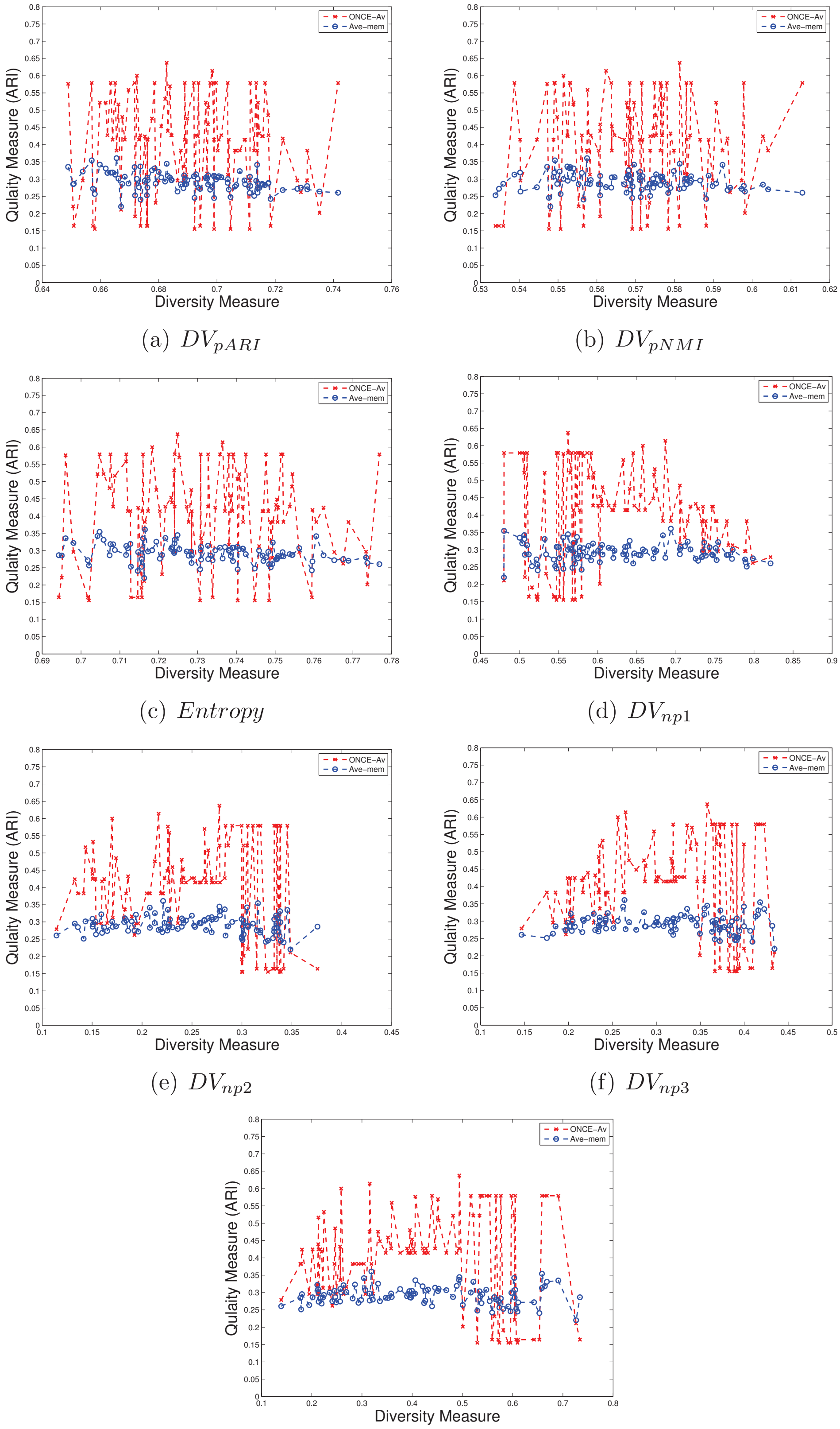
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.16: The seven diversity measures from Thyroid dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

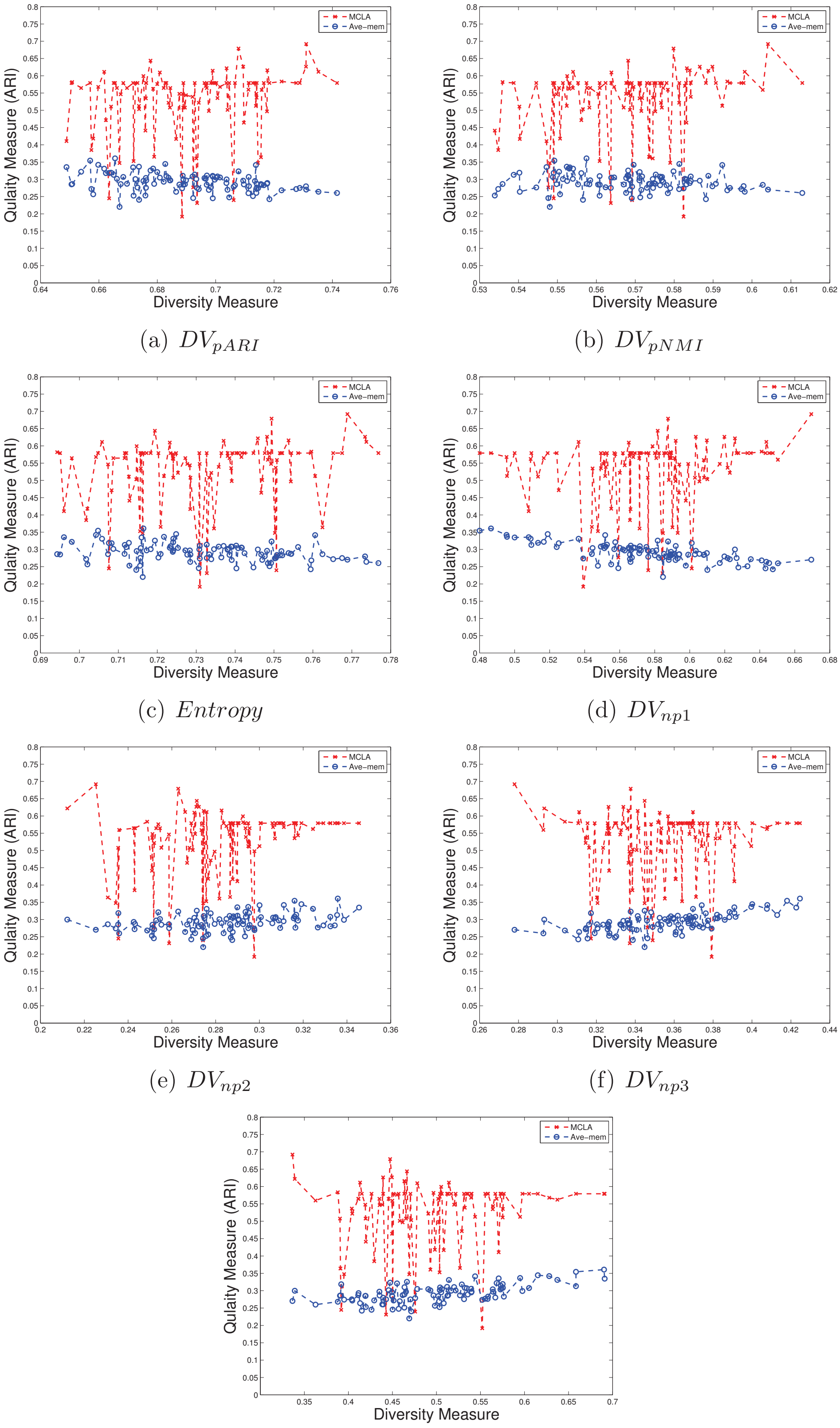
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.17: The seven diversity measures from Thyroid dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

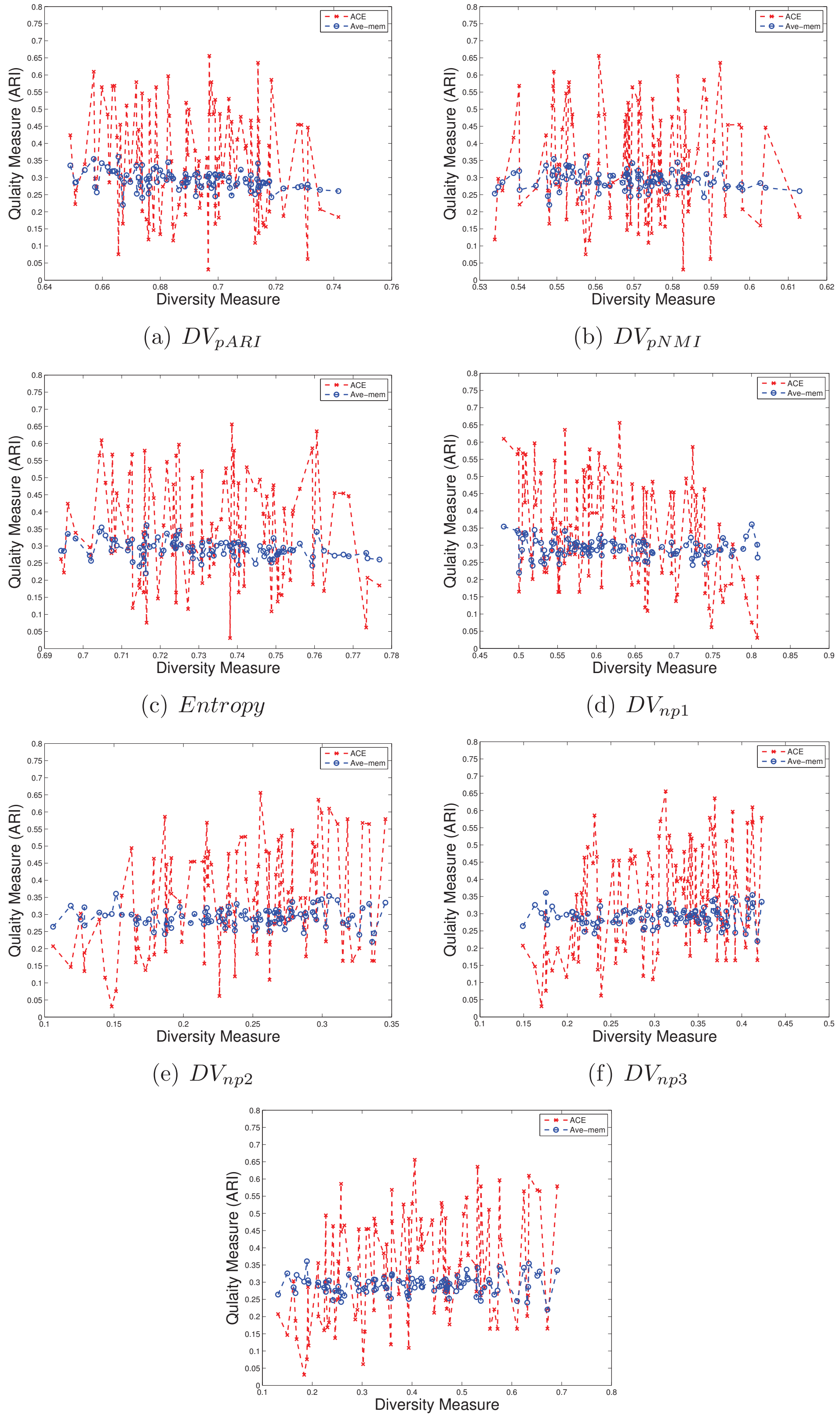
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.18: The seven diversity measures from Thyroid dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

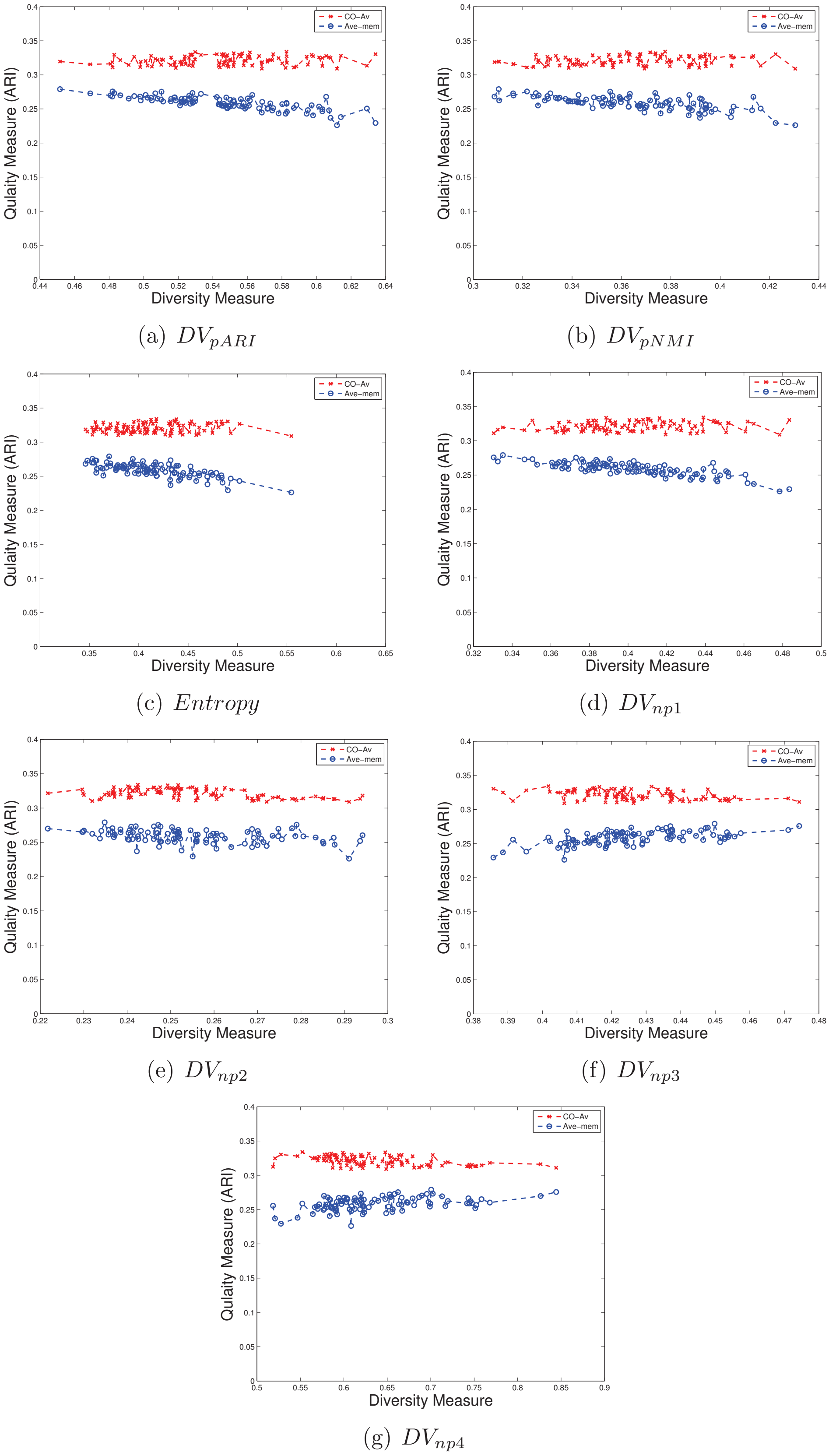
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.19: The seven diversity measures from Thyroid dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

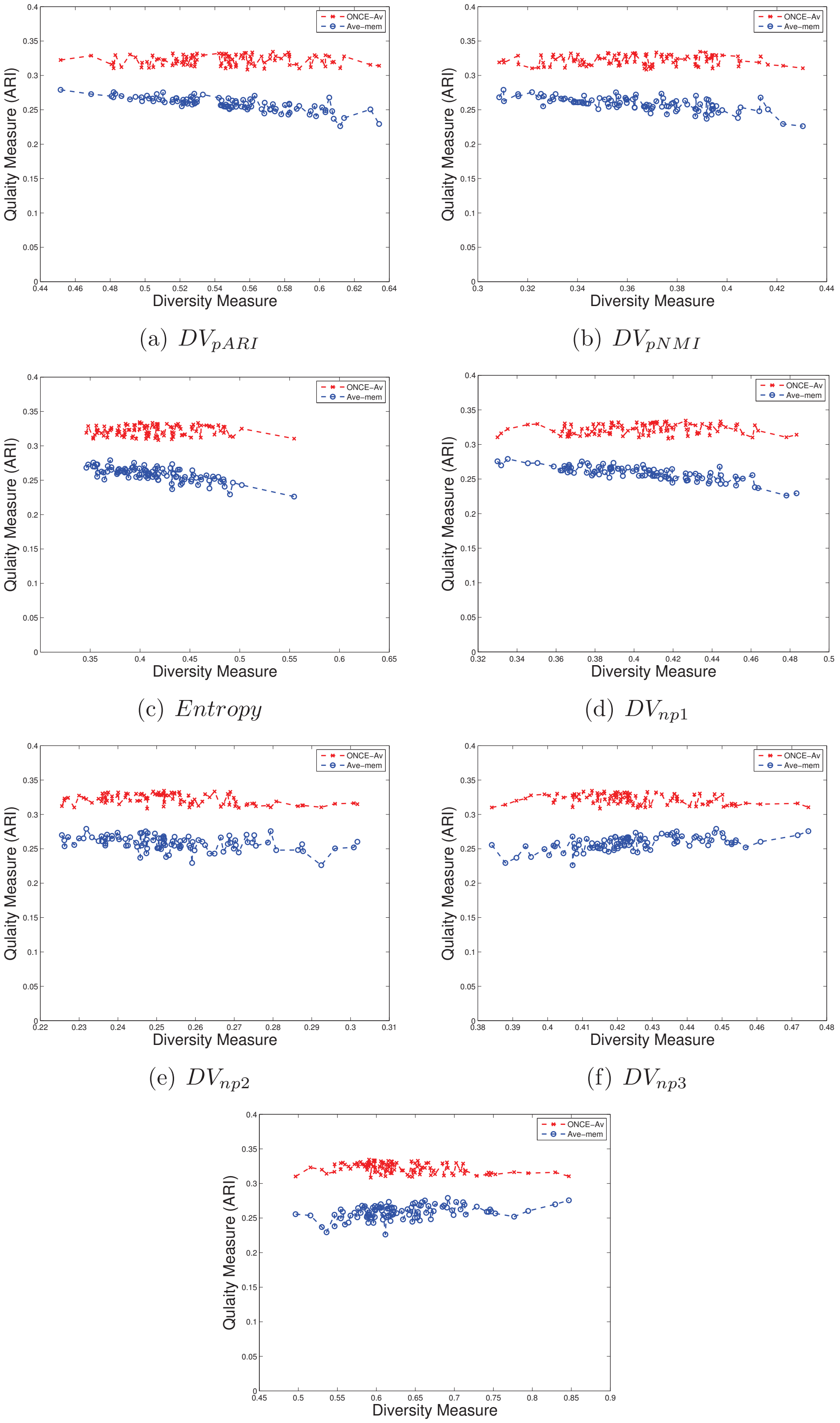
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.20: The seven diversity measures from Mfeatures dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

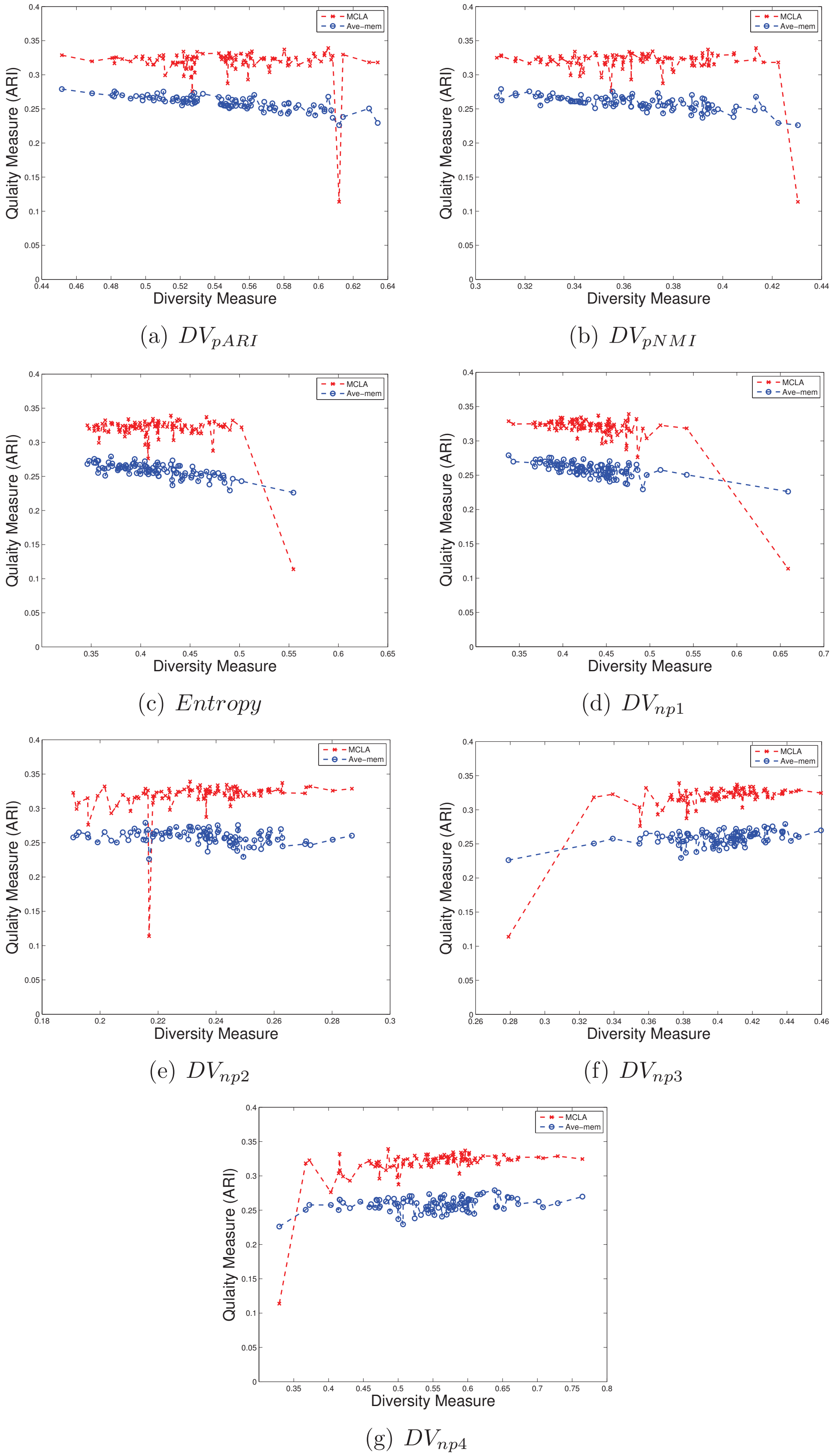
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.21: The seven diversity measures from Mfeatures dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

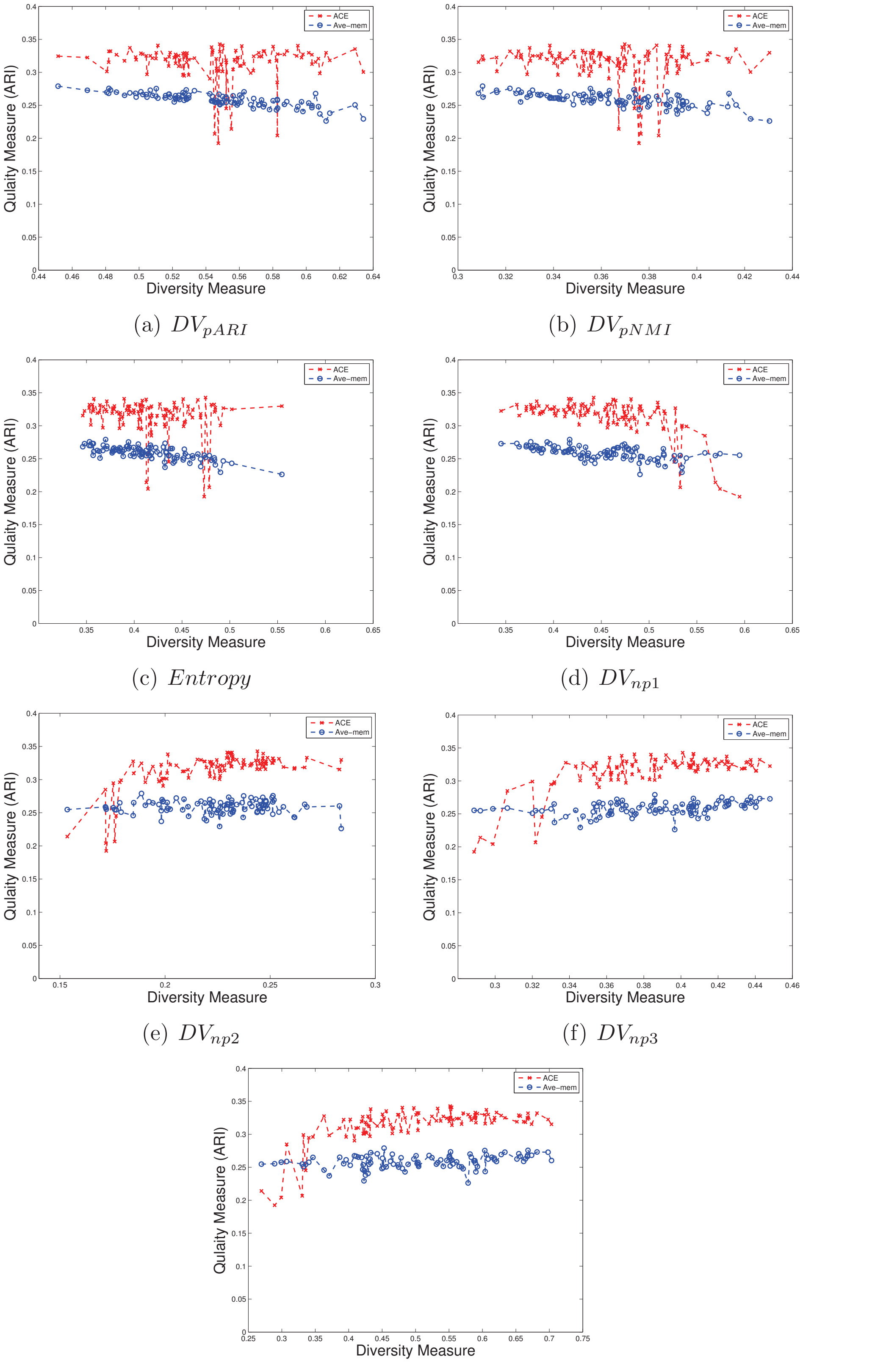
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.22: The seven diversity measures from Mfeatures dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

