Data Mining and Knowledge Discovery manuscript No.
(will be inserted by the editor)

Projective Clustering Ensembles

Francesco Gullo · Carlotta Domeniconi · Andrea Tagarelli

the date of receipt and acceptance should be inserted later

Abstract A considerable amount of work has been done in data clustering research during the last four decades, and a myriad of methods has been proposed focusing on different data types, proximity functions, cluster representation models, and cluster presentation. However, clustering remains a challenging problem due to its ill-posed nature: it is well known that off-the-shelf clustering methods may discover different patterns in a given set of data, mainly because every clustering algorithm has its own bias resulting from the optimization of different criteria. This bias becomes even more important as in almost all real-world applications, data is inherently high-dimensional and multiple clustering solutions might be available for the same data collection. In this respect, the problems of *projective clustering* and *clustering ensembles* have been recently defined to deal with the high dimensionality and multiple clusterings issues, respectively. Nevertheless, despite such two issues can often be encountered together, existing approaches to the two problems have been developed independently of each other.

In our earlier work [35] we introduced a novel clustering problem, called Projective Clustering Ensembles (PCE): given a set (ensemble) of projective clustering solutions, the goal is to derive a projective consensus clustering, i.e., a projective clustering that complies with the information on object-to-cluster and the feature-to-cluster assignments given in the ensemble. In this paper, we enhance our previous study and provide theoretical and experimental insights into the PCE problem. PCE is formalized as an optimization problem and is designed to satisfy desirable requirements on independence from the specific clustering ensemble algorithm, ability to handle hard as well as soft data clustering, and different feature weightings. Two PCE formula-

F. Gullo
DEIS Department, University of Calabria, 87036 Rende (CS), Italy, E-mail: fgullo@deis.unical.it

C. Domeniconi
Department of Computer Science, George Mason University, 22030 Fairfax - VA, USA,
E-mail: carlotta@cs.gmu.edu

A. Tagarelli
DEIS Department, University of Calabria, 87036 Rende (CS), Italy, E-mail: tagarelli@deis.unical.it

tions are defined: a two-objective optimization problem, in which the two objective functions respectively account for the object- and feature-based representations of the solutions in the ensemble, and a single-objective optimization problem, in which the object- and feature-based representations are embedded into a single function to measure the distance error between the projective consensus clustering and the projective ensemble. The significance of the proposed methods for solving the PCE problem has been shown through an extensive experimental evaluation based on several datasets and comparatively with projective clustering and clustering ensemble baselines.

Keywords Clustering · Clustering ensembles · Projective clustering · Multi-objective optimization

1 Introduction

Given a set of data objects as points in a multi-dimensional space (or *feature space*), the problem of *clustering* is to discover a number of homogeneous subsets of data, called *clusters*, which are well-separated from each other. Clustering is the key step for many tasks in data management and mining that require the discovery of unknown relationships and patterns in large sets of data [42, 31].

Most of the existing approaches to clustering provide single clustering solutions and/or use the same (typically very large) feature space. The latest advances in clustering research have focused on methods for solving issues that are concerned with two major aspects: (i) dealing with high dimensionality, and (ii) handling multiple organizations (clustering solutions) of the data.

Almost all problems of practical interest are high-dimensional, i.e., they involve data objects represented by large sets of features. A common scenario with high-dimensional data is that several clusters may exist in different subspaces that correspond to different combinations of features. In general, it is unlikely that all features of the data objects are equally relevant to form meaningful clusters.

Another challenge in the clustering process is due to the fact that, in many real-life domains, multiple, potentially meaningful groupings of the input data can be available, hence providing different views of the data. For instance, in genomics, multiple clustering solutions would be needed to capture the multiple functional roles of genes. In text mining, documents inherently discuss multiple topics, hence their grouping by content should reflect different informative views which correspond to multiple (possibly alternative) clusterings. In evolving data (streams) management, users could be interested in different views of the data that may correspond to different informative needs.

Recent advances in data clustering have led to the definition of the problems of *projective clustering* (to deal with high dimensionality) and *clustering ensembles* (for handling multiple clustering solutions).

Projective clustering. Projective clustering (or projected clustering) [63, 85, 1, 23, 56] aims to discover *projective clusters*, i.e., subsets of the input data having different (possibly overlapping) subsets of features (subspaces) associated to them. Projective clustering is closely related to the *subspace clustering* problem [4, 65, 47, 57], as

both detect clusters of data points that exist in subspaces of a dataset; however, the problem definition in subspace clustering is actually to search for all clusters of data points in all meaningful subspaces of a data set.

Being able to discover clusters of data points in subsets of features, projective clustering aims to solve issues that typically arise in high-dimensional data, namely sparsity and concentration of distances. The former issue is inherent in high-dimensional datasets, since the number of data points required to represent any distribution exponentially grows with the number of dimensions. The latter issue refers to a lack of distances in distinguishing between data points as dimensionality increases [11, 41, 73]. These problems are sometimes also referred to as the *curse-of-dimensionality* [10]. The identification of compact clusters in high-dimensional feature spaces can hence be meaningful only if the assigned objects are projected onto the corresponding natural subspaces.

Any projective cluster is hence coupled with a twofold information: the *object-to-cluster assignment* (whether an object belongs to that cluster) and the *feature-to-cluster assignment* (whether a feature belongs to the subspace assigned to that cluster). Projective clusters tend to be less noisy—because each group of data is represented over a subspace which ideally does not contain features that are irrelevant or redundant for that group—and more understandable—because the exploration of a cluster is much easier when only few, descriptive features are involved.

This ability of projective clustering fits well an important characteristic of most real-life application domains, in which the clusters of data objects depend on the type of information (i.e., the subset of features) used to represent/group the data. For instance, in the context of object recognition, projective clustering methods can provide better solutions to the image segmentation problem as they are able to identify dense regions into an image, where the associated subspaces are based on features like pixel color, intensity, or texture. Moreover, in wireless sensor networks and environmental monitoring applications, sensor nodes can be differently partitioned according to their readings (time series) that capture different behavioral trends of the sensors in response to well-distinguished environmental events. In customer segmentation applications, customers can be differently partitioned depending on which part of their demographic profile (e.g., education, occupation) or behavioral profile (e.g., purchase habits, needs expressed through preferences manifested in everyday behavior).

Clustering ensembles. Clustering ensembles [71, 75, 22, 33], also known as consensus clustering [64] or aggregation clustering [34], are concerned with the following problem: Given a set of clustering solutions (called *ensemble*), derive a *consensus clustering* as a (new) clustering by the optimization of a certain objective function (the *consensus function*) which expresses how well any candidate consensus clustering complies with the solutions in the ensemble.

Clustering ensemble methods resort to the idea of combining multiple classifiers [68, 15], and adapt it to a clustering context: due to algorithmic peculiarities of any specific clustering method, a single clustering solution may not be able to capture all facets of a given clustering problem. In this case, it is useful to generate an ensemble by varying one or more aspects of the clustering process, such as the

clustering algorithm, the parameter setting, or the number of clusters, and eventually obtain a consensus clustering by properly "aggregating" the information in the ensemble.

Moreover, clustering ensemble methods find application for multi-view data, where a set of clustering solutions is naturally available rather than a single one, but the results to present should have the form of a unique solution. Typical examples of such scenarios are distributed clustering, where each node in the distributed environment stores its own portion of the entire data and outputs a clustering that considers only that portion; or privacy preserving clustering, where different representations of the same set of objects are maintained in different sites for privacy purposes.

Projective clustering ensembles. Projective clustering and clustering ensembles have been so far developed independently of one another: projective clustering avoids the curse-of-dimensionality in high-dimensional feature spaces, but cannot handle sets of multiple clusterings, whereas clustering ensemble methods address the multi-view nature of clustering, but do not face in general the high dimensionality issue. The state-of-the-art of research in clustering lacks a unified framework capable of handling both issues, thus we aim to bridge them through the *projective clustering ensembles* (PCE) problem [35]: Given a set of projective clustering solutions, or *projective ensemble*, derive a *projective consensus clustering* that complies with the information available from the projective ensemble. This information is only expressed in terms of object-to-cluster and feature-to-cluster assignments, and hence feature relevance values are assumed to be unavailable.

Intuitively, each projective cluster is characterized by a distribution of memberships of the objects as well as a distribution over the features that belong to the subspace of that cluster. Figure 1 illustrates a projective clustering ensemble with three clustering solutions, which are obtained according to different views over the same database. A projective cluster is graphically represented as a rectangle filled with a color gradient, where higher intensities correspond to higher memberships of objects to the cluster. Clusters of the same clustering may overlap with their gradient (i.e., objects can have multiple assignments with different degrees of membership), and colors change to denote that different groupings of objects are associated with different feature subspaces. In the figure, a projective consensus clustering is derived by suitably "aggregating" the ensemble members. Indeed, the first projective consensus cluster is derived from C_1', C_2'', and C_2''', the second projective consensus cluster is derived from C_2', C_3'', and C_3''', finally the last cluster is derived from C_4', C_1'', and C_1'''. Note that the resulting color in each projective consensus cluster resembles a merge of colors in the original projective clusters, which means that a projective consensus cluster is associated with a subset of features shared by the objects in the original clusters.

The PCE problem faces a major challenge that lies in the twofold nature of the information to be aggregated: both the objects and the subspace associated to each cluster need to be identified, and the two sides of the aggregation should depend on one another. This challenge is emphasized by the fact that existing clustering ensemble methods cannot be applied to solve the PCE problem, since they cannot derive a projective consensus clustering starting from an ensemble of solutions each of

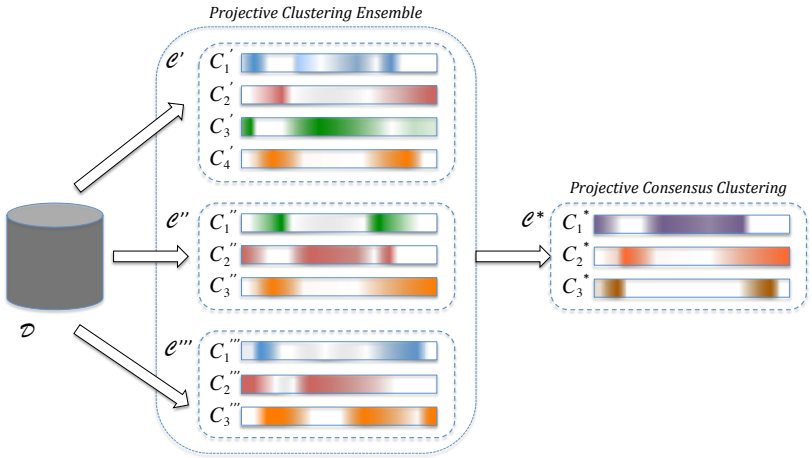


Fig. 1 Illustration of a projective clustering ensemble and derived consensus clustering. Each gradient refers to the cluster memberships over all objects. Colors denote different feature subspaces associated with the projective clusters.

which is outputted by a projective clustering method—traditional clustering ensemble methods can only aggregate the information about the object-to-cluster assignments, whereas they are not conceived to aggregate the information about the feature-to-cluster assignments.

