Diversity-based Weighting Schemes for Clustering Ensembles

Francesco Gullo * Andrea Tagarelli * Sergio Greco *

Abstract

Clustering ensembles has been recently recognized as an emerging approach to provide more robust solutions to the data clustering problem. Current methods of clustering ensembles typically fall into *instance-based*, *cluster-based*, or *hybrid* approaches; however, most of such methods fail in discriminating among the various clusterings that participate to the ensemble. In this paper, we address the problem of weighting clustering ensembles by proposing general weighting approaches based on different implementations of the notion of *diversity*. We introduce three weighting schemes for clustering ensembles, called *Single Weighting*, *Group Weighting* and *Dendrogram Weighting*, which are independent of the particular method of clustering ensembles and designed to take into account correlations among the individual clustering solutions in different ways. We show how these schemes can be instantiated into any instance-based, cluster-based and hybrid clustering ensembles methods. Experiments have shown that the performance of the clustering ensembles algorithms increases when the proposed weighting schemes are employed.

1 Introduction

Clustering ensembles, also known as *consensus clustering* or *aggregation clustering*, has recently emerged as a powerful tool to face traditional issues of the clustering problem. In particular, clustering ensembles aims to make a clustering solution more robust against the bias due to the peculiarities of the specific clustering algorithm.

Basically, clustering ensembles resorts to the idea of combining multiple classifiers, which has received increased attention in the last years [3]. Given a data collection, a set of clustering solutions, or *ensemble*, can be generated by varying one or more aspects, such as the clustering algorithm, the parameter setting, and the number of features, objects or clusters. Given a clustering ensembles, a major goal is to extract a *consensus partition*, i.e., a clustering solution that maximizes some objective function (the *consensus function*) defined by taking into account different information available from the given set of clustering solutions. Moreover, any clustering ensembles method should derive the consensus partition without accessing the original features of the objects in the data collection.

In recent years, various consensus functions have been defined (e.g., [25, 26, 1, 4, 30]) and coupled with heuristic algorithms for maximizing them (e.g., [17, 29, 2, 4, 12, 5, 34, 15, 14]). Heuristic clustering ensembles algorithms are commonly based on three main approaches, namely *instance-based clustering ensembles*, *cluster-based clustering ensembles*, and *hybrid clustering ensembles*. Instance-based clustering ensembles methods require a notion of distance measure that possibly employs information available from the ensemble to group the data objects. Cluster-based clustering ensembles is based on the principle "to cluster clusters", i.e., to apply a clustering algorithm on the set of all the clusters produced by the clustering solutions in the ensemble, in order to compute a set of *meta-clusters*; finally, the consensus partition is computed to assign each data object to the meta-cluster that maximizes some assignment criterion (e.g., majority voting). Hybrid clustering ensembles methods attempt to combine ideas coming from both instance-based and cluster-based approaches.

A common limit of all the above approaches is that most of the consensus functions proposed in literature are defined by *equally* considering the various clustering solutions in the ensemble. This is clearly a weak assumption for a number of reasons; for instance, an ensemble may be comprised of very different clusterings, as well as clusters that are somehow correlated with each other may appear in distinct clusterings of the ensemble. As a consequence, treating the constituent solutions of an ensemble equally and averaging over them to extract the consensus partition may not be effective.

In this paper we address the *weighted clustering ensembles* problem by leveraging the importance of employing weighting schemes to discriminate among the clustering solutions in an ensemble in extracting a proper consensus partition. Our research focuses on the development of general schemes for weighting clusterings which can be applied to any clustering ensembles method regardless of a specific approach. Our contributions can be summarized as follows:

*Dept. of Electronics, Computer and System Sciences, University of Calabria, Via P. Bucci 41/C, Arcavacata di Rende (CS) I87036, Italy. Email: {fgullo,tagarelli,greco}@deis.unical.it

1. We propose three weighting schemes, called *Single Weighting*, *Group Weighting*, and *Dendrogram Weighting*, which are designed to take into account correlations among the individual clustering solutions to different levels.

2. We describe how any instance-based, cluster-based, and hybrid clustering ensembles approach can be easily reformulated to include a weighting for the clustering solutions that participate to an ensemble.

3. We experimentally evaluated various state-of-the-art methods, for each one of the mentioned clustering ensembles approaches, with and without employing weighting schemes. Results have shown the beneficial impact of using a weighting scheme in improving the quality of the consensus partition from clustering ensembles.

The next section discusses the state-of-the-art in clustering ensembles. Section 3 provides basic definitions and notations used in this paper. Section 4 presents our proposal in details. Section 5 describes an experimental evaluation, finally Section 6 concludes the paper.

2 Related Work

Instance-based clustering ensembles. Most instance-based methods operate on the *co-occurrence* or *co-association* matrix ($\mathbf{M}$). For each pair of objects (x_i, x_j), this matrix stores the number of partitions of the ensemble in which x_i and x_j appear in the same cluster divided by the size of the ensemble. In the *Majority Voting* (MV) algorithm [14], $\mathbf{M}$ is "cut" at a given threshold θ, i.e., all the objects whose pairwise entry in $\mathbf{M}$ is greater than θ are joined into the same cluster. This approach has been proved to be equivalent to applying the Agglomerative Hierarchical Clustering algorithm (AHC) with single linkage on $\mathbf{M}$, cutting the resulting dendrogram according to θ [15].

Other algorithms are based on using $\mathbf{M}$ directly as a pair-wise distance matrix into a specific clustering algorithm. The *Agglomeration* (AGGL) algorithm [17] uses Expectation Maximization or AHC with average linkage, whereas the *Iterative Voting Consensus* (IVC) algorithm [29] uses k-Means. In [34], the AHC algorithm is applied to a pair-wise distance matrix derived from $\mathbf{M}$ by taking into account the statistical "signal" of the clusters in the ensemble.

In [30], clustering ensembles is mapped to a graph/hypergraph partitioning problem. The authors present two instance-based clustering ensembles methods, namely the *Cluster-based Similarity Partitioning Algorithm* (CSPA) and the *HyperGraph Partitioning Algorithm* (HPGA). CSPA induces a weighted graph from $\mathbf{M}$ and partitions it using the well-known graph partitioning algorithm METIS [23]. HPGA builds a hypergraph whose vertices are the data objects and the hyperedges are given by the clusters of all the clustering solutions in the ensemble; the consensus partition is then obtained by partitioning the hypergraph using HMETIS [22].

More recent graph-partitioning-based approaches are proposed in [2, 1]. In [2], the weight of each edge (x_i, x_j) in the induced graph is defined in terms of the size of the nearest neighbor list shared between the data objects x_i and x_j. In [1], the *Weighted Similarity Partitioning Algorithm* (WSPA) is proposed to combine multiple partitions that result from different runs of the projective clustering algorithm *Locally Adaptive Clustering* (LAC) [9].

Cluster-based clustering ensembles. The study in [5] proposes a two-stage clustering procedure. In the first stage, clustering solutions are obtained by multiple runs of the k-Means algorithm. Then, the output centroids from these clustering solutions are clustered by a further run of k-Means, and the resulting meta-centroids are used for the data assignment step.

The *Meta-CLustering Algorithm* (MCLA) [30] builds a graph whose vertices are the clusters of the various clustering solutions in the ensemble, and each edge (v_i, v_j) has a weight equal to the Jaccard similarity value between the clusters relatively associated to the vertices v_i and v_j. The set of meta-clusters is computed by applying METIS on the graph, whereas the objects are assigned to the meta-clusters according to a majority voting criterion.

In [4], a *MetaCluster Search* (MCS) algorithm is formulated as a linear optimization problem to compute the optimum set of meta-clusters. The inter-cluster similarity is defined in terms of the Jaccard coefficient, and the assignment of the objects to the meta-clusters is accomplished by majority voting.