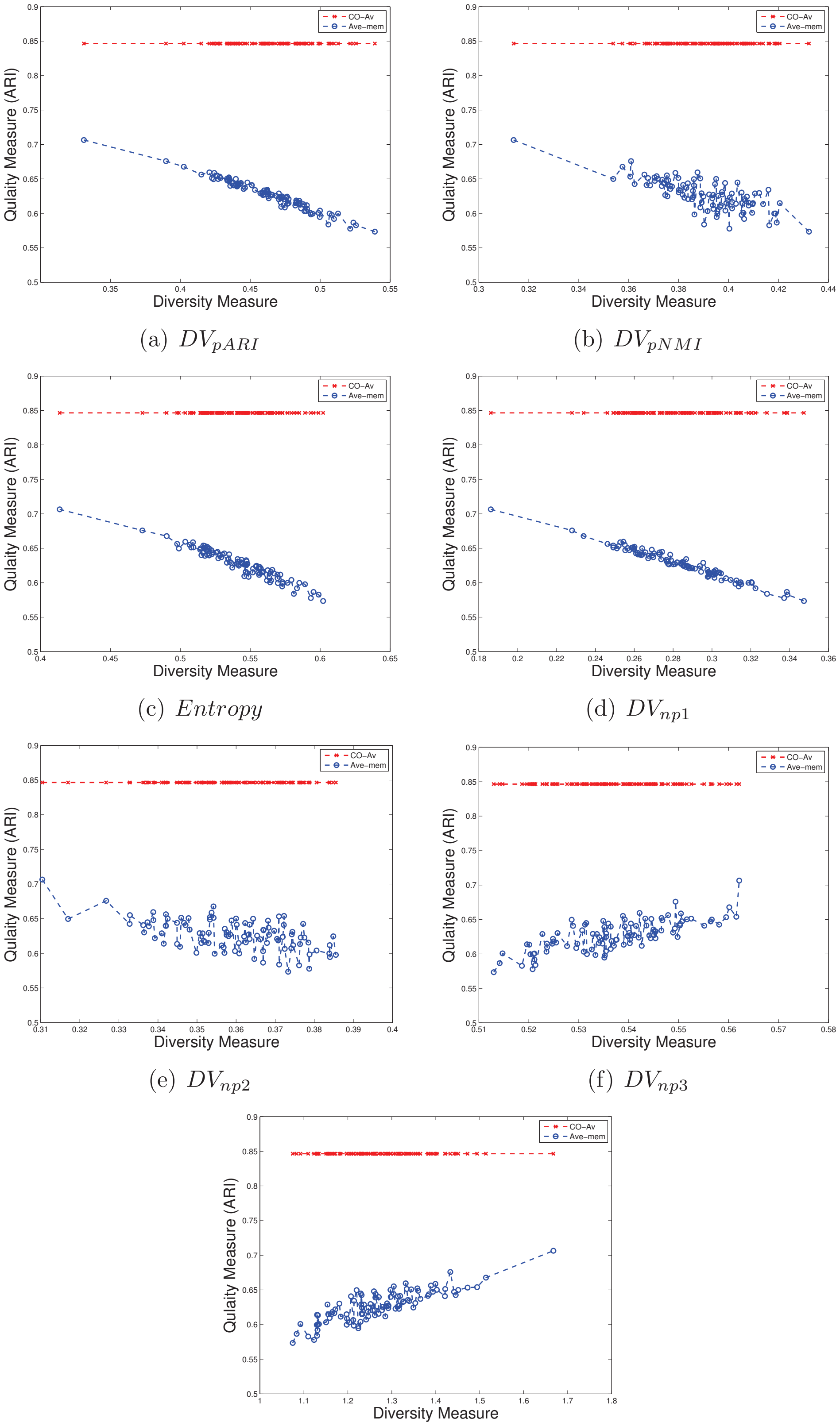
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.23: The seven diversity measures from Mfeatures dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

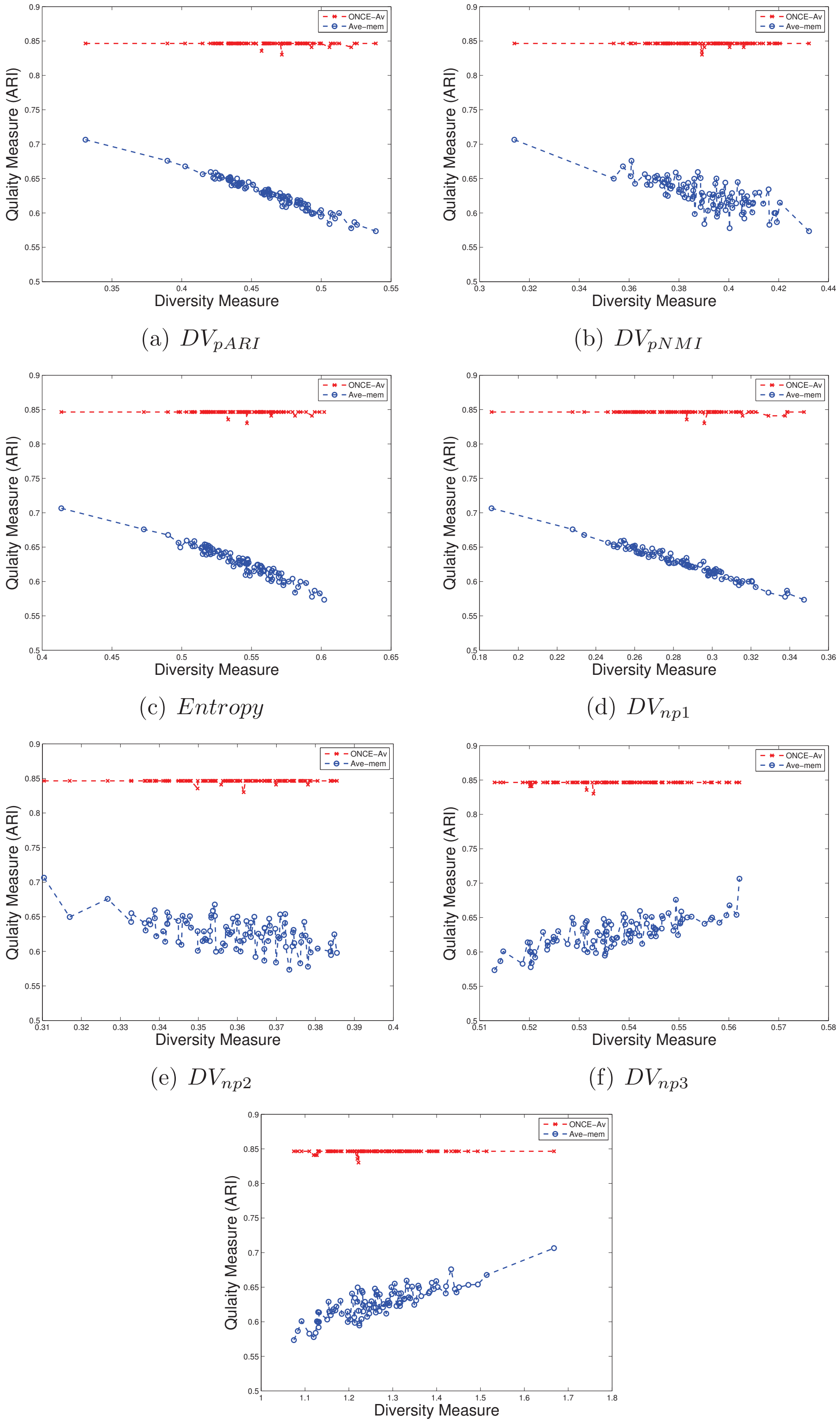
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.24: The seven diversity measures from Bcw dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

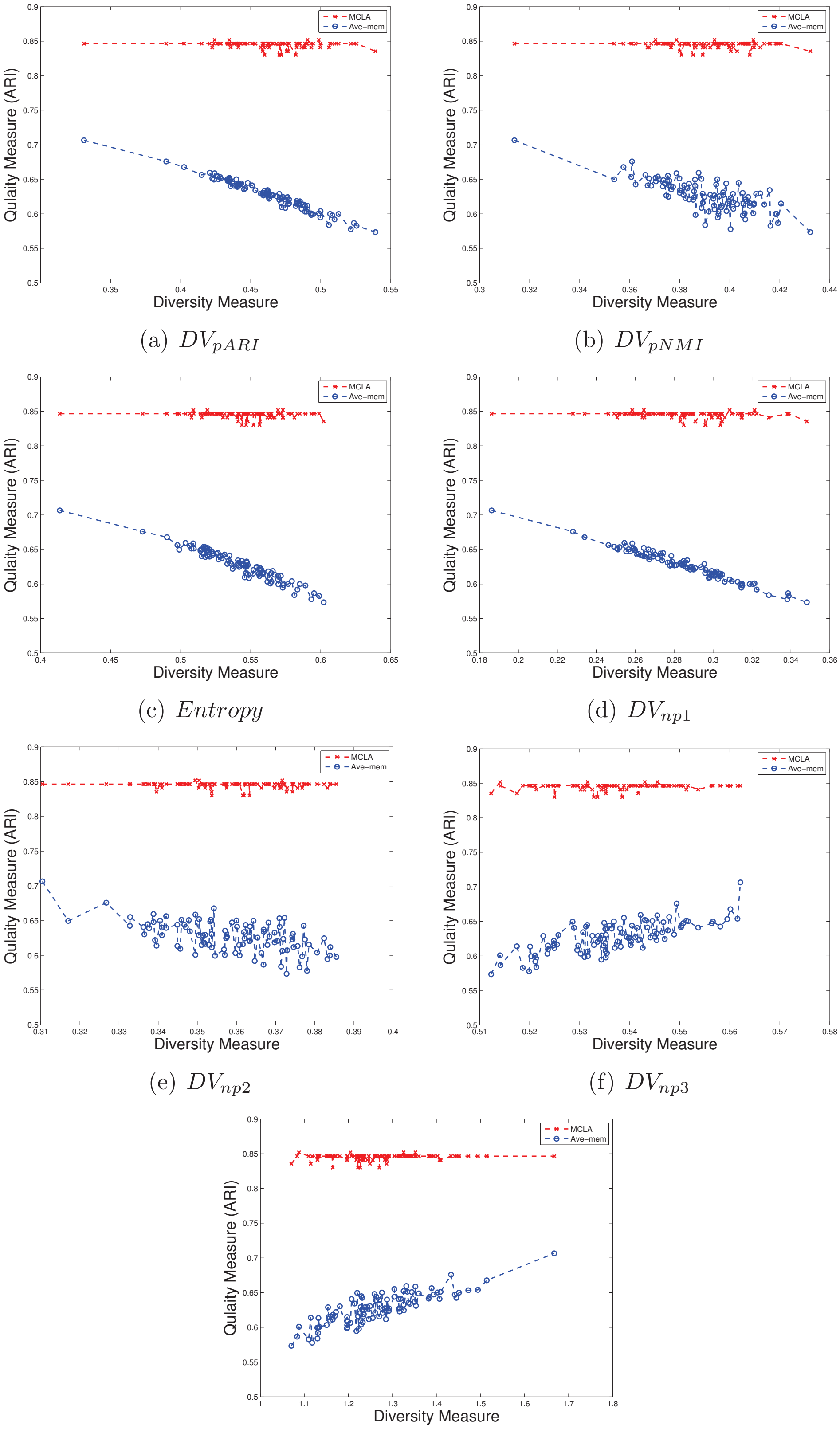
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.25: The seven diversity measures from Bcw dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

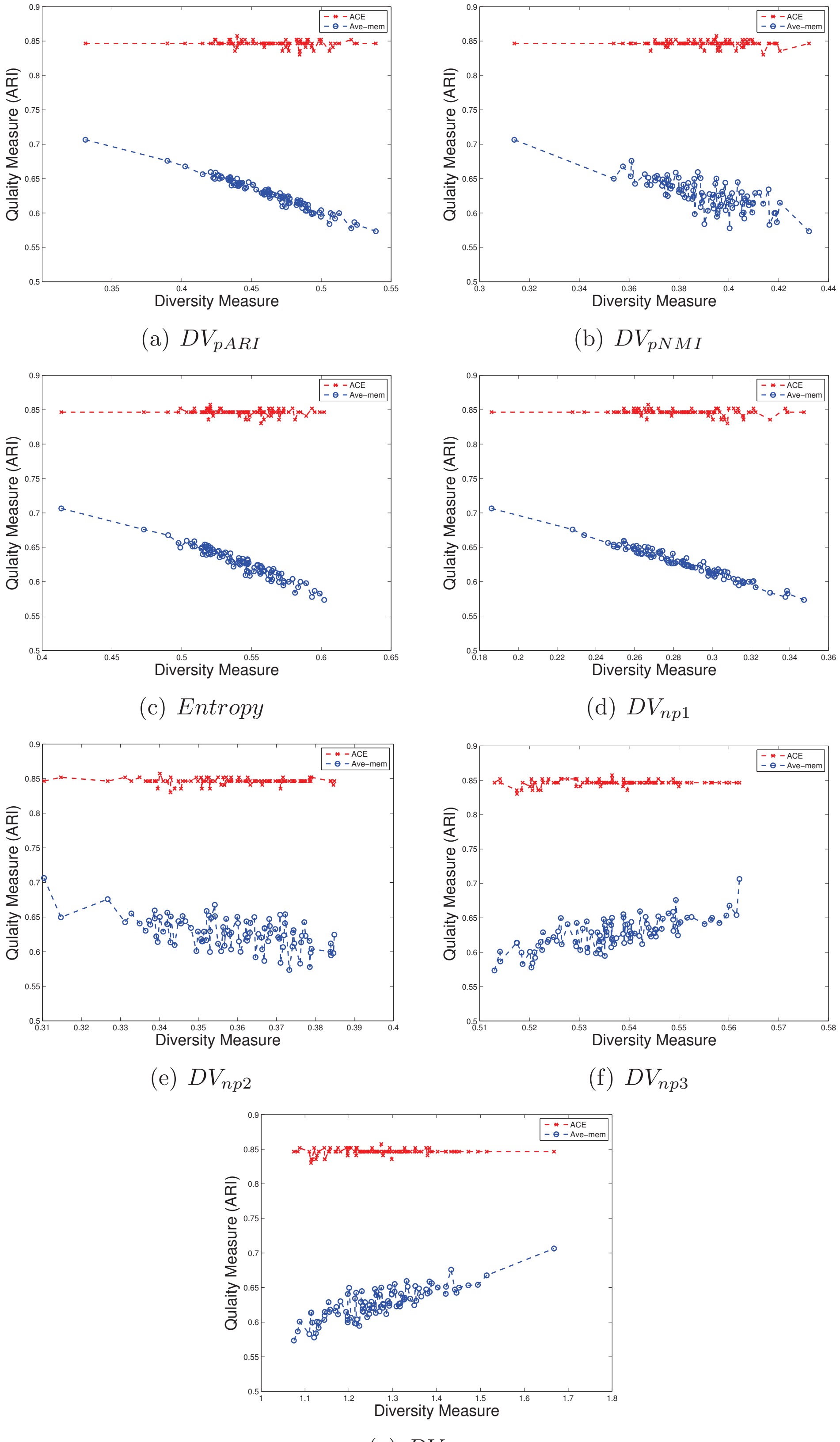
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.26: The seven diversity measures from Bcw dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

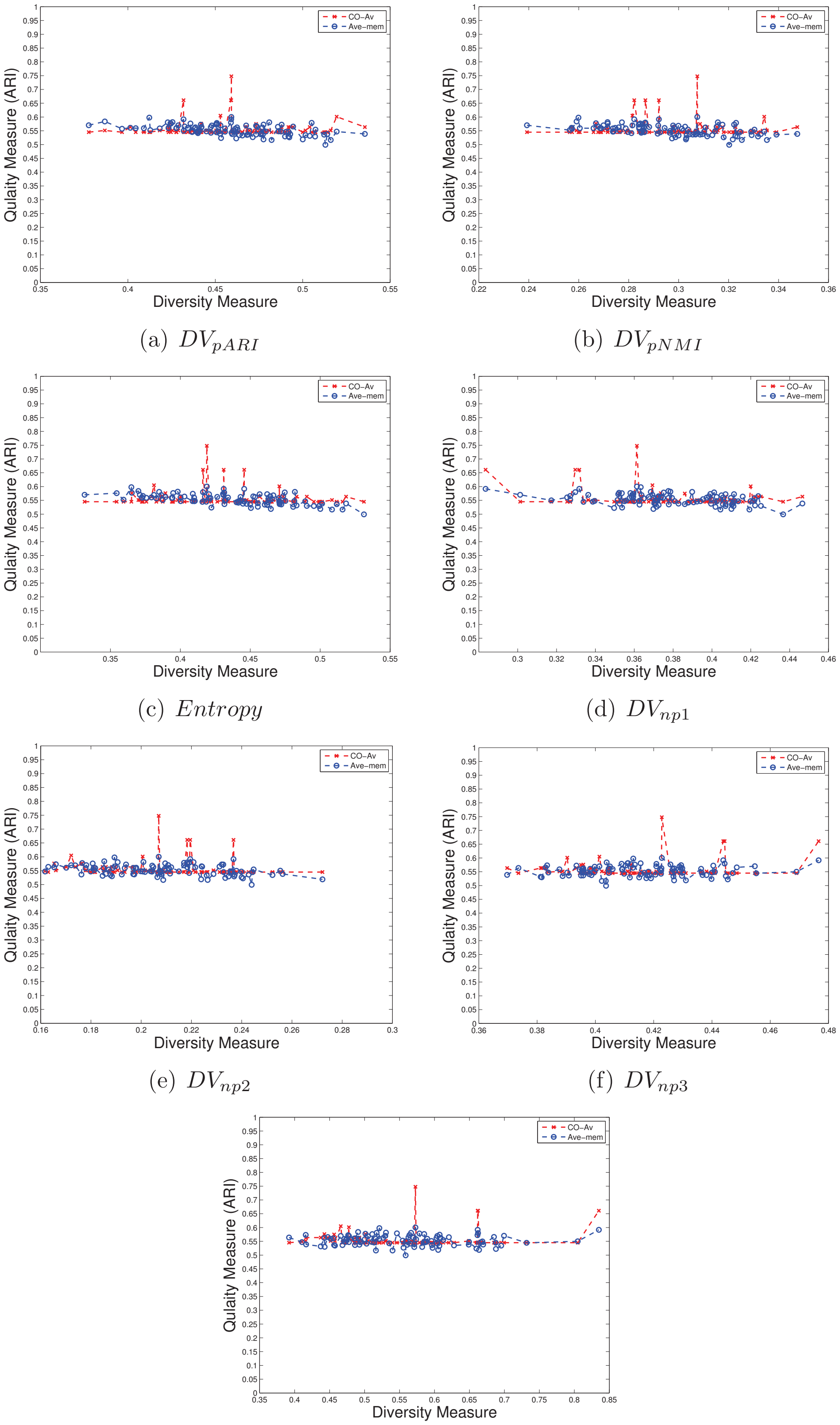
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.27: The seven diversity measures from Bcw dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

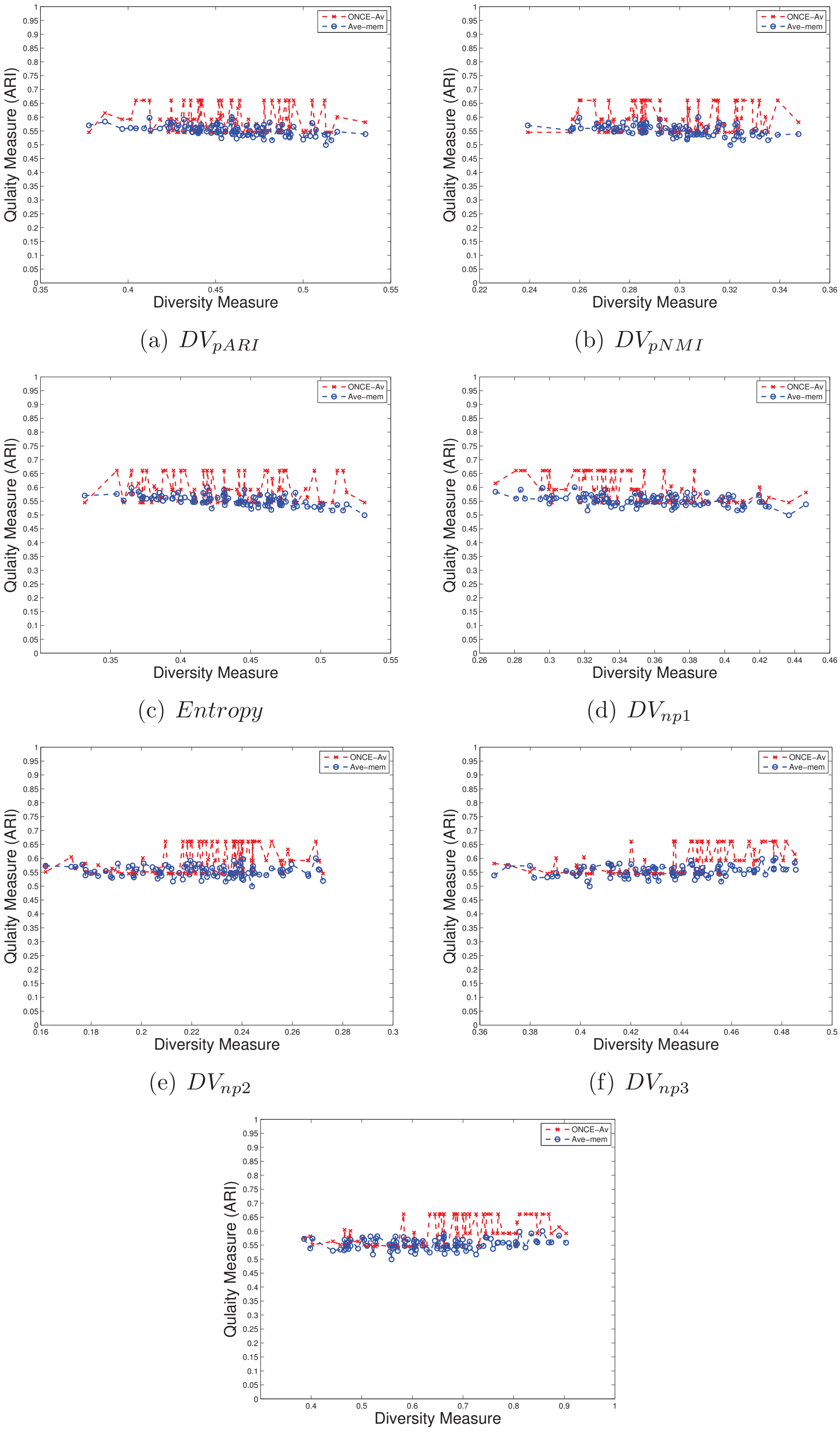
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.28: The seven diversity measures from Soybean dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

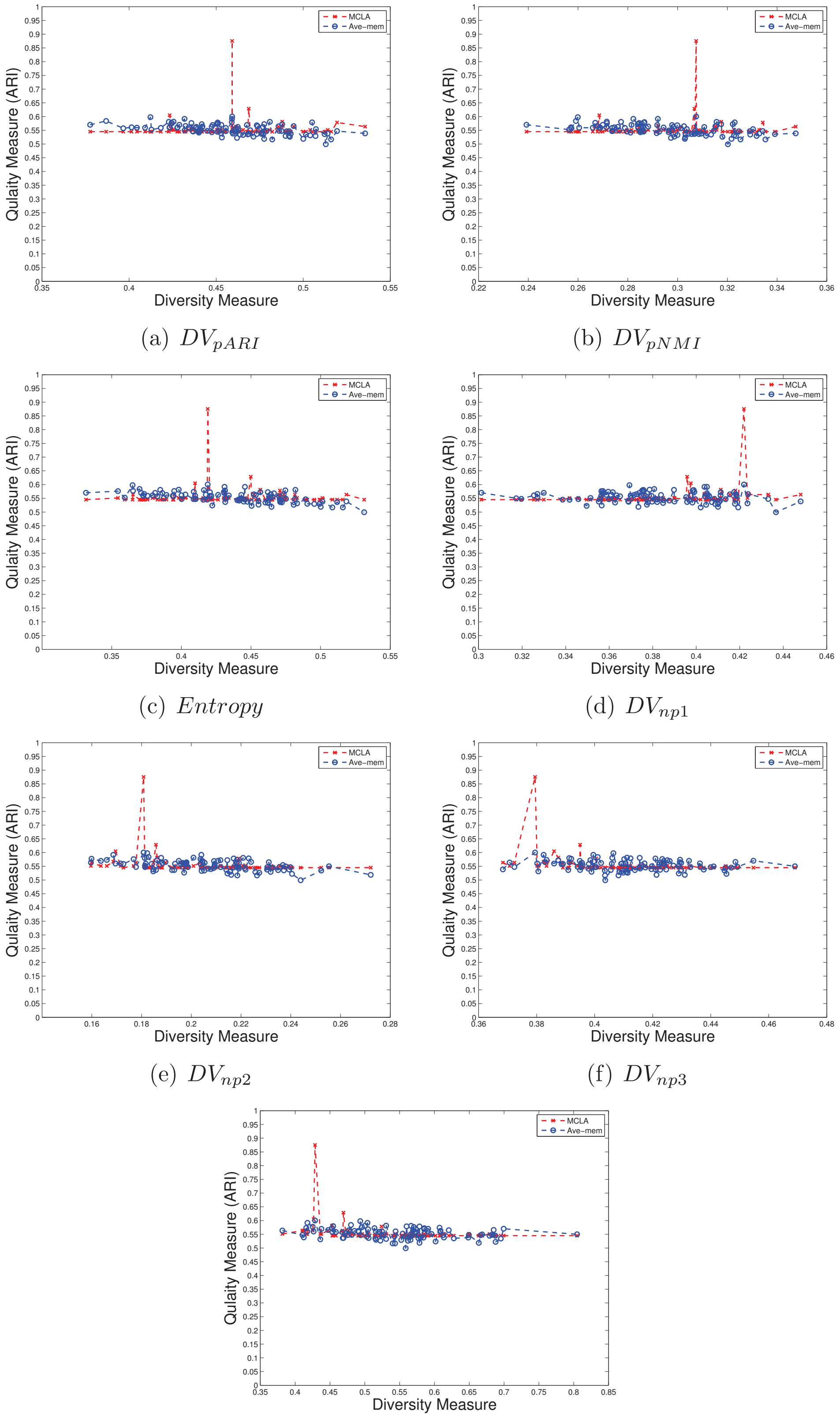
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.29: The seven diversity measures from Soybean dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

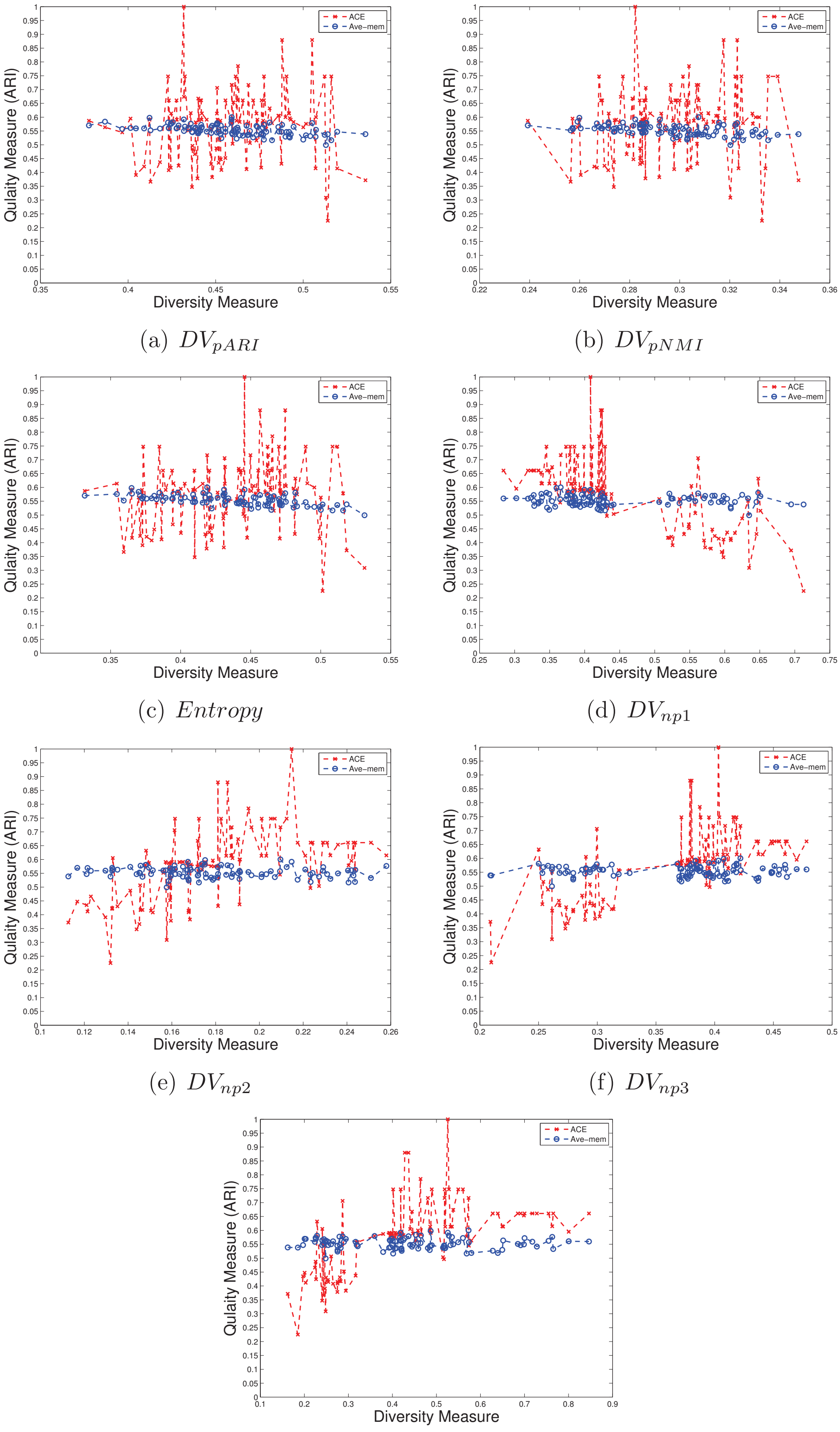
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.30: The seven diversity measures from Soybean dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