Any application domain that needs or can reuse multiple organizations that reflect different perspectives on the data *and* a high number of features are used to represent data can be profitably assisted by a PCE method. Applications previously discussed respectively for projective clustering and clustering ensembles could be likely devised in an integrated way. Here we illustrate another scenario in which users want to skim a given collection of news from an online repository. The news are organized in two levels. A first-level page only provides the highlights of the news, such as titles and headlines. A read-more link must be followed to reach a second-level page, containing the full-text content (body) of the news; second-level pages however could not be freely available, as, e.g., users must register and pay a fee to access them. Each news is tagged with one or more keywords concerning the themes discussed, the countries involved in the events described, or political/financial companies that are implicitly or explicitly mentioned in the news. These tags can be used to provide different perspectives on the news categorization (e.g., one categorization by "theme", or by "company"), and hence multiple alternative groupings (clusterings) of the news could be made available to the users. Now, suppose that a user wants to take advantage of an online retrieval system that has two main functionalities: (i) exploiting a new clustering of the news that takes into account all the predefined categorizations in an integrated way, and (ii) enabling the search for news through the exploration of a cluster-based index which stores a summary of the representative content of any cluster in the form of descriptive terms of the cluster. In particular, the latter functionality of the system requires the availability of a subset of features of the news assigned to a given cluster.

A PCE approach is definitely required to enable an application like the one above described. Indeed, without the possibility of accessing the entire content of the news, it would be hard for any standard (projective) clustering method to provide effective solutions to this problem (as only a summarized content of the news would be used), and more importantly it would be infeasible to provide a single clustering of the news that can be orthogonal to all existing categorizations. On the other hand, any traditional clustering ensemble approach would certainly be able to reuse the various theme-/company-/country-oriented categorizations of the news by exploiting them as clustering solutions to be incorporated into an ensemble. But traditional clustering ensembles would not be able of handling the most-relevant terms related to the various clusters of news; consequently, the assignment of news to clusters would not be necessarily consistent with the relevance of terms that represent the news of a cluster.

Analogous examples of real-life application of PCE include the categorization of scientific papers given the availability of only summarized information such as title, abstract, keywords, and taxonomic subject fields (e.g., Medical Subject Headings (MeSH), or ACM Computing Classification categories and terms). Yet, in order to develop a system that automatically groups web bookmarks given only their URLs, while ignoring the content of the sites hosting the bookmarks, PCE approaches would simply refer to the implicit categorizations provided by web directory services, such as Google Directory or Yahoo! Directory, to build a projective ensemble, and properly derive an output projective consensus clustering.

Contributions. Projective ensembles are composed by *axis-aligned* (or *axis-parallel*) projective clustering solutions, i.e., solutions in which the subspace associated to each projective cluster is given by a subset of the original feature space. This subset may be expressed either by specifying whether each feature is part of that subspace [56, 63, 83, 3, 12] (*unweighted* feature-to-cluster assignment), or by defining proper feature weights (probabilities) that express the strength according to which a given feature participates to that subspace [23, 17] (*weighted* feature-to-cluster assignment).

Since we are interested in developing PCE methods that are as general as possible, we require the PCE objective functions have to fulfil the following desiderata:

1) Independence from the original feature values of the input data. Any valid PCE formulation should take into account only information about object- and feature-to-cluster assignments, while discarding any information about the specific feature values of data objects, because in many applications this is not available;
2) Independence from the specific clustering algorithm(s) and from any prior knowledge on the setup (e.g., setting of possible input parameters), and from the strategy (e.g., number of output clusters or subsets of features and/or objects) used to generate the projective ensemble, since, again, this information is usually not available;
3) The ability to handle projective ensembles whose solutions have *hard* as well as *soft* object-to-cluster assignments;
4) The ability to deal with feature-to-cluster assignments that are both *unweighted* and *weighted*.

To this end, we define two formulations of PCE: as a *two-objective* and as a *single-objective* optimization problem. The multi-objective formulation involves two

distinct objective functions, each embedding one side of the projective ensemble components: the data clusterings and the assignments of features to clusters. The second formulation aims at solving some issues that arise from the two-objective formulation, such as poor efficiency and/or hard interpretation of the results. In particular, it is based on a single-objective function which acts as an error criterion in the computation of a candidate projective consensus clustering. It involves the object-based representation and the feature-based representation of the various projective clusters simultaneously. For each of the two proposed formulations of PCE, we develop well-founded heuristics, in which a *multi-objective evolutionary strategy* [18] and an EM-like approach are respectively employed.

We conducted an extensive experimental evaluation on 22 datasets, including benchmark, synthetic and time series datasets, and involving both external and internal cluster validity criteria. In addition, we considered a case study to assess how well our PCE approach is suited for a real-world application. We compared the results of the proposed PCE methods with those achieved by four baselines, which also include projective clustering and clustering ensemble methods. Results have shown that both the proposed algorithms produce more accurate projective consensus clusterings than the baseline methods, both in terms of similarity w.r.t. the reference classifications (external assessment criteria) and in terms of similarity w.r.t. the solutions in the projective ensemble (internal assessment criteria). Comparing the two proposed methods to one another, the two-objective-based approach generally leads to more accurate results, whereas the single-objective-based one is more efficient.

We would like to point out that the first formal definition of the PCE problem and heuristics to solve it were introduced in our earlier work [35]. Besides a new and extensive experimental evaluation, in this paper we have thoroughly deepened our understanding of the PCE problem, providing theoretical insights into both the proposed formulations, comparing their features and highlighting their differences.

The next section briefly overviews the state-of-the-art for projective clustering and clustering ensembles. Section 3 provides the problem definition, and Section 4 presents our solutions. Section 5 describes our experimental evaluation, and Section 6 concludes the paper. Finally, we report in Appendix the proofs of all the results derived in the paper.

2 Related Work

In this section, we overview the main literature on projective clustering and clustering ensembles. We point out that none of the following discussed methods is closely related to ours, since each of them develops a solution for either one or the other problem.

2.1 Projective Clustering

Existing (axis-aligned) projective clustering methods can be classified into four main approaches, namely *bottom-up*, *top-down*, *soft*, and *hybrid* [47, 57].

Bottom-up projective clustering. Bottom-up methods are based on two steps: finding subspaces recognized as "interesting", and assigning each data object to the most similar subspace. The projective cluster structure is computed in a bottom-up fashion by searching for the subspaces to be associated to the discovered projective clusters.

The *Projected Clustering via Cluster Cores* (P3C) algorithm [56] deals with numeric as well as categorical data, and is designed to work with projective clusters that exist in subspaces spanned by very few features. P3C is also able to compute overlapping projective clusters.

In [69], the *Support and Chernoff-Hoeffding bound-based Interesting Subspace Miner* (SCHISM) algorithm is proposed to mine interesting subspaces rather than projective clusters; hence, SCHISM is not an actual projective clustering algorithm, since it solves a related problem: finding subspaces to look for clusters.

Top-down projective clustering. Top-down approaches aim to find the subspaces starting from the full feature space.

Efficient Projective Clustering by Histograms (EPCH) [63] identifies dense regions in each low-dimensional histogram. In [54], the *CLustering based on decision Trees* (CLTree) algorithm assigns a common class label to all input objects and adds additional objects uniformly distributed over the data space and labeled by a different class. Then, a decision tree is trained to separate the two classes.

Further approaches belong to hierarchical, partitional relocation, and density-based categories. Hierarchical algorithms are proposed in [83, 1]. HARP (*a Hierarchical approach with Automatic Relevant dimension selection for Projected clustering*) [83] follows an agglomerative hierarchical clustering (AHC) scheme with single link, and requires two main parameters to control the cluster construction: the minimum number of selected features and a threshold for selecting a feature in a cluster being formed. Unlike HARP, the *Hierarchical Subspace Clustering* (HiSC) algorithm [1] produces a hierarchy of nested projective clusters, i.e., a dendrogram storing relationships of lower dimensional projective clusters that are embedded within higher-dimensional projective clusters.

Partitional relocation methods [3, 84] follow a classic iterative relocation scheme. *PROjected CLUStering algorithm* (PROCLUS) [3] is a K-Medoids algorithm that makes a clustering initialization over the full feature space and, besides the number of desired clusters, requires an additional parameter concerning the average dimensionality of a projective cluster, which is not trivial to set. For this reason, PROCLUS may fail in detecting projective clusters of very different sizes. Variants of PROCLUS include FINDIT (*a Fast and INtelligent subspace clustering algorithm using DImension voTing*) [81], which employs some heuristics to enhance efficiency and clustering accuracy, and *SemiSupervised Projected Clustering* (SSPC) [84], which is able to further enhance accuracy by using domain knowledge in the form of labeled objects and/or labeled features.

The *PreDeCon* algorithm proposed in [12] follows a density-based approach, as it adapts the basic DBSCAN [25] using a specialized subspace distance measure that captures the subspace of each cluster.

Soft projective clustering. All above methods provide clustering solutions that are hard at data clustering level and have unweighted feature-to-cluster assignments. However, a recent corpus of study has focused on algorithms able to produce soft data clusterings [56, 17], and/or clusterings having differently weighted feature-to-cluster assignments [23, 17].

Locally Adaptive Clustering (LAC) [23] performs local feature selection by assigning weights to features, and thus enables distance measures to reflect local correlations of data. A parameter h controls the incentive for clustering on more features depending on the strength of the local correlation of data. The study proposed in [17] focuses on a probabilistic modeling of projective clusters and proposes a *Fuzzy Projective Clustering* (FPC) algorithm, which can produce overlapping clusters, like P3C, and can also assign different weights to the subspace features.

Hybrid projective clustering. Hybrid projective clustering involves methods that may in principle be considered as both projective and subspace clustering approaches. Most hybrid algorithms follow a density-based approach. *Density-based Optimal Projective Clustering* (DOC) [67] greedily discovers projective clusters. It can handle variable-size clusters and does not require the number of clusters as input parameter; however, it is sensitive to a user-defined parameter required to control the cluster quality, and assumes that the projective clusters are hypercubes with same side-length over all features. In [85], *MINECLUS* is proposed to enhance the efficiency of DOC based on an optimized adaptation of the frequent pattern tree growth method. The key idea is to model any input data object as an itemset comprised of the features in which that object is within a certain distance from a given pivot data object.

Detecting Subspace cluster Hierarchies (DiSH) [2] follows a similar idea as PreDeCon, but uses a hierarchical clustering model which is inspired by the density-based hierarchical clustering algorithm OPTICS [5].

FIlter REfinement Subspace clustering (FIRES) [46] computes one-dimensional projective clusters using any clustering technique provided in input by the user. These one-dimensional projective clusters are then merged by applying a "clustering of clusters" step. The clusters discovered by FIRES may overlap, but, unlike classic subspace clustering methods, the algorithm is not able to produce all clusters in all interesting subspaces.

A crucial issue in subspace clustering is redundancy, since exponentially many subspace clusters are usually detected in arbitrary projections. The study in [58] deals with global redundancy removal, by introducing a twofold model of relevance for subspace clustering, based on interestingness and non-redundancy functions via a new definition of cluster gain. This relevance model enables a heuristic for detecting only non-redundant yet possibly overlapping subspace clusters.

Another issue in subspace clustering (related to redundancy) is scalability, since the commonly adopted Apriori-style search of possible subspaces is exponential in the number of dimensions. Moreover, density-based clustering methods have to compute the neighborhood around each object in each subspace under consideration, resulting in poor scalability w.r.t. dimensionality as well as database size. The EDSC algorithm [6] improves the efficiency of density-based clustering using a multistep filter-and-refine procedure, while guaranteeing lossless pruning of the search space.

The approach in [59] explicitly focuses on the scalability problem, particularly for the density-based paradigm. The proposed best-first-way steering of the clustering ensures a reduction of the search subspace processing by directly finding the interesting subspace clusters, while avoiding the majority of redundant subspaces and repeated database scans.