Hybrid clustering ensembles. The *Hybrid Bipartite Graph Formulation* (HBGF) algorithm [12] builds a bipartite graph whose edges (v_i^O, v_j^C) have weights equal to 1, if the object v_i^O belongs to the cluster v_j^C according to the clustering $\mathcal{C} \supset v_j^C$; otherwise, the weights are equal to zero. The clustering ensembles result is obtained by partitioning the graph according to standard methods (e.g., METIS) or spectral graph partitioning algorithms (e.g., [28]).

The *Weighted Bipartite Partitioning Algorithm* (WBPA) [1] follows the same overall scheme of [12], although it allows for extending the range of weight values from $\{0, 1\}$ to $[0, 1]$.

Recently, there has been an increasing interest for some problems related to clustering ensembles; in particular, the *cluster ensemble selection* problem [6, 13] is to select a proper subset of solutions from an ensemble, and the *weighted consensus clustering* problem [25] is to automatically determine a proper weight for each solution in the ensemble. The key motivation for both problems arises from the fact that selecting a proper subset of clustering solutions (resp. assigning a proper weight to each clustering solution) allows for extracting a more accurate consensus partition than using the whole ensemble (resp. the unweighted version of the algorithm).

Our proposal belongs to the class of weighted consensus clustering problems. A major difference between the approach in [25] and ours is that the former proposes an optimization of an objective function which is derived from a specific formulation of the problem of clustering ensembles based on *Nonnegative Matrix Factorization* (NMF) [26]; by contrast, we do not focus on any specific formulation of the clustering ensembles problem, but rather we consider general properties of the ensemble focusing on the notion of diversity.

3 Background

Definition 1. (clustering solution) *Given a set of data objects D, a* clustering solution *(or* partition*) $\mathcal{C} = \{C_1, \ldots, C_k\}$ defined over D is a partition of D into k disjoint groups (clusters).*

Definition 2. (ensemble) *Given a set of data objects D, an* ensemble *is a set $E = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$, where $\mathcal{C}_i$ is a clustering solution defined over D, for each $i \in [1..m]$.*

Definition 3. (consensus partition) *Given a clustering ensemble E, a* consensus partition *derived from E is a clustering solution $\mathcal{C}^*_E$ that maximizes a given* consensus function *by exploiting information available from E.*

Building up an ensemble can be addressed by various ways, such as using different subsets of features [30, 18], using different clustering algorithms [33], varying one or more (random) parameters of the clustering algorithm [5, 11, 31], or using different datasets obtained, e.g., by re-sampling the original dataset [30, 10, 16, 27]. A crucial factor in the ensemble generation is the notion of *diversity*, which is used to quantify how the various clustering solutions in an ensemble are dissimilar to each other. This notion has been recognized as highly related to the accuracy of the consensus partition derived from an ensemble [11, 24, 20].

Definition 4. (partition-through diversity) *Given a clustering ensemble E, a* partition-through diversity *measure defined over E is a function $\delta_P : E \times E \rightarrow \Re$ that quantifies, for each pair of clustering solutions $\mathcal{C}_i, \mathcal{C}_j \in E$, how $\mathcal{C}_i$ and $\mathcal{C}_j$ are dissimilar to each other.*

In literature, partition-through diversity functions have been defined by resorting to external criteria used for assessing the quality of a clustering solution. One of the commonest criterion used for measuring partition-through diversity is based on *Normalized Mutual Information (NMI)* [7]. This measure has been involved into a lot of research focusing on clustering ensembles [11, 24, 20, 19, 8]. In this work, we also resort to *F-Measure (FM)* [32], which is another assessment criterion widely used in Information Retrieval and Machine Learning.

Given a set $D = \{x_1, \ldots, x_n\}$ of n data objects, let $\mathcal{A} = \{A_1, \ldots, A_h\}$ and $\mathcal{B} = \{B_1, \ldots, B_k\}$ be two clustering solutions defined over D. NMI is a symmetric measure to quantify the statistical information shared between two distributions. NMI can be used to express a sound indication of the degree of shared information between any pair of clustering solutions:

(3.1)
$$NMI(\mathcal{A}, \mathcal{B}) = \frac{\displaystyle\sum_{i=1}^{h}\sum_{j=1}^{k} |A_i \cap B_j| \log\left(\frac{n|A_i \cap B_j|}{|A_i||B_j|}\right)}{\sqrt{\left(\displaystyle\sum_{i=1}^{h} |A_i| \log\frac{|A_i|}{n}\right)\left(\displaystyle\sum_{j=1}^{k} |B_j| \log\frac{|B_j|}{n}\right)}}$$

F-Measure $FM(\mathcal{A}, \mathcal{B})$ is defined as the harmonic mean between two values that express the notions of precision (P) and recall (R), respectively:

$$P_{ij}(\mathcal{A}, \mathcal{B}) = \frac{|B_j \cap A_i|}{|B_j|}, \quad R_{ij}(\mathcal{A}, \mathcal{B}) = \frac{|B_j \cap A_i|}{|A_i|},$$

$$P(\mathcal{A}, \mathcal{B}) = \frac{1}{h}\sum_{i=1}^{k} \max_{j \in [1..k]} P_{ij}(\mathcal{A}, \mathcal{B}),$$

$$R(\mathcal{A}, \mathcal{B}) = \frac{1}{h}\sum_{i=1}^{k} \max_{j \in [1..k]} R_{ij}(\mathcal{A}, \mathcal{B}),$$

(3.2)
$$FM(\mathcal{A}, \mathcal{B}) = \frac{2P(\mathcal{A}, \mathcal{B})R(\mathcal{A}, \mathcal{B})}{P(\mathcal{A}, \mathcal{B}) + R(\mathcal{A}, \mathcal{B})}$$

NMI and FM are similarity measures that range within $[0, 1]$. The NMI- and FM-based partition-through diversity measures between two clustering solutions $\mathcal{A}$ and $\mathcal{B}$ can be defined as $1 - NMI(\mathcal{A}, \mathcal{B})$ and $1 - \sqrt{FM(\mathcal{A}, \mathcal{B}) \times FM(\mathcal{B}, \mathcal{A})}$, respectively. Note that the geometric mean is used in the definition of the FM-based partition-through diversity since FM is originally not symmetric, unlike NMI.

Starting from the notion of partition-through diversity, we provide the definitions of *clustering* and *ensemble* diversity notions used in this work.

DEFINITION 5. (CLUSTERING DIVERSITY) *Given a clustering ensemble $E = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ and a partition-through diversity measure δ_P defined over E, the* clustering diversity *measure is a function $\delta_{\mathcal{C}} : E \to \Re$ such that:*

$$\delta_{\mathcal{C}}(\mathcal{C}_i) = \frac{1}{m-1} \sum_{\substack{j \in [1..m], \\ j \neq i}} \delta_P(\mathcal{C}_i, \mathcal{C}_j), \quad i \in [1..m]$$

DEFINITION 6. (ENSEMBLE DIVERSITY) *Given a clustering ensemble $E = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ and a partition-through diversity measure δ_P defined over E, the* ensemble diversity *of E is defined as:*

$$\delta_E = \frac{2}{m(m-1)} \sum_{i=1}^{m-1} \sum_{j=i+1}^{m} \delta_P(\mathcal{C}_i, \mathcal{C}_j)$$

4 Weighting Clustering Ensembles

In this section we present three general weighting schemes for clustering ensembles, and we show how the proposed weighting schemes can be involved into any instance-based, cluster-based and hybrid clustering ensembles algorithm.