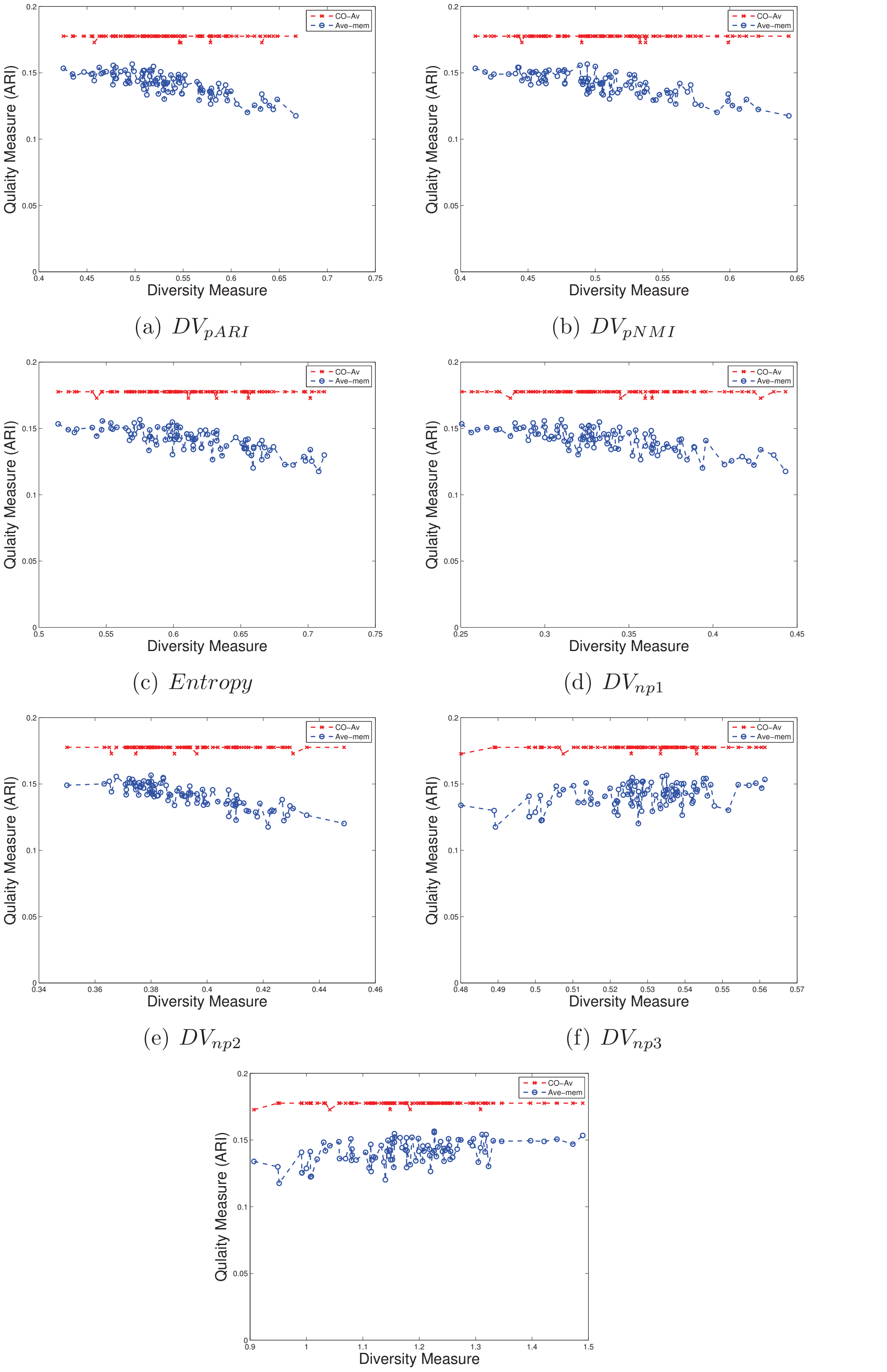
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.31: The seven diversity measures from Soybean dataset using ACE.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

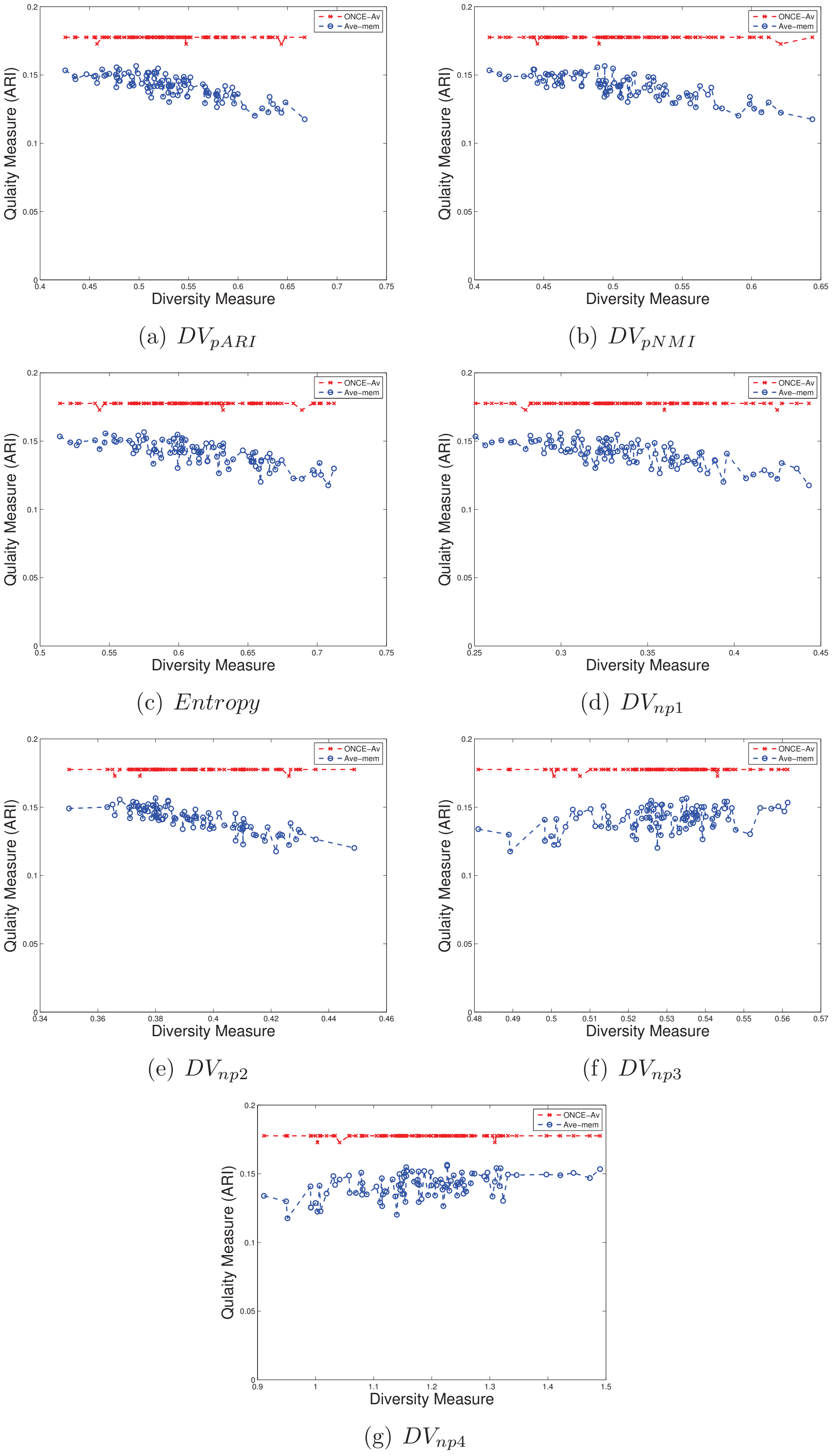
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.32: The seven diversity measures from Ionosphere dataset using CO-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

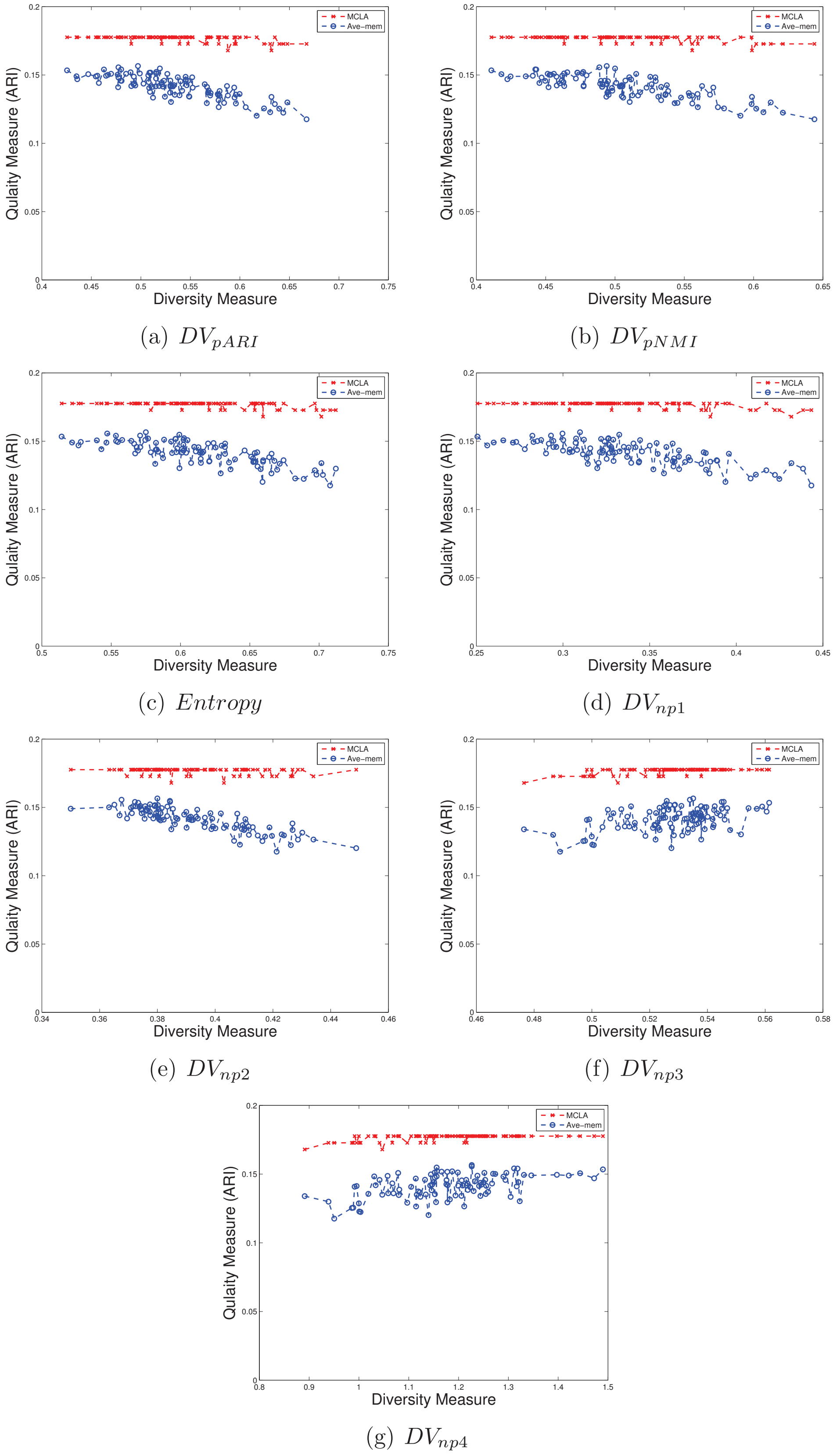
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.33: The seven diversity measures from Ionosphere dataset using ONCE-Av.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

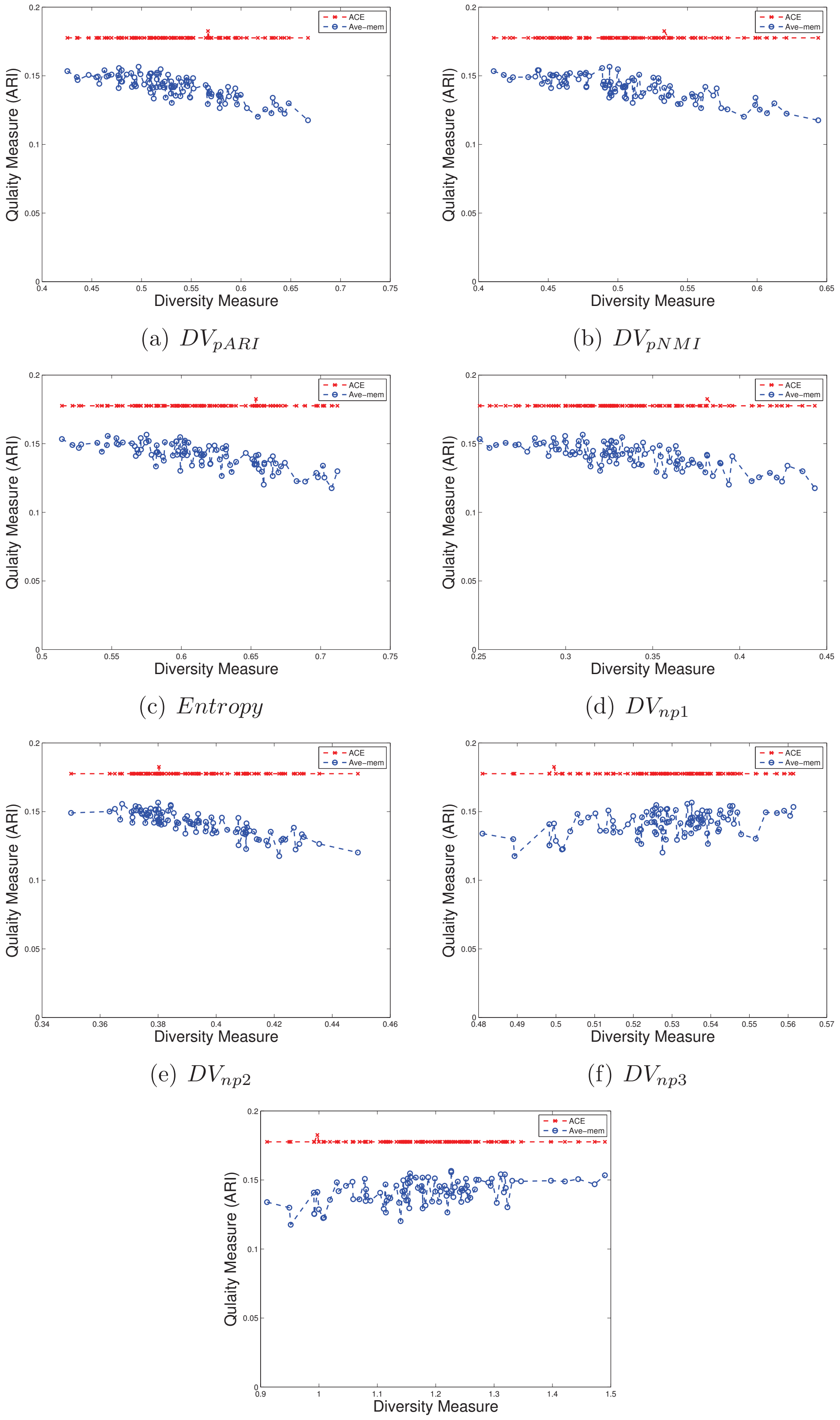
(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.34: The seven diversity measures from Ionosphere dataset using MCLA.



(a) DV_{pARI}

(b) DV_{pNMI}

(c) $Entropy$

(d) DV_{np1}

(e) DV_{np2}

(f) DV_{np3}

(g) DV_{np4}

Figure A.35: The seven diversity measures from Ionosphere dataset using ACE.

Table A.2: The p-value of the Correlation coefficient at the 95% confidence interval, where the correlation is presented in Table 6.1

Datasets	**CF**	DV_{pARI}	DV_{pNMI}	$Entropy$	DV_{np1}	DV_{np2}	DV_{np3}	DV_{np4}
Iris	**CO**	0.219	0.125	0.152	0.166	0.429	0.197	0.224
	ONCE	0.001	0.000	0.000	0.000	0.000	0.000	0.000
	ACE	0.134	0.127	0.158	0.012	0.001	0.002	0.001
	MCLA	0.528	0.543	0.569	0.603	0.704	0.646	0.637
Wine	**CO**	0.004	0.020	0.005	0.000	0.000	0.000	0.000
	ONCE	0.197	0.547	0.277	0.000	0.000	0.000	0.000
	ACE	0.035	0.056	0.050	0.000	0.000	0.000	0.000
	MCLA	0.927	0.848	0.673	0.072	0.000	0.000	0.000
Thyroid	**CO**	0.420	0.577	0.412	0.631	0.127	0.633	0.448
	ONCE	0.905	0.446	0.916	0.399	0.639	0.767	0.716
	ACE	0.360	0.747	0.519	0.000	0.001	0.000	0.001
	MCLA	0.059	0.006	0.069	0.168	0.013	0.786	0.238
Mfeatures	**CO**	0.127	0.065	0.170	0.027	0.000	0.000	0.000
	ONCE	0.731	0.621	0.598	0.275	0.000	0.003	0.000
	ACE	0.488	0.434	0.148	0.000	0.000	0.000	0.000
	MCLA	0.246	0.062	0.005	0.000	0.004	0.000	0.000
Glass	**CO**	0.998	0.649	0.993	0.060	0.568	0.085	0.134
	ONCE	0.176	0.597	0.175	0.201	0.047	0.110	0.172
	ACE	0.000	0.002	0.000	0.000	0.018	0.000	0.000
	MCLA	0.472	0.875	0.444	0.000	0.000	0.000	0.000
Bcw	**CO**	1.000	1.000	1.000	1.000	1.000	1.000	1.000
	ONCE	0.235	0.747	0.338	0.093	0.756	0.088	0.093
	ACE	0.165	0.421	0.227	0.096	0.896	0.075	0.099
	MCLA	0.097	0.308	0.112	0.060	0.264	0.162	0.112
Soybean	**CO**	0.905	0.891	0.476	0.019	0.364	0.233	0.375
	ONCE	0.292	0.942	0.326	0.000	0.001	0.000	0.000
	ACE	0.636	0.261	0.152	0.000	0.000	0.000	0.000
	MCLA	0.751	0.433	0.835	0.028	0.027	0.004	0.008
Ionosphere	**CO**	0.428	0.490	0.437	0.308	0.803	0.141	0.206
	ONCE	0.596	0.641	0.748	0.444	0.710	0.223	0.332
	ACE	0.539	0.583	0.377	0.289	0.504	0.074	0.109
	MCLA	0.000	0.000	0.000	0.000	0.045	0.000	0.000

Appendix B

In this appendix, we give a complete results of the experiments in Section 6.2.1.

B.1 The Complete Results of Analysis of the Positive and Negative Effects of Diversity

Each Figure from B.1 to B.15 is related to a particular pair of ensemble cases (blue is a negative and red is a positive ensemble case), and in each one of them we have three subfigures (a, b, and c). Figure (a) represents the diversity measures, the ensemble quality, the average member quality and the standard deviation of the member quality (all qualities measured by *ARI*) of a particular pair; Figure (b) represents the distributions of the related individual members' qualities to a particular pair; and Figure (c) shows the number of members whose quality is as poor as the negative ensemble quality in a particular pair (Poor Q-mem), the number whose quality is as good as the positive ensemble quality (Good Q-mem), and the number of members that have a medium quality (between the poor and the good quality levels, Medium Q-mem). In Figure (a), we plot only one of the consensus functions that has been identified as a negative case, along with its related diversity measures. Figures from B.16 to B.26 show the similar results that shown in figure (a) but with the other consensus functions. The following sections explain these two situations in more detail:

Results of the First Situation

In this situation, we looked for negative cases that occurred on at least two consensus functions. From case 1 to case 4, the negative cases were discovered in ACE and MCLA (Figures B.1 and B.5), while case 5 and case 6 were discovered in CO and ACE (Figure B.6). The results show that whichever diversity measures are used in cases 1 and 2, the two ensembles have the same level of diversity, and their average member quality is more or less the same, as is the level of standard deviation, however case 2 is positive and case 1 is a negative case. Looking at the quality of the individual members we found that members in case 2 clearly had higher quality than case 1, and by classifying these members in terms of their quality, it is noticeable that case 2 had less poor members and more members of a medium quality than case 1. Due to the implemented techniques in ACE and MCLA, we noticed that the members in case 1 had a pattern of failure, while the same pattern was successful when the CO and ONCE were used, as both of them had a very good performance (0.579) in case 1 as shown in Table 6.2.

Figures B.2 and B.3 show the heat maps of the similarity matrices of CO and ONCE respectively. When compared with the heat map of the Thyroid true label in Figure B.4, we find that more object pairs are similar in case 1 than in case 2, particularly in the cluster placed in the middle of the similarity matrix. Applying the average linkage over these similarity matrices results in cluster labels with better quality in case 1 than in case 2. ACE and MCLA apply the pairwise similarity between clusters and not objects; this is why they do not perform well in case 1.

The other two pairs in this situation are different from pair one in terms of the quality of the other consensus functions used, which are ONCE and ACE in case 3 and ONCE and MCLA in case 5. They also did not perform well compared to their second pair (positive case). When considering the quality of each individual member in case 3 and case 5, we found that their second pair (case 4 and case 6) had more good quality members. Furthermore, the number of poor-quality members in case 4 was four less than in case 3, while case 6 had three more poor members than in case

5, but on other hand it had double the number of good members — one of them had a quality higher than 0.6. It was observed that this very good quality member ($Q(P_q) > 0.6$) was also one of the members in case 3 (negative case), but as the quality of the other members was not good enough to support it, the overall quality of the ensemble was poor. Therefore, it is clear that how the members resemble each other in terms of quality has an influence on the ensemble quality.

Results of the Second Situation

In this situation, the negative cases occurred on at least three consensus functions, and we have 8 pairs (from pair number 4 to 11 as shown in Table 6.2). In all of them the negative cases occurred on CO, ONCE and ACE, except pair number 9, where the negative case (Cases 17) occurred on CO, ACE and MCLA. The results of the individual members' qualities in case 8, case 10, case 12 and case 14 (positive cases), which are shown in Figures B.7, B.9, B.10 and B.11 respectively, show that the members have higher quality than the members in the negative cases, and the total number of good-quality members in the positive cases is larger than in the negative cases. Obviously, good-quality members with a high level of diversity (measured by most measurements except DV_{np2}, DV_{np3}, where they had a medium level) had contributed to improving the quality of the ensemble in the positive cases, while the poor-quality members with the same level of diversity had a negative effect on the ensemble performance for these particular consensus functions (CO, ONCE, ACE). Therefore, the reason behind these negative cases is the number of poor-quality members with high diversity among them. This indicates that each member made different errors in terms of cluster structure in the dataset, leading to lower/zero similarity between the correctly classified objects and to poor performance ensemble results for these consensus functions.

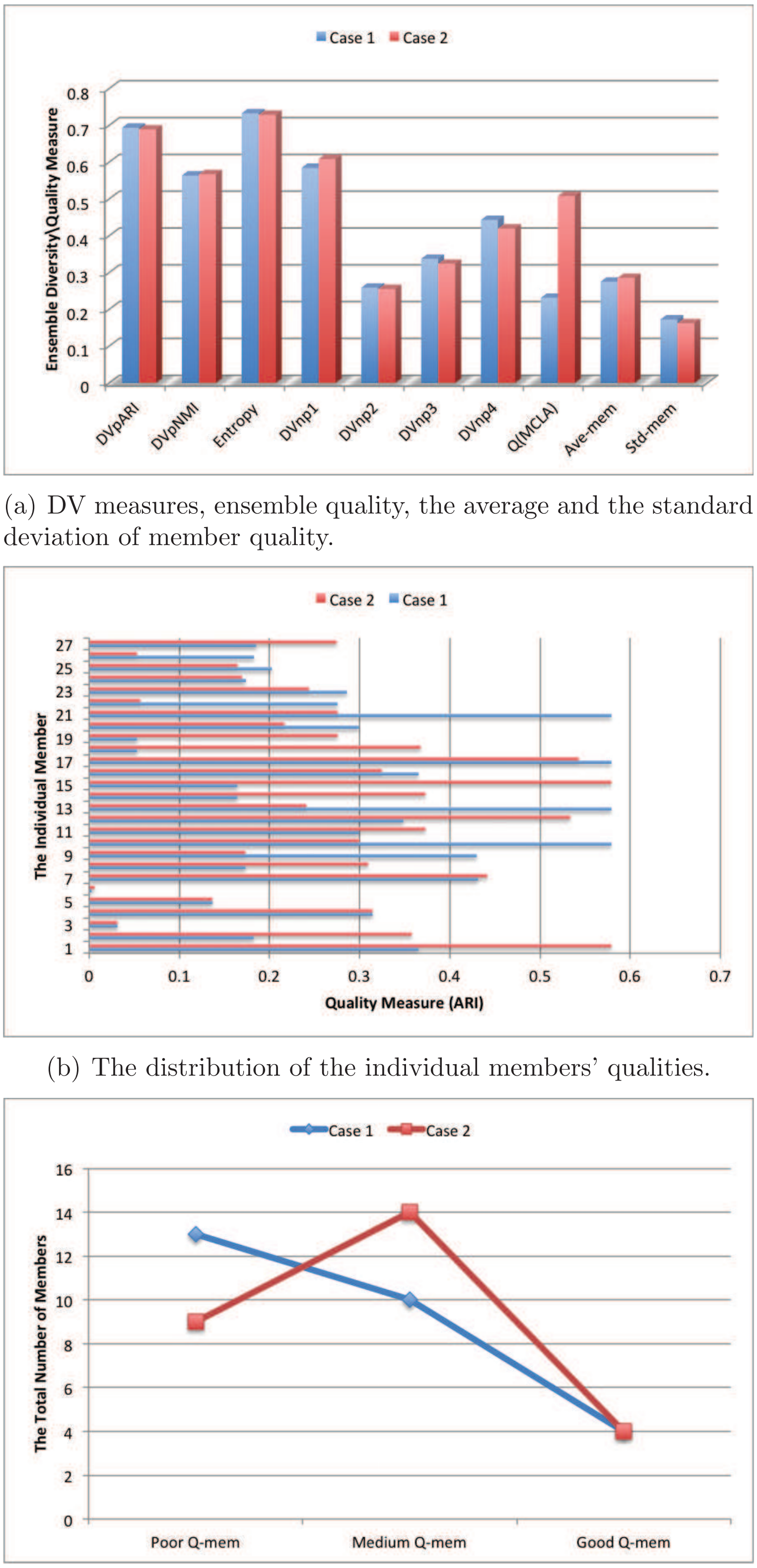
Figure B.7(b) shows that cases 8 and 7 had more or less the same number of members in each category, with only one less poor member, which moved to the medium-quality category in case 8, with the same level of diversity in both cases. The ensemble performance (CO, ONCE and ACE) in case 8 is much better than in case 7. Interestingly, MCLA had not been affected by the change in the category of this member or the improvement of the members' quality in case 8. Investigating the similarity matrix of CO in case 7 and 8 as shown in Figures B.8(a) and B.8(b) respectively, it is shown that more pairs of objects in case 8 are more similar to each other than in case 7.