The study in [39] has been recently introduced to deal with joining the two problems of density-based subspace clustering and dense subgraph mining, which can be useful in several application domains such as social networks and genomics.

2.2 Clustering Ensembles

Clustering ensemble methods are here presented under four main categories: *direct* methods, *instance-based*, *cluster-based*, and *hybrid* approaches.

Direct clustering ensembles. Direct clustering ensemble methods are defined according to optimization criteria or probabilistic models that involve a direct comparison between the solutions in the ensemble and any candidate consensus clustering. The algorithms proposed in [21, 24, 28] explicitly solve the *label correspondence problem* to find a correspondence between the cluster labels across the clusterings. The clustering ensemble problem has been mapped to other well-known problems, such as correlation clustering [34] and nonnegative matrix factorization (NNMF) [53, 52]. Heuristic search procedures to formulate the consensus clustering have also been developed as genetic algorithms [32] and multi-ant colony optimization methods [82].

A basic mixture model for clustering ensembles is proposed in [74], where a certain number of consensus clusters is assumed and a multinomial distribution is drawn for each consensus cluster and clustering in the ensemble. Model parameters are estimated through a maximum log-likelihood problem that can be solved by an EM-like procedure. More recently, model-based approaches to clustering ensembles have improved over the basic multinomial mixture model by developing Bayesian models [77, 78], where methods such as collapsed Gibbs sampling and variational Bayesian inference are used for inference and parameter estimation. Moreover, nonparametric Bayesian models [79, 80] have been also proposed to avoid requiring the apriori specification of the size of the consensus clustering.

Instance-based clustering ensembles. Instance-based methods are developed to carry out a direct comparison between data objects. Most instance-based methods operate on the *co-occurrence* or *co-association* matrix $\mathbf{M}$. For each pair of objects $(\mathbf{o}', \mathbf{o}'')$, this matrix stores the number of solutions of the ensemble in which $\mathbf{o}'$ and $\mathbf{o}''$ are placed in the same cluster, divided by the size of the ensemble. In the *Majority Voting* (MV) algorithm [29], $\mathbf{M}$ is "cut" at a given threshold, i.e., all objects whose pairwise entry in $\mathbf{M}$ is greater than the threshold are joined in the same cluster. This approach has been proved to be equivalent to an AHC algorithm with single link metric on $\mathbf{M}$, cutting the resulting dendrogram according to the given threshold [30].

Other algorithms are based on using $\mathbf{M}$ either as a new data matrix [50] or as a pair-wise distance matrix involved in a specific clustering algorithm. In [34], Expectation Maximization or AHC with average linkage are employed, whereas the *Iterative*

Voting Consensus (IVC) algorithm [64] uses K-Means. In [86], 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 [71], the clustering ensemble problem is mapped to a graph/hypergraph partitioning problem. The authors present two instance-based clustering ensemble methods, namely the *Cluster-based Similarity Partitioning Algorithm* (CSPA) and the *HyperGraph Partitioning Algorithm* (HGPA). CSPA induces a weighted graph from $\mathbf{M}$ and partitions it using the well-known graph partitioning algorithm METIS [44]. HGPA 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 clustering is then obtained by partitioning the hypergraph using HMETIS [43].

More recent graph-partitioning-based approaches are proposed in [8, 22]. In [8], the weight of each edge $(\mathbf{o}', \mathbf{o}'')$ in the induced graph is defined in terms of the size of the nearest neighbor list shared between the data objects $\mathbf{o}'$ and $\mathbf{o}''$. In [22], the *Weighted Similarity Partitioning Algorithm* (WSPA) is proposed to combine multiple clusterings that result from different runs of the LAC projective clustering algorithm [23].

In [75], the features of the input data are re-defined according to the information available from the ensemble (e.g., by considering the specific cluster label, for each clustering of the ensemble) and are involved into EM-like procedures.

Cluster-based clustering ensembles. Cluster-based clustering ensemble approaches are based on the principle "to cluster clusters". The key idea is to run a clustering algorithm on the set of clusters contained in all clustering solutions of the ensemble, in order to compute a set of *meta-clusters*. The consensus clustering is finally computed to assign each data object to the meta-cluster that maximizes some assignment criterion (e.g., majority voting).

The study in [14] 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 an additional run of K-Means, and the resulting meta-centroids are used as initial points for a complete run of EM or K-Means.

The *Meta-CLustering Algorithm* (MCLA) [71] builds a graph whose vertices are the clusters of the various clustering solutions in the ensemble, and each edge (C', C'') has a weight equal to the Jaccard similarity coefficient [42] between the clusters associated to the vertices C' and C''. 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 [13], 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. Hybrid clustering ensemble methods attempt to combine ideas coming from both instance-based and cluster-based approaches. The ob-

jective is to build a *hybrid* bipartite graph whose vertices belong to the sets of objects and clusters.

The *Hybrid Bipartite Graph Formulation* (HBGF) algorithm [26] builds a bipartite graph where each edge $(\mathbf{o}, C)$ has weight equal to 1, if the object $\mathbf{o}$ belongs to the cluster C, 0 otherwise. The clustering ensemble result is obtained by partitioning the graph according to standard methods such as METIS, or spectral graph partitioning algorithms (e.g., [62]). The *Weighted Bipartite Partitioning Algorithm* (WBPA) [22] follows the same overall scheme of HBGF, although it extends the range of weight values from $\{0, 1\}$ to $[0, 1]$.

Recently, there has been an increasing interest in selecting and weighting the components of an ensemble. In particular, the *cluster ensemble selection* problem [16, 27] is to select a proper subset of solutions from an ensemble, and the *weighted consensus clustering* problem [52, 38] 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 (or assigning a proper weight to each clustering solution) allows for extracting a more accurate consensus clustering than using the whole ensemble (or the unweighted version of the algorithm).

3 Problem Definition

We present here our formal definition of the problem of projective clustering ensembles (PCE). The objective is to define methods to exploit the information provided by an ensemble of projective clustering solutions (i.e., *projective ensemble*) to compute a projective consensus clustering. The information provided by any projective ensemble is two-fold: on the one hand data are grouped in clusters, and on the other, features are assigned to clusters. This lies in the following notion of *projective cluster*.

Definition 1 (projective cluster) Let $\mathcal{D}$ be a set of data objects, where each $\mathbf{o} \in \mathcal{D}$ is an $|\mathcal{F}|$-dimensional point defined over a feature space $\mathcal{F}$. A *projective cluster* C defined over $\mathcal{D}$ is a pair $\langle \mathbf{\Gamma}_C, \mathbf{\Delta}_C \rangle$, where

- $\mathbf{\Gamma}_C$ denotes the *object-based* representation of C. It is a $|\mathcal{D}|$-dimensional real-valued vector whose components $\Gamma_{C,\mathbf{o}} \in [0, 1]$, $\forall \mathbf{o} \in \mathcal{D}$, represent the *object-to-cluster* assignment of $\mathbf{o}$ to C, i.e., the probability $\Pr(\mathbf{o}|C)$ that the object $\mathbf{o}$ belongs to C;
- $\mathbf{\Delta}_C$ denotes the *feature-based* representation of C. It is an $|\mathcal{F}|$-dimensional real-valued vector whose components $\Delta_{C,f} \in [0, 1]$, $\forall f \in \mathcal{F}$, represent the *feature-to-cluster* assignment of the f-th feature to C, i.e., the probability $\Pr(f|C)$ that the feature f is *informative* for cluster (f belongs to the subspace associated with C).

□

Note that the above definition addresses all possible types of projective clusters handled by existing projective clustering algorithms (cf. Sect. 2.1). Moreover, the definition enables any PCE formulation to satisfy requirements 3) and 4) reported in the Introduction. In fact, both *soft* and *hard* object-to-cluster assignments are taken into

Table 1 Notation used in this paper

Symbol	Description
$\mathbf{o}$	data object
$\mathcal{D}$	collection of data objects
C	projective cluster (set of data objects)
$\mathcal{C}$	projective clustering (set of projective clusters)
$\mathcal{E}$	projective ensemble (set of projective clusterings)
$\mathcal{C}^*$	projective consensus clustering
K	number of clusters in the projective consensus clustering
f	feature
$\mathcal{F}$	set of features
$\mathbf{\Gamma}_C$	object-based representation vector of projective cluster C
$\Gamma_{C,\mathbf{o}}$	object-to-cluster assignment of object $\mathbf{o}$ to projective cluster C
$\mathbf{\Delta}_C$	feature-based representation vector of projective cluster C
$\Delta_{C,f}$	feature-to-cluster assignment of feature f to projective cluster C
$\mathbf{\Lambda}_\mathbf{o}$	feature-based representation vector of object $\mathbf{o}$
$\Lambda_{\mathbf{o},f}$	probability that feature f is informative for object $\mathbf{o}$
$\Psi_o, \overline{\psi}_o, \psi_o$	object-based optimization functions
$\Psi_f, \overline{\psi}_f, \psi_f$	feature-based optimization functions
Q	object- and feature-based optimization function
J	extended Jaccard (Tanimoto) similarity coefficient
ρ	Pareto ranking function
t	population size
$\mathcal{I}, I$	numbers of iterations
$F1_{of}, F1_o, F1_f$	external assessment criteria
$\overline{F1}_{of}, \overline{F1}_o, \overline{F1}_f$	internal assessment criteria

account—the assignment is hard when $\Gamma_{C,\mathbf{o}} \in \{0,1\}$ rather than $[0,1]$, $\forall \mathbf{o} \in \mathcal{D}$. Similarly, feature-to-cluster assignments may be unweighted, i.e., $\Delta_{C,f} = 1$, if f is associated to C, $\Delta_{C,f} = 0$, otherwise; the latter representation is suited for dealing with the output of projective clustering algorithms that only select the most informative features for each cluster, without specifying any feature-to-cluster assignment probability distribution.

Definition 2 (projective clustering solution) Let $\mathcal{D}$ be a set of data objects, where each $\mathbf{o} \in \mathcal{D}$ is defined over a feature space $\mathcal{F}$. A *projective clustering solution* $\mathcal{C}$ for $\mathcal{D}$ is a set of projective clusters such that $\sum_{C \in \mathcal{C}} \Gamma_{C,\mathbf{o}} = 1$, $\forall \mathbf{o} \in \mathcal{D}$. □

According to Def. 1, both objects and features have a probabilistic assignment to any given projective cluster. Furthermore, when a set of projective clusters forms a projective clustering solution $\mathcal{C} = \{C_1, \ldots, C_K\}$ according to Def. 2, the assignment of each object $\mathbf{o} \in \mathcal{D}$ to the various clusters within $\mathcal{C}$ is implicitly described by a random variable. No random variable is instead assigned to the features. The motivation for this lies in that the events "$\mathbf{o}$ *belongs to* C_1", ..., "$\mathbf{o}$ *belongs to* C_K" are mutually exclusive and, therefore, represent a valid event space for the random variable associated to $\mathbf{o}$. The same reasoning does not hold for features. Indeed, any feature f may naturally be informative for a number of different clusters at the same time, leading to events that can be in general non-mutually exclusive. Nevertheless, regarding features, a choice adopted by many projective clustering methods (such as [23, 17]) is to force mutual exclusion from a cluster perspective, i.e., requiring $\sum_{f \in \mathcal{F}} \Delta_{C,f} = 1$, $\forall C \in \mathcal{C}$. This acts like a $[0,1]$-normalization on the feature-based representation vector of any projective cluster, allowing the formulas to be more readable and/or easier

to express. It is easy to see that Def. 2 complies with the latter feature-to-cluster assignment model as well, thus still guaranteeing full generality.