4.1 Clustering ensembles weighting schemes

Given a clustering ensemble $E = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$, we are interested in defining a vector of weights $W = (w_1, \ldots, w_m)$, in such a way that each component w_i in W is assigned to the clustering solution $\mathcal{C}_i$ and reflects the relevance of $\mathcal{C}_i$ in determining the consensus partition.

In principle, W can be defined by resorting to traditional criteria that assess the validity of a clustering solution. Unfortunately, this way is not applicable in this context, since neither external nor internal clustering validity criteria can be employed. Indeed, external criteria require prior knowledge of the ideal classification, whereas internal criteria can be used only if the original features of the clustered objects are available.

We propose three general schemes to compute the vector W, called *Single Weighting* (SW), *Group Weighting* (GW) and *Dendrogram Weighting* (DW). Each of these schemes is based on theoretical considerations on ensemble diversity and computes the vector $W = (w_1, \ldots, w_m)$ in such a way that $w_i \in [0, 1]$, for each $i \in [1..m]$, and $\sum_{i=1}^{m} w_i = 1$.

Single Weighting. The SW scheme takes into account each $\mathcal{C}_i \in E$ individually. The key idea consists in evaluating the ensemble diversity of $E \setminus \{\mathcal{C}_i\}$, $\delta_{E\setminus\{\mathcal{C}_i\}}$, and defining w_i proportionally to $\delta_{E\setminus\{\mathcal{C}_i\}}$.

Most research works focusing on clustering ensembles diversity suggest generating ensembles according to a *maximum diversity criterion*, which states that the higher the ensemble diversity, the better the accuracy of the consensus partition extracted from the ensemble [11, 24, 8]. This is an empirical assumption, which may not hold in general. Indeed, the study in [20] shows that, in some cases, ensembles generated by a *median diversity criterion* (i.e., ensembles that exhibit a moderate level of diversity) produce a more accurate consensus partition than ensembles having higher diversity.

We take into account both the above intuitions and define W in such a way that it follows a mixture density composed by

- a linearly increasing distribution, W', which defines weights according to a maximum diversity criterion,
- a Normal distribution, W'', which computes weights according to a median diversity criterion.

Precisely, we define W as:

$$W = \alpha\ W' + (1 - \alpha)\ W'' \quad (4.3)$$

$W' = (w'_1, \ldots, w'_m)$ is a vector whose component values w'_i linearly increase as the quantity $\delta_{E\setminus\{\mathcal{C}_i\}}$ increases. By contrast, the normally distributed components of $W'' = (w''_1, \ldots, w''_m)$ are defined in such a way that the maximum value in W'' corresponds to the clustering solution $\mathcal{C}_i$ having the median $\delta_{E\setminus\{\mathcal{C}_i\}}$. The importance of the two vectors W' and W'' in computing W is established by the user-defined parameter α (ranging within $[0, 1]$). Formally, each w'_i of W' ($i \in [1..m]$) is defined as:

$$w'_i = \frac{\delta_{E\setminus\{\mathcal{C}_i\}}}{\sum_{l=1}^{m} \delta_{E\setminus\{\mathcal{C}_l\}}} \quad (4.4)$$

and each w''_i of W'' ($i \in [1..m]$) is defined as:

$$w''_i = \frac{N_{\mu,\sigma}(\delta_{E\setminus\{\mathcal{C}_i\}})}{\sum_{l=1}^{m} N_{\mu,\sigma}(\delta_{E\setminus\{\mathcal{C}_l\}})} \quad (4.5)$$

where $N_{\mu,\sigma}$ is the Normal probability density function having mean μ and standard deviation σ, i.e.,

$$N_{\mu,\sigma}(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Let us denote with Δ the set $\{\delta_{E\setminus\{\mathcal{C}\}} \mid \mathcal{C} \in E\}$, and with $\overline{d}$, d_{min}, d_{max} the median, the minimum and the maximum value in Δ. We define $\mu = \overline{d}$ and $\sigma = \hat{d}/3$, where $\hat{d} = \max_{d\in\{d_{min},d_{max}\}}|\overline{d} - d|$. This choice of σ guarantees that the condition $\int_{\overline{d}-\hat{d}}^{\overline{d}+\hat{d}} N_{\mu,\sigma}(x)\mathrm{d}x \approx 1$ holds.

Group Weighting. The SW scheme leads to the construction of the vector W by considering each clustering solution individually. However, an ensemble may not contain only solutions that are totally dissimilar to each other; instead, in a real scenario, an ensemble comprises a number of subsets of (highly) correlated clusterings, which tend to bias the consensus partition.

Within this view, an intuitive refinement of SW can work as follows. The subsets of correlated clusterings are initially detected, then a *macro*-weight is preliminarily assigned to each of these subsets to quantify the importance of the whole corresponding group of clusterings. Based on the macro-weight assigned to the specific subset, a *micro*-weight is finally computed for each clustering of that subset.

The aforementioned idea is at the basis of the proposed GW scheme, whose outline can be summarized as follows:

1: partition the ensemble E into a set of clusters (of clusterings) $\mathbf{C} = \{\mathrm{C}_1, \ldots, \mathrm{C}_k\}$
2: compute the weight vector $W_{\mathbf{C}} = (w_{\mathbf{C}}^{(1)}, \ldots, w_{\mathbf{C}}^{(k)})$, where each $w_{\mathbf{C}}^{(l)}$, $l \in [1..k]$, is assigned to the cluster (of clusterings) $\mathrm{C}_l \in \mathbf{C}$
3: compute the weight vector $W = (w_1, \ldots, w_m)$ from $W_{\mathbf{C}}$, in which each w_i, $i \in [1..m]$, is assigned to the clustering solution $\mathcal{C}_i \in E$

As shown in the outline, the task of detecting the subsets of correlated clusterings is accomplished by *clustering the clustering solutions*. This idea is not new in the context of clustering ensemble, since it has been previously involved into the cluster ensemble selection problem [6, 13]. However, in this work we bring out for the first time this idea for solving the weighted consensus clustering problem.

Once the "to cluster clusterings" step has been performed, the vector $W_{\mathbf{C}}$ is computed in a way similar to the SW scheme. The only difference is that GW considers the ensemble diversity of the sets obtained by subtracting the clustering solutions in the various clusters from the whole ensemble; by contrast, SW takes into account the diversity of the ensemble when a single clustering solution is subtracted from it. Formally, we compute $W_{\mathbf{C}}$ as:

$$W_{\mathbf{C}} = \alpha\ W'_{\mathbf{C}} + (1-\alpha)\ W''_{\mathbf{C}} \tag{4.6}$$

where each $w_{\mathbf{C}}^{(l)'}$ of $W'_{\mathbf{C}}$ ($l \in [1..k]$) is defined as:

$$w_{\mathbf{C}}^{(l)'} = \frac{\delta_{E\setminus\mathrm{C}_l}}{\sum_{u=1}^{k} \delta_{E\setminus\mathrm{C}_u}} \tag{4.7}$$

and each $w_{\mathbf{C}}^{(l)''}$ of $W''_{\mathbf{C}}$ ($l \in [1..k]$) is defined as:

$$w_{\mathbf{C}}^{(l)''} = \frac{N_{\mu,\sigma}(\delta_{E\setminus\mathrm{C}_l})}{\sum_{u=1}^{k} N_{\mu,\sigma}(\delta_{E\setminus\mathrm{C}_u})} \tag{4.8}$$

From $W_{\mathbf{C}}$, we compute the vector of *micro*-weights W, which is the output of the GW scheme. Each w_i of W ($i \in [1..m]$) is defined as:

$$w_i = w_{i,j}^{SW} \times \overline{w}_j \tag{4.9}$$

where

- $w_{i,j}^{SW}$ is the weight assigned to the clustering solution $\mathcal{C}_i$ according to the SW scheme, when the ensemble is given by $\mathrm{C}_j \in \mathbf{C}$. C_j is the cluster such that $\mathcal{C}_i \in \mathrm{C}_j$,
- $\overline{w}_j \in W_{\mathbf{C}}$ is the weight assigned to C_j in the first step of GW.