Pair 8 consists of very interesting cases as shown in Figure B.12, in which half of their members were classified as poor-quality members (their quality was less than or equal to 0.2) with approximately the same average member quality in both of them, and a slightly increasing diversity in case 16. It was observed that case 16 had the highest number of poor members among all the positives cases that we had. This indicates that having a high number of poor-quality members in the members is not always a sign of poor ensemble performance — if the right diversity among them is achieved combining them can produce a high performance ensemble.

Cases 18, 20 and 22 (positive cases) as shown in figures B.13, B.14 and B.15 respectively also had a higher number of quality members than their second pairs (cases 17, 19, and 21 respectively). Case 18 had also fewer poor-quality members than case 17, and both of them had three good-quality members, one of which had quality higher than 0.6 (case 18). In case 19, the number of poor-quality members is lower than in case 20, which is also the lowest among all the negative cases in this analysis, and as the number of good-quality members in this case is also low, there was no room for the ensemble to improve upon its members, whereas there were six good-quality members in case 20, so the ensembles were improved in terms of quality for all the consensus functions used in this case. In case 22, there were five good-quality members in the members, while in case 21 there were only three good-quality members. Among these good members in both cases, one member had quality higher than 0.6, because in case 22 there were more members to support this high-quality member than case 21. The ensembles were improved in all the used consensus functions in case 22.

In summary, in most cases, the poor-quality members with a high level of DV_{pARI}

had affected the ensemble quality, and thus in the next section we will design an experiment to see how the different consensus functions perform as the poor-quality members are gradually removed. We will also see if the gradual removal of this deterioration in the diversity in the members leads to a successful ensemble performance or not.

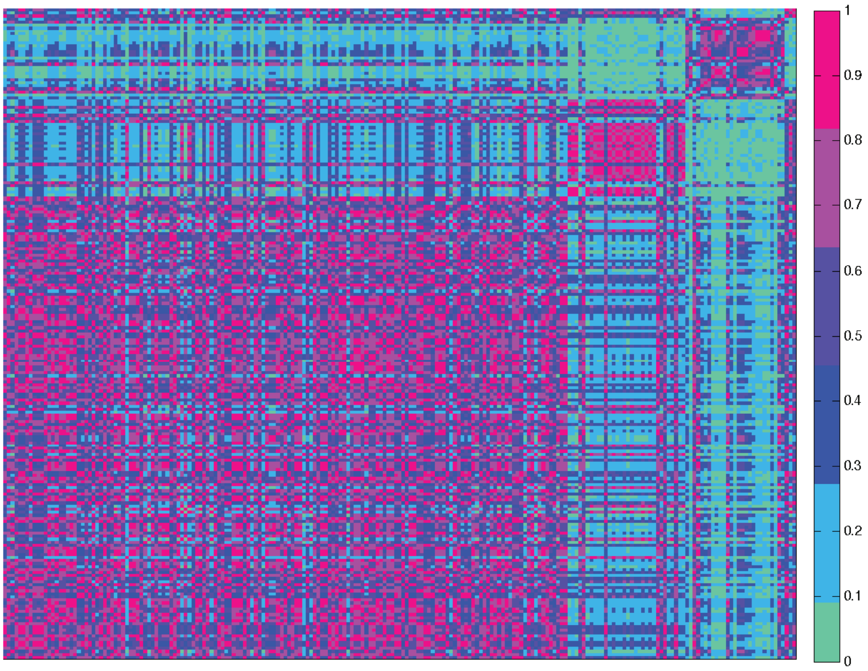


(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

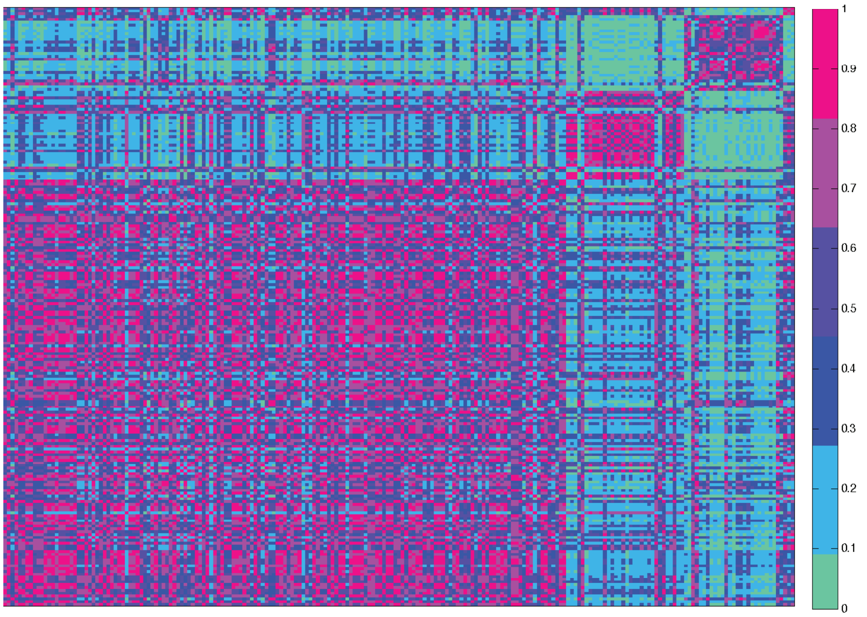
(b) The distribution of the individual members' qualities.

(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.1: Pair # 1 consists of Case 1 and Case 2.

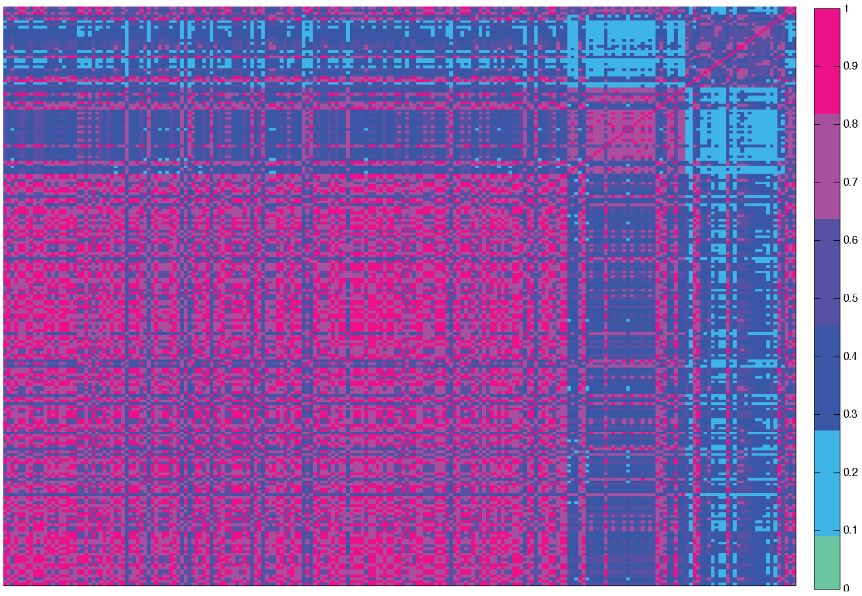


(a) Case 1.

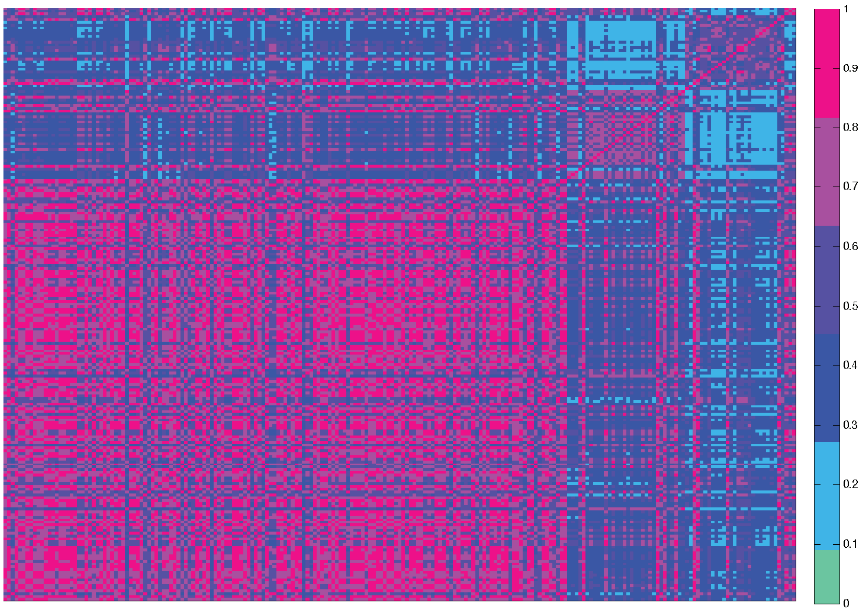


(b) Case 2.

Figure B.2: The heat map of the CO similarity matrix for Case 1 and Case 2.



(a) Case 1.



(b) Case 2.

Figure B.3: The heat map of the ONCE similarity matrix for Case 1 and Case 2.

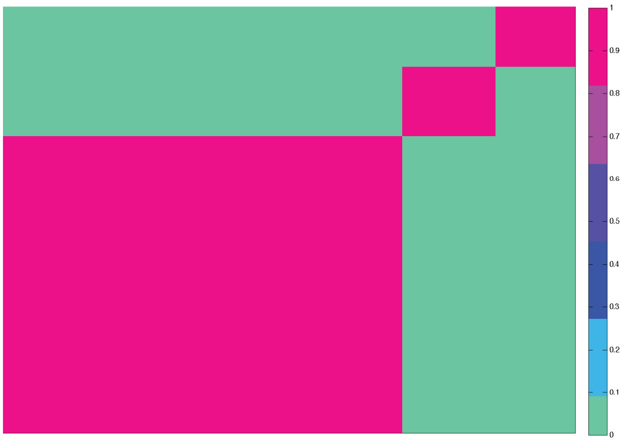
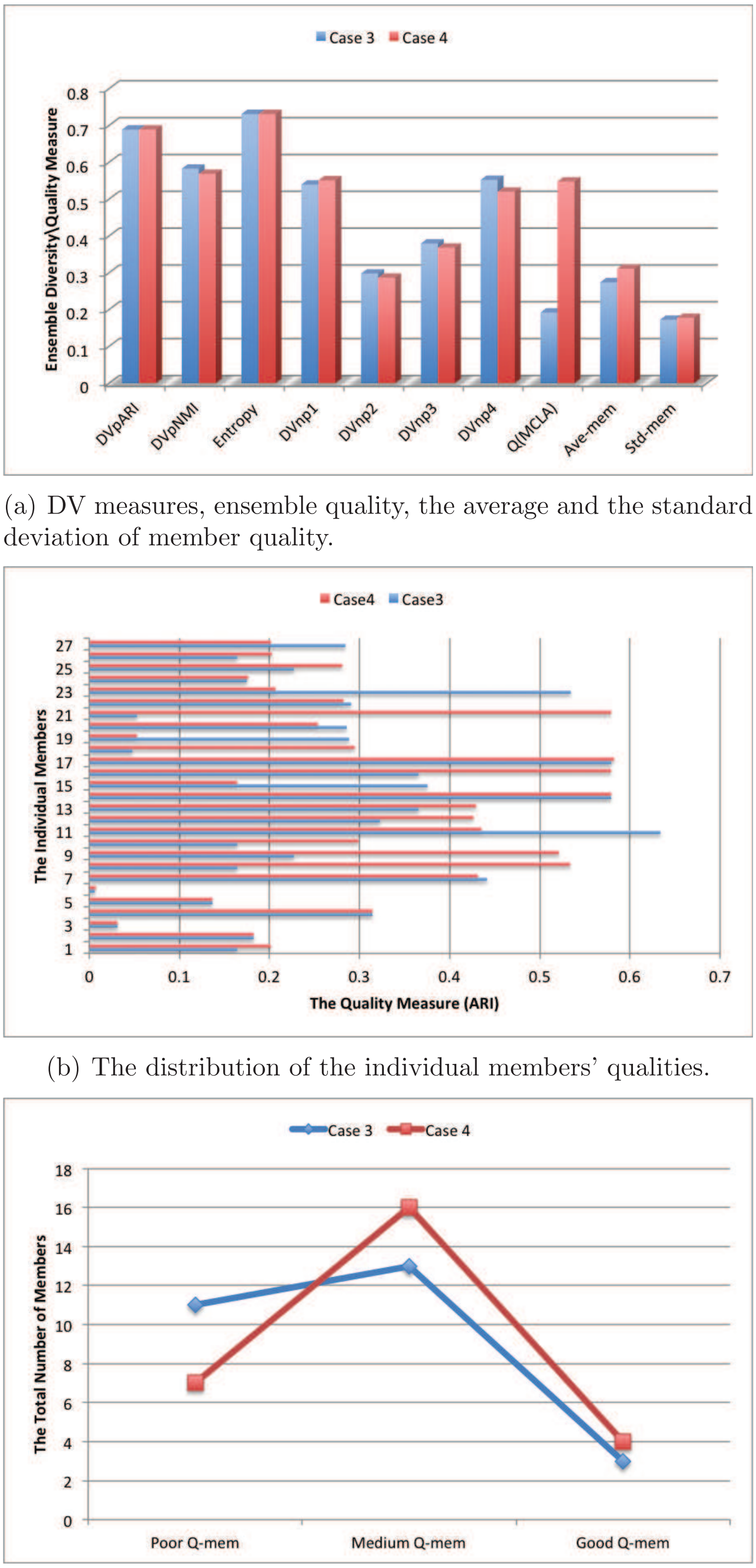


Figure B.4: The heat map of the true label of the Thyroid dataset.

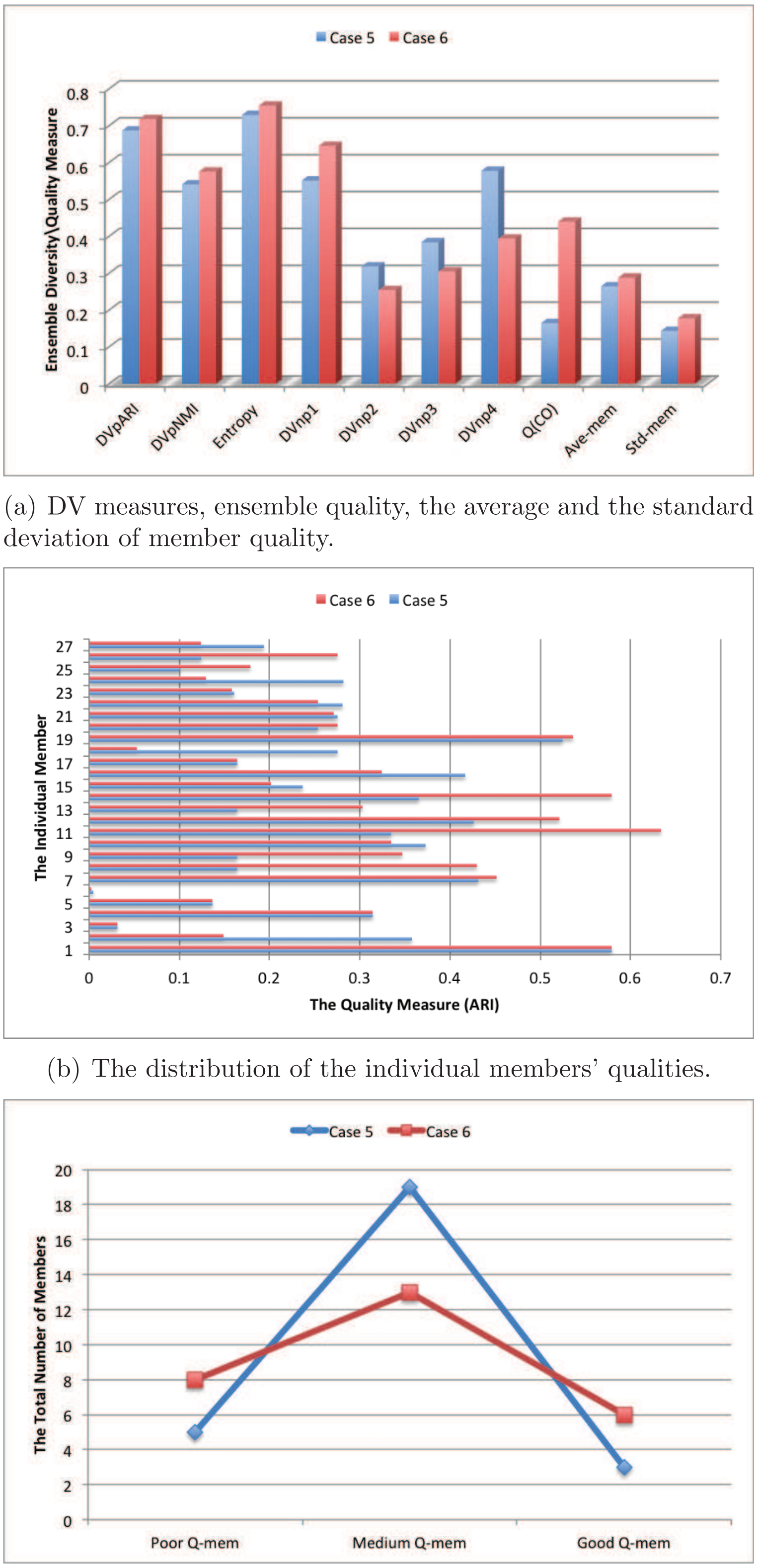


(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

(b) The distribution of the individual members' qualities.

(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.5: Pair # 2 consists of Case 3 and Case 4.

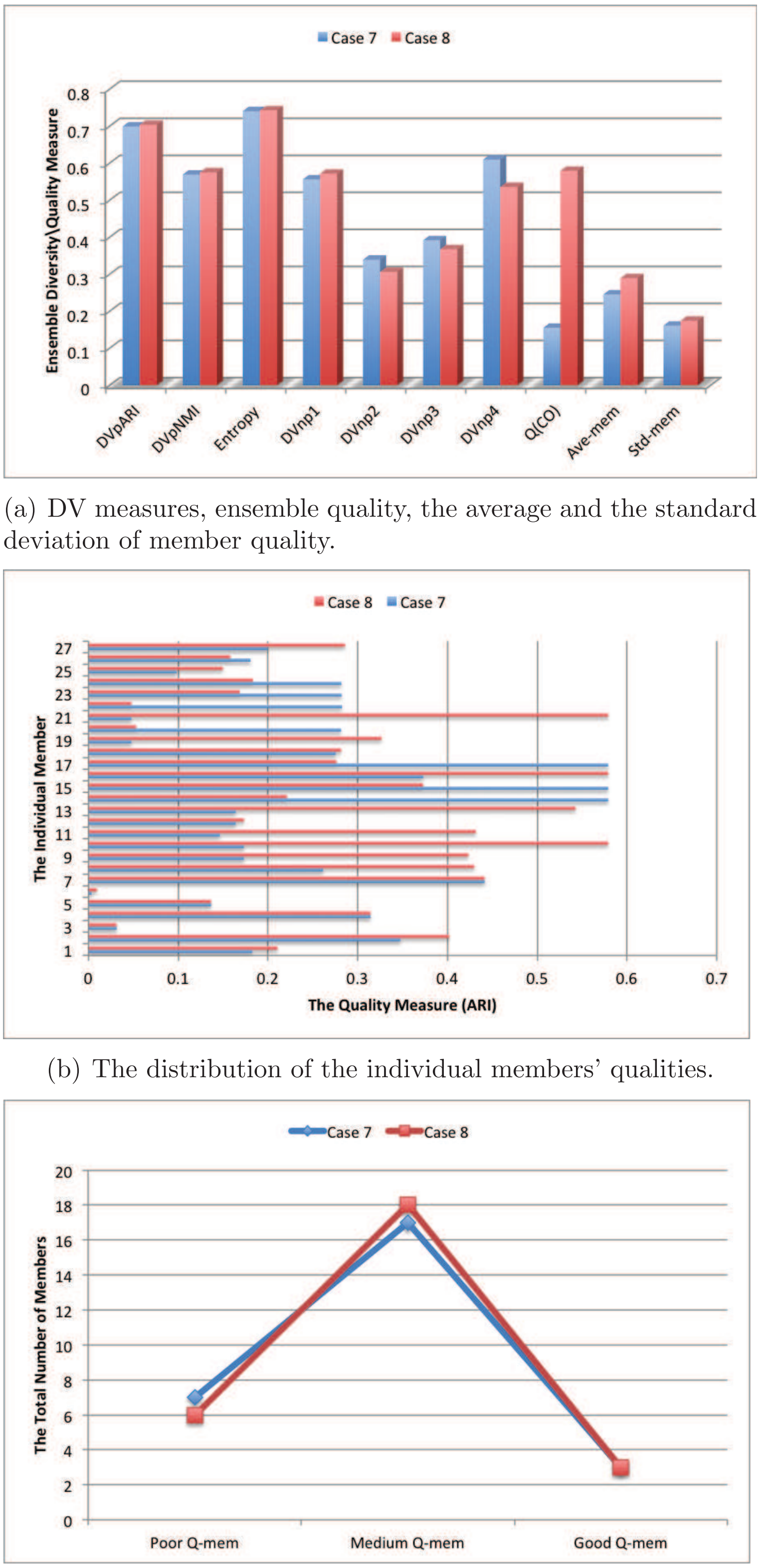


(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

(b) The distribution of the individual members' qualities.

(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.6: Pair # 3 consists of Case 5 and Case 6.

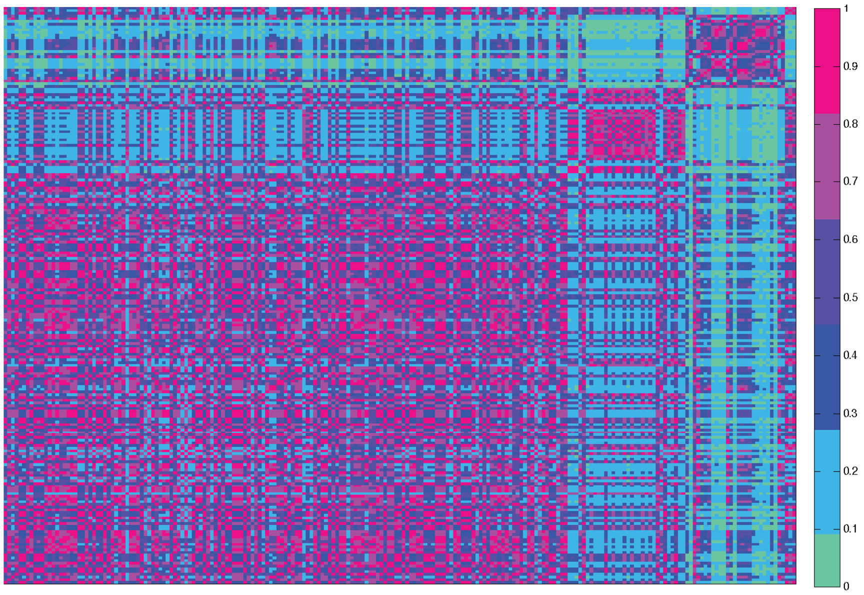


(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

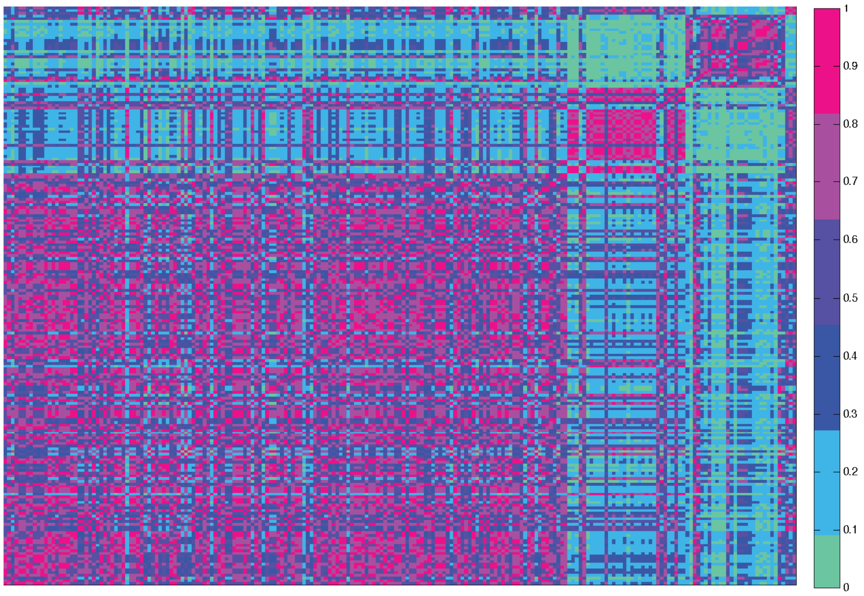
(b) The distribution of the individual members' qualities.

(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.7: Pair # 4 consists of Case 7 and Case 8.

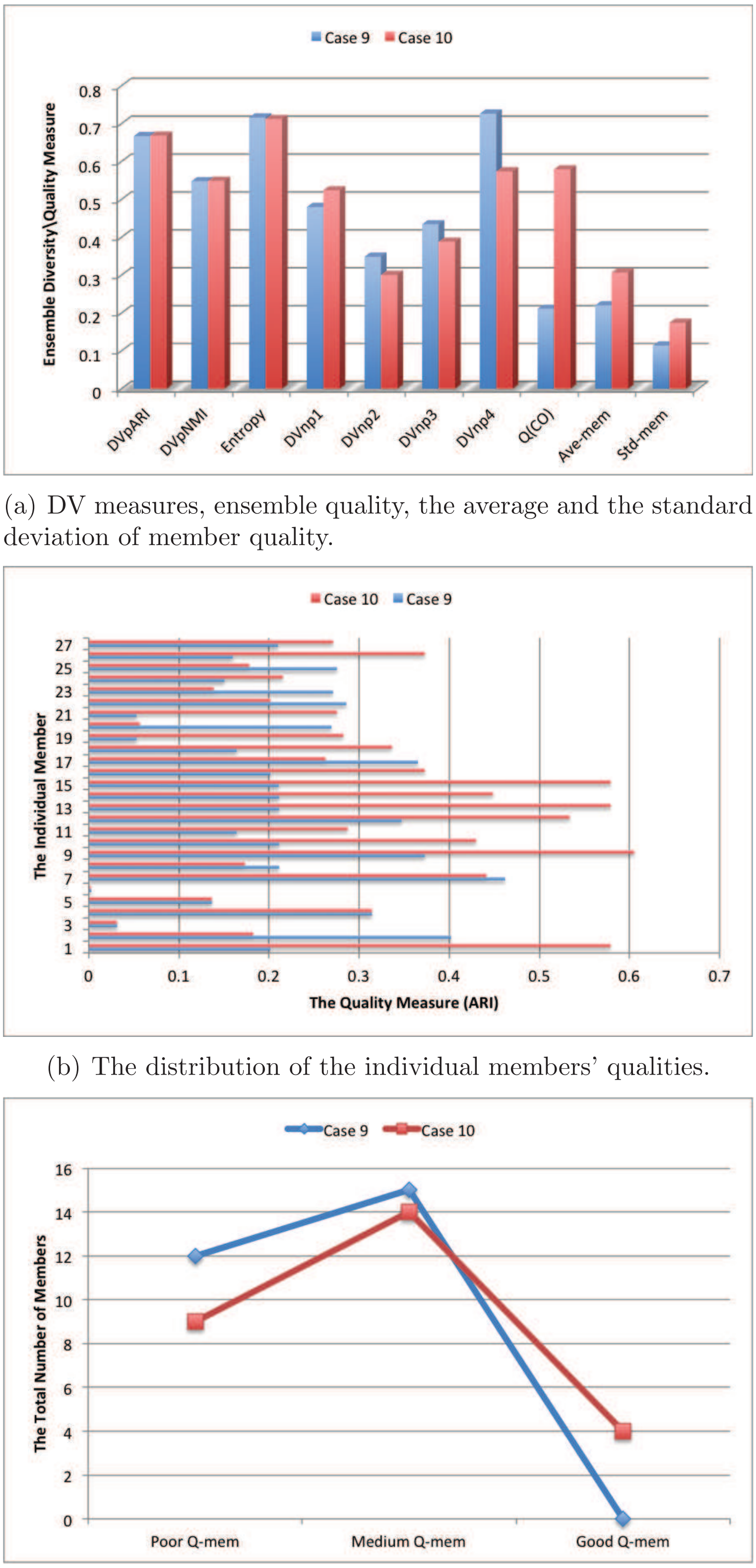


(a) Case 7.