Definition 3 (projective ensemble) Given a set $\mathcal{D}$ of data objects, a *projective ensemble* $\mathcal{E}$ defined over $\mathcal{D}$ is a set of projective clustering solutions for $\mathcal{D}$. □

We observe that Def. 3 satisfies the first two requirements needed for developing general PCE methods, as described in the Introduction. In fact, no information about the projective ensemble generation strategy (algorithms and/or setups), nor original feature values of the objects within $\mathcal{D}$ are provided along with the projective ensemble. Moreover, each clustering solution in $\mathcal{E}$ may contain in general a different number of clusters.

4 Projective Clustering Ensembles Formulations

4.1 Two-objective PCE

The traditional clustering ensemble problem is typically formulated as a special version of the *median partition problem*, which is defined as follows [9]: given a number of partitions (clusterings) defined over the same set of objects and a function that measures the distance between clusterings, find a (new) clustering that minimizes the distance from all the input clusterings. Formally, given an ensemble $\mathcal{E}_{CE}$, the consensus clustering is derived by solving the following:

$$\arg\min_{\mathcal{C}_{CE}} \Psi(\mathcal{C}_{CE}, \mathcal{E}_{CE}) \tag{1}$$

where $\Psi(\mathcal{C}_{CE}, \mathcal{E}_{CE}) = \sum_{\hat{\mathcal{C}}_{CE} \in \mathcal{E}_{CE}} \psi(\mathcal{C}_{CE}, \hat{\mathcal{C}}_{CE})$, and ψ is a distance function between clusterings.

Within this view, a natural way to formulate PCE is to extend (1) by taking into account that any optimal projective consensus clustering $\mathcal{C}^*$ for PCE should meet two different requirements, rather than the only one of standard clustering ensembles. Such requirements refer to the object-to-cluster and feature-to-cluster assignments of the solutions within the input projective ensemble $\mathcal{E}$, respectively. Thus, in order to capture both these aspects, PCE can be formulated as a two-objective optimization problem:

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} \ \{\Psi_o(\mathcal{C}, \mathcal{E}),\ \Psi_f(\mathcal{C}, \mathcal{E})\} \tag{2}$$

where Ψ_o and Ψ_f are optimization functions that account for the object- and the feature-to-cluster assignments of the projective clusterings in $\mathcal{E}$, respectively. Note that the only constraint in the above formulation is that $\mathcal{C}$ must be a well-defined projective clustering solution, as given in Def. 2.

Similarly to the function Ψ in standard clustering ensembles, the functions Ψ_o and Ψ_f can be defined using a *clustering-based* approach, which involves a direct comparison with the projective clustering solutions of the projective ensemble. Formally, we have:

$$\Psi_o(\mathcal{C}, \mathcal{E}) = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \overline{\psi}_o(\mathcal{C}, \hat{\mathcal{C}}) \tag{3}$$

$$\Psi_f(\mathcal{C},\mathcal{E}) = \sum_{\hat{\mathcal{C}}\in\mathcal{E}} \overline{\psi}_f(\mathcal{C},\hat{\mathcal{C}}) \tag{4}$$

where $\overline{\psi}_o$ (respectively, $\overline{\psi}_f$) is a function that measures the distance between any two projective clustering solutions in terms of their corresponding object-to-cluster assignment (respectively, feature-to-cluster assignments).

In principle, $\overline{\psi}_o$ and $\overline{\psi}_f$ might be defined by resorting to any well-known external criterion for comparing clusterings (see, e.g., [55, 66]), and adapting such criteria in order to focus on either the object-based or feature-based representation of the projective clusters within the projective clusterings to be compared. In this respect, as both hard and soft object- and feature-to-cluster assignments should be taken into account, a reasonable choice would be the well-founded *clustering error* measure. This measure is defined as the scaled sum of the non-diagonal elements of the so-called *confusion matrix* (the matrix storing the pairwise distances between the clusters of the two clusterings to be compared), minimized over all possible permutations of rows and columns. The fastest method known for computing clustering error is the Hungarian algorithm [49], which has a complexity that is cubic in the sizes of the clusterings to be compared. However, due to its computational cost, the algorithm does not scale well for many applications. We therefore adopt here a slightly modified version of the clustering error measure, which has the advantage of being quadratic w.r.t. the sizes of the clusterings to be compared. In particular, we define $\overline{\psi}_o$ and $\overline{\psi}_f$ as follows:

$$\overline{\psi}_o(\mathcal{C}',\mathcal{C}'') = \frac{1}{2}\Big(\psi_o(\mathcal{C}',\mathcal{C}'') + \psi_o(\mathcal{C}'',\mathcal{C}')\Big) \tag{5}$$

$$\overline{\psi}_f(\mathcal{C}',\mathcal{C}'') = \frac{1}{2}\Big(\psi_f(\mathcal{C}',\mathcal{C}'') + \psi_f(\mathcal{C}'',\mathcal{C}')\Big) \tag{6}$$

where

$$\psi_o(\mathcal{C}',\mathcal{C}'') = \frac{1}{|\mathcal{C}'|}\sum_{C'\in\mathcal{C}'}\Big(1-\max_{C''\in\mathcal{C}''} J\big(\mathbf{\Gamma}_{C'},\mathbf{\Gamma}_{C''}\big)\Big)$$

$$\psi_f(\mathcal{C}',\mathcal{C}'') = \frac{1}{|\mathcal{C}'|}\sum_{C'\in\mathcal{C}'}\Big(1-\max_{C''\in\mathcal{C}''} J\big(\mathbf{\Delta}_{C'},\mathbf{\Delta}_{C''}\big)\Big)$$

with $J\big(\mathbf{u},\mathbf{v}\big) = \big(\mathbf{u}\cdot\mathbf{v}\big)/\big(\|\mathbf{u}\|^2+\|\mathbf{v}\|^2-\mathbf{u}\cdot\mathbf{v}\big) \in [0,1]$ denoting the extended Jaccard similarity coefficient between any two real-valued vectors $\mathbf{u}$ and $\mathbf{v}$. The extended Jaccard coefficient J is one of the most used distance measures between real-valued vectors, being in general a trade-off solution between Euclidean and Cosine measures in terms of scale/translation invariance [72]; moreover, it has a fixed-range codomain, $[0,1]$, as required in the proposed two-objective PCE formulation in order to make the values of the two objective functions Ψ_o and Ψ_f scale-independent w.r.t. each other.

Defining ψ_o and ψ_f according to (5) and (6) allows for overcoming the complexity of computing clustering error, as both (5) and (6) can be computed in $\mathcal{O}(|\mathcal{C}'|\,|\mathcal{C}''|)$. As a side effect, ψ_o and ψ_f are asymmetric; this essentially depends on the "cluster alignment" between $\mathcal{C}'$ and $\mathcal{C}''$, which may not be one-to-one unlike classic clustering error measure; indeed, a cluster of $\mathcal{C}''$ could be mapped to multiple clusters of $\mathcal{C}'$, implying that some clusters of $\mathcal{C}''$ could not be assigned at all.

The MOEA-PCE algorithm. As stated previously, each of the two objectives of the optimization problem reported in (2) is close to the classic formulation employed by standard clustering ensembles. Thus, both the objectives refer to the median partition problem, which has been proved to be NP-hard in [48]. Due to the hardness of the problem at hand, we focus on the development of heuristics in order to compute accurate approximations. The definition of such heuristics would be easier if the two objectives were not conflicting; in that case, in fact, the two-objectives could be easily collapsed into a single one. Unfortunately, the problem in (2) involves two conflicting objectives, as shown next.

Proposition 1 *The two objective functions Ψ_o and Ψ_f of the problem defined in (2) are conflicting w.r.t. one another.* □

Traditional optimization methods are not good choices for multi-objective problems whose objectives are conflicting. For example, a conventional approach to solving this kind of problems consists in defining a single-objective problem whose optimization function is computed as a weighted linear combination of the functions in the original problem. Unfortunately, this approach has several drawbacks [18]: it mixes non-commensurable objectives, involves a hard setting of the weights to assign to each function, and requires prior knowledge of the application domain.

A more refined approach that has been recognized as particularly appropriate in providing valid solutions for the problem at hand is given by Pareto-based *Multi-Objective Evolutionary Algorithms (MOEAs)* [18]. This class of methods is able to solve a multi-objective problem while maintaining the underlying multi-objective structure of the given problem, i.e., without combining the various objective functions into a single one.

Within this view, we define the *MOEA-based Projective Clustering Ensembles (MOEA-PCE)* algorithm. In particular, we resort to the *Nondominated Sorting Genetic Algorithm-II (NSGA-II)* [19], whose multi-objective quality assessment strategy is in part based on the notion of *Pareto-ranking*. In the following we provide the necessary definitions.

Definition 4 (Domination) Let P be a multi-objective optimization problem of the form: $x^* = \arg\min_x \{\mathrm{f}_1(x), \ldots, \mathrm{f}_n(x)\}$. Let x' and x'' be two candidate solutions of P. x' dominates x'' ($x' \prec x''$) if and only if $\mathrm{f}_i(x') \leq \mathrm{f}_i(x''), \ \ \forall i \in \{1, \ldots, n\}$, and $\exists j \in \{1, \ldots, n\}$ s.t. $\mathrm{f}_j(x') < \mathrm{f}_j(x'')$. □

Definition 5 (Pareto-nondominated) Let P be a multi-objective optimization problem of the form: $x^* = \arg\min_x \{\mathrm{f}_1(x), \ldots, \mathrm{f}_n(x)\}$. Let $\mathcal{S}$ be a population of individuals for P, i.e., a set of candidate solutions of P. $\mathcal{S}^*_P \subseteq \mathcal{S}$ is a Pareto-nondominated solution set of P w.r.t. $\mathcal{S}$ if and only if $x \not\prec x^*, \forall x \in \mathcal{S}, \forall x^* \in \mathcal{S}^*_P$. □

Definition 6 (Pareto-ranking) Let P be a multi-objective optimization problem of the form: $x^* = \arg\min_x \{\mathrm{f}_1(x), \ldots, \mathrm{f}_n(x)\}$. Let $\mathcal{S}$ be a population of individuals for P. The Pareto-ranking function $\rho : \mathcal{S} \rightarrow \mathbb{N}^+$ for P is defined recursively as follows. Let $\mathcal{S}_1 = \mathcal{S}$. For any given set of individuals $\mathcal{S}_i$, the Pareto rank of any x belonging to the maximal Pareto-nondominated solution set $\mathcal{S}^*_{P,i}$ of P w.r.t. $\mathcal{S}_i$ is defined to be i (i.e., $\rho(x) = i, \forall x \in \mathcal{S}^*_{P,i}$), and $\mathcal{S}_{i+1} = \mathcal{S}_i \setminus \mathcal{S}^*_{P,i}$. □

Algorithm 1 MOEA-PCE

Input: A projective ensemble $\mathcal{E}$ defined over a set $\mathcal{D}$ of data objects; the number K of clusters in the output projective consensus clusterings; the population size t; the maximum number I of iterations
Output: A projective consensus clustering $\mathcal{C}^*$