Dendrogram Weighting. A major issue in the GW scheme is the requirement of a clustering algorithm to partition the ensemble and its relative parameter settings, such as the number of output clusters. To this purpose, we define a further weighting scheme, named Dendrogram Weighting (DW), to maintain the advantageous features of the GW scheme while overcoming the problem of choosing a clustering algorithm. The DW scheme is based on theoretical considerations on the dendrogram which can be built over the clustering solutions in the ensemble. In particular, DW consists of two main steps. First, the clusterings in the ensemble are clustered by using a hierarchical algorithm in order to organize them into a dendrogram. Then, the dendrogram is used as an intuitive tool for understanding relationships among the clusterings; this information is eventually exploited for properly defining the clustering weights.

The dendrogram required by the first step of DW can be defined as follows.

DEFINITION 7. (DENDROGRAM) *[21] Given a set of data objects D, a* dendrogram *T defined over D is a set of cluster pairs $T = \{P_1, \ldots, P_t\}$, where each $P_r = (\mathrm{C}_{r_1}, \mathrm{C}_{r_2})$, $r \in [1..t]$, and:*

1. $\mathrm{C}_{r_1}, \mathrm{C}_{r_2} \in 2^D$*, for each* $P_r \in T$
2. $\mathrm{C}_{r_2} \subset \mathrm{C}_{r_1}$*, for each* $P_r \in T$

3. $\mathrm{C}_{r_2} \cap \mathrm{C}_{s_2} = \emptyset$ *and* $\mathrm{C}_{r_1} \supseteq \mathrm{C}_{r_2} \cup \mathrm{C}_{s_2}$*, for each* $P_r, P_s \in T$ *such that* $\mathrm{C}_{r_1} = \mathrm{C}_{s_1}$*.*

A dendrogram T, as defined in Def. 7, can be also organized in levels.

Definition 8. (level-organized dendrogram) *Let T be a dendrogram defined over a set of data objects D. A* level-organized dendrogram *derived from T is a set $\mathcal{D} = \{\mathcal{L}_0, \ldots, \mathcal{L}_\tau\}$, where each $\mathcal{L}_u$, $u \in [0..\tau]$, is a set of clusters $\{\mathrm{C}_1^{(u)}, \ldots, \mathrm{C}_{k_u}^{(u)}\}$ corresponding to the level u, such that:*

1. $\bigcup_{v=1}^{k_u} \mathrm{C}_v^{(u)} = D$
2. $\mathrm{C}_v^{(u)} \cap \mathrm{C}_w^{(u)} = \emptyset, \forall \mathrm{C}_v^{(u)}, \mathrm{C}_w^{(u)} \in \mathcal{L}_u$
3. $|\mathcal{L}_0| = |D|$, $|\mathcal{L}_\tau| = 1$, $|\mathcal{L}_u| > |\mathcal{L}_{u+1}|, u \in [0..\tau-1]$

Hereinafter we refer to a dendrogram as a level-organized dendrogram.

Once a dendrogram has been defined over the set of clustering solutions E (i.e., the ensemble), the weight vector W is finally computed by associating each clustering solution $\mathcal{C}_i \in E$ with a coefficient γ_i. This coefficient expresses the correlation of $\mathcal{C}_i$ with the other clusterings in the ensemble, based on the set S_i (i.e., the set of different clusters of the dendrogram that contain $\mathcal{C}_i$). Precisely, γ_i is defined as inversely proportional to the size of S_i and directly proportional to the sum of the dendrogram levels that contain the clusters in S_i:

$$\gamma_i = \frac{\tau(\tau-1)}{2} - \sum_{h=1}^{\tau-1} (\tau - h) I(\mathcal{D}, \mathcal{C}_i, \mathcal{L}_h) \quad (4.10)$$

where $I(\mathcal{D}, \mathcal{C}_i, \mathcal{L}_h)$ is an indicator function that returns 1 if there exists some "new" cluster at the level $\mathcal{L}_h$ of $\mathcal{D}$ that contains $\mathcal{C}_i$, otherwise the function returns 0. Formally, let $\overline{\mathrm{C}}_h \in \mathcal{L}_h$ and $\overline{\mathrm{C}}_{h-1} \in \mathcal{L}_{h-1}$ be the clusters such that $\mathcal{C}_i \in \overline{\mathrm{C}}_h$ and $\mathcal{C}_i \in \overline{\mathrm{C}}_{h-1}$:

$$I(\mathcal{D}, \mathcal{C}_i, \mathcal{L}_h) = \begin{cases} 1 & \text{if } \overline{\mathrm{C}}_h \neq \overline{\mathrm{C}}_{h-1} \\ 0 & \text{otherwise} \end{cases}$$

The intuition underlying γ_i can be explained as follows. If the clustering $\mathcal{C}_i$ belongs to a large number of different clusters in the dendrogram, then $\mathcal{C}_i$ is expected to be correlated with a large number of other clusterings in the ensemble; therefore, γ_i should be low. On the other hand, the higher the dendrogram level of the clusters containing $\mathcal{C}_i$, the lower the correlation of $\mathcal{C}_i$ with the other clusterings in the ensemble; indeed, a high dendrogram level means that the corresponding clusters are less compact than the clusters formed at the lower levels.

Algorithm 1 WICE: Weighted Instance-based Clustering Ensembles

Input: a set of data objects $D = \{x_1, \ldots, x_n\}$, where $x_j = (f_{j1}, \ldots, f_{jp})$, $j \in [1..n]$;
an ensemble $E = \{\mathcal{C}_1, \ldots, \mathcal{C}_m\}$ defined over D;
a weight vector $W = (w_1, \ldots, w_m)$
Output: the consensus partition $\mathcal{C}_E^*$

```
1: for all j ∈ [1..n] do
2:     replace x_j with x'_j = (f'_j1, ..., f'_jm)
3: end for
4: for all a ∈ [1..n], b ∈ [1..n] do
5:     (M')_ab = Φ(w_1 φ(f'_a1, f'_b1), ..., w_m φ(f'_am, f'_bm),
                 Γ(x'_a, x'_b))
6: end for
7: C*_E ← cluster(D, M')
8: return C*_E
```

In order to define the final weight vector W, we resort to a similar approach used for the SW scheme. In particular, we employ the same equations used for SW (Eq. (4.3)-(4.5)), where $\delta_{E \setminus \{\mathcal{C}_i\}}$ is replaced with the coefficients γ_i, $i \in [1..m]$.

Computational aspects. Given an ensemble of size m, the computational complexity of the proposed weighting schemes is the following.

- *Single Weighting* performs in $\mathcal{O}(m^2)$
- *Group Weighting* performs in $\mathcal{O}(\max\{A, km^2\})$, where A is the execution cost required by the "to cluster clusterings" step, and k is the number of output clusters of clusterings. If we assume that k is a constant and A is $\mathcal{O}(m^2)$, then the complexity of the GW scheme is $\mathcal{O}(m^2)$
- *Dendrogram Weighting* performs in $\mathcal{O}(m^2)$

4.2 Involving weights in clustering ensembles algorithms

In this section we provide a formulation of the instance-based, cluster-based and hybrid clustering ensembles methods which takes into account weights for the clustering solutions in the ensembles.