(b) Case 8.

Figure B.8: The heat map of the CO similarity matrix for Case 7 and Case 8.

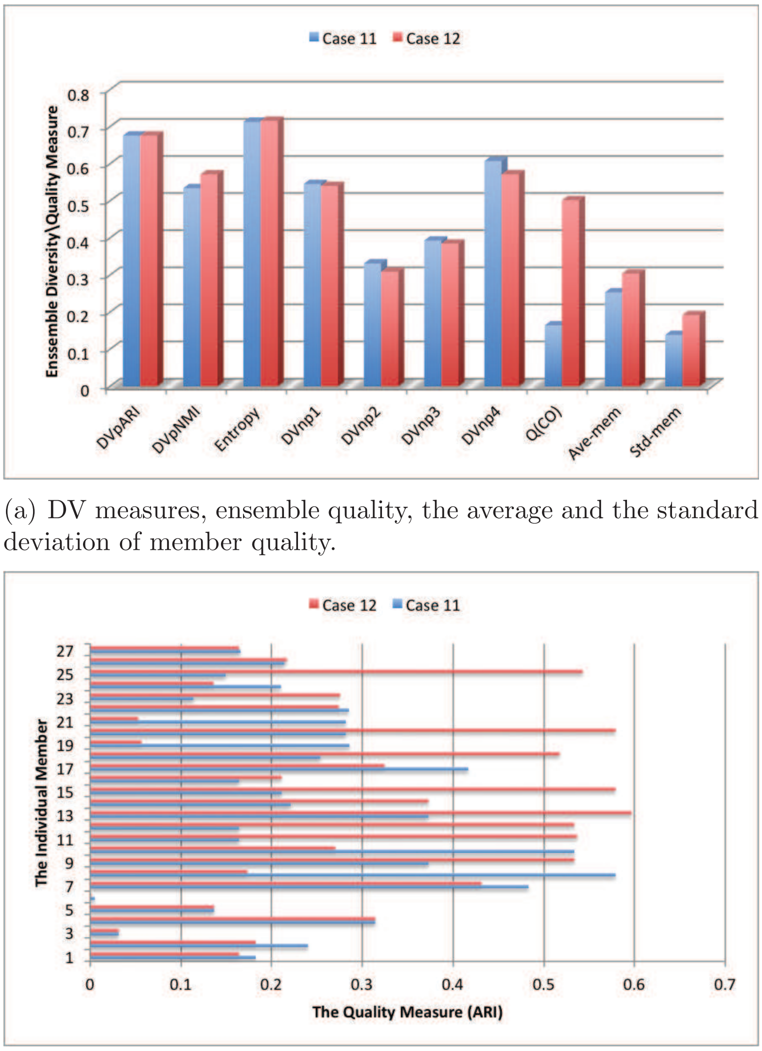


(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

(b) The distribution of the individual members' qualities.

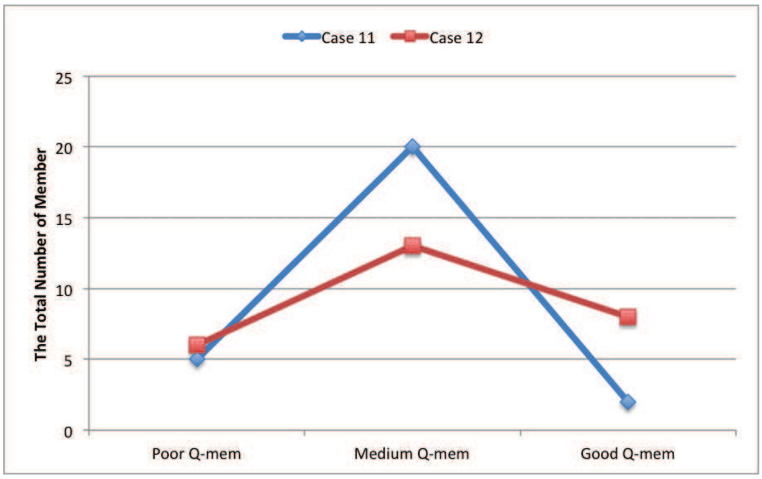
(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.9: Pair # 5 consists of Case 9 and Case 10.



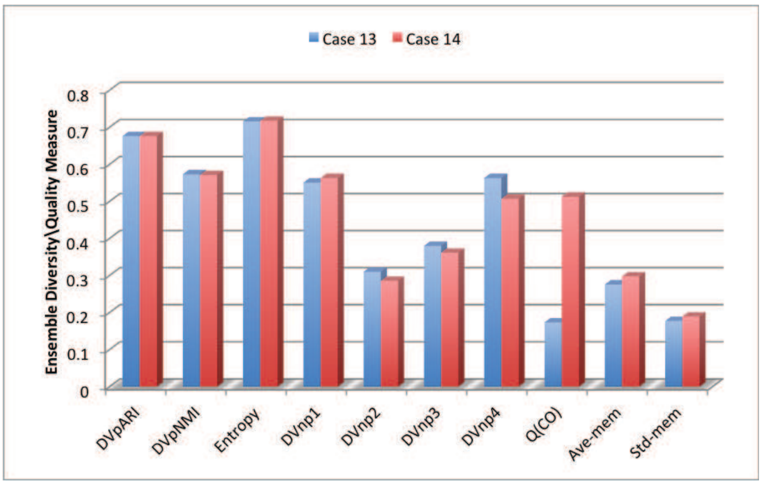
(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

(b) The distribution of the individual members' qualities.

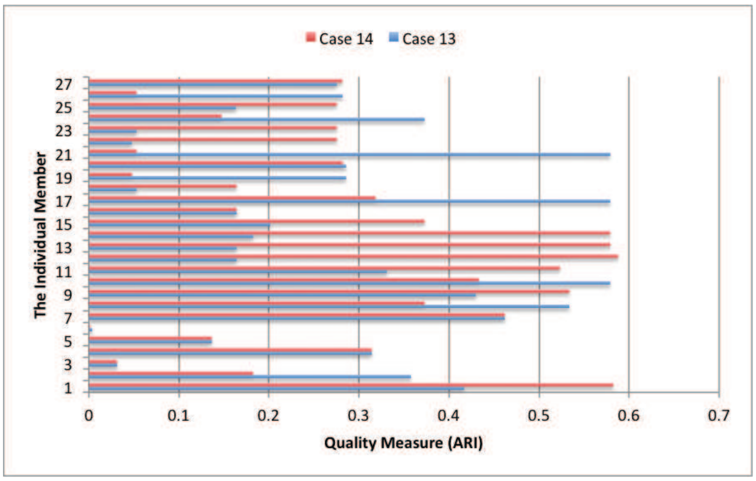


(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

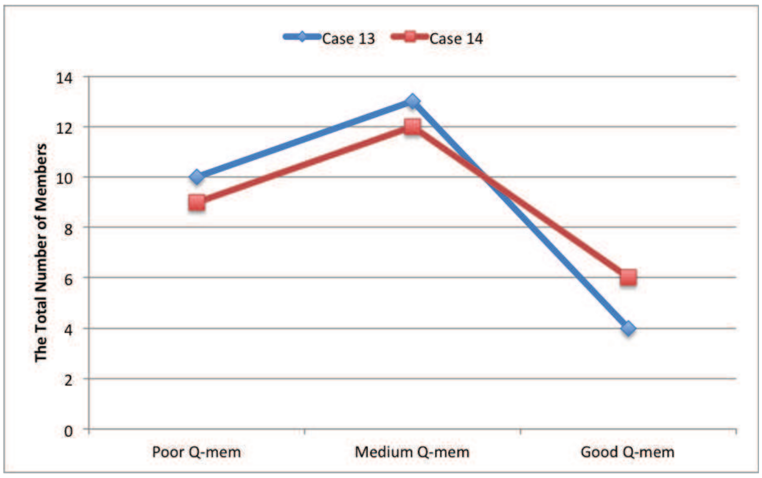
Figure B.10: Pair # 6 consists of Case 11 and Case 12.



(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

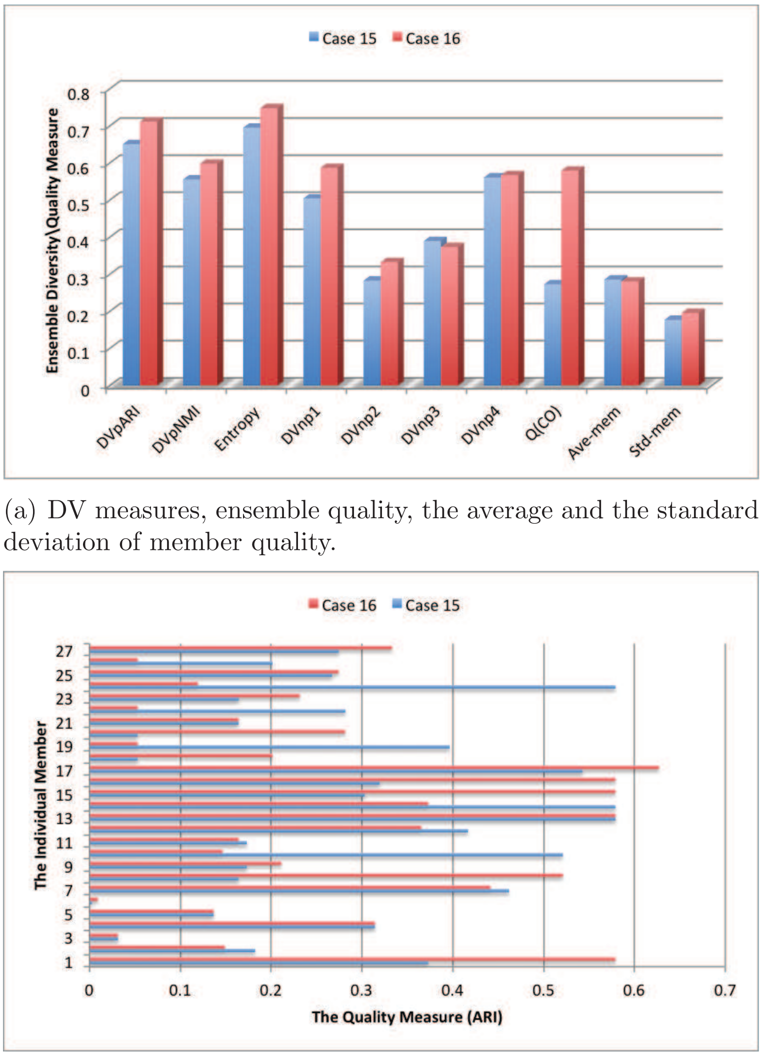


(b) The distribution of the individual members' qualities.



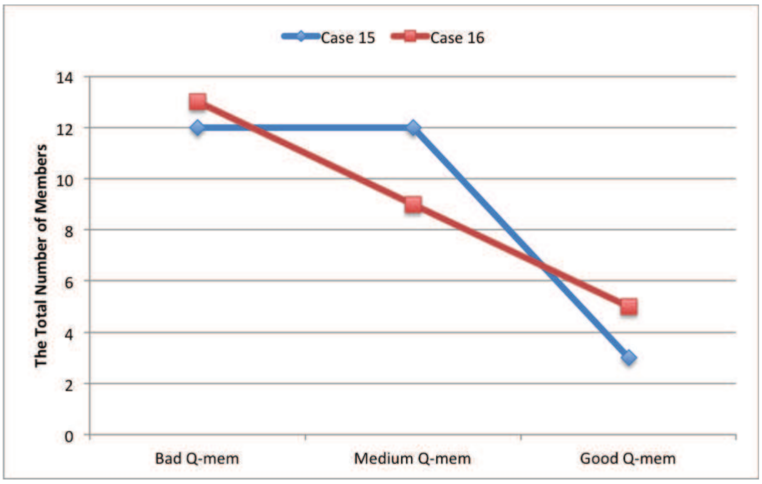
(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.11: Pair # 7 consists of Case 13 and Case 14.



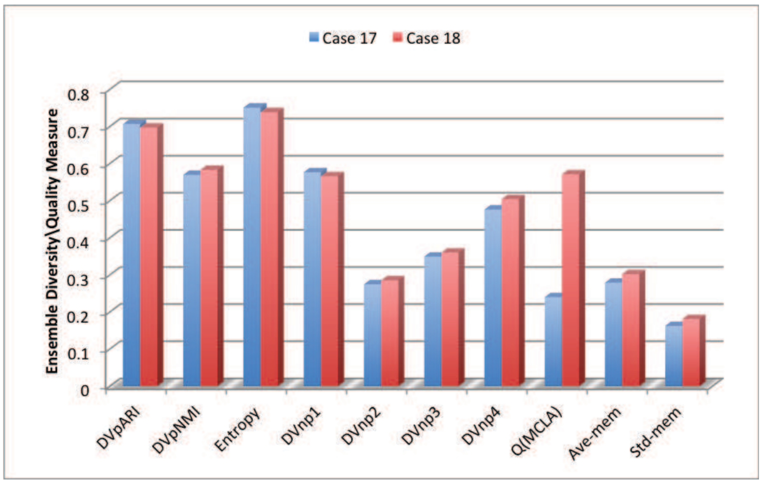
(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

(b) The distribution of the individual members' qualities.

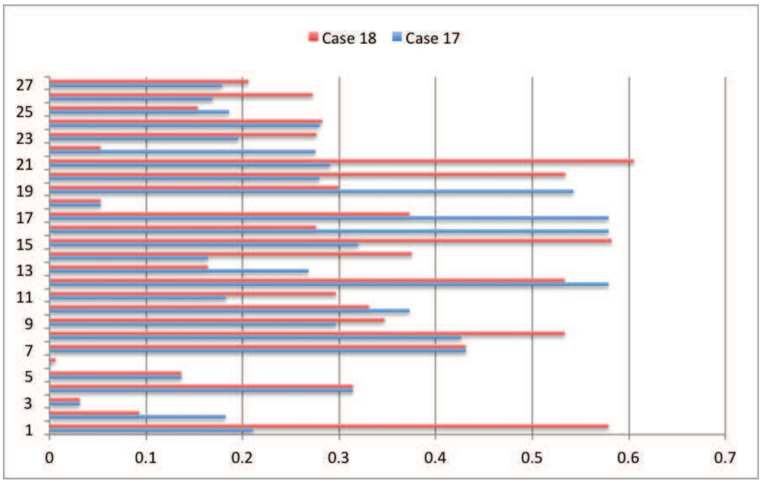


(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

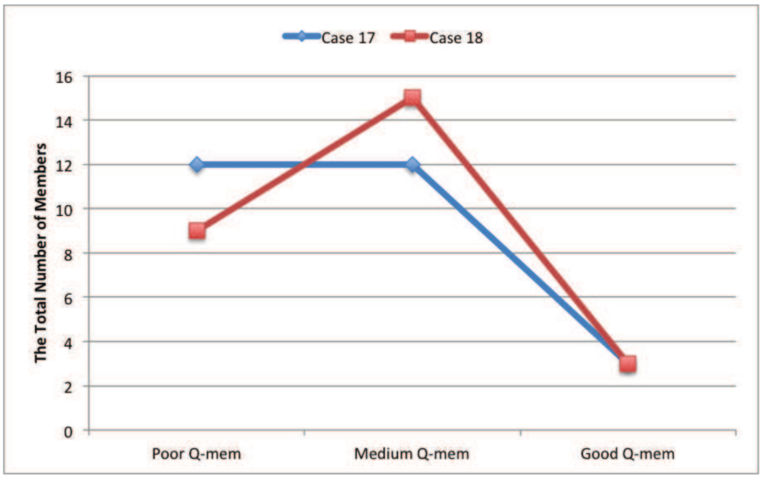
Figure B.12: Pair # 8 consists of Case 15 and Case 16.



(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

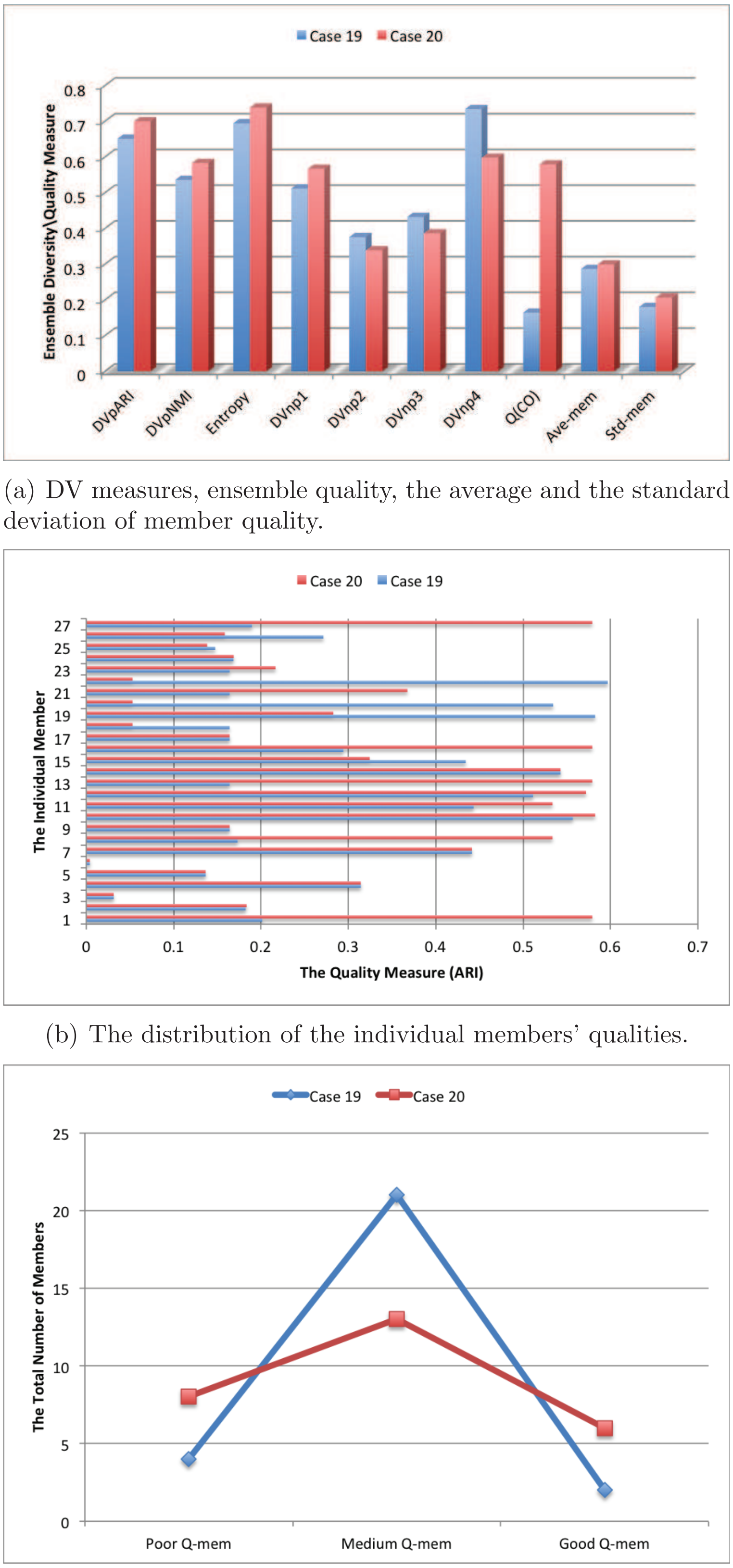


(b) The distribution of the individual members' qualities.



(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.13: Pair # 9 consists of Case 17 and Case 18.

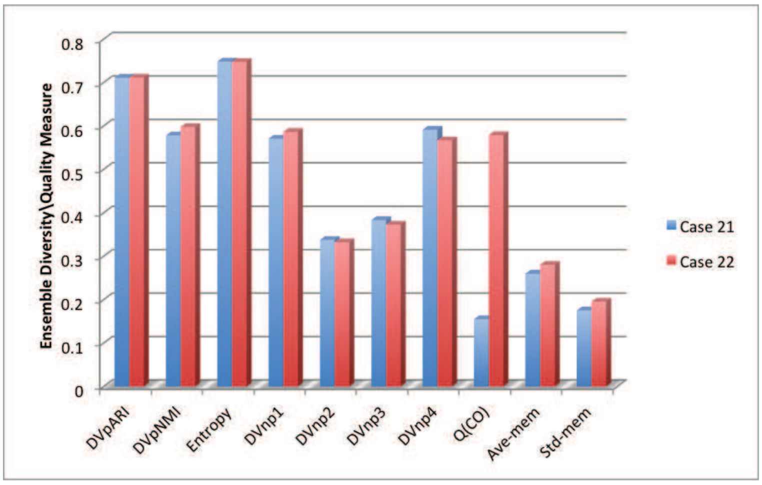


(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

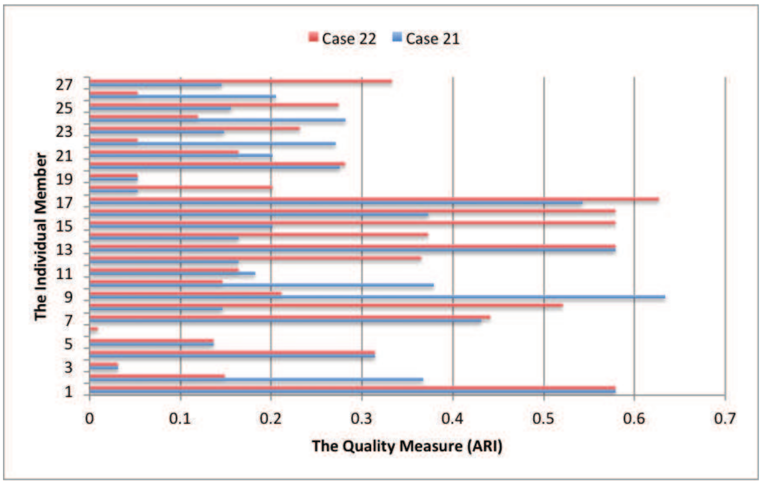
(b) The distribution of the individual members' qualities.

(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

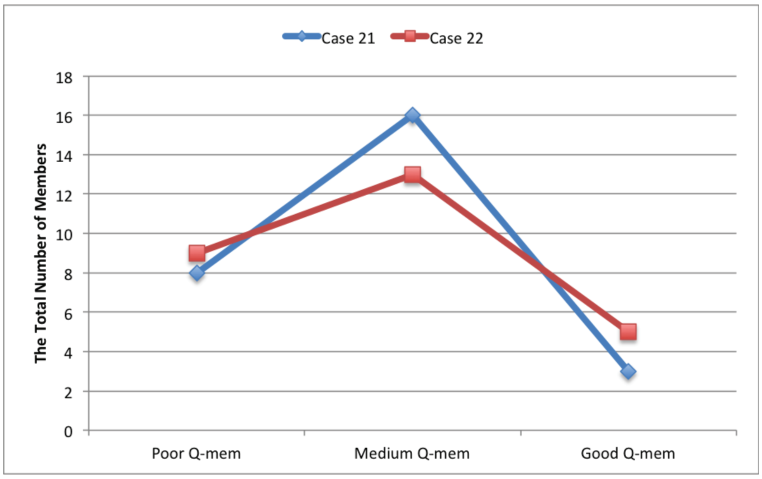
Figure B.14: Pair # 10 consists of Case 19 and Case 20.



(a) DV measures, ensemble quality, the average and the standard deviation of member quality.

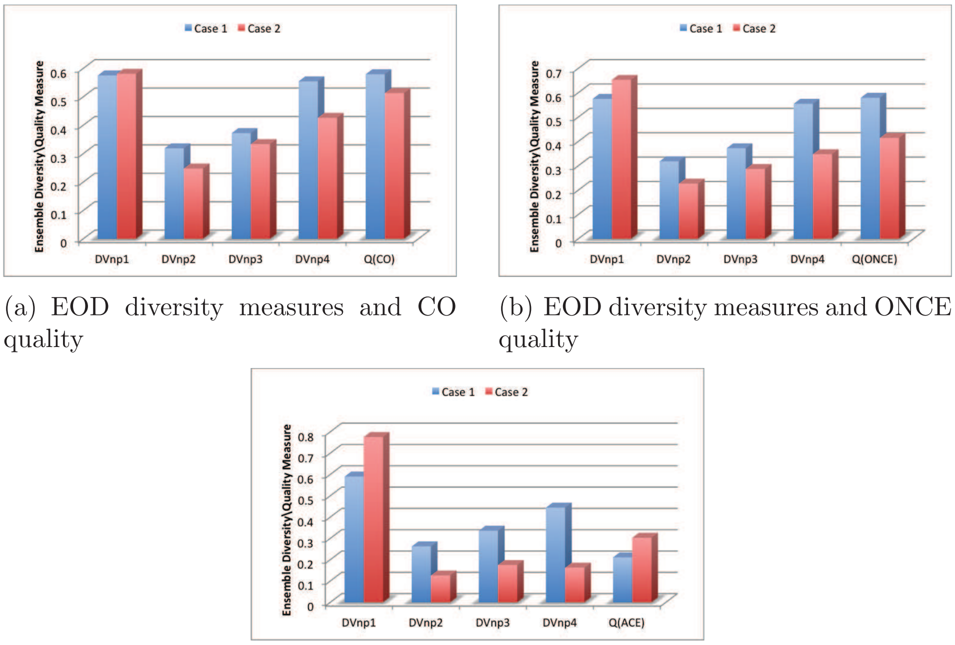


(b) The distribution of the individual members' qualities.



(c) The Number of members whose Poor, Good and Medium Q-mem compared to ensembles qualities in the two cases.

Figure B.15: Pair # 11 consists of Case 21 and Case 22.

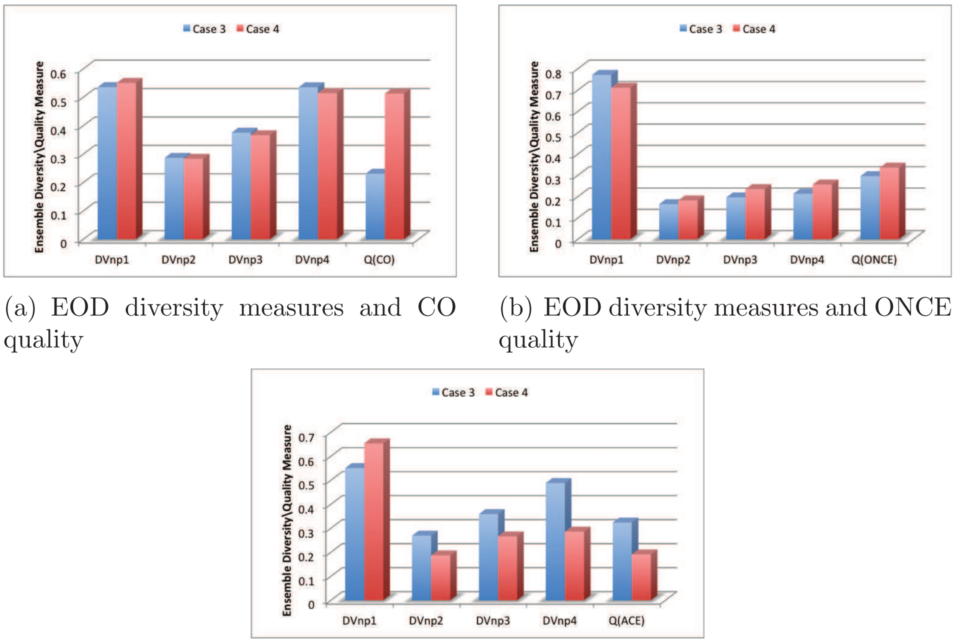


(a) EOD diversity measures and CO quality

(b) EOD diversity measures and ONCE quality

(c) EOD diversity measures and ACE quality

Figure B.16: Pair #1 consists of Case 1 and Case 2.