1: $\mathcal{S} \leftarrow populationRandomGen(\mathcal{E}, t, K)$
2: $it \leftarrow 1$
3: **repeat**
4: $\quad \rho \leftarrow computeParetoRanking(\mathcal{S})$ {cf. Def. 6}
5: $\quad \langle \mathcal{S}', \mathcal{S}'' \rangle \leftarrow \langle \check{\mathcal{S}}' \subset \mathcal{S}, \check{\mathcal{S}}'' \subset \mathcal{S} \rangle : |\check{\mathcal{S}}'| = |\mathcal{S}|/2, |\check{\mathcal{S}}''| = |\mathcal{S}|/2, \check{\mathcal{S}}' \cup \check{\mathcal{S}}'' = \mathcal{S}, \rho(x') \leq \rho(x''), \forall x' \in \check{\mathcal{S}}', x'' \in \check{\mathcal{S}}''$
6: $\quad \mathcal{S}'_{CM} \leftarrow crossoverAndMutation(\mathcal{S}')$
7: $\quad \mathcal{S} \leftarrow \mathcal{S}' \cup \mathcal{S}'_{CM}$
8: $\quad it \leftarrow it + 1$
9: **until** $it = I$
10: $\rho \leftarrow computeParetoRanking(\mathcal{S})$
11: $\mathcal{S}^* \leftarrow \{x' \in \mathcal{S} : \rho(x') \leq \rho(x''), \forall x'' \in \mathcal{S}, x'' \neq x'\}$
12: select $\mathcal{C}^*$ from $\mathcal{S}^*$

Let us informally explain Defs. 4-6. Essentially, Def. 4 states that a solution x' dominates any other solution x'' if and only if the value of x' is strictly lower than that of x'' according to at least one objective function, while the two values are at most equal to each other according to the remaining functions. Given a set of solutions, the corresponding Pareto-nondominated subset (Def. 5) is composed by all individuals that are not dominated by any other solution in the given set. The Pareto-ranking function (Def. 6) aims to assign a "score" (i.e., a rank in the form of a positive integer) to each solution of any given population $\mathcal{S}$. In particular, all the nondominated solutions in $\mathcal{S}$ (denoted by $S^*_{P,1}$) have rank 1, and the rank of the remaining solutions is recursively assigned by considering the nondominated solutions that do not have a rank yet. For instance, all the nondominated solutions in $\mathcal{S} \setminus \mathcal{S}^*_{P,1}$ have rank 2 and form the set $\mathcal{S}^*_{P,2}$, all the nondominated solutions in $(\mathcal{S} \setminus \mathcal{S}^*_{P,1}) \setminus \mathcal{S}^*_{P,2}$ have rank 3 and form the set $\mathcal{S}^*_{P,3}$, and so on until a rank is assigned to all the solutions in $\mathcal{S}$.

The Pareto-ranking function ρ is exploited by the proposed MOEA-PCE algorithm (Algorithm 1) to perform the steps of *selection*, *crossover*, and *mutation*. The algorithm starts by randomly generating an initial population (set of candidate solutions) $\mathcal{S}$ (Line 1), and proceeds by performing the main loop until a maximum number I of iterations has been reached (Lines 3-9). At each iteration, the Pareto-ranking function ρ, defined w.r.t. the current population $\mathcal{S}$, is computed according to Def. 6, where the problem denoted with P is the one reported in (2) (Line 4). The ranking function ρ is computed by following the procedure described in [19]. The ρ values of each candidate solution in $\mathcal{S}$ are then exploited for sorting $\mathcal{S}$, and partitioning it into two equally-sized subsets, i.e., $\mathcal{S}'$ and $\mathcal{S}''$, such that each candidate in $\mathcal{S}'$ has a ρ value not greater than any candidate in $\mathcal{S}''$ (Line 5). The subset $\mathcal{S}'$ undergoes a crossover-and-mutation step, which is performed by adding random Gaussian noise to each candidate solution in $\mathcal{S}'$ [70] (Line 6). The result of this step is the "offspring" set $\mathcal{S}'_{CM}$ of new candidates, which, together with $\mathcal{S}'$, form the new population (Line 7). Finally, the Pareto-optimal solution set $\mathcal{S}^*$ (i.e., the set of the output projective consensus clusterings) is derived from the population $\mathcal{S}$ computed at the last iteration of the algorithm (Line 11). From this set, one Pareto-optimal solution is eventually selected to provide a single projective consensus clustering (Line 12).

4.2 Single-objective PCE

Pareto-based MOEAs provide a valuable solution to the proposed two-objective PCE formulation in terms of effectiveness and adaptability to the multi-objective optimization context. Nevertheless, MOEAs typically incur a number of intrinsic issues:

1) Inefficiency, mostly due to the fact that the number I of iterations needed for achieving good solutions is usually large;
2) Difficult setting for the parameters I and t (population size), for which no guidance or prior knowledge is available;
3) Difficult interpretation of the results. The algorithm outputs a set of projective consensus clusterings and, although one of such solutions can be always selected (e.g., randomly, as each one is however Pareto-optimal), the task of selection could in general be non-trivial.

As stated in the previous section, overcoming the above drawbacks by combining the two objective functions into a single one is not feasible. Therefore, the solution here adopted aims to provide an alternative formulation to PCE in terms of a single-objective function $Q(\mathcal{C}, \mathcal{E})$, whose details are given next.

4.2.1 Deriving the Single-objective PCE Optimization Function

A key notion for the definition of Q is the feature-based representation of any data object $\mathbf{o}$, which is provided next.

Definition 7 (feature-based representation) Let $\mathcal{E}$ be a projective ensemble defined over a set $\mathcal{D}$ of data objects, where each $\mathbf{o} \in \mathcal{D}$ is described by a set $\mathcal{F}$ of features. Moreover, let $A_{\mathbf{o},f}$ denote the event "*feature f is informative for $\mathbf{o}$ w.r.t. the projective ensemble $\mathcal{E}$*", $\forall \mathbf{o} \in \mathcal{D}$, $\forall f \in \mathcal{F}$. The *feature-based* representation of any object $\mathbf{o} \in \mathcal{D}$ is an $|\mathcal{F}|$-dimensional probability vector $\mathbf{\Lambda_o}$, where each component $\Lambda_{\mathbf{o},f}$ corresponds to the probability $\Pr(A_{\mathbf{o},f}|\mathcal{E})$ of the event $A_{\mathbf{o},f}$ given the information available from the projective ensemble $\mathcal{E}$. □

Proposition 2 *In reference to the expression $\Lambda_{\mathbf{o},f}$ and the event $A_{\mathbf{o},f}$ introduced in Def. 7, it holds that:*

$$\Lambda_{\mathbf{o},f} = \frac{1}{|\mathcal{E}|} \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} \Gamma_{\hat{C},\mathbf{o}} \, \Delta_{\hat{C},f} \tag{7}$$

□

The single-objective function Q is defined as an error criterion to be minimized that accounts for both the object- and feature-based representations of the solutions in the projective ensemble. To this end, for each cluster C within any candidate projective consensus clustering $\mathcal{C}$, Q should fulfill the following requirements:

1) Q is such that the object-to-cluster assignment $\Gamma_{C,\mathbf{o}}$ of any object $\mathbf{o}$ to C is directly proportional to how well $\mathbf{o}$ complies with the information from the projective ensemble about the object-to-cluster assignments $\{\Gamma_{\hat{C},\mathbf{o}'} \mid \hat{C} \in \hat{\mathcal{C}}, \hat{\mathcal{C}} \in \mathcal{E}\}$ of the other objects $\mathbf{o}' \neq \mathbf{o}$ within C;

2) larger values of Q correspond to a lower agreement of the feature-based representation $\mathbf{\Delta}_C$ of C with the information from the projective ensemble about the feature-based representations $\mathbf{\Lambda_o}$ of the objects $\mathbf{o}$ within C.

To satisfy the above requirements, we first take into account the error, denoted as $E_{C,\mathbf{o}}$, between the feature-based representation $\mathbf{\Delta}_C$ of cluster C and the feature-based representation $\mathbf{\Lambda_o}$ of any object $\mathbf{o}$. This error can be trivially defined by applying any distance measure to the vectors $\mathbf{\Delta}_C$ and $\mathbf{\Lambda_o}$. Therefore, choosing the squared Euclidean distance for its simplicity, it holds that $E_{C,\mathbf{o}} = \|\mathbf{\Delta}_C - \mathbf{\Lambda_o}\|^2 = \sum_{f\in\mathcal{F}}(\Delta_{C,f} - \Lambda_{\mathbf{o},f})^2$. $E_{C,\mathbf{o}}$ refers to the error of C w.r.t. only one object $\mathbf{o}$. To compute the overall error E_C, it is sufficient to sum up $E_{C,\mathbf{o}}$ over all objects, and weight each individual $E_{C,\mathbf{o}}$ by the probability $\Gamma_{C,\mathbf{o}}$ that $\mathbf{o}$ belongs to C, i.e., $E_C = \sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{C,\mathbf{o}}\, E_{C,\mathbf{o}} = \sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{C,\mathbf{o}} \sum_{f\in\mathcal{F}}(\Delta_{C,f} - \Lambda_{\mathbf{o},f})^2$.

It is easy to see that E_C gives a measure of how well the feature-based representation $\mathbf{\Delta}_C$ of the cluster C in the candidate projective consensus clustering complies with the feature-based representations $\mathbf{\Lambda_o}$ of all objects $\mathbf{o}$ within C, according to the information from the projective ensemble. E_C hence meets the aforementioned requirement 2). Moreover, it is straightforward to see that E_C implicitly measures how the feature-based representations $\mathbf{\Lambda_o}$, $\mathbf{\Lambda_{o'}}$ of any two objects $\mathbf{o}$, $\mathbf{o}'$ within C are close to each other. Since Proposition 3 reported next shows that the latter condition is strictly related to the closeness of $\mathbf{o}$ to $\mathbf{o}'$ also in terms of their corresponding object-to-cluster assignments, thus E_C fulfills requirement 1) as well.

Proposition 3 *Let $\mathcal{E}$ be a projective ensemble defined over a set $\mathcal{D}$ of data objects, where each $\mathbf{o} \in \mathcal{D}$ is described by a set $\mathcal{F}$ of features. Given any two objects $\mathbf{o}, \mathbf{o}' \in \mathcal{D}$, let $d_{\mathbf{o},\mathbf{o}'}$ be the squared Euclidean distance between the object-to-cluster assignments of $\mathbf{o}$ and $\mathbf{o}'$ to the various clusters of all the solutions in $\mathcal{E}$, i.e., $d_{\mathbf{o},\mathbf{o}'} = \sum_{\hat{\mathcal{C}}\in\mathcal{E}} \sum_{\hat{C}\in\hat{\mathcal{C}}} (\Gamma_{\hat{C},\mathbf{o}} - \Gamma_{\hat{C},\mathbf{o}'})^2$. It holds that the squared Euclidean distance $\|\mathbf{\Lambda_o} - \mathbf{\Lambda_{o'}}\|^2$ between the feature-based representations of $\mathbf{o}$ and $\mathbf{o}'$ is directly proportional to $d_{\mathbf{o},\mathbf{o}'}$.* □

In summary, E_C puts in relation both the object- and feature-to-cluster assignments of any cluster C within the candidate projective consensus clustering $\mathcal{C}$ with the information available from the projective ensemble. Summing up E_C over all clusters in $\mathcal{C}$ provides the desired error criterion to be used in the resulting objective function Q, as it satisfies both the requirements previously discussed.