Weighted instance-based clustering ensembles. Algorithm 1 outlines the general scheme of a weighted instance-based clustering ensembles method (WICE). Initially, each data object $x_j \in D$ is replaced with a new one x'_j which is defined over the space of features according to the information stored in E (Lines 1-3).

Algorithm 2 WCCE: Weighted Cluster-based Clustering Ensembles

Input: a set of data objects $D = \{x_1, \dots, x_n\}$;
an ensemble $E = \{\mathcal{C}_1, \dots, \mathcal{C}_m\}$ defined over D;
a weight vector $W = (w_1, \dots, w_m)$
Output: the consensus partition $\mathcal{C}^*_E$

1: $\mathcal{C}^*_E = \{\mathrm{C}^*_1, \dots, \mathrm{C}^*_k\} \leftarrow \{\emptyset, \dots, \emptyset\}$
2: $D_M \leftarrow \bigcup_{\mathcal{C} \in E} \mathcal{C}$
3: $\mathbf{M}_{D_M} \leftarrow \mathit{pair\text{–}wise\text{–}distances}(D_M)$
4: $M = \{M_1, \dots, M_k\} \leftarrow \mathit{cluster}(D_M, \mathbf{M}_{D_M})$
5: **for all** $j \in [1..n]$ **do**
6: find $M_l \in M$ such that $M_l \leftarrow \mathit{assign}(x_j, M, W)$
7: $\mathrm{C}^*_l \leftarrow \mathrm{C}^*_l \cup \{x_j\}$
8: **end for**
9: **return** $\mathcal{C}^*_E$

Algorithm 3 WHCE: Weighted Hybrid Clustering Ensembles

Input: a set of data objects $D = \{x_1, \dots, x_n\}$;
an ensemble $E = \{\mathcal{C}_1, \dots, \mathcal{C}_m\}$ defined over D;
a weight vector $W = (w_1, \dots, w_m)$
Output: the consensus partition $\mathcal{C}^*_E$

1: $\mathcal{V}_o \leftarrow D$
2: $\mathcal{V}_c \leftarrow \bigcup_{\mathcal{C} \in E} \mathcal{C}$
3: $\mathcal{E} \leftarrow \emptyset$
4: **for all** $v_o \in \mathcal{V}_o$ **do**
5: **for all** $v_c \in \mathcal{V}_c$ **do**
6: $\omega = \mathit{weight}(v_o, v_c, E, W)$
7: $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_o, v_c, \omega)\}$
8: **end for**
9: **end for**
10: $G_H \leftarrow \langle \mathcal{V}_o \cup \mathcal{V}_c, \mathcal{E} \rangle$
11: $\mathcal{C}^*_E \leftarrow \mathit{partition}(G_H)$
12: **return** $\mathcal{C}^*_E$

Each feature f'_{ji} of x'_j ($i \in [1..m]$) is defined according to the clustering solution $\mathcal{C}_i \in E$ and depends on the specific instance-based algorithm. Once the objects x'_j have been defined, the matrix $\mathbf{M}'$ storing the (weighted) pair-wise distances for the data objects is computed (Lines 4-6). Each entry $(\mathbf{M}')_{ab}$ is computed in terms of the functions $\Phi : \Re^{m+1} \to \Re$, $\phi : \Re^2 \to \Re$, and $\Gamma : \Re^{2m} \to \Re$. Note that Φ, ϕ and Γ are properly defined depending on the specific instance-based algorithm. The entries in $\mathbf{M}'$ should give more weight to the information coming from clusterings whose associated weights are higher. Finally, the output consensus partition $\mathcal{C}^*_E$ is computed by performing a further clustering task on the objects in D, where $\mathbf{M}'$ is used as the pair-wise distance matrix (Line 7).

According to Algorithm 1, any instance-based method provides a specific way of computing the objects x'_j, and the functions Φ, ϕ and Γ. As an example, a clustering ensembles algorithm using a co-occurrence matrix with Euclidean distance values (e.g., [17, 29]) should be equipped with $x'_j = (\lambda_1(x_j), \dots, \lambda_m(x_j))$, where each $\lambda_i(x)$ returns the identifier of the cluster in $\mathcal{C}_i$ that contains x, whereas $\Phi(y_1, \dots, y_{m+1}) = \sqrt{y_1 + \cdots + y_m}$, $\phi(y_1, y_2) = (y_1 - y_2)^2$, and $\Gamma(y_1, \dots, y_{2m}) = 0$.

Weighted cluster-based clustering ensembles. The weighted cluster-based clustering ensembles algorithm, WCCE, consists of two main phases (Algorithm 2). First, a preliminary task of clustering is performed over the union set D_M of all the clusters belonging to the clustering solutions in E, in order to obtain a set M of *meta-clusters* (Lines 2-4). The clustering procedure involves $\mathbf{M_{D_M}}$ as a matrix storing the distances between the pairs of clusters in D_M (Line 3). $\mathbf{M_{D_M}}$ is properly defined according to the specific cluster-based algorithm (e.g., by using the Jaccard coefficient). Then, the output consensus partition is derived by assigning each $x_j \in D$ to one and only one meta-cluster in M (Lines 5-8) based on: *i)* some criterion of object-to-meta-cluster assignment (which depends on the specific cluster-based method) and *ii)* the weight vector W defined over the ensemble.

Most cluster-based algorithms adopt the so-called *majority voting* [30, 4] as an object-to-meta-cluster assignment criterion. Precisely, each $x_j \in D$ is assigned to the meta-cluster $\overline{M}_j = \arg\max_{M_l \in M} \sum_{\mathrm{C}' \in M_l} I[x_j \in \mathrm{C}']$. The function $I[A]$ is equal to 1 when the event A occurs, otherwise 0.

A weighted version of the majority voting criterion can be easily derived inasmuch as each $x_j \in D$ is assigned to the meta-cluster $\overline{M}_j = \arg\max_{M_l \in M} \sum_{\mathrm{C}' \in M_l} w' I[x_j \in \mathrm{C}']$, where w' is the weight associated to the clustering $\mathcal{C} \in E$ such that $\mathrm{C}' \in \mathcal{C}$.

Weighted hybrid clustering ensembles. Any hybrid clustering ensembles method exploits information coming from both instance-based and cluster-based approaches, and can be described by the outline reported in Algorithm 3.

Initially, a *hybrid* bipartite graph G_H is built (Lines 1-9). The vertex set of G_H contains both the data objects in D (the set $\mathcal{V}_o$) and the clusters of each clustering solution in E (the set $\mathcal{V}_c$) (Lines 1-2). The weighted edge set $\mathcal{E}$ is comprised of links between vertices in $\mathcal{V}_o$ and vertices in $\mathcal{V}_c$, whereas the weight of each edge is defined according to the specific hybrid algorithm and takes into account the weight vector W (Lines 3-8). The weights in W are used to enhance the edge weights; precisely, given any two nodes $v_1 \in \mathcal{V}_o$, $v_2 \in \mathcal{V}_c$ and the corresponding edge $e = (v_1, v_2, \omega)$, the associated weight ω is multiplied to $(1 + w_i)$, where w_i in W is the weight associated to the clustering $\mathcal{C}_i$ such that $v_2 \in \mathcal{C}_i$. Finally, the output consensus partition is derived by partitioning G_H by means of a suitable procedure (e.g., METIS [23]) that depends on the specific hybrid algorithm.

5 Experiments

We devised an experimental evaluation in order to assess the impact of employing the proposed weighting schemes in clustering ensembles. To this purpose, we evaluated and compared the performances of instance-based, cluster-based and hybrid clustering ensemble algorithms with and without each weighting scheme. Specifically, in the experiments we involved the following clustering algorithms which have been discussed in Section 2:

- CSPA [30], HPGA [30], WSPA [1], MV [14], AGGL [17], and IVC [29], as instance-based methods;
- MCLA [30] and MCS [4], as cluster-based methods;
- HBGF [12] and WBPA [1], as hybrid methods.