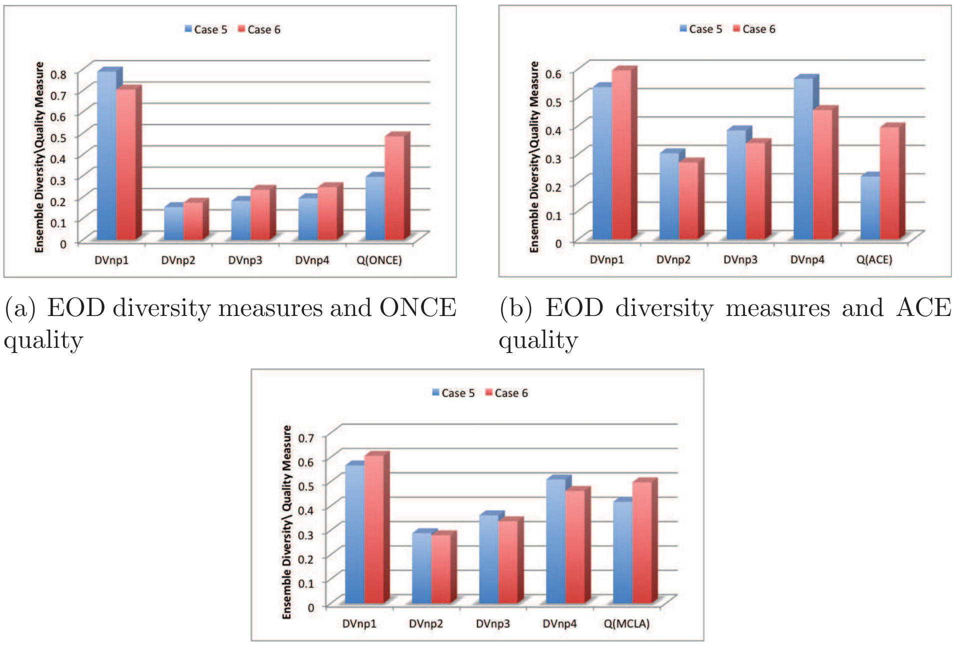


(a) EOD diversity measures and CO quality

(b) EOD diversity measures and ONCE quality

(c) EOD diversity measures and ACE quality

Figure B.17: Pair #2 consists of Case 3 and Case 4.

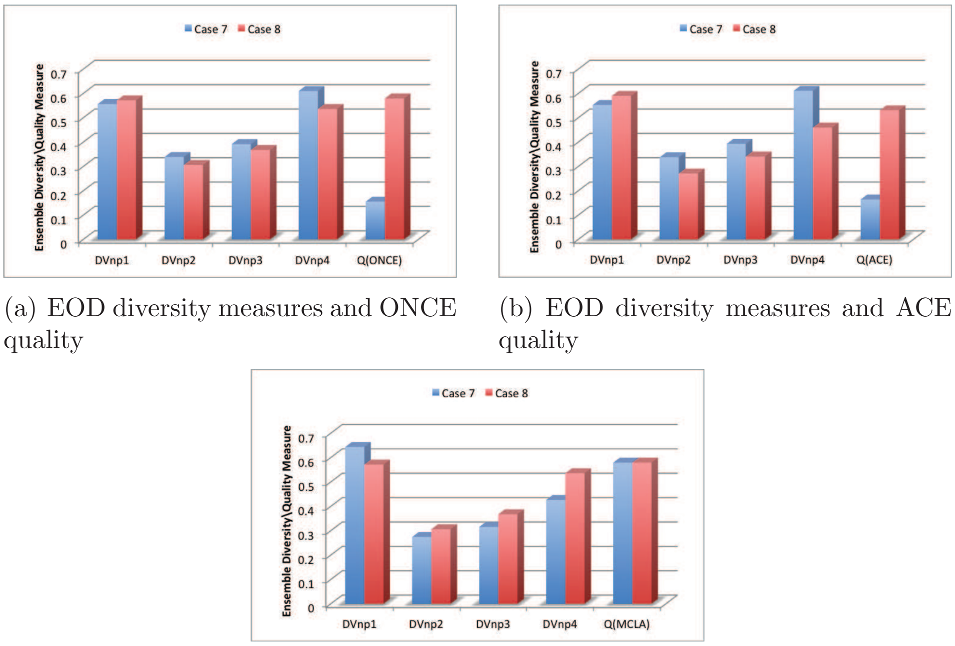


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.18: Pair #3 consists of Case 5 and Case 6.

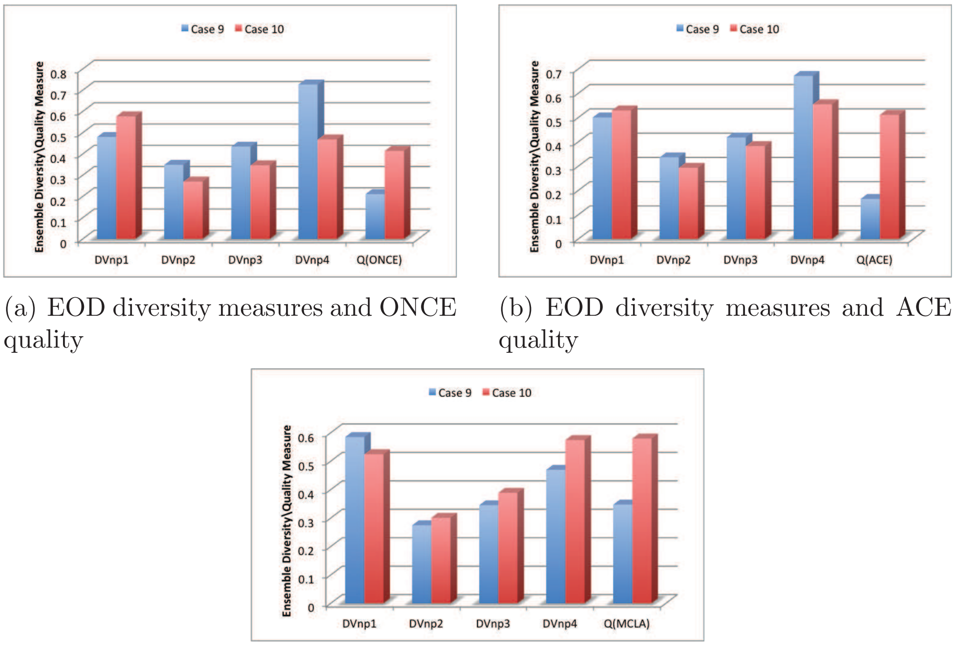


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.19: Pair #4 consists of Case 7 and Case 8.

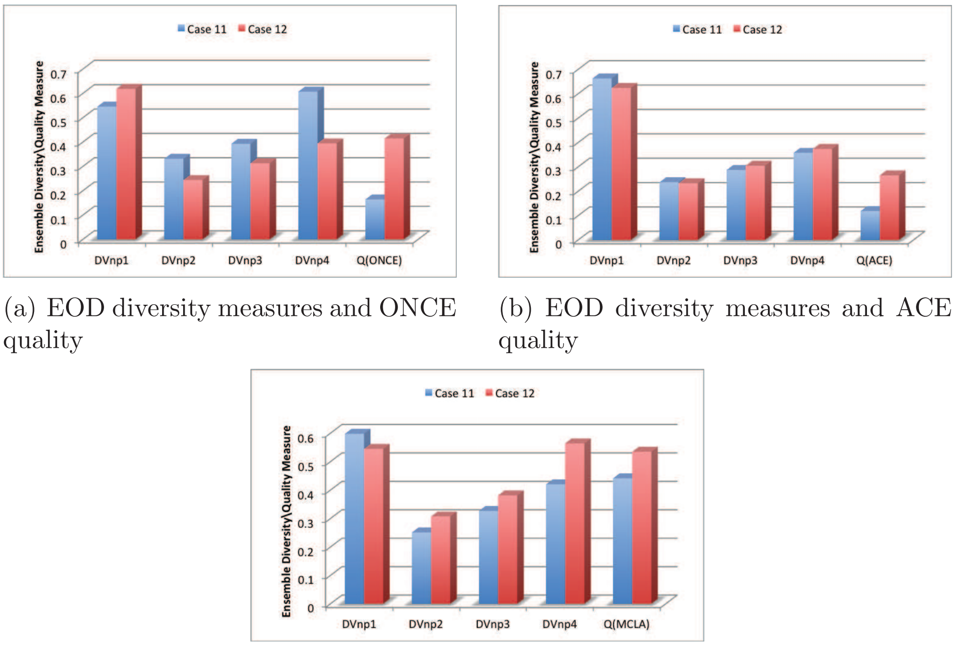


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.20: Pair #5 consists of Case 9 and Case 10.

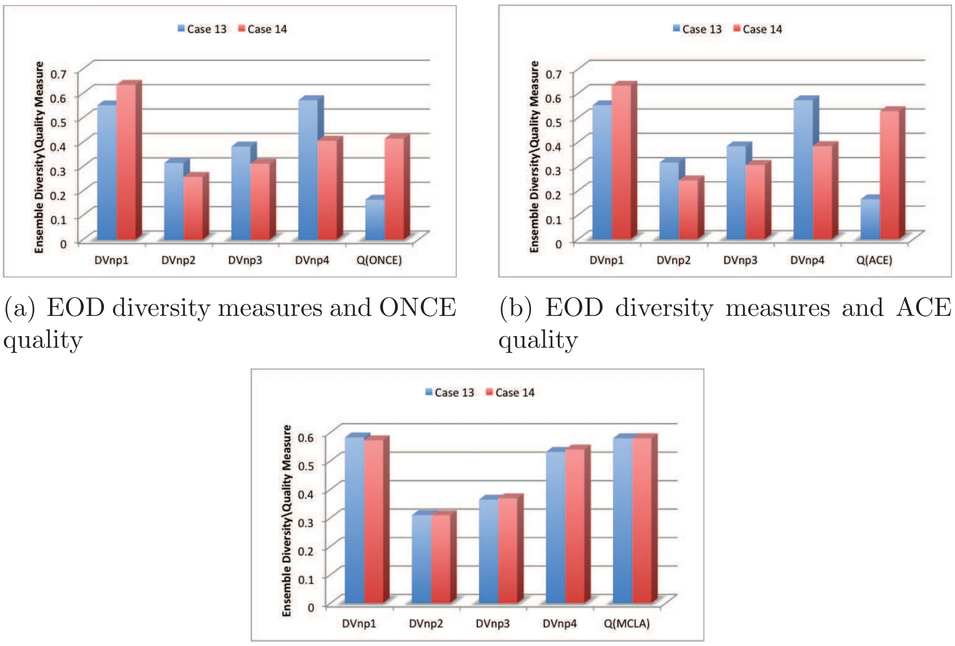


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.21: Pair #6 consists of Case 11 and Case 12.

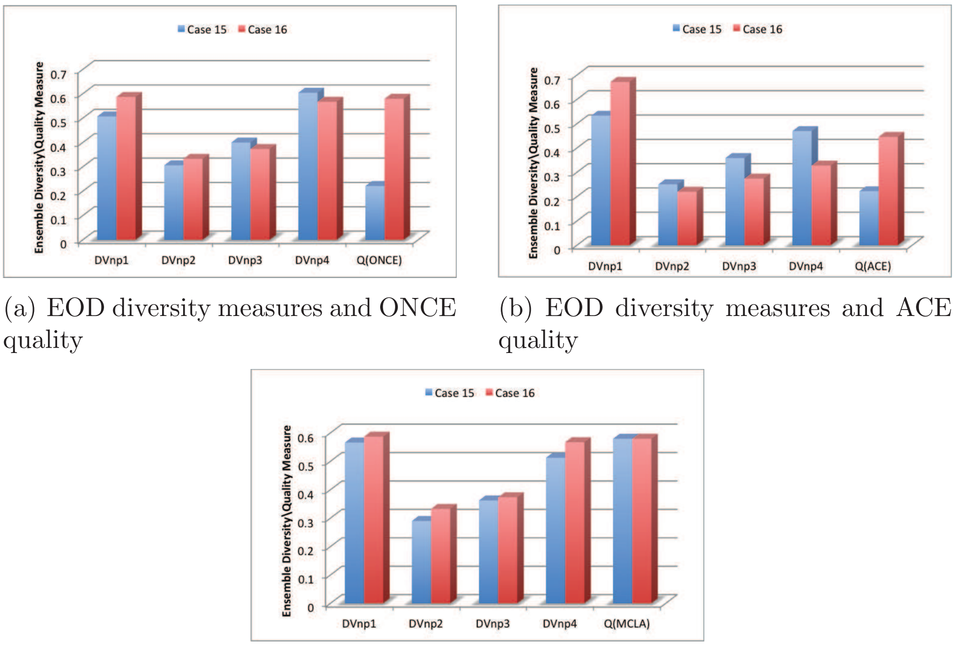


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.22: Pair #7 consists of Case 13 and Case 14.

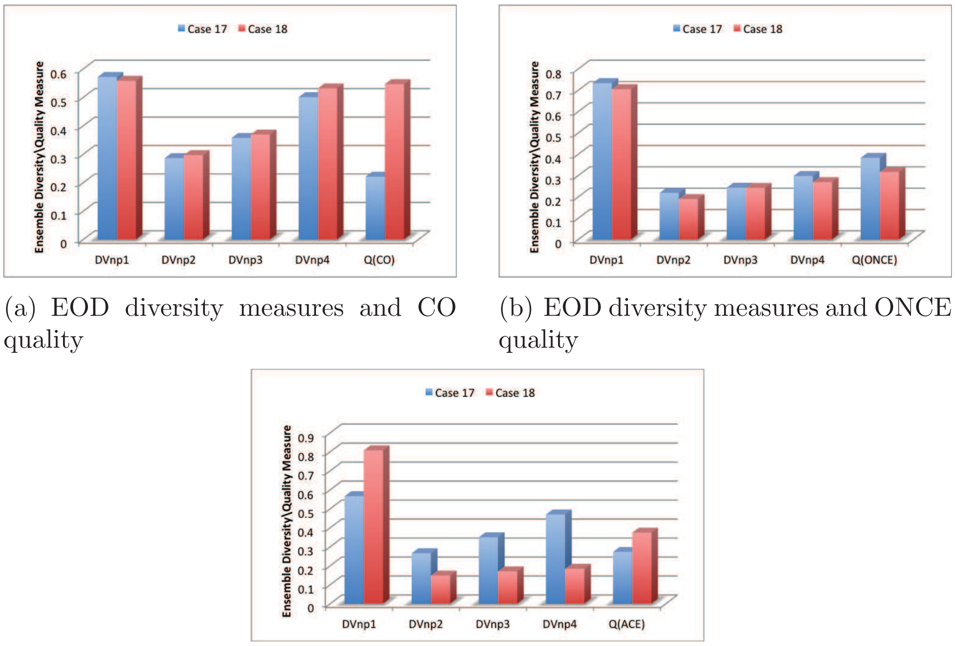


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.23: Pair #8 consists of Case 15 and Case 16.

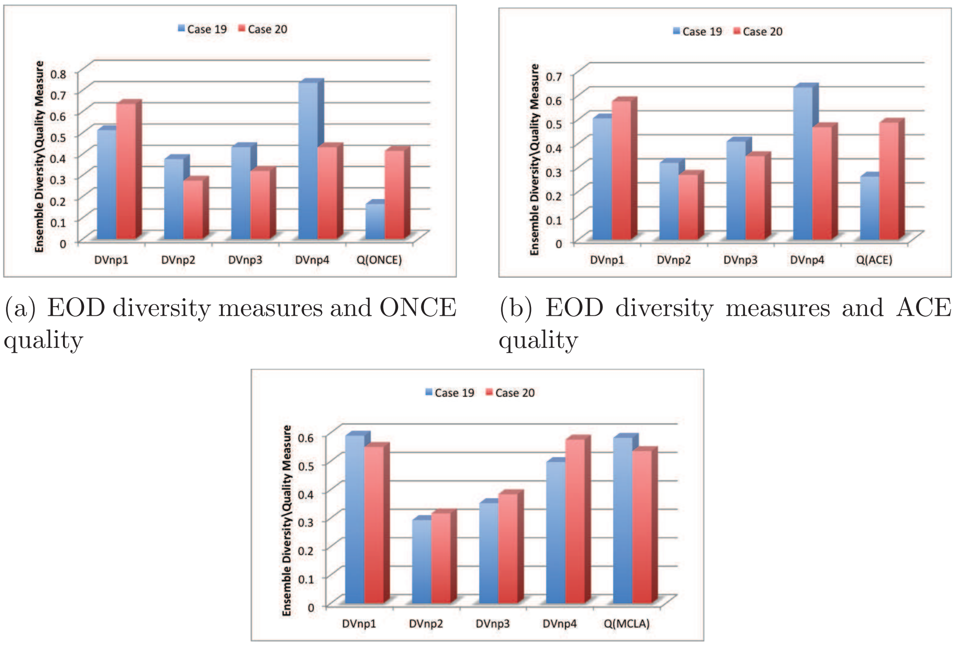


(a) EOD diversity measures and CO quality

(b) EOD diversity measures and ONCE quality

(c) EOD diversity measures and ACE quality

Figure B.24: Pair #9 consists of Case 17 and Case 18.

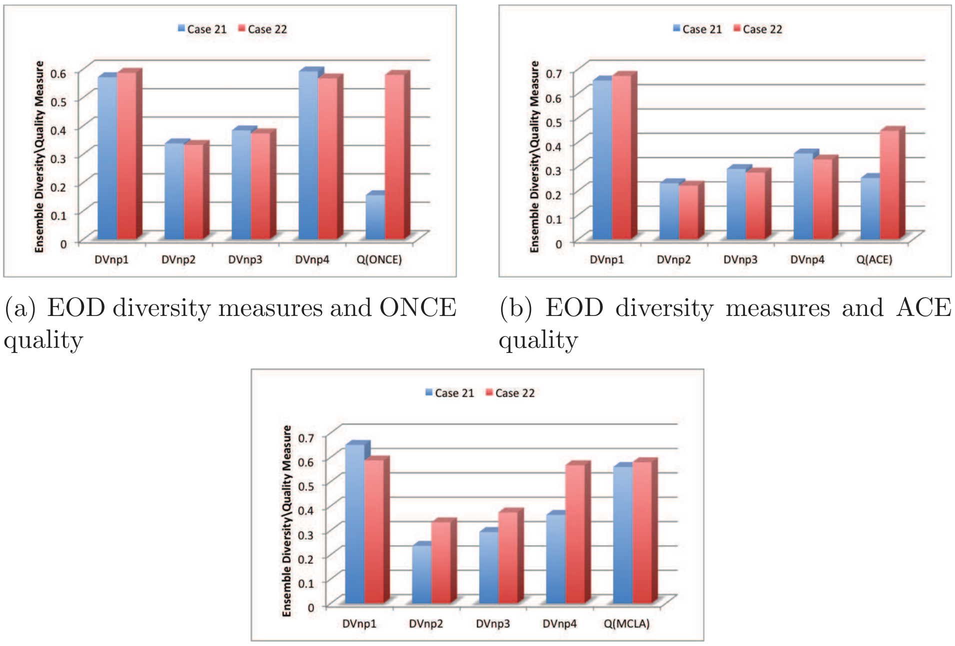


(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.25: Pair #10 consists of Case 19 and Case 20.



(a) EOD diversity measures and ONCE quality

(b) EOD diversity measures and ACE quality

(c) EOD diversity measures and MCLA quality

Figure B.26: Pair #11 consists of Case 21 and Case 22.

B.2 The Complete Results of Eliminating Poor Members

Figures B.27 to B.37 show the results of experiment in Section 6.2.1.3, and each figure has 4 subfigures: subfigure (a) represents the 25 runs on the negative case, and subfigure (b) shows the comparative positive case. The first run in these figures shows the original results of the cases (of combining 27 members), and the bar chart shows the quality of the individual members $Q(m), m = \{1, 2, \ldots, 27\}$ sorted in ascending order. Subfigures (c) and (d) show the diversity values for the negative and positive cases respectively.

In the First Situation, we have cases 1 to 6, where the negative case 1 was associated with ACE and MCLA consensus functions, while case 3 was associated with the CO and MCLA, and case 5 was associated with CO and ACE.

In case 1 shown in Figure B.27(a), the results show that the performance of MCLA improved in run 2, as the poorest quality member was removed, while ACE improved in run 4 in terms of being better than the average member quality, and ACE gradually improved as the poorest member was removed in each run (up to run 16, after which it remained stable). On the other hand, the performance of MCLA remained stable from run 2 to run 24, but in the last run its performance dropped to 0.2. In this run, 3 members were combined; these members had equally high qualities (0.579) and DV_{pARI} equal to 0. This means that this particular pattern of the members can cause MCLA to perform poorly, but not the other consensus functions. In case 2 shown in Figure B.27(b), the performance of CO was stable until run 10, when it then dropped slightly below the average member quality. We think the reason for this is that there is a greater occurrence of the wrongly classified object pairs, compared to the correctly classified object pairs, in the members in runs 11, 12, 13 and 14.

On the other hand, in the same case, ACE performance improved from the point of removing the poorest quality member in run 1. MCLA performance remained stable until run 18, when it then fluctuated below the average member quality to almost equal to the average member quality. Looking closely at runs 18 and 19, the MCLA performance dropped from 0.550 (being positive) to 0.235 (being negative); the diversity measured by DV_{pARI} was 0.428 and 0.432 respectively, and when measured by DV_{np3} it was 0.428 and 0.432 respectively. The performances of CO, ONCE and ACE were the same in these two runs. This indicates that the performance of MCLA in run 19 is not affected by the quality of the members or by their diversity, and that it is in fact due to its implementing techniques, which makes it sensitive to this members' pattern.

In case 3 shown in Figure B.28(a), from run 2 to 4 the MCLA quality gradually improved as the poor members were removed, then it was not stable until run 16, after which it gradually decreased below the average member quality as the latter increased and the diversity (measured by DV_{np3}) slightly decreased. In contrast, CO performance remained below the average member quality until run 7, by which point 6 poor-quality members with a quality of below 0.2 had been removed. In run 7 it improved slightly and then dropped slightly lower than the average member, but in run 19 as the average member quality increased, the performance of CO improved to a higher level until it reached 0.597 in the last run. The diversity in this run reached a medium level measured by DV_{pARI}. In case 4 shown Figure B.28(b), the performance of CO was stable until run 13 where it slightly improved as the average quality increased and the diversity measured by DV_{pARI} slightly decreased.

In case 5 shown in Figure B.29(a), the performance of ACE improved as the poorest member was removed in run 2, and it then remained stable until run 13 when it improved to a high level as the average member quality increased and the diversity slightly decreased. The results show that in run 19 the performance of MCLA decreased to below the average member quality as the latter increased and the diversity decreased to reach a value of 0.4 (measured by DV_{pARI}).

In the positive case 6 shown in Figure B.29(b), the quality of CO remained almost stable as the average quality of the members increased and the diversity decreased, until run 20, when its quality was slightly below the average member and diversity reached 0.4. The performance of ACE improved further as a result of removing poor-quality members, and reached over 0.6. The performance of MCLA in this case was not stable; it had small fluctuations as the average member quality increased until run 17, after which it fluctuated greatly from above to below the average member quality as diversity decreased.

The Second Situation includes cases 7 to 22, and as mentioned previously, the negative cases occurred in three consensus functions, which are CO, ONCE and MCLA in case 17, and in the remaining cases they occurred with CO, ONCE and ACE. However, in these negative cases (7, 9, 11, 13, 15, 17, 19, and 21) there was a clear cut-off point for CO, ONCE, and ACE improvements from below the average member quality to higher than the average member quality. It is therefore clear that removing some poor-quality members, and consequently increasing the average member quality and decreasing the level of diversity, positively influenced the performances of CO, ONCE and ACE. The positions of the cut-off point were different for each consensus function in each case, but the remarkable features about them are as follows:

1. CO improved in run 9 (in cases 13, 15, 17 and 19), 11 (in cases 7 and 11), 13 (in case 21), and 17 (in case 9). It is clear that CO improves when most of the poor-quality members are removed, compared to other consensus functions. This is explained by the fact that CO measures the degree of agreement between members when clustering a pair of objects, and in the members of these cases the poor-quality members increase the certainty of wrongly classified pairs of objects more than correctly classified pairs of objects. In addition, as CO only considers the object pairwise information, it produced poor-quality clustering until we had removed some of the poor-quality members. For example, in case 15 shown in Figure B.35(a), CO had a very poor performance until run 9. Comparing run 7 to run 10, CO performance improved from 0.221 to 0.579 (as good as its performance in its compared positive case number 16),

and the average member quality increased slightly from 0.346 to 0.376. The diversity measured by DV_{pARI} decreased slightly from 0.539 to 0.525, while that measured by DV_{np3} was equal to 0.411 in the two runs. This is shown in Figure B.39, the heat map of the CO similarity matrix, in runs 7 and 10 in case 15. As we can see, the certainty of the correctly classified object pairs in the first cluster (from the left) in run 10 is higher than in run 7, as well as some object pairs in the third cluster. Thus, the clustering results produced by CO in run 10 are much better than in run 7 in terms of quality; neither diversity measures nor the average member quality are able to give an explanation for this difference in performance.

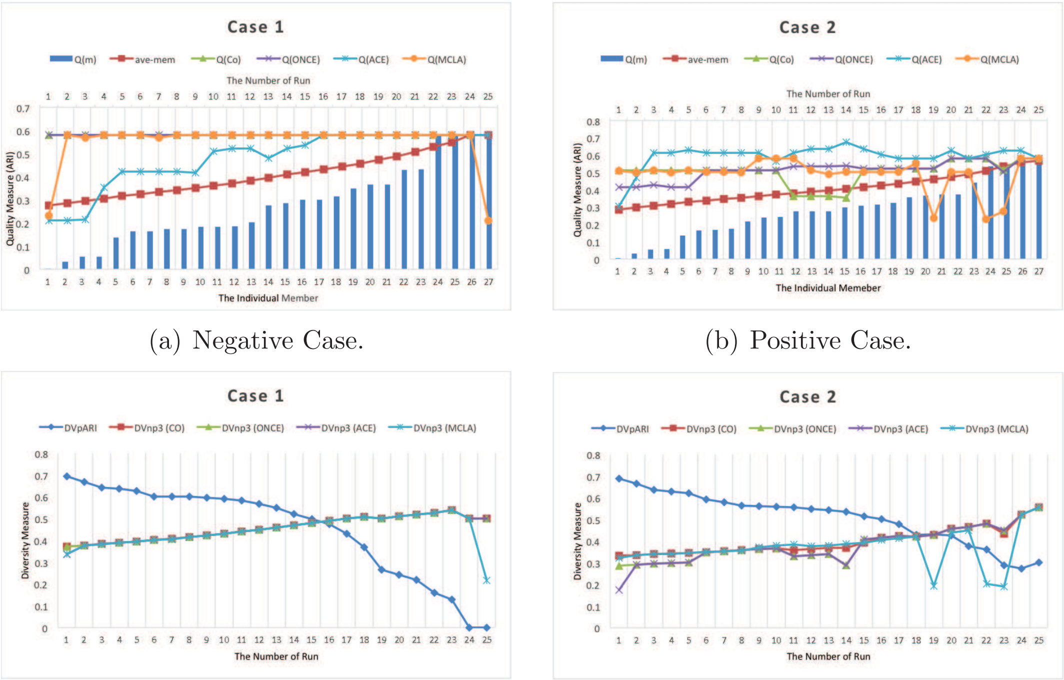
In all the negative cases, the performance of CO at some point of removing the poor-quality members improved to a level that was as good as or above its performance in its compared positive case. The exception to this was case 9 shown in Figure B.31(a), where in all the 25 runs CO performance did not reach the same level as in case 10. The highest quality of CO in case 9 was 0.516, which was in run 25, where it was built by combining 3 members which each had a quality of 0.373, 0.402, and 0.462, making an average of 0.412 and DV_{pARI} of 0.536. In case 10 shown in Figure B.31 (b), the highest performance of CO was 0.579, which occurred in a number of runs, but mostly from run 15 to the last run, the average member quality increased from 0.452 to reach 0.588, while the diversity decreased from 0.412 to 0.190. This clearly indicates that this high-quality performance of CO in the positive case is influenced by the high average member quality and lower diversity, which makes the certainty between the correctly classified objects pairs higher than between the wrongly classified ones.