According to the above reasoning, we introduce the following single-objective PCE formulation:

$$\mathcal{C}^* = \arg\min_{\mathcal{C}} Q(\mathcal{C}, \mathcal{E}) \tag{8}$$

$$s.t.$$

$$\sum_{C\in\mathcal{C}} \Gamma_{C,\mathbf{o}} = 1, \quad \forall \mathbf{o} \in \mathcal{D} \tag{9}$$

$$\Gamma_{C,\mathbf{o}} \geq 0, \Delta_{C,f} \geq 0, \Delta_{C,f} \leq 1, \quad \forall C, \forall \mathbf{o}, \forall f \tag{10}$$

where

$$Q(\mathcal{C}, \mathcal{E}) = \sum_{C\in\mathcal{C}} \sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{C,\mathbf{o}}^{\alpha} \sum_{f\in\mathcal{F}} (\Delta_{C,f} - \Lambda_{\mathbf{o},f})^2 \tag{11}$$

and α is a positive integer whose rationale is as follows. Denoting by P the optimization problem defined in (8)-(10), if we set $\alpha = 1$ both the objective function and the constraints of P become linear w.r.t. $\mathit{\Gamma}_{C,\mathbf{o}}$. This would be sufficient for the fundamental property of linear programming, which states that the optimal solution of a linear programming problem is at a vertex of the hyper-polygon of the feasible region, to apply for P. In this case, the optimal solution for P would have hard object-to-cluster assignments, i.e., $\mathit{\Gamma}_{C,\mathbf{o}} \in \{0,1\}$. Thus, in order to have more general solutions (i.e., solutions whose object-to-cluster assignments range within $[0,1]$), we require the parameter α to be a positive integer greater than 1. The value of α controls the softness of the optimal solutions of P, i.e., the larger α is, the higher the sparsity of the $\mathit{\Gamma}_{C,\mathbf{o}}$ values is, and vice versa.

4.2.2 The EM-PCE Algorithm

The optimization problem defined in (8)-(10) is close to a formulation of the traditional clustering problem based on a Sum of Squared Error (SSE) minimization. Thus, such a problem can be easily proved to be NP-hard. We provide a heuristic solution by defining a novel procedure inspired by the popular *Expectation Maximization (EM)* algorithm [20].

The proposed algorithm, called *EM-based Projective Clustering Ensembles (EM-PCE)* (Alg. 2), consists of two main EM-like steps (i.e., *expectation* step and *maximization* step), which are iteratively repeated until a convergence criterion is met. Function Q (11) is used to find the optimal values for $\mathit{\Gamma}_{C,\mathbf{o}}$ ($\mathit{\Delta}_{C,f}$, respectively), while keeping $\mathit{\Delta}_{C,f}$ ($\mathit{\Gamma}_{C,\mathbf{o}}$, respectively) fixed. The optimal solutions for $\mathit{\Gamma}_{C,\mathbf{o}}$ and $\mathit{\Delta}_{C,f}$ ($\forall C, \forall \mathbf{o}, \forall f$) are given by the following equations:

$$\mathit{\Gamma}^*_{C,\mathbf{o}} = \left[\sum_{C' \in \mathcal{C}} \left(\frac{X_{C,\mathbf{o}}}{X_{C',\mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-1} \tag{12}$$

$$\mathit{\Delta}^*_{C,f} = \frac{Z_{C,f}}{Y_C} \tag{13}$$

where

$$X_{C,\mathbf{o}} = \sum_{f \in \mathcal{F}} \left(\mathit{\Delta}_{C,f} - \mathit{\Lambda}_{\mathbf{o},f} \right)^2 \tag{14}$$

$$Y_C = \sum_{\mathbf{o} \in \mathcal{D}} \mathit{\Gamma}^{\alpha}_{C,\mathbf{o}} \tag{15}$$

$$Z_{C,f} = \sum_{\mathbf{o} \in \mathcal{D}} \mathit{\Gamma}^{\alpha}_{C,\mathbf{o}} \, \mathit{\Lambda}_{\mathbf{o},f} \tag{16}$$

In the following we prove that *(i)* (12) and (13) provide the optimal solution of the problem P defined in (8)-(10), where either $\mathit{\Delta}_{C,f}$ or $\mathit{\Gamma}_{C,\mathbf{o}}$ are kept fixed, and *(ii)* the convergence of Alg. 2. All the following results refer to a set $\mathcal{D}$ of data objects, a set $\mathcal{F}$ of features, a candidate projective clustering solution $\mathcal{C}$, and a projective ensemble $\mathcal{E}$.

Algorithm 2 EM-PCE

Input: A projective ensemble $\mathcal{E}$; the number K of clusters in the output projective consensus clustering;
Output: the projective consensus clustering $\mathcal{C}^*$

1: $\mathcal{C}^* \leftarrow randomGen(\mathcal{E}, K)$
2: **repeat**
3: **for all** $C^* \in \mathcal{C}^*$ **do**
4: compute Γ_{C^*} according to (12)
5: compute Δ_{C^*} according to (13)
6: **end for**
7: **until** $convergence$

Theorem 1 *For the problem P defined in (8)-(10), it holds that:*

1) Given the current values for $\Delta_{C,f}$, (12) computes the optimal $\Gamma^*_{C,\mathbf{o}}$, $\forall C$, $\forall \mathbf{o}$
2) Given the current values for $\Gamma_{C,\mathbf{o}}$, (13) computes the optimal $\Delta^*_{C,f}$, $\forall C$, $\forall f$

□

Theorem 2 *The EM-PCE algorithm (Alg. 2) converges to a local minimum of the function Q defined in (11) in a finite number of steps.* □

4.3 Complexity Analysis

We now discuss the computational complexity of the proposed MOEA-PCE (Alg. 1) and EM-PCE (Alg. 2) algorithms; such complexities are summarized in Table 2. We are given: a set $\mathcal{D}$ of data objects, each one defined over a feature space $\mathcal{F}$, a projective ensemble $\mathcal{E}$ defined over $\mathcal{D}$, a positive integer K representing the number of clusters in the output projective consensus clustering, and the size t of the population (for MOEA-PCE). We also assume that the number of clusters of each solution in $\mathcal{E}$ is bounded by $\mathcal{O}(K)$.

MOEA-PCE. The costs of the various steps of MOEA-PCE algorithm (Alg. 1) are summarized as follows:

- the random initialization step (Line 1) is $\mathcal{O}(t\ K\ (|\mathcal{D}| + |\mathcal{F}|))$;
- the *computeParetoRanking* function (Line 4) has two steps: *(i)* the computation of the values of the functions Ψ_o (cf. (3)) and Ψ_f (cf. (4)), for each of the t new individuals in $\mathcal{S}$, which costs $\mathcal{O}(t\ K^2\ |\mathcal{E}|\ (|\mathcal{D}| + |\mathcal{F}|))$, and *(ii)* the computation of the ρ values for $\mathcal{S}$, which is performed in $\mathcal{O}(t^2)$, according to the procedure described in [19]. Therefore, since t is $\mathcal{O}(|\mathcal{E}|)$, the total cost of *computeParetoRanking* is $\mathcal{O}(t\ K^2\ |\mathcal{E}|\ (|\mathcal{D}| + |\mathcal{F}|))$;
- the partitioning of $\mathcal{S}$ into the subsets $\mathcal{S}'$ and $\mathcal{S}''$ (Line 5) costs $\mathcal{O}(t\ \log t)$;
- the crossover & mutation operations (Line 6) are performed in $\mathcal{O}(t\ K\ (|\mathcal{D}| + |\mathcal{F}|))$;
- the computation of the set $\mathcal{S}^*$ and the output $\mathcal{C}^*$ (Lines 10-11 and 12) costs $\mathcal{O}(t\ K^2\ |\mathcal{E}|\ (|\mathcal{D}| + |\mathcal{F}|))$ and $\mathcal{O}(t)$, respectively.

Since the steps of the main loop (Lines 3-9) are repeated I times, MOEA-PCE has an overall complexity of $\mathcal{O}(I\ t\ K^2\ |\mathcal{E}|\ (|\mathcal{D}| + |\mathcal{F}|))$.

Table 2 Computational complexities of the proposed algorithms

	MOEA-PCE	*EM-PCE*
offline	—	$\mathcal{O}(K \, \lvert\mathcal{E}\rvert \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert)$
online	$\mathcal{O}(I \, t \, K^2 \, \lvert\mathcal{E}\rvert \, (\lvert\mathcal{D}\rvert + \lvert\mathcal{F}\rvert))$	$\mathcal{O}(\mathcal{I} \, K \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert)$
total	$\mathcal{O}(I \, t \, K^2 \, \lvert\mathcal{E}\rvert \, (\lvert\mathcal{D}\rvert + \lvert\mathcal{F}\rvert))$	$\mathcal{O}(K \, \lvert\mathcal{E}\rvert \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert)$

We point out that each step of MOEA-PCE is performed *online*, for each run of the algorithm, in case of a multi-run execution.

EM-PCE. It consists of two phases (cf. Alg. 2): an *offline* phase, with operations to be executed only once in case of multi-run executions, and an *online* phase, whose operations are repeated for each iteration of the algorithm, until convergence. Let us analyze both phases in detail.

- *Offline phase*: it computes (7) (i.e., $\mathit{\Lambda}_{\mathbf{o},f} = \sum_{\hat{C} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} \mathit{\Gamma}_{\hat{C},\mathbf{o}} \, \mathit{\Delta}_{\hat{C},f}$), $\forall \mathbf{o}$, $\forall f$ at a total cost of $\mathcal{O}(K \, \lvert\mathcal{E}\rvert \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert)$;
- *Online*: it computes (12) and (13), which require in turn the computation of $X_{C,\mathbf{o}} = \sum_{f \in \mathcal{F}} (\mathit{\Delta}_{C,f} - \mathit{\Lambda}_{\mathbf{o},f})^2$, $Y_C = \sum_{\mathbf{o} \in \mathcal{D}} \mathit{\Gamma}^{\alpha}_{C,\mathbf{o}}$, and $Z_{C,f} = \sum_{\mathbf{o} \in \mathcal{D}} \mathit{\Gamma}^{\alpha}_{C,\mathbf{o}} \, \mathit{\Lambda}_{\mathbf{o},f}$ (cf. (14)-(16)). The individual $X_{C,\mathbf{o}}$, Y_C, and $Z_{C,f}$ take $\mathcal{O}(\lvert\mathcal{F}\rvert)$, $\mathcal{O}(\lvert\mathcal{D}\rvert)$, and $\mathcal{O}(\lvert\mathcal{D}\rvert)$, respectively, as the $\mathit{\Lambda}_{\mathbf{o},f}$ terms values are already available from the offline phase. Since these terms must be computed $\forall C$, $\forall \mathbf{o}$ ($X_{C,\mathbf{o}}$), $\forall C$ (Y_C), and $\forall C$, $\forall f$ ($Z_{C,f}$), the total cost of the online phase is $\mathcal{O}(2 \, K \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert + K \, \lvert\mathcal{D}\rvert)$, i.e., $\mathcal{O}(K \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert)$.

The online steps are repeated $\mathcal{I}$ times, where $\mathcal{I}$ is the number of iterations needed for convergence, where typically $\mathcal{I} \ll I$ (i.e., the number of iterations $\mathcal{I}$ required for EM-PCE to converge is typically much smaller than the number of iterations I needed by MOEA-PCE). In conclusion, the overall computational complexity of EM-PCE is $\mathcal{O}(K \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert \, (\mathcal{I} + \lvert\mathcal{E}\rvert))$, i.e., $\mathcal{O}(K \, \lvert\mathcal{E}\rvert \, \lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert)$, as typically $\mathcal{I}$ is $\mathcal{O}(\lvert\mathcal{E}\rvert)$.