Note that, in case of weighted clustering ensembles, we adapted each of the selected clustering methods according to the algorithm schemes presented in Section 4.2.

In the following, we discuss the evaluation methodology used in this work, which includes: *(i)* a description of the selected datasets, *(ii)* the strategy used for generating the ensemble, *(iii)* the setups of the proposed weighting schemes and of *(iv)* the various clustering ensembles methods, *(v)* the measures to assess the quality of the consensus partition derived from the ensemble. Finally, we present the main experimental results obtained on the various datasets.

5.1 Evaluation methodology

Datasets. We used five benchmark datasets from the UCI Machine Learning Repository,[1] namely Glass, Ecoli, ImageSegmentation, ISOLET, and LetterRecognition. Such datasets have been involved in several research works focusing on clustering ensembles such as, e.g., [25, 13, 26, 8, 1, 6, 12, 30]. In addition to UCI datasets, we used two time-series datasets coming from different application domains,[2] namely Tracedata and ControlChart.

Table 1: Datasets used in the experiments

dataset	*objects*	*attributes*	*classes*
Glass	214	10	7
Ecoli	327	7	5
ImageSegmentation	2,310	19	7
ISOLET	1,800	617	6
LetterRecognition	7,000	16	10
Tracedata	200	275	4
ControlChart	600	60	6

Table 1 reports on the main characteristics of the selected datasets. Glass contains glass instances which are described by their chemical components. Ecoli contains data on the Escherichia Coli bacterium, which are identified with values coming from different analysis techniques. ImageSegmentation contains objects that were drawn from a database of seven outdoor images randomly; the images (3x3 regions) were hand-segmented to create a classification for each pixel. ISOLET contains objects representing letters of the alphabet spoken by certain subjects; we selected a subset of objects representing the letters A, B, C, D, E, and G. LetterRecognition contains character images corresponding to the capital letters in the English alphabet; we selected a subset of 700 objects for each letter from A to J. Tracedata simulates signals representing instrumentation failures. ControlChart contains synthetically generated control charts that are classified into one of the following: normal, cyclic, increasing trend, decreasing trend, upward shift, and downward shift.

Ensemble generation. To generate an ensemble we varied the clustering algorithm, the setting of the selected clustering algorithm, the number of features of the original data objects, and the number of output clusters. Precisely, the ensemble for each dataset was built up as follows:

1. We computed a set of clustering solutions, which were obtained by performing multiple runs of the k-Means algorithm on the specific dataset with different random initializations.
2. We completed the ensemble by adding a further set of clustering solutions obtained by varying the feature selection of the original data, the clustering algorithm, and the number of output clusters. In particular, the feature set was varied by randomly selecting subsets having 40%, 50%, 60%, 70%, 80%,

[1] http://archive.ics.uci.edu/ml/

[2] http://www.cis.temple.edu/~latecki/TestData/TS-Koegh/

90%, and 100% size of the original feature space. The k-Means and AHC with group-average-linkage algorithms were used for the clustering task, and the output clustering solutions were composed by a number of clusters equal to 2 and 50%, 75%, 100%, 150%, and 200% of the number of ideal classes of the specific dataset.

Setting of weighting schemes. For each of the proposed weighting schemes, we used both NMI and FM in order to measure the ensemble diversity. We also performed a preliminary phase of tuning of the parameter α and finally presented the clustering performance corresponding to the relative best setting of α for each weighting scheme. Although details are not reported in this paper for the sake of brevity of presentation, we make mention here that varying α seemed not to have a significant impact on the overall results—a generally valid setting was $\alpha = 0.65$.

In addition to the setting of α, the GW scheme also requires the selection of a clustering algorithm for the step "to cluster clusterings". We tried different well-known algorithms, such as k-Means and AHC with single-linkage, group-average-linkage and complete-linkage. In this paper we present results obtained by using the AHC with group-average-linkage.

Setting of clustering ensembles methods. For each method and dataset, we set the number of output clusters k equal to the number of ideal classes of the specific dataset. Also, as far as the graph-partitioning-based methods (i.e., CSPA, HPGA, WSPA, MCLA, HBGF, WBPA), we set the METIS parameters as suggested in [23]; WSPA and WBPA additionally require the number of LAC iterations, which was set equal to the size of the ensemble generated for each dataset.

It should be remarked that setting the clustering methods had a marginal importance in this work, since the main focus of our experimental evaluation was on assessing the effectiveness of the methods with and without employing the proposed weighting schemes.

Clustering validity criteria. To assess the quality of a consensus partition belonging to an ensemble, we exploited the availability of reference classifications for the datasets. The objective was to evaluate how well a clustering fits a predefined scheme of known classes (natural clusters). To this purpose, we resorted to the previously defined NMI and FM measures.

5.2 Results

Tables 2–8 show the accuracy results obtained by the various clustering ensembles algorithms, with and without employing weighting schemes, on the selected datasets. Accuracy results are reported in terms of NMI and FM (cf. Section 3).

Table 2: Results on Glass

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.56	0.61	0.65	0.66
	FM	0.77	0.78	0.74	0.78
HPGA	NMI	0.55	0.57	0.56	0.57
	FM	0.66	0.71	0.75	0.72
WSPA	NMI	0.65	0.67	0.73	0.71
	FM	0.75	0.71	0.73	0.72
MV	NMI	0.65	0.65	0.64	0.70
	FM	0.71	0.72	0.69	0.74
AGGL	NMI	0.67	0.67	0.65	0.68
	FM	0.70	0.70	0.70	0.70
IVC	NMI	0.55	0.68	0.65	0.70
	FM	0.71	0.77	0.75	0.80
MCLA	NMI	0.56	0.61	0.62	0.65
	FM	0.64	0.70	0.65	0.67
MCS	NMI	0.58	0.58	0.57	0.60
	FM	0.70	0.68	0.67	0.70
HBGF	NMI	0.64	0.64	0.62	0.64
	FM	0.76	0.77	0.76	0.77
WBPA	NMI	0.66	0.69	0.70	0.73
	FM	0.68	0.70	0.73	0.71

Table 3: Results on Ecoli

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.66	0.68	0.65	0.69
	FM	0.70	0.70	0.68	0.70
HPGA	NMI	0.64	0.64	0.66	0.65
	FM	0.70	0.71	0.74	0.71
WSPA	NMI	0.59	0.62	0.59	0.62
	FM	0.62	0.64	0.63	0.63
MV	NMI	0.68	0.69	0.66	0.69
	FM	0.85	0.85	0.84	0.86
AGGL	NMI	0.63	0.67	0.67	0.69
	FM	0.74	0.75	0.77	0.75
IVC	NMI	0.65	0.63	0.66	0.68
	FM	0.80	0.74	0.86	0.80
MCLA	NMI	0.59	0.62	0.65	0.67
	FM	0.69	0.70	0.73	0.73
MCS	NMI	0.58	0.58	0.57	0.60
	FM	0.70	0.68	0.67	0.70
HBGF	NMI	0.62	0.62	0.59	0.62
	FM	0.72	0.74	0.70	0.75
WBPA	NMI	0.71	0.72	0.74	0.72
	FM	0.73	0.75	0.75	0.77

Evaluation of weighted clustering ensembles. Looking at the tables, a first important remark is that, for each of the clustering algorithms, weighted settings led to better performance in general.

Regardless of the specific weighting scheme or clustering ensembles algorithm, we observed the following maximum improvements of clustering quality with respect to the case no weighting scheme was used: 24% on ControlChart, 22% on ISOLET, 18% on ImageSegmentation, 15% on Glass, 12% on LetterRecognition, 11% on Tracedata, and 8% on Ecoli.