2. ONCE improved at an earlier stage than CO, and specifically in run 9 (in cases 7, 11 and 19), 15 (in case 9), 5 (in case 13), and run 4 (in cases 15 and 21). In fact, there are no negative cases in this experiment where ONCE improved after CO improvement, it is always the case that ONCE improves

before CO improves, after removing poor-quality members. The reason for this is that ONCE considers the similarity between the common neighbours of a pair of objects, as well as the similarity between the pair itself, so there is more information to be constructed from the members in ONCE than in CO. For example, in case 15 shown in Figure B.34, ONCE started with a low quality of 0.22; it then improved in run 4 of 0.579 after removing only three poor-quality members, and then it remained stable at the same quality until the last run. In the same case, in runs 2 and 5, the ONCE quality was 0.221 and 0.579 respectively, the average member quality increased from 0.296 to 0.329 and the DV_{pARI} decreased from 0.623 to 0.566. Plotting the similarity matrix of ONCE in both cases, as shown in Figure B.41, we found that the certainty between the correctly classified object pairs, in particular in the third clusters (from the left) in run 5, is higher than in run 2, and there is lower certainty between wrongly classified object pairs, in particular the one that is not truly classified in the first and the second clusters (from the left). It is also noticeable in some of the positive cases, that the performance of ONCE also improved after removing the poorest members to become as good as CO, or in some cases better than CO.

3. ACE improved in run 2 in most of the negative cases, except in case 17 as shown in Figure B.35 (a), where it improved in run 4 and in case 9 in run 10. It is noticeable that the improvement of ACE occurred gradually as we removed one poor-quality member at a time; this was also noticeable with some positive cases such as cases 12, 16 and 22. The highest quality in this experiment was achieved by ACE in case 17 (run 16), which was 0.703 with an average member quality of 0.418 and diversity measured by DV_{pARI} of 0.581. The performances of other consensus functions in this run were of 0.579. It is therefore obvious that these members had the right diversity among them, and that this represents a pattern of success for ACE. It is noticeable that in 10 cases there was one run that had the same average member quality and a value for DV_{pARI} between 0.515 to 0.581 (medium level), and that ACE always

had a high performance of between 0.608 and 0.703 (cases 2, 4, 12, 14, 16, 17, 18, 19 ,20, and 22). In the other cases, we had in 6 cases a run with also the same level of average member quality and DV_{pARI} between 0.504 and 0.548, and ACE achieved a quality of 0.579.



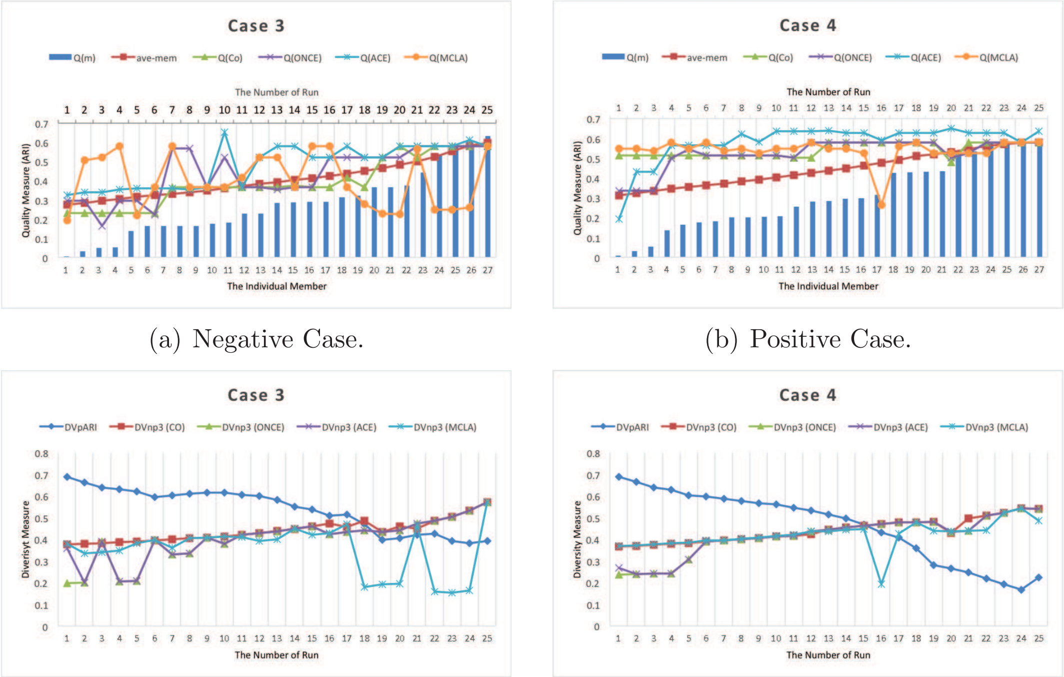
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.27: 25 ensemble runs for case 1 & 2, in each run one member is removed.



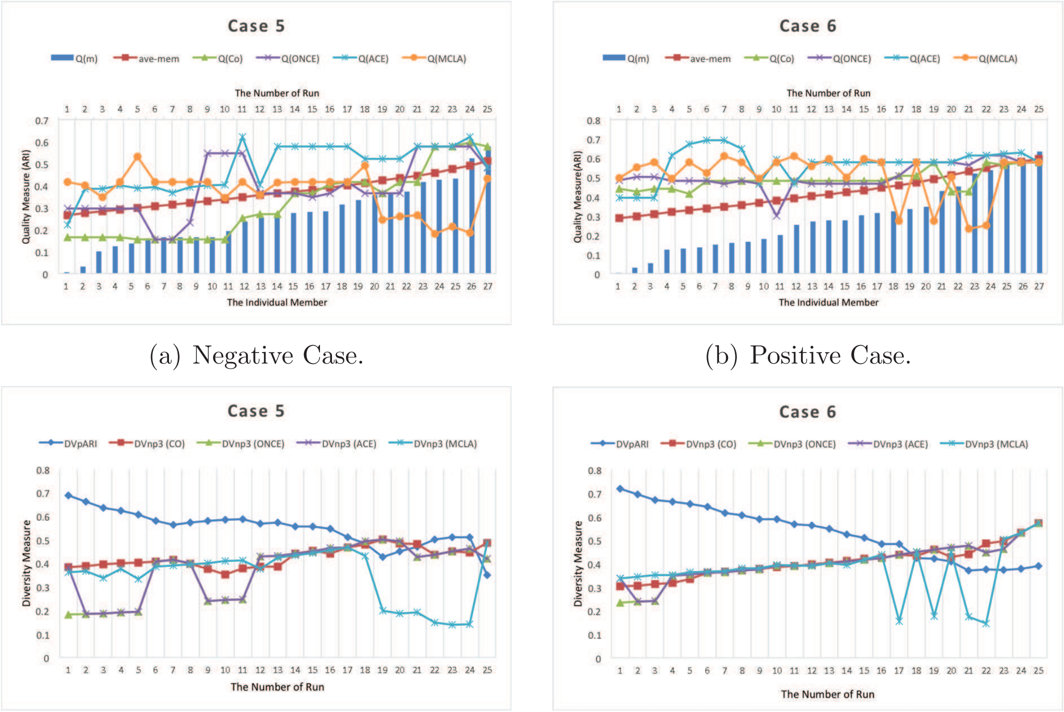
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.28: 25 ensemble runs for case 3 & 4, in each run one member is removed.



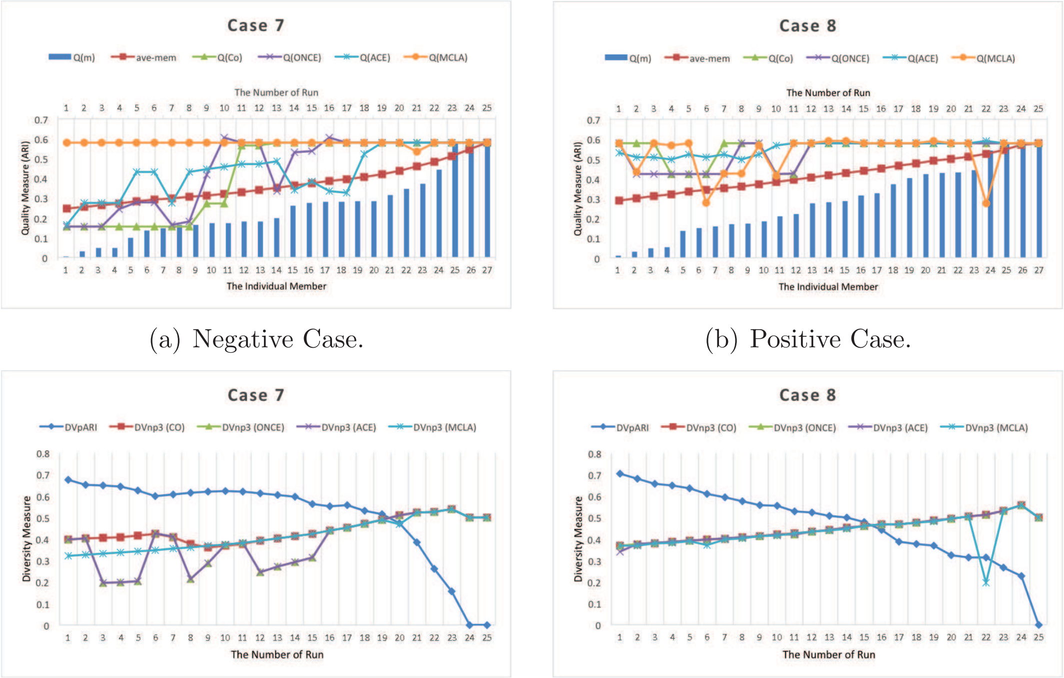
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.29: 25 ensemble runs for case 5 & 6, in each run one member is removed.



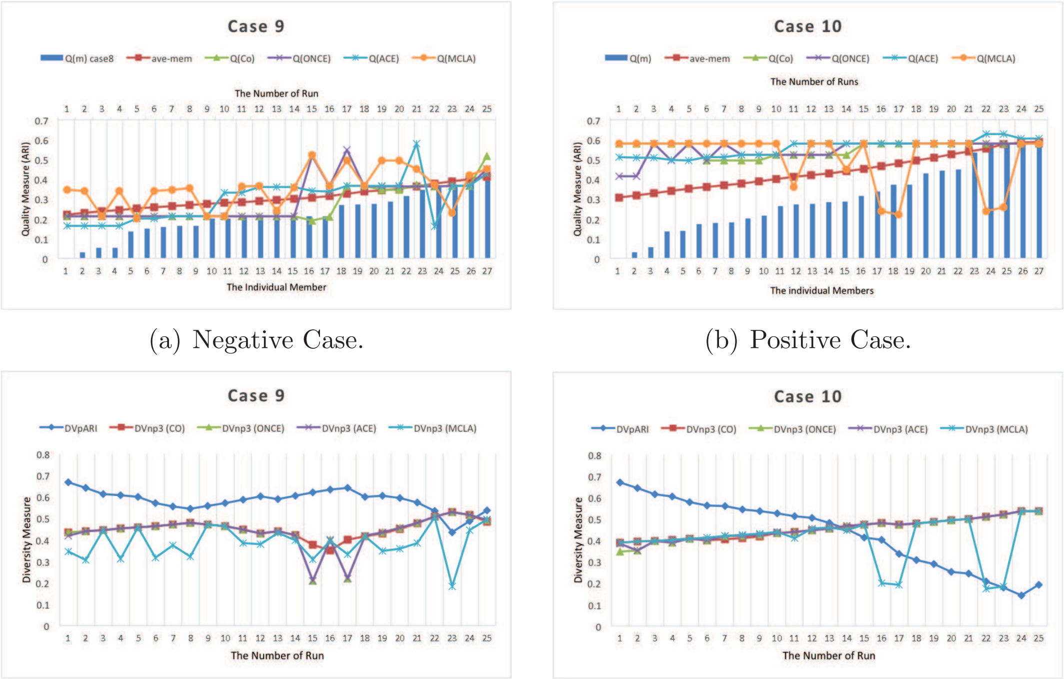
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.30: 25 ensemble runs for case 7 & 8, in each run one member is removed.



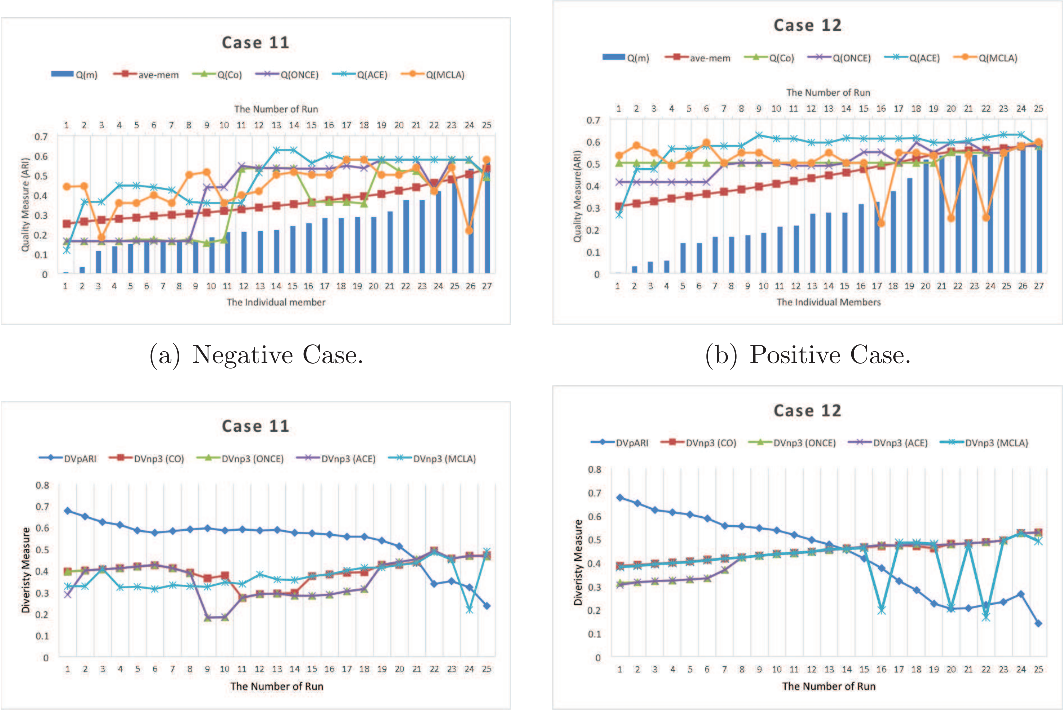
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.31: 25 ensemble runs for case 9 & 10, in each run one member is removed.



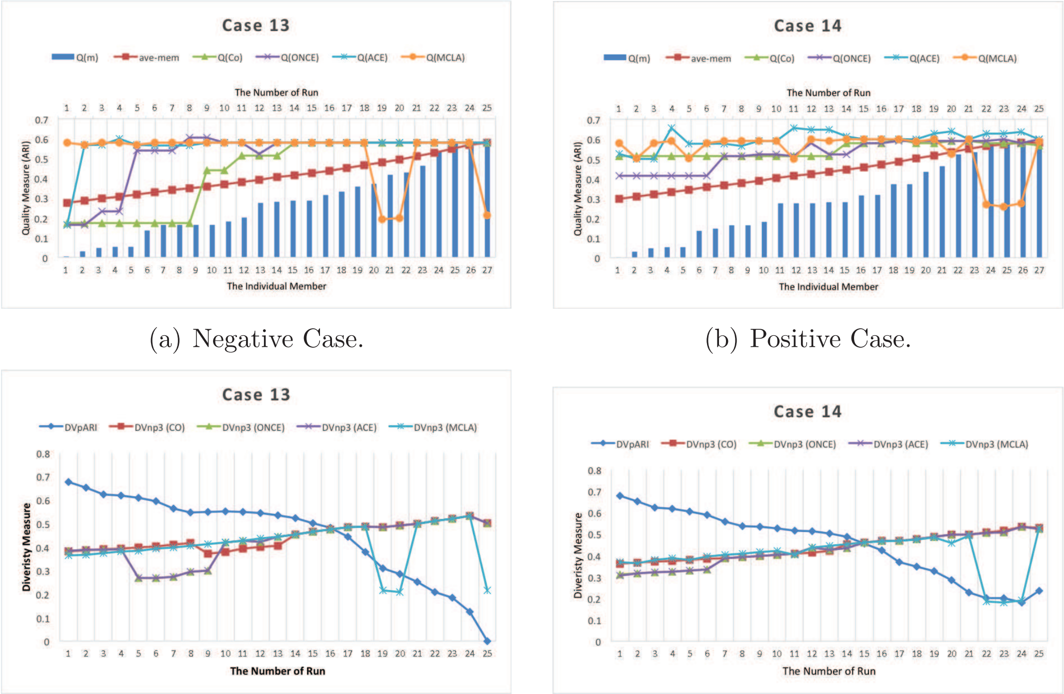
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.32: 25 ensemble runs for case 11 & 12, in each run one member is removed.



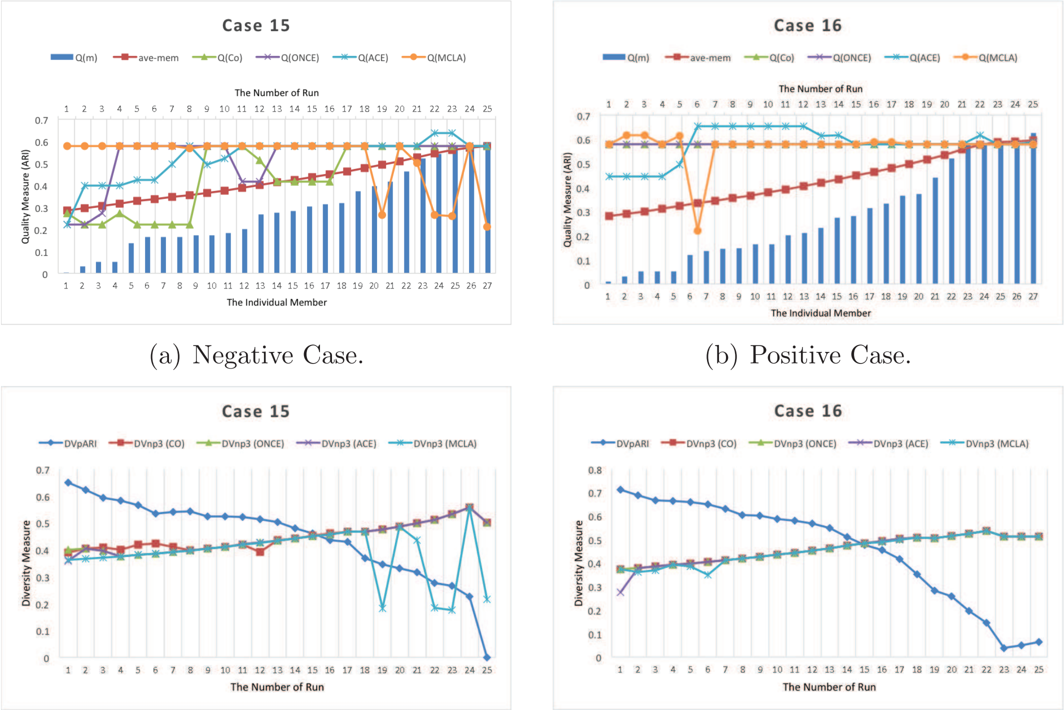
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.33: 25 ensemble runs for case 13 & 14, in each run one member is removed.



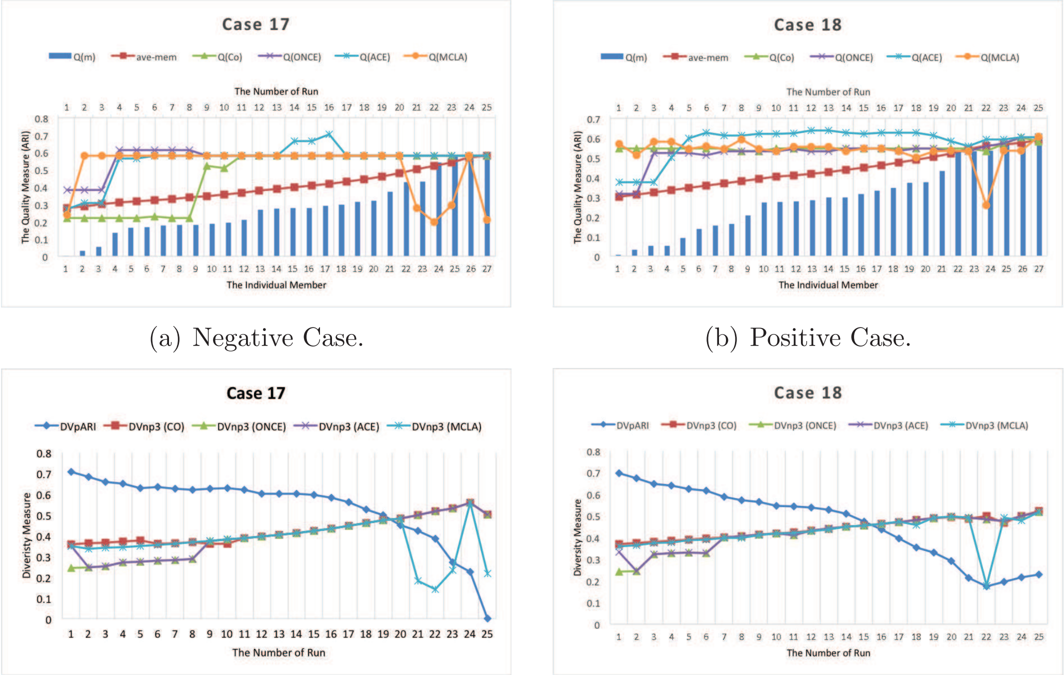
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.34: 25 ensemble runs for case 15 &16, in each run one member is removed.



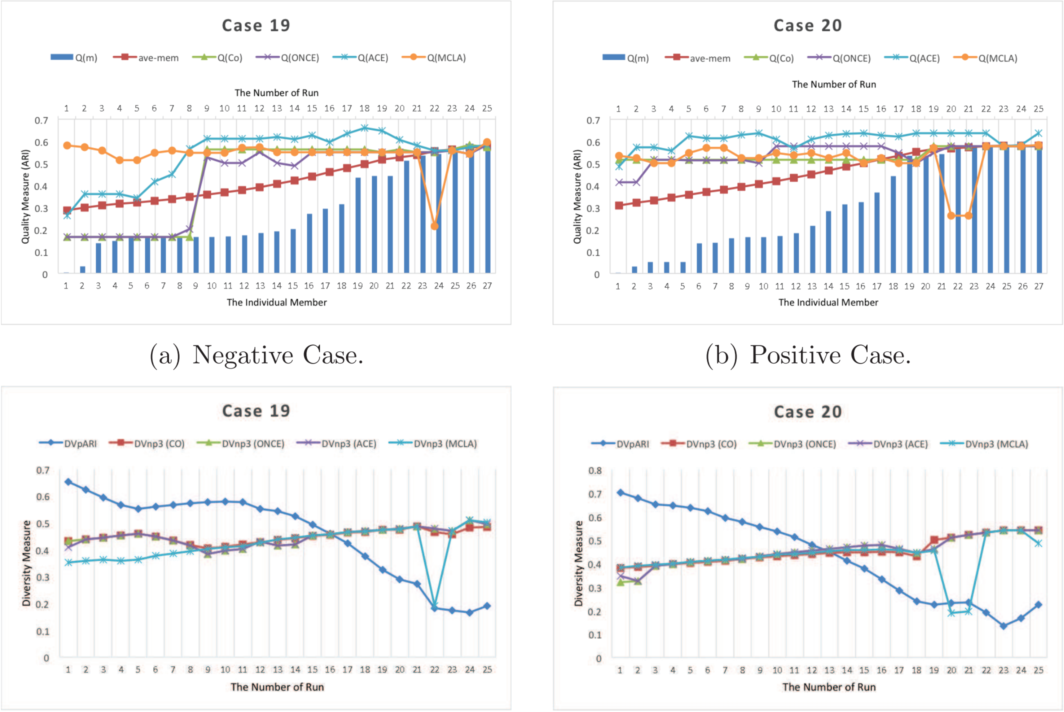
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.35: 25 ensemble runs for case 17 & 18, in each run one member is removed.



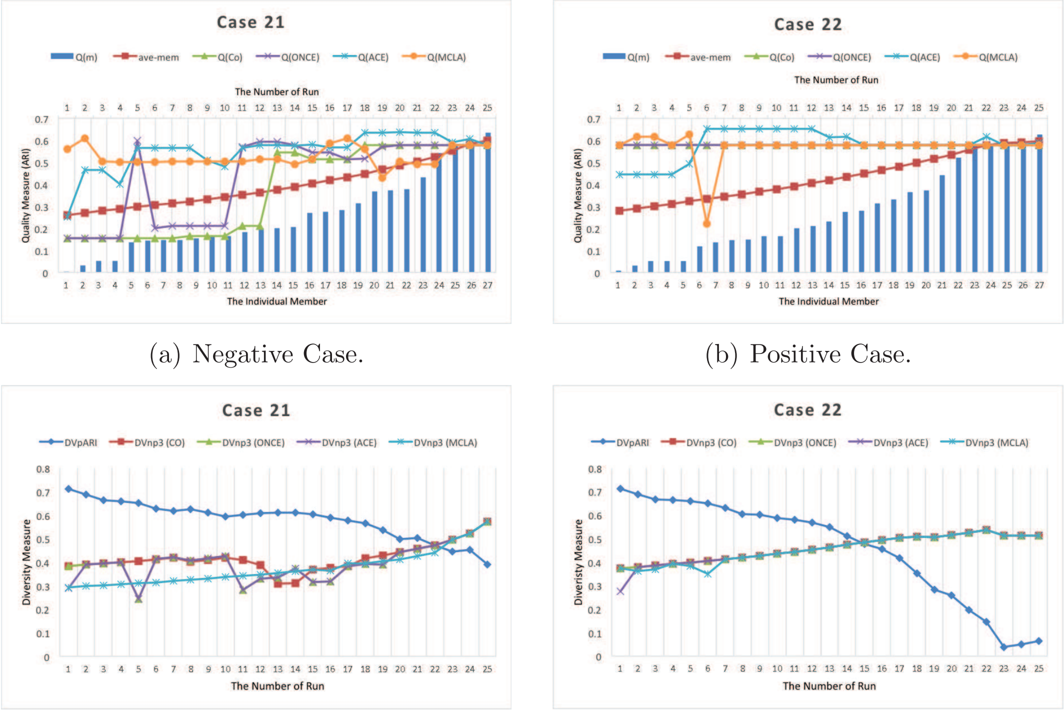
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

Figure B.36: 25 ensemble runs for case 19 & 20, in each run one member is removed



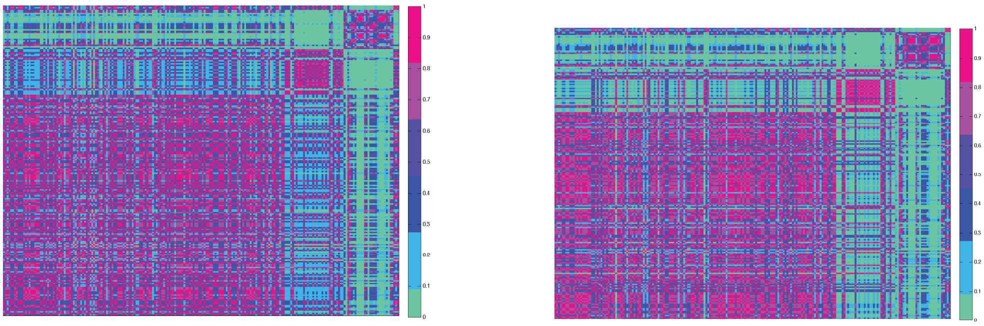
(a) Negative Case.

(b) Positive Case.

(c) Diversity Measures of the Negative Case.

(d) Diversity Measures of the Positive Case.