Interpretation of the complexity results. To analyze in detail the computational costs of the proposed MOEA-PCE and EM-PCE algorithms, we interpret the (total) complexity results reported in Table 2. The relative complexity "gap" of MOEA-PCE w.r.t. EM-PCE is defined as the ratio between the corresponding complexities. Noting that, among the parameters used for expressing the computational complexities, $\lvert\mathcal{D}\rvert$, $\lvert\mathcal{F}\rvert$ and K vary in accordance with the selected dataset, whereas I and t do not, we are interested in expressing such a gap right in terms of $\lvert\mathcal{D}\rvert$, $\lvert\mathcal{F}\rvert$ and K; hence, this gap is equal to $\mathcal{O}(r(\lvert\mathcal{D}\rvert, \lvert\mathcal{F}\rvert, K))$, where r is a function of $\lvert\mathcal{D}\rvert$, $\lvert\mathcal{F}\rvert$ and K expressed as:

$$r(\lvert\mathcal{D}\rvert, \lvert\mathcal{F}\rvert, K) = \frac{I \, t \, K \, (\lvert\mathcal{D}\rvert + \lvert\mathcal{F}\rvert)}{\lvert\mathcal{D}\rvert \, \lvert\mathcal{F}\rvert}$$

Let us analyze the conditions under which $r(\lvert\mathcal{D}\rvert, \lvert\mathcal{F}\rvert)$ is greater than one, i.e., when EM-PCE is less expensive than MOEA-PCE.

Proposition 4 *It holds that* $r(\lvert\mathcal{D}\rvert, \lvert\mathcal{F}\rvert) > 1$ *if* $(\lvert\mathcal{D}\rvert + \lvert\mathcal{F}\rvert) / K < 4 \, I \, t$. □

Table 3 Datasets used in the experiments

dataset	*objects*	*features*	*classes*
Iris	150	4	3
Wine	178	13	3
Glass	214	10	6
Ecoli	327	7	5
Yeast	1,484	8	10
Multiple-Features	2,000	585	10
Segmentation	2,310	19	7
Abalone	4,124	7	17
Waveform	5,000	40	3
Letter	7,648	16	10
Isolet	7,797	617	26
Gisette	13,500	5,000	2
p53-Mutants	300	5,409	2
Amazon	120	10,000	4
Arcene	200	10,000	2
Shapes	160	1,614	9
Tracedata	200	275	4
ControlChart	600	60	6
Twopat	800	128	4
N30	1,356	20	8
D75	1,365	75	7
S2500	2,262	20	8

Proposition 4 states that EM-PCE is less expensive than MOEA-PCE when the ratio $(|\mathcal{D}| + |\mathcal{F}|)/K$ is below $4\ I\ t\ K$. Thus, the smaller the sum $|\mathcal{D}| + |\mathcal{F}|$ and/or the larger K is, the more efficient EM-PCE w.r.t. MOEA-PCE is. However, it can be noted that the condition $(|\mathcal{D}| + |\mathcal{F}|)/K < 4\ I\ t\ K$ is true in a large number of real cases. As an example, in fact, considering the numerical values of I and t employed in our experiments (cf. Sect. 5), i.e., $I = 200$, $t = 60$, and varying K within $\{8, \ldots, 26\}$ (i.e., the range bounded by the average and the maximum number of clusters over all datasets considered in our experiments), it results that: $|\mathcal{D}| + |\mathcal{F}|$ should be upper bounded by a value within the range $[384,000, \ldots, 1,248,000]$ to have $r(|\mathcal{D}|, |\mathcal{F}|, K) > 1$. This condition is satisfied in many real cases, making EM-PCE less expensive than MOEA-PCE in practice. Nevertheless, for huge datasets, it could happen for EM-PCE to be outperformed by MOEA-PCE.

5 Experimental Evaluation

Our experimental evaluation was aimed to assess accuracy and efficiency of the projective consensus clusterings obtained by the proposed MOEA-PCE and EM-PCE algorithms. In the following, we introduce the evaluation methodology which includes the selected datasets, the strategy used for generating the projective ensembles, the setup of the proposed algorithms, the measures to assess the quality of the projective consensus clusterings, and the baselines adopted to validate the results of the proposed methods. Next, we present and discuss experimental results obtained on the evaluation datasets, and finally conclude this section with a real-life case study that demonstrates the applicability of our PCE approach.

5.1 Evaluation Methodology

5.1.1 Datasets

We selected 22 publicly available datasets having different characteristics in terms of number of objects, features and classes, which are summarized in Table 3. A brief description for each dataset is given next.

- Fifteen datasets from the UCI Machine Learning Repository [7], namely Iris, Wine, Glass, Ecoli, Yeast, Multiple-Features, Segmentation, Abalone, Waveform, Letter, Isolet, Gisette, p53-Mutants, Amazon, and Arcene. Iris refers to measurements on different iris plants. Wine represents results of a chemical analysis of Italian wines derived from three different cultivars. Glass contains glass instances that are described by their chemical components. Ecoli contains data on the Escherichia Coli bacterium, which are identified with values coming from different analyses. Multiple-Features concerns binary images representing handwritten digits (0-9) extracted from a collection of Dutch utility maps. Yeast objects describe the main features and the localization of various proteins. Segmentation represents objects that were randomly drawn from a database of seven outdoor images; the images (3x3 regions) were hand-segmented to create a classification for each pixel. Abalone is about different types of abalone shells. Waveform contains data synthetically generated as a combination of two among three "base" waves. Letter contains character images corresponding to the capital letters in the English alphabet. Isolet contains recording of the name of each letter of the alphabet spoken by several subjects. Gisette is about the handwritten digit recognition problem of separating the highly confusible digits '4' and '9'. p53-Mutants concerns biophysical models of mutant p53 proteins and yields features which can be used to predict p53 transcriptional activity. Amazon dataset is derived from the customers' reviews in Amazon Commerce Website for authorship identification. Arcene is a mass-spectrometry dataset where the features indicate the abundance of proteins in human sera having a given mass value.
- Four time-series datasets from the UCR Time Series Classification/Clustering Page [45], namely Shapes, Tracedata, ControlChart, and Twopat. Shapes contains time series derived from shapes of nine different objects, i.e., bone, cup, device, fork, glass, hand, pencil, rabbit and tool. Tracedata simulates signals representing instrumentation failures. ControlChart represents synthetically generated control charts that are classified into one of the following: normal, cyclic, increasing trend, decreasing trend, upward shift, and downward shift. Twopat includes time series generated combining two different patterns (upward step and downward step); these patterns are used to define the classes down-down, up-down, down-up, and up-up.
- Three synthetically generated datasets, which were selected from [60, 61]. Three categories, namely *dimscale*, *dbsizescale*, and *noisescale*, were originally used in [60] for testing the scalability w.r.t. dataset size (dbsizescale), dimensionality (dimscale), and noise (noisescale), respectively. We selected one dataset for each category, denoted as S2500 for dbsizescale, D75 for dimscale, and N30 for

noisescale. As such datasets are synthetically generated, the natural subspaces assigned to the various groups of data objects are known. Moreover, since the datasets are originally provided in such a way that the corresponding clusters may overlap, we selected for each dataset the maximal subset of data objects forming a partition and the corresponding natural subspace for each cluster in the partition.

5.1.2 Projective Ensemble Generation

We adopted a basic strategy for projective ensemble generation, which consists in selecting a (projective) clustering algorithm and varying the parameter(s) of that algorithm in order to guarantee the diversity of the solutions within the projective ensemble. We would like to point out that we were not interested in comparing projective clustering algorithms and assessing the impact of their performance on projective ensemble generation, since generating projective ensembles with the highest quality is not a goal of this work; nevertheless, we resorted to a state-of-the-art algorithm, LAC, which has been experimentally proved as very effective in the context of projective clustering [23]. The diversity of the projective clustering solutions was ensured by randomly choosing the initial centroids and varying the parameter h (cf. Sect. 2). LAC yields projective clusterings that have hard object-to-cluster assignments and have weighted feature-to-cluster assignments. Therefore, in order to test the ability of the proposed algorithms to also deal with soft clustering solutions and with solutions having unweighted feature-to-cluster assignments, we generated each projective ensemble $\mathcal{E}$ as a composition of four equally-sized subsets, denoted as $\mathcal{E}_1$, $\mathcal{E}_2$, $\mathcal{E}_3$, and $\mathcal{E}_4$ and defined as follows:

- $\mathcal{E}_1$ is comprised of solutions that have hard object-to-cluster assignments and weighted feature-to-cluster assignments, i.e., solutions obtained by standard LAC;
- $\mathcal{E}_2$ is comprised of solutions that have hard object-to-cluster assignments and unweighted feature-to-cluster assignments. Starting from a LAC solution $\mathcal{C}$ defined over a set $\mathcal{D}$ of data objects and a set $\mathcal{F}$ of features, a projective clustering $\mathcal{C}'$ having unweighted feature-to-cluster assignments is derived such that $\Delta_{C',f} = \mathbf{I}\left[\Delta_{C',f} \geq |\mathcal{F}|^{-1} \sum_{f' \in \mathcal{F}} \Delta_{C',f'}\right], \forall C' \in \mathcal{C}',\ \forall f \in \mathcal{F}$, where $\mathbf{I}[A]$ is the *indicator function*, which is equal to 1 when the event A is true, 0 otherwise;
- $\mathcal{E}_3$ is comprised of solutions that have soft object-to-cluster assignments and weighted feature-to-cluster assignments. Starting from a LAC solution $\mathcal{C}$ defined over a set $\mathcal{D}$ of data objects and a set $\mathcal{F}$ of features, a soft projective clustering $\mathcal{C}''$ is derived by computing the $\Gamma_{C'',\mathbf{o}}$ values ($\forall C'' \in \mathcal{C}''$, $\forall \mathbf{o} \in \mathcal{D}$), proportionally to the distance of $\mathbf{o}$ from the centroids $\overline{C}''$ of the clusters C'':

$$\Gamma_{C'',\mathbf{o}} = \frac{\sum_{f \in \mathcal{F}} \left(o_f - \overline{C}''_f\right)^2}{\sum_{C \in \mathcal{C}''} \sum_{f \in \mathcal{F}} \left(o_f - \overline{C}_f\right)^2}$$

  where the f-th feature $\overline{C}_f$ of the centroid of any cluster C is defined as $\overline{C}_f = |C|^{-1} \sum_{\mathbf{o} \in C} o_f$.

– $\mathcal{E}_4$ is comprised of solutions that have soft object-to-cluster assignments and unweighted feature-to-cluster assignments. The solutions are derived from standard LAC solutions according to the methods employed for generating $\mathcal{E}_2$ and $\mathcal{E}_3$, respectively.

For each dataset, we generated 10 different projective ensembles; all results we present in the following refer to averages over these projective ensembles.

Unless otherwise specified, all experiments refer to the aforementioned ensemble generation strategy. Nevertheless, in order to analyze more deeply the performance w.r.t. the properties of the input ensemble, we also carried out a study varying other aspects in ensemble generation, such as the projective clustering method used for deriving the ensemble solutions and/or the number of output clusters. We report on this study at the end of Sect. 5.2.1.