Table 4: Results on ImageSegmentation

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.49	0.51	0.53	0.51
	FM	0.55	0.56	0.60	0.59
HPGA	NMI	0.38	0.50	0.53	0.56
	FM	0.45	0.54	0.55	0.59
WSPA	NMI	0.53	0.63	0.63	0.61
	FM	0.61	0.71	0.70	0.68
MV	NMI	0.45	0.45	0.46	0.45
	FM	0.69	0.78	0.77	0.75
AGGL	NMI	0.58	0.58	0.58	0.58
	FM	0.68	0.69	0.66	0.69
IVC	NMI	0.51	0.57	0.55	0.60
	FM	0.59	0.65	0.60	0.69
MCLA	NMI	0.51	0.52	0.52	0.54
	FM	0.63	0.63	0.68	0.67
MCS	NMI	0.58	0.58	0.57	0.60
	FM	0.70	0.68	0.67	0.70
HBGF	NMI	0.48	0.53	0.51	0.53
	FM	0.57	0.58	0.60	0.59
WBPA	NMI	0.51	0.52	0.53	0.52
	FM	0.53	0.55	0.57	0.57

Table 5: Results on ISOLET

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.55	0.55	0.58	0.58
	FM	0.62	0.62	0.65	0.64
HPGA	NMI	0.45	0.59	0.55	0.61
	FM	0.50	0.62	0.57	0.63
WSPA	NMI	0.55	0.62	0.63	0.66
	FM	0.64	0.70	0.70	0.74
MV	NMI	0.49	0.50	0.46	0.50
	FM	0.75	0.81	0.74	0.84
AGGL	NMI	0.66	0.67	0.63	0.68
	FM	0.72	0.72	0.71	0.72
IVC	NMI	0.44	0.60	0.66	0.66
	FM	0.53	0.70	0.74	0.73
MCLA	NMI	0.55	0.62	0.60	0.62
	FM	0.67	0.67	0.71	0.73
MCS	NMI	0.58	0.58	0.57	0.60
	FM	0.70	0.68	0.67	0.70
HBGF	NMI	0.56	0.69	0.68	0.66
	FM	0.63	0.69	0.69	0.65
WBPA	NMI	0.58	0.60	0.63	0.61
	FM	0.65	0.66	0.71	0.70

Table 6: Results on LetterRecognition

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.40	0.48	0.47	0.48
	FM	0.51	0.60	0.61	0.62
HPGA	NMI	0.41	0.40	0.38	0.41
	FM	0.51	0.48	0.52	0.53
WSPA	NMI	0.43	0.45	0.45	0.45
	FM	0.52	0.53	0.49	0.53
MV	NMI	0.72	0.70	0.68	0.70
	FM	0.80	0.80	0.80	0.80
AGGL	NMI	0.63	0.64	0.63	0.65
	FM	0.68	0.70	0.74	0.74
IVC	NMI	0.38	0.43	0.41	0.43
	FM	0.46	0.56	0.55	0.58
MCLA	NMI	0.45	0.49	0.51	0.53
	FM	0.56	0.59	0.59	0.62
MCS	NMI	0.48	0.50	0.50	0.53
	FM	0.50	0.52	0.48	0.55
HBGF	NMI	0.40	0.41	0.42	0.42
	FM	0.51	0.52	0.50	0.55
WBPA	NMI	0.46	0.48	0.50	0.51
	FM	0.52	0.52	0.56	0.57

Table 7: Results on Tracedata

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.50	0.51	0.48	0.50
	FM	0.53	0.54	0.51	0.54
HPGA	NMI	0.53	0.55	0.56	0.58
	FM	0.64	0.65	0.67	0.67
WSPA	NMI	0.50	0.50	0.50	0.50
	FM	0.52	0.55	0.55	0.57
MV	NMI	0.50	0.54	0.57	0.54
	FM	0.53	0.59	0.62	0.63
AGGL	NMI	0.50	0.57	0.58	0.57
	FM	0.54	0.64	0.60	0.64
IVC	NMI	0.50	0.58	0.56	0.59
	FM	0.54	0.63	0.60	0.64
MCLA	NMI	0.58	0.60	0.63	0.64
	FM	0.71	0.70	0.73	0.75
MCS	NMI	0.57	0.58	0.57	0.60
	FM	0.63	0.68	0.65	0.66
HBGF	NMI	0.50	0.51	0.53	0.54
	FM	0.53	0.60	0.62	0.62
WBPA	NMI	0.45	0.50	0.53	0.56
	FM	0.52	0.53	0.56	0.62

Evaluation of weighting schemes. Comparing the proposed weighting schemes, the DW scheme led to the maximum quality improvements on all the datasets. Moreover, the DW-based weighted version of each clustering ensembles method performed as good as or better than the original (unweighted) clustering method in most cases (i.e., except WSPA on Glass and MV on LetterRecognition, with FM and NMI as diversity measures, respectively).

As far as the other two weighting schemes, the adoption of GW led to better maximum performance than SW in nearly all datasets. However, considering the average performance (i.e., the average increase in accuracy with respect to the unweighted settings, over all the algorithms), GW behaved less reliably than SW. This can be explained since GW requires a phase of parameter tuning which is more critical than in the SW case; however, GW is in principle designed as a refinement of SW and is really effective in improving the performance of the clustering ensembles algorithms.

For instance, on ISOLET (Table 5), GW allowed the clustering ensembles algorithms to achieve up to 22% (resp. 21%) of maximum quality improvement according to NMI (resp. FM), against the 16% (resp. 17%) improvement obtained by employing the SW scheme. However, on the same dataset, no benefit resulted from the adoption of the GW scheme in six out of twenty cases (over all the algorithms and the reported performance).

Table 8: Results on ControlChart

method	*diversity*	*accuracy*			
		no weights	*SW*	*GW*	*DW*
CSPA	NMI	0.78	0.81	0.75	0.82
	FM	0.82	0.82	0.80	0.82
HPGA	NMI	0.68	0.80	0.77	0.81
	FM	0.72	0.87	0.83	0.85
WSPA	NMI	0.51	0.61	0.72	0.75
	FM	0.60	0.75	0.79	0.79
MV	NMI	0.83	0.84	0.86	0.84
	FM	0.84	0.86	0.87	0.87
AGGL	NMI	0.74	0.74	0.75	0.74
	FM	0.76	0.77	0.78	0.77
IVC	NMI	0.69	0.74	0.82	0.82
	FM	0.75	0.76	0.80	0.82
MCLA	NMI	0.63	0.64	0.70	0.66
	FM	0.69	0.71	0.75	0.75
MCS	NMI	0.58	0.58	0.57	0.60
	FM	0.70	0.68	0.67	0.70
HBGF	NMI	0.72	0.72	0.76	0.73
	FM	0.78	0.77	0.80	0.78
WBPA	NMI	0.56	0.69	0.72	0.73
	FM	0.62	0.78	0.79	0.78

Evaluation of diversity measures. Using FM as diversity criterion, the accuracy results were generally higher than in the NMI setting, i.e., the maximum quality of the consensus partition observed on all the datasets always referred to FM values. However, from the perspective of the advantages that can be derived from using a weighting scheme, the highest average gains (over the performance of all the methods) were obtained in terms of NMI on four out of seven datasets (i.e., Glass, Ecoli, ISOLET, and ControlChart).

For instance, on ControlChart, using the DW scheme led to a maximum increase in quality (with respect to unweighted clustering methods) which was equal to 24% and 19% in terms of NMI and FM, respectively; the average increase in quality was 8% (NMI) and 6.5% (FM). On ImageSegmentation, the maximum gain was achieved in terms of NMI (18%, against 14% by FM) by using the DW scheme; the average improvement instead referred to the FM diversity (5.2%, against 4.3% by NMI).