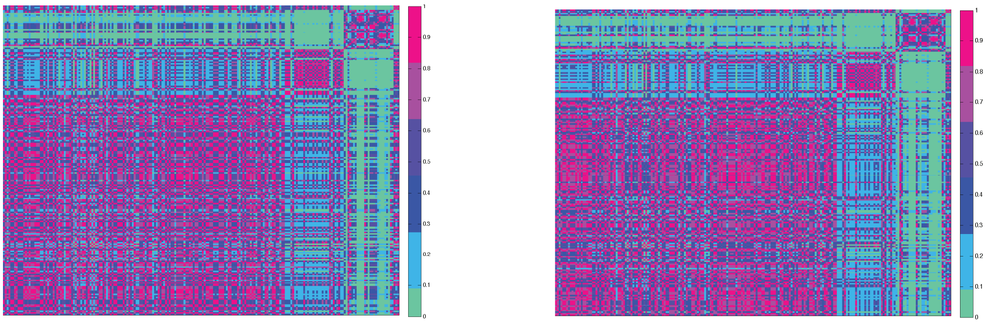
Figure B.37: 25 ensemble runs for case 21 & 22, in each run one member is removed



(a) Case 7

(b) Case 8

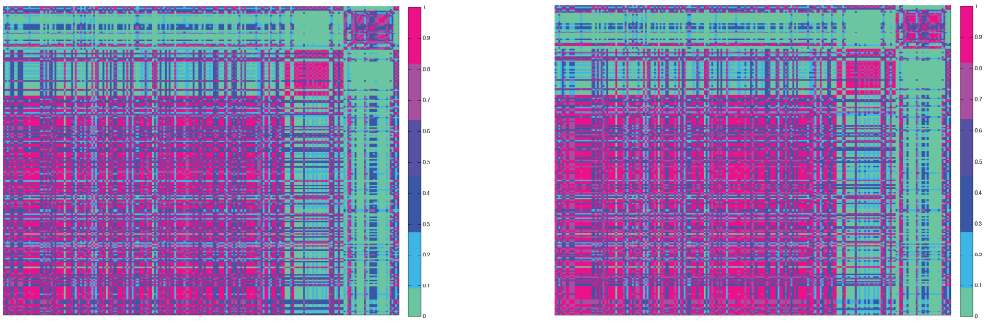
Figure B.38: The heat map of the CO similarity matrix for Run 7 at case 7 and 8.



(a) Run 7

(b) Run 11

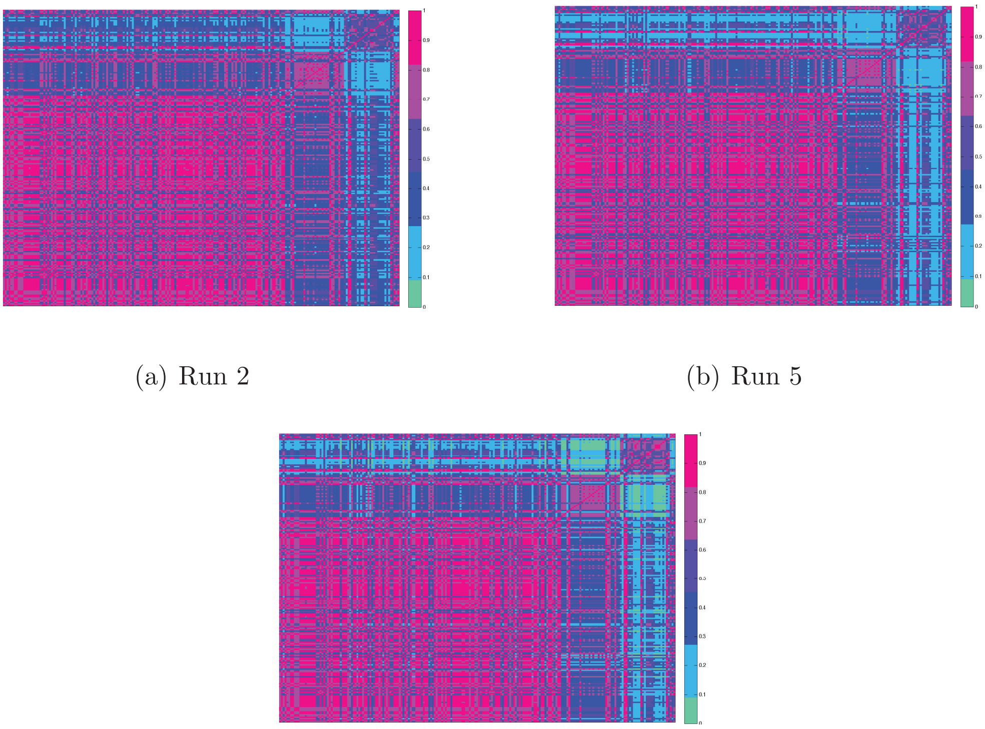
Figure B.39: The heat map of the CO similarity matrix for Run 7 and 11 at case 7.



(a) Run 7

(b) Run 10

Figure B.40: The heat map of the CO similarity matrix for Run 7 and 10 at case 15.



(a) Run 2

(b) Run 5

(c) Run 7

Figure B.41: The heat map of the ONCE similarity matrix for Run 2 and 7 at case 15.

Appendix C

Checking the ANOVA Assumptions

Before we conduct the ANOVA test in our experiment in Section 6.2.2.2, we have to check its assumptions, which is the normality and the homogeneity of variances.

For normality, we used the Anderson-Darling test [93], which is a statistical test used to test whether the data follows a specified distribution; in our case it is normal distribution. The Anderson-Darling test detected a violation of the normality assumptions (p-values always less than 0.05) for all the used consensus functions in both of the datasets.

For the homogeneity of variances, we used the Levene test [13], because it is more robust when the sampled data deviate from normality [71]. This tests the null hypothesis that the variances of all conditions are all equal, and it was found that we could reject the null hypotheses for all the used consensus functions in both of the datasets.

For the non-normal samples, Montgomery [75] recommends applying a Box-Cox transformation method [11] to the sample data to recover the normality and to obtain a constant variance (after transformation). The Box-Cox method is a parametric power transformation technique to estimate a value for the transformation parameter

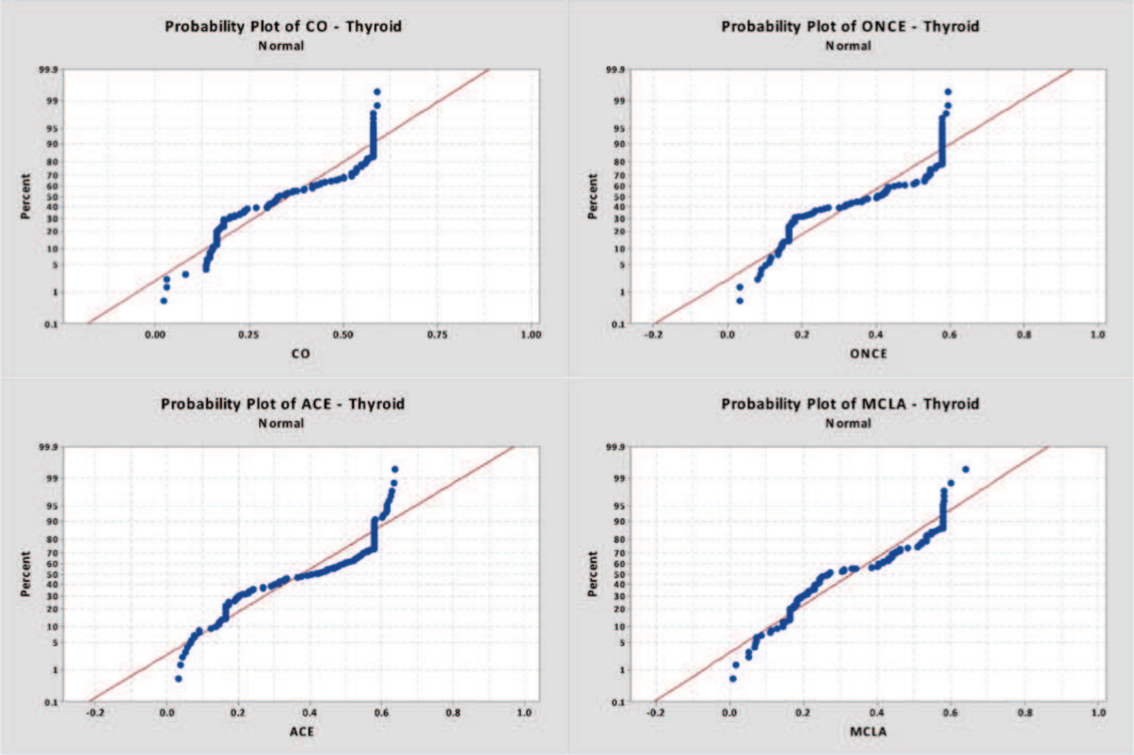


Figure C.1: The normal probability plot of the response variables (CO, ONCE, ACE and MCLA) for Thyroid dataset.

λ, and it can also suggest the best transformation function to be applied to the sample data. We applied the Box-Cox method, and it suggested that for most of the examined sample data there is no need to transform the sampled data. The only exceptions were the samples data of Wine using CO and ONCE and the estimated λ were 1.477 and 1.534 respectively. Figures C.1 and C.2 show the normal probability plot of response variables on the original sample data for the Thyroid and Wine datasets respectively.

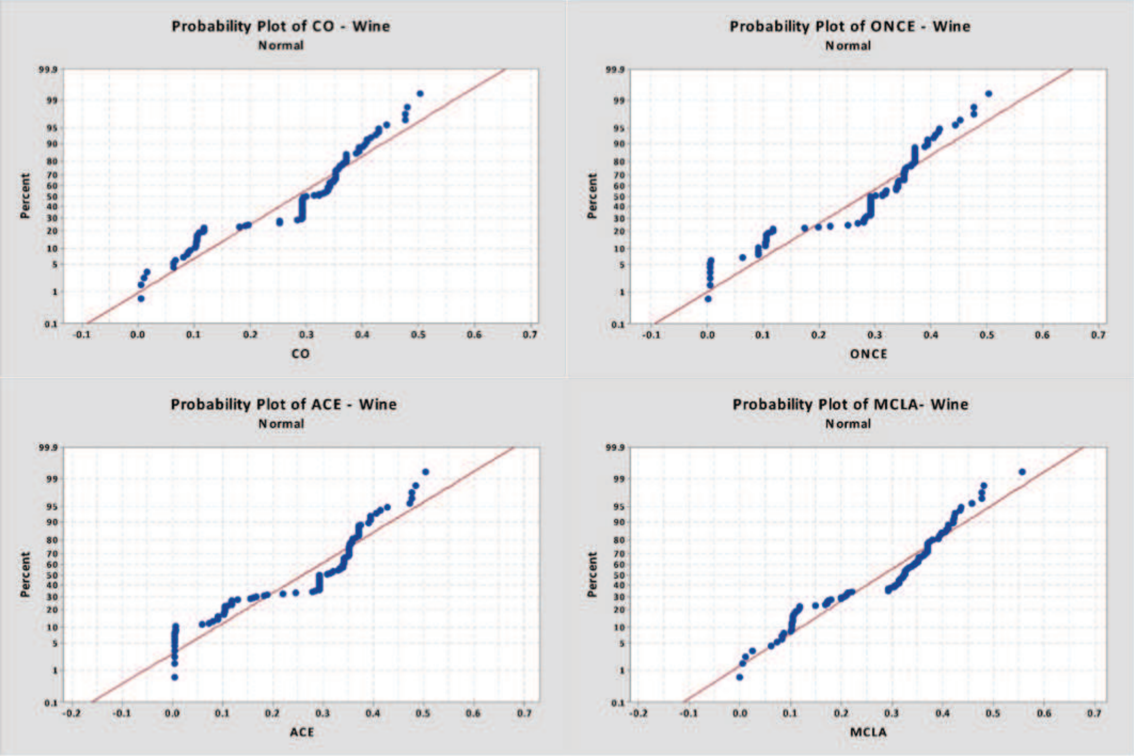


Figure C.2: The normal probability plot of the response variables (CO, ONCE, ACE and MCLA) for Wine dataset.

Bibliography

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[2] Muna Al-Razgan and Carlotta Domeniconi. Weighted clustering ensembles. In *Proceedings of the SIAM International Conference on Data Mining*, pages 258–269, 2006.

[3] Tahani Alqurashi and Wenjia Wang. Object-neighbourhood clustering ensemble method. In *Intelligent Data Engineering and Automated Learning (IDEAL)*, pages 142–149. Springer, 2014.

[4] Hanan G. Ayad. *Voting-Based Consensus of Data Partitions*. PhD thesis, University of Waterloo, Ontario, Canada, 2008.

[5] Hanan G Ayad and Mohamed S Kamel. Cluster-based cumulative ensembles. In *Multiple Classifier Systems*, pages 236–245. Springer, 2005.

[6] Hanan G Ayad and Mohamed S Kamel. On voting-based consensus of cluster ensembles. *Pattern Recognition*, 43(5):1943–1953, 2010.

[7] Javad Azimi and Xiaoli Fern. Adaptive cluster ensemble selection. In *The International Joint Conference on Artificial Intelligence (IJCAI)*, volume 9, pages 992–997, 2009.

[8] James C Bezdek. *Pattern recognition with fuzzy objective function algorithms*. Springer, 1981.

[9] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Occam's razor. *Information Processing Letters*, 24(6):377–380, 1987.

[10] George EP Box. Some theorems on quadratic forms applied in the study of analysis of variance problems, I. effect of inequality of variance in the one-way classification. *The Annals of Mathematical Statistics*, 25(2):290–302, 1954.

[11] George EP Box and David R Cox. An analysis of transformations. *Journal of the Royal Statistical Society*, 26(2):211–252, 1964.

[12] Gavin Brown, Jeremy Wyatt, Rachel Harris, and Xin Yao. Diversity creation methods: a survey and categorisation. *Information Fusion*, 6(1):5–20, 2005.

[13] Morton B Brown and Alan B Forsythe. Robust tests for the equality of variances. *Journal of the American Statistical Association*, 69(346):364–367, 1974.

[14] Arjun Chandra and Xin Yao. Divace: Diverse and accurate ensemble learning algorithm. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 619–625. Springer, 2004.

[15] Yuchou Chang, Dah-Jye Lee, Yi Hong, James Archibald, and Dong Liang. A robust color image quantization algorithm based on knowledge reuse of k-means clustering ensemble. *Journal of Multimedia*, 3(2):20–27, 2008.

[16] Lane M David. Online statistics education: A multimedia course of study. `http://onlinestatbook.com/`, 2008. Online; accessed 30 September 2015.

[17] David L Davies and Donald W Bouldin. A cluster separation measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1(2):224–227, 1979.

[18] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research*, 7:1–30, 2006.

[19] Evgenia Dimitriadou, Andreas Weingessel, and Kurt Hornik. A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(07):901–912, 2002.

[20] Carlotta Domeniconi and Muna Al-Razgan. Weighted cluster ensembles: Methods and analysis. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(4):17, 2009.

[21] Sandrine Dudoit and Jane Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[22] Joseph C Dunn. A fuzzy relative of the isodata process and its use in detecting compact well-separated clusters. *Journal of Cybernetics*, 3(3):32–57, 1973.

[23] Joseph C Dunn. Well-separated clusters and optimal fuzzy partitions. *Journal of Cybernetics*, 4(1):95–104, 1974.

[24] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.

[25] Xiaoli Z Fern and Carla E Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proceedings of the 20th International Conference on Machine Learning*, pages 186–193, 2003.

[26] Xiaoli Z Fern and Carla E Brodley. Solving cluster ensemble problems by bipartite graph partitioning. In *Proceedings of the 21st International Conference on Machine learning*, page 36. ACM, 2004.

[27] Xiaoli Z Fern and Carla E Brodley. Cluster ensembles for high dimensional clustering: an empirical study. Technical report, Oregon State University, Dept. of Computer Science, http://hdl.handle.net/1957/35655, 2006.

[28] Xiaoli Z Fern and Wei Lin. Cluster ensemble selection. *Statistical Analysis and Data Mining*, 1(3):787–797, 2008.

[29] András Frank. On Kuhn's Hungarian method a tribute from Hungary. *Naval Research Logistics (NRL)*, 52(1):2–5, 2005.

[30] Ana LN Fred. Finding consistent clusters in data partitions. In *Multiple classifier systems*, pages 309–318. Springer, 2001.

[31] Ana LN Fred and Anil K Jain. Data clustering using evidence accumulation. In *Proceedings of the16th International Conference on Pattern Recognition*, volume 4, pages 276–280. IEEE, 2002.

[32] 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.

[33] Milton Friedman. The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937.

[34] Joydeep Ghosh and Ayan Acharya. Cluster ensembles. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(4):305–315, 2011.

[35] Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):4, 2007.

[36] Derek Greene and Pádraig Cunningham. Efficient ensemble methods for document clustering. Technical report, Department of Computer Science, Trinity College Dublin, 2006.

[37] Derek Greene, Alexey Tsymbal, Nadia Bolshakova, and Padraig Cunningham. Ensemble clustering in medical diagnostics. In *Proceedings of the 17th IEEE Symposium on Computer-Based Medical Systems (CBMS)*, pages 576–581. IEEE, 2004.

[38] Saikat Guha, Rajeev Rastogi, and Kyuseok Shim. Rock: A robust clustering algorithm for categorical attributes. In *Proceedings of the 15th International Conference on Data Engineering.*, pages 512–521. IEEE, 1999.

[39] Stefan T Hadjitodorov, Ludmila I Kuncheva, and Ludmila P Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[40] Maria Halkidi, Yannis Batistakis, and Michalis Vazirgiannis. On clustering validation techniques. *Journal of Intelligent Information Systems*, 17(2-3):107–145, 2001.

[41] Maria Halkidi and Michalis Vazirgiannis. Clustering validity assessment: Finding the optimal partitioning of a data set. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 187–194. IEEE, 2001.

[42] Maria Halkidi, Michalis Vazirgiannis, and Yannis Batistakis. Quality scheme assessment in the clustering process. In *Proceedings of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD)*, pages 265–276, London, UK, 2000. Springer-Verlag.

[43] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining: Concepts and techniques*. Morgan Kaufmann, 2006.

[44] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.

[45] Prodip Hore, Lawrence O Hall, and Dmitry B Goldgof. A scalable framework for cluster ensembles. *Pattern Recognition*, 42(5):676–688, 2009.

[46] Michael E Houle. The relevant-set correlation model for data clustering. *Statistical Analysis and Data Mining*, 1(3):157–176, 2008.

[47] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2(1):193–218, 1985.

[48] Natthakan Iam-On, Tossapon Boongeon, Simon Garrett, and Chris Price. A link-based cluster ensemble approach for categorical data clustering. *IEEE Transactions on Knowledge and Data Engineering*, 24(3):413–425, 2012.

[49] Natthakan Iam-On, Tossapon Boongoen, and Simon Garrett. Refining pairwise similarity matrix for cluster ensemble problem with cluster relations. In *Discovery Science*, pages 222–233. Springer, 2008.

[50] Natthakan Iam-on, Tossapon Boongoen, and Simon Garrett. LCE: a link-based cluster ensemble method for improved gene expression data analysis. *Bioinformatics*, 26(12):1513–1519, 2010.

[51] Natthakan Iam-On, Tossapon Boongoen, Simon Garrett, and Chris Price. A link-based approach to the cluster ensemble problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2396–2409, 2011.

[52] Natthakan Iam-on, Simon Garrett, et al. LinkCluE: A MATLAB package for link-based cluster ensembles. *Journal of Statistical Software*, 36(9):1–36, 2010.

[53] Ronald L. Iman and James M. Davenport. Approximations of the critical region of the fbietkan statistic. *Communications in Statistics - Theory and Methods*, 9(6):571–595, 1980.

[54] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[55] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[56] Raymond A Jarvis and Edward A Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers*, 100(11):1025–1034, 1973.

[57] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[58] George Karypis and Vipin Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[59] Leonard Kaufman and Peter J Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. John Wiley & Sons, 2009.

[60] Ludmila I Kuncheva and Stefan Todorov Hadjitodorov. Using diversity in cluster ensembles. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 2, pages 1214–1219. IEEE, 2004.

[61] Ludmila I Kuncheva, Stefan Todorov Hadjitodorov, and Ludmila P Todorova. Experimental comparison of cluster ensemble methods. In *Proceedings of the 9th International Conference on Information Fusion*, pages 1–7. IEEE, 2006.

[62] Ludmila I Kuncheva and Christopher J Whitaker. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, 51(2):181–207, 2003.

[63] Ludmila I Kuncheva, Christopher J Whitaker, Catherine A Shipp, and Robert PW Duin. Limits on the majority vote accuracy in classifier fusion. *Pattern Analysis and Applications*, 6(1):22–31, 2003.

[64] Louisa Lam and Ching Y Suen. Application of majority voting to pattern recognition: an analysis of its behavior and performance. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 27(5):553–568, 1997.

[65] Tilman Lange, Mikio L Braun, Volker Roth, and Joachim M Buhmann. Stability-based model selection. In *Proceedings of the Advances in Neural Information Processing Systems*, pages 617–624, 2002.

[66] Tilman Lange, Volker Roth, Mikio L Braun, and Joachim M Buhmann. Stability-based validation of clustering solutions. *Neural computation*, 16(6):1299–1323, 2004.

[67] Tao Li, Chris Ding, Michael Jordan, et al. Solving consensus and semi-supervised clustering problems using nonnegative matrix factorization. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, pages 577–582. IEEE, 2007.

[68] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *ArXiv*, 2010.

[69] André Lourenço, Samuel Rota Bulò, Nicola Rebagliati, Ana LN Fred, Mário AT Figueiredo, and Marcello Pelillo. Probabilistic consensus clustering using evidence accumulation. *Machine Learning*, 98(1-2):331–357, 2015.

[70] Huilan Luo, Furong Jing, and Xiaobing Xie. Combining multiple clusterings using information theory based genetic algorithm. In *Proceedings of the International Conference on Computational Intelligence and Security*, volume 1, pages 84–89. IEEE, 2006.

[71] Paul G Mathews. *Design of Experiments with MINITAB*. ASQ Quality Press, 2005.

[72] Srujana Merugu and Joydeep Ghosh. A distributed learning framework for heterogeneous data sources. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pages 208–217. ACM, 2005.

[73] Selim Mimaroglu and Emin Aksehirli. DICLENS: Divisive clustering ensemble with automatic cluster number. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 9(2):408–420, 2012.

[74] Behrouz Minaei-Bidgoli, Alexander Topchy, and William F Punch. Ensembles of partitions via data resampling. In *Proceedings of the International Conference on Information Technology: Coding and Computing ITCC*, volume 2, pages 188–192. IEEE, 2004.

[75] Douglas C Montgomery. *Design and analysis of experiments*. John Wiley (New York), 1984.

[76] Stefano Monti, Pablo Tamayo, Jill Mesirov, and Todd Golub. Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning*, 52(1-2):91–118, 2003.

[77] Lichman Moshe. UCI machine learning repository. `http://archive.ics.uci.edu/m`, 2013.

[78] Weiliang Qiu and Harry Joe. Generation of random clusters with specified degree of separation. *Journal of Classification*, 23(2):315–334, 2006.

[79] Weiliang Qiu and Harry Joe. ClusterGeneration: random cluster generation (with specified degree of separation). `https://cran.r-project.org/web/packages/clusterGeneration/clusterGeneration.pdf`, 2009.

[80] William M Rand. Objective criteria for the evaluation of clustering methods. *Journal of the American Statistical Association*, 66(336):846–850, 1971.

[81] Yazhou Ren, Guoji Zhang, Carlotta Domeniconi, and Guoxian Yu. Weighted-object ensemble clustering. In *Proceedings of the IEEE 13th International Conference on Data Mining (ICDM)*, pages 627–636. IEEE, 2013.

[82] Eréndira Rendón, Itzel Abundez, Alejandra Arizmendi, and Elvia M Quiroz. Internal versus External cluster validation indexes. *International Journal of Computers and Communications*, 5(1):27–34, 2011.

[83] Dorota Rozmus. Analysis of diversity-accuracy relations in cluster ensemble. In *Classification as a Tool for Research*, pages 217–225. Springer, 2010.

[84] Faisal Saeed, Naomie Salim, Ammar Abdo, and Hamza Hentabli. Graph-based consensus clustering for combining multiple clusterings of chemical structures. *Molecular Informatics*, 32(2):165–178, 2013.

[85] Robert E Schapire. The strength of weak learnability. *Machine learning*, 5(2):197–227, 1990.

[86] Robert E Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. *Annals of statistics*, pages 1651–1686, 1998.

[87] Magne Setnes and Valerie Cross. Compatibility-based ranking of fuzzy numbers. In *Fuzzy Information Processing Society*, pages 305–310. IEEE, 1997.

[88] Xavier Sevillano, Germán Cobo, Francesc Alías, and Joan Claudi Socoró. Feature diversity in cluster ensembles for robust document clustering. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 697–698. ACM, 2006.

[89] Xavier Sevillano, Joan Claudi Socoró, and Francesc Alıas. Fuzzy clusterers combination by positional voting for robust document clustering. *Procesamiento del lenguaje natural*, 43:245–253, 2009.

[90] John J Shaughnessy and Eugene B Zechmeister. *Research methods in psychology.* Alfred A. Knopf, 1985.

[91] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis.* Cambridge University Press, 2004.

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

[93] Michael A Stephens. EDF statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.

[94] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[95] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Addison Wesley (Boston), 2006.

[96] E Ke Tang, Ponnuthurai N Suganthan, and Xin Yao. An analysis of diversity measures. *Machine Learning*, 65(1):247–271, 2006.

[97] Alexander Topchy, Anil K Jain, and William Punch. A mixture model of clustering ensembles. In *Proceedings of the SIAM International Conference of Data Mining*. Citeseer, 2004.

[98] 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.

[99] John. W. Tukey. The problem of multiple comparisons, mimeographed notes. (Note: This is a secondary citation. Many statistics texts cite this work, although the original monograph is no longer accessible), 1953.

[100] Cor J. Veenman, Marcel J. T. Reinders, and Eric Backer. A maximum variance cluster algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1273–1280, 2002.

[101] Sandro Vega-Pons, Jyrko Correa-Morris, and José Ruiz-Shulcloper. Weighted partition consensus via kernels. *Pattern Recognition*, 43(8):2712–2724, 2010.

[102] Sandro Vega-Pons and José Ruiz-Shulcloper. A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, 25(03):337–372, 2011.

[103] Sandro Vega-Pons, José Ruiz-Shulcloper, and Alejandro Guerra-Gandón. Weighted association based methods for the combination of heterogeneous partitions. *Pattern Recognition Letters*, 32(16):2163–2170, 2011.

[104] Nguyen X Vinh and Michael E Houle. A set correlation model for partitional clustering. In *Advances in Knowledge Discovery and Data Mining*, pages 4–15. Springer, 2010.

[105] Hongjun Wang, Hanhuai Shan, and Arindam Banerjee. Bayesian cluster ensembles. *The Journal of Statistical Analysis and Data Mining*, 4(1):54–70, 2011.

[106] Wenjia Wang. Some fundamental issues in ensemble methods. In *Proceedings of the IEEE International Joint Conference on Neural Networks.*, pages 2243–2250. IEEE World Congress on Computational Intelligence., 2008.

[107] Xi Wang, Chunyu Yang, and Jie Zhou. Clustering aggregation by probability accumulation. *Pattern Recognition*, 42(5):668–675, 2009.

[108] David H Wolpert and William G Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, 1997.

[109] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678, 2005.

[110] Ka Yee Yeung, David R. Haynor, and Walter L. Ruzzo. Validating clustering for gene expression data. *Bioinformatics*, 17(4):309–318, 2001.

[111] Jinfeng Yi, Tianbao Yang, Rong Jin, Anil K Jain, and Mehrdad Mahdavi. Robust ensemble clustering by matrix completion. In *Proceedings of the IEEE 12th International Conference on Data Mining (ICDM)*, pages 1176–1181. IEEE, 2012.

[112] Hye-Sung Yoon, Sun-Young Ahn, Sang-Ho Lee, Sung-Bum Cho, and Ju Han Kim. Heterogeneous clustering ensemble method for combining different cluster results. In *Data Mining for Biomedical Applications*, pages 82–92. Springer, 2006.

[113] Zhiwen Yu, Le Li, Hau-San Wong, Jane You, Guoqiang Han, Yunjun Gao, and Guoxian Yu. Probabilistic cluster structure ensemble. *Information Sciences*, 267:16–34, 2014.

[114] Zhi-Hua Zhou and Wei Tang. Clusterer ensemble. *Knowledge-Based Systems*, 19(1):77–83, 2006.