Evaluation of clustering ensembles methods. Instance-based methods showed better performance with respect to methods belonging to the other two clustering ensembles approaches, on all datasets (except for Tracedata). For instance, considering the results based on NMI, we observed the following differences between the maximum NMI values scored by the best and the worst approach: 19% on LetterRecognition, 16% on ControlChart, 10% on ImageSegmentation, 9% on Ecoli, 8% on Tracedata and Glass, and 7% on ISOLET.

Concerning the algorithms, MV ranked first followed by IVC and HPGA, according to the FM-based diversity criterion; by contrast, in the NMI-based evaluation, more algorithms alternated with each other as best performer on the various datasets.

However, looking at the average performance over all the methods for each clustering approach and dataset, we observed that there was no approach prevailing against the remaining ones. In particular, the best average results were achieved by the instance-based methods on LetterRecognition and ControlChart, the hybrid methods on Glass, Ecoli and ISOLET, and the cluster-based methods on ImageSegmentation and Tracedata.

6 Conclusion

In this paper we proposed general schemes for weighting clustering ensembles. These schemes were designed to be independent of any specific method of clustering ensembles, which represents an important advantage due to the variety of clustering ensembles approaches and methods. We showed how the weighting schemes can be easily employed in algorithm models of the most currently used approaches for clustering ensembles. We conducted an extensive experimental evaluation aimed to assess a number of aspects, such as the beneficial impact of using a weighting scheme in clustering ensembles algorithms, a comparison between different notions of diversity as clustering ensemble validity criteria, and a comparison of existing clustering ensembles algorithms. Results have shown that clustering ensembles algorithms improve their ability in finding a consensus partition when equipped with the proposed weighting schemes.

References

[1] M. Al-Razgan and C. Domeniconi. Weighted Clustering Ensembles. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, 2006.

[2] H. Ayad and M. S. Kamel. Finding Natural Clusters Using Multi-Clusterer Combiner Based on Shared Nearest Neighbors. In *Proc. Int. Workshop on Multiple Classifier Systems (MCS)*, pages 166–175, 2003.

[3] E. Bauer and R. Kohavi. An Empirical Comparison of Voting Classification Algorithms: Bagging, Boosting, and Variants. *Machine Learning*, 36(1-2):105–139, 1999.

[4] C. Boulis and M. Ostendorf. Combining Multiple Clustering Systems. In *Proc. European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 63–74, 2004.

[5] P. S. Bradley and U. M. Fayyad. Refining Initial Points for K-Means Clustering. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 91–99, 1998.

[6] R. Caruana, M. F. Elhawary, N. Nguyen, and C. Smith.

Meta Clustering. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 107–118, 2006.

[7] T.M. Cover and J.A. Thomas. *Elements of Information Theory (Second Edition).* Wiley, 2006.

[8] C. Domeniconi and M. Al-Razgan. Weighted Cluster Ensembles: Methods and Analysis. *ACM Trans. on Knowledge Discovery from Data (TKDD)*, To appear, 2009.

[9] C. Domeniconi, D. Gunopulos, S. Ma, B. Yan, M. Al-Razgan, and D. Papadopoulos. Locally Adaptive Metrics for Clustering High Dimensional Data. *Data Mining and Knowledge Discovery*, 14(1):63–97, 2007.

[10] S. Dudoit and J. Fridlyand. Bagging to improve the accuracy of a clustering procedure. *Bioinformatics*, 19(9):1090–1099, 2003.

[11] X. Z. Fern and C. E. Brodley. Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 186–193, 2003.

[12] X. Z. Fern and C. E. Brodley. Solving Cluster Ensemble Problems by Bipartite Graph Partitioning. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 281–288, 2004.

[13] X. Z. Fern and W. Lin. Cluster Ensemble Selection. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 787–797, 2008.

[14] A. L. N. Fred. Finding Consistent Clusters in Data Partitions. In *Proc. Int. Workshop on Multiple Classifier Systems (MCS)*, pages 309–318, 2001.

[15] A. L. N. Fred and A. K. Jain. Data Clustering using Evidence Accumulation. In *Proc. Int. Conf. on Pattern Recognition (ICPR)*, pages 276–280, 2002.

[16] A. L. N. Fred and A. K. Jain. Robust Data Clustering. In *Proc. IEEE Computer Society Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 128–136, 2003.

[17] A. Gionis, H. Mannila, and P. Tsaparas. Clustering Aggregation. *ACM Trans. on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.

[18] D. Greene, A. Tsymbal, N. Bolshakova, and P. Cunningham. Ensemble Clustering in Medical Diagnostics. In *Proc. IEEE Int. Symposium on Computer-Based Medical Systems (CBMS)*, pages 576–581, 2004.

[19] S. T. Hadjitodorov and L. I. Kuncheva. Selecting Diversifying Heuristic for Cluster Ensembles. In *Proc. Int. Workshop on Multiple Classifier Systems (MCS)*, pages 200–209, 2007.

[20] S. T. Hadjitodorov, L. I. Kuncheva, and L. P. Todorova. Moderate diversity for better cluster ensembles. *Information Fusion*, 7(3):264–275, 2006.

[21] D. Ienco and R. Meo. Exploration and Reduction of the Feature Space by Hierarchical Clustering. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 577–587, 2008.

[22] G. Karypis, R. Aggarwal, V. Kumar, and S. Shekhar. Multilevel Hypergraph Partitioning: Applications in VLSI Domain. In *Proc. Design Automation Conf. (DAC)*, pages 526–529, 1997.

[23] G. Karypis and V. Kumar. A fast and high quality multilevel scheme for partitioning irregular graphs. *SIAM Journal on Scientific Computing*, 20(1):359–392, 1998.

[24] L. I. Kuncheva and S. T. Hadjitodorov. Using Diversity in Cluster Ensembles. In *Proc. IEEE Int. Conf. on Systems, Man and Cybernetics (SMC)*, volume 2, pages 1214–1219, 2004.

[25] T. Li and C. Ding. Weighted Consensus Clustering. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 798–809, 2008.

[26] T. Li, C. Ding, and M. I. Jordan. Solving Consensus and Semi-supervised Clustering Problems Using Nonnegative Matrix Factorization. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 577–582, 2007.

[27] B. Minaei, A. Topchy, and W. Punch. Ensembles of Partitions via Data Resampling. In *Proc. Int. Conf. on Information Technology: Coding and Computing (ITCC)*, pages 188–192, 2004.

[28] A. Y. Ng, M. I. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an algorithm. In *Proc. Int. Conf. on Neural Information Processing Systems (NIPS)*, pages 849–856, 2001.

[29] N. Nguyen and R. Caruana. Consensus Clustering. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 607–612, 2007.

[30] A. Strehl and J. Ghosh. Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research (JMLR)*, 3:583–617, 2002.

[31] A. Topchy, A. K. Jain, and W. Punch. Combining Multiple Weak Clusterings. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 331–338, 2003.

[32] C.J. van Rijsbergen. *Information Retrieval.* Butterworths, 1979.

[33] X. Yu and I. Yoo. Cluster ensemble and its applications in gene expression analysis. In *Proc. Asia-Pacific Bioinformatics Conf. (APBC)*, pages 297–302, 2004.

[34] Y. Zeng, J. Tang, J. Garcia-Frias, and G. R. Gao. An Adaptive Meta-Clustering Approach: Combining the Information from Different Clustering Results. In *Proc. IEEE Computer Society Bioinformatics Conf. (CSB)*, pages 330–332, 2002.