5.1.3 Assessment Criteria

We assessed the quality of a projective consensus clustering $\mathcal{C}$ using both external and internal cluster validity approaches: the former is based on the similarity of $\mathcal{C}$ w.r.t. a reference classification, whereas the latter is based on the average similarity w.r.t. the solutions in the input projective ensemble $\mathcal{E}$.

Similarity w.r.t. the reference classification (external evaluation). This evaluation stage exploits the availability of a reference classification, denoted as $\widetilde{\mathcal{C}}$, for any given dataset $\mathcal{D}$. The object-to-cluster assignments, i.e., the $\Gamma_{\widetilde{C},\mathbf{o}}$ values, $\forall \widetilde{C} \in \widetilde{\mathcal{C}}$, $\forall \mathbf{o} \in \mathcal{D}$, are specified in a hard way. The $\Delta_{\widetilde{C},f}$ feature-to-cluster assignments are instead defined according to the following approaches:

– For the synthetic datasets N30, D75, and S2500, which already provide information about the subspaces assigned to each group of objects identified by the reference classification, these subspaces are directly employed to define unweighted $\Delta_{\widetilde{C},f}$ feature-to-cluster assignments in $\widetilde{C}$.
– For all remaining datasets, the $\Delta_{\widetilde{C},f}$ values are derived by applying the procedure suggested in [23] to the groups of objects identified by $\widetilde{\mathcal{C}}$. Formally, given the $\Gamma_{\widetilde{C},\mathbf{o}}$ values ($\forall \widetilde{C} \in \widetilde{\mathcal{C}}$, $\forall \mathbf{o} \in \mathcal{D}$) originally provided along with the reference classification $\widetilde{C}$, the $\Delta_{\widetilde{C},f}$ values are computed as:

$$\Delta_{\widetilde{C},f} = \frac{\exp\Big(-U(\widetilde{C},f)/h\Big)}{\sum_{f' \in \mathcal{F}} \exp\Big(-U(\widetilde{C},f')/h\Big)}$$

where the LAC parameter h is set equal to 0.2 and:

$$U(\widetilde{C},f) = \left(\sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{\widetilde{C},\mathbf{o}}\right)^{-1} \sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{\widetilde{C},\mathbf{o}} \big(\overline{C}_f - o_f\big)^2 \qquad \overline{C}_f = \left(\sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{\widetilde{C},\mathbf{o}}\right)^{-1} \sum_{\mathbf{o}\in\mathcal{D}} \Gamma_{\widetilde{C},\mathbf{o}} \times o_f$$

In order to compute similarity between a projective consensus clustering $\mathcal{C}$ and a reference classification $\widetilde{\mathcal{C}}$, we resort to the popular *F1-measure* [76]. This measure

has been previously used to evaluate subspace/projective clustering (e.g., [60]) and also been subject of examination in [40] for developing a (symmetric) F1-measure able to fulfill a number of desiderata specifically for subspace clustering, named as object, subspace, redundancy, and identification awareness. Here we provide a definition of F1-measure that enables a comparison between projective clustering having *soft* object/feature-to-cluster assignments. Given a projective cluster $C \in \mathcal{C}$, the precision $P(C)$ and the recall $R(C)$ are defined as:

$$P(C) = \frac{\max_{\widetilde{C} \in \widetilde{\mathcal{C}}} overlap(\widetilde{C}, C)}{size(C)} \qquad R(C) = \frac{\max_{\widetilde{C} \in \widetilde{\mathcal{C}}} overlap(\widetilde{C}, C)}{size(\arg\max_{\widetilde{C} \in \widetilde{\mathcal{C}}} overlap(\widetilde{C}, C))}$$

the $F1$-measure is defined as:

$$F1(\widetilde{\mathcal{C}}, \mathcal{C}) = \frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \frac{2\ P(C)\ R(C)}{P(C) + R(C)}$$

$F1$-measure ranges within $[0, 1]$, where higher values indicate more accurate projective consensus clusterings. The $overlap(\cdot, \cdot)$ and $size(\cdot)$ functions, which quantify the overlap degree between any two projective clusters and the size of any projective cluster, respectively, are defined according to a comparison performed among the various projective clusters. We considered all possible comparisons, i.e., *object-based (o)*, *feature-based (f)*, and *object* & *feature-based (of)*, which respectively account for the object-based representations only of the projective clusters to be compared, the feature-based representation only, or both. We define ad-hoc versions of the $overlap(\cdot, \cdot)$ and $size(\cdot)$ functions to handle all three types of comparison:

- object-based (measure $F1_o$): $overlap(C', C'') = \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C', \mathbf{o}}\ \Gamma_{C'', \mathbf{o}}$, $size(C) = \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C, \mathbf{o}}$
- feature-based (measure $F1_f$): $overlap(C', C'') = \sum_{f \in \mathcal{F}} \Delta_{C', f}\ \Delta_{C'', f}$, $size(C) = \sum_{f \in \mathcal{F}} \Delta_{C, f}$
- object & feature-based (measure $F1_{of}$): $overlap(C', C'') = \left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C', \mathbf{o}}\ \Gamma_{C'', \mathbf{o}}\right) \left(\sum_{f \in \mathcal{F}} \Delta_{C', f}\ \Delta_{C'', f}\right)$, $size(C) = \left(\sum_{\mathbf{o} \in \mathcal{D}}\ \Gamma_{C, \mathbf{o}}\right) \left(\sum_{f \in \mathcal{F}} \Delta_{C, f}\right)$

Similarity w.r.t. the projective ensemble solutions (internal evaluation). Any valid projective consensus clustering $\mathcal{C}$ should comply with the information available from the input projective ensemble $\mathcal{E}$. In this respect, we carried out an evaluation stage to measure the average similarity between any projective consensus clustering and the solutions within $\mathcal{E}$. We define the following object & feature-based measure $\overline{F1}_{of}$ (object-based $\overline{F1}_o$ and feature-based $\overline{F1}_f$ are defined similarly):

$$\overline{F1}_{of}(\mathcal{C}) = \frac{1}{|\mathcal{E}|} \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \max\{F1(\mathcal{C}, \hat{\mathcal{C}}), F1(\hat{\mathcal{C}}, \mathcal{C})\}$$

Clearly, the larger the values $\overline{F1}_{of}$, $\overline{F1}_o$, or $\overline{F1}_f$ are, the larger the similarity between the projective consensus clustering $\mathcal{C}$ and the solutions within the projective ensemble is, and hence the better the quality of $\mathcal{C}$. All these measures range within $[0, 1]$ as well.

5.1.4 Setting of the Proposed Algorithms

We performed a *leave-one-dataset-out* approach to set a specific parameter of a method based on the method performance in the remaining datasets. Roughly speaking, for each dataset the performance of a particular method on the other datasets was assessed for different values of the parameter, and the value that achieved the maximum $F1_{of}$ was then used to obtain a projective clustering solution for the left-out dataset.

In general, we observed that however the settings of the proposed methods were scarcely influenced by any specific dataset, which would indicate that a very easy setup can be performed on new datasets for which a reference classification or other a-priori knowledge is not available. Specifically, for the MOEA-PCE algorithm, the population size (t) was set to 50% of the projective ensemble size (i.e., equal to 60), and the number I of maximum iterations was set to 200; the random noise needed for the mutation step was obtained via *Monte Carlo* sampling on a standard Gaussian distribution. For the EM-PCE algorithm, the best-performance setting of parameter α of the objective function Q (cf. (11)) resulted in $\alpha = 2$, which leads to a minimal softness degree in the object-to-cluster assignments of the consensus clustering. Finally, the number of clusters in the projective ensemble solutions and in the projective consensus clusterings computed by MOEA-PCE and EM-PCE, was chosen as the same as the number of classes in the reference classification associated with each dataset.

5.1.5 Baselines

In order to comparatively evaluate the proposed PCE algorithms, we considered the following baselines:

- A method that computes a projective consensus clustering by randomly selecting a solution from the input projective ensemble $\mathcal{E}$. Here, the rationale is that when no additional information is provided along with $\mathcal{E}$, randomly extracting a projective clustering solution from $\mathcal{E}$ is likely the simplest and fairest comparison, in case no PCE method can be employed.
  To improve the robustness of this baseline, one should repeat a reasonably large number of times the process of selecting solutions from $\mathcal{E}$. It can be easily shown that this is equivalent to taking the average result across all solutions in $\mathcal{E}$. In fact, for the baseline, $F1_{of}$ can be computed as $\sum_{\mathcal{C}\in\mathcal{E}} F1_{of}(\mathcal{C})\ \Pr(\mathcal{C})$.[1] Since the probability $\Pr(\mathcal{C})$ of randomly selecting a solution $\mathcal{C}$ from $\mathcal{E}$ can be estimated as $|\mathcal{E}|^{-1}$, $\forall \mathcal{C} \in \mathcal{E}$, the previous expression becomes equal to the average $|\mathcal{E}|^{-1} \sum_{\mathcal{C}\in\mathcal{E}} F1_{of}(\mathcal{C})$ of the $F1_{of}$ results achieved by the various solutions in $\mathcal{E}$. For this purpose, this baseline is hereinafter referred to as *AVG-ensemble*.
- A method that computes the projective consensus clusterings so that: *(i)* the object-to-cluster assignment $\mathbf{\Gamma}$ values are derived by resorting to any traditional clustering ensemble algorithm which takes into account only the object-based representation information of the solutions in the projective ensemble;

[1] Analogous considerations hold for the other criteria $F1_o$, $F1_f$, $\overline{F1}_{of}$, $\overline{F1}_o$, and $\overline{F1}_f$.

Table 4 Evaluation w.r.t. the reference classification ($F1_{of}$)

dataset	*AVG-ensemble*	*MAX-CE*	*PROCLUS*	*LAC*	**MOEA-PCE**	**EM-PCE**
Iris	.574	.326	.436(.083)	.574(.051)	.649(.025)	.588(.002)
Wine	.273	.163	.393(.049)	.265(.019)	.345(.025)	.300(.003)
Glass	.224	.134	.251(.041)	.216(.039)	.279(.009)	.298(.007)
Ecoli	.454	.244	.481(.066)	.451(.050)	.518(.020)	.564(.013)
Yeast	.254	.137	.186(.028)	.256(.036)	.288(.009)	.237(.004)
Mult.-Feat.	.157	.083	.010(.006)	.028(.021)	.270(.019)	.300(.015)
Segmentation	.205	.132	.313(.070)	.085(.101)	.334(.018)	.400(.008)
Abalone	.110	.075	.102(.014)	.111(.004)	.116(.003)	.112(.003)
Waveform	.107	.313	.409(.027)	.057(.010)	.339(.056)	.338(.006)
Letter	.094	.183	.199(.029)	.084(.019)	.181(.025)	.155(.007)
Isolet	.125	.060	.287(.103)	.112(.019)	.141(.004)	.138(.001)
Gisette	.505	.136	.235(.148)	.507(.039)	.595(.015)	.532(.006)
p53-Mutants	.381	.205	.047(.052)	.377(.036)	.464(.021)	.411(.020)
Amazon	.370	.238	.280(.079)	.382(.057)	.441(.019)	.388(.006)
Arcene	.275	.013	.122(.040)	.288(.054)	.367(.012)	.142(.002)
Shapes	.204	.133	.157(.038)	.198(.023)	.243(.009)	.294(.007)
Tracedata	.387	.240	.458(.078)	.386(.022)	.438(.010)	.432(.012)
ControlChart	.019	.153	.354(.053)	.018(.002)	.092(.013)	.203(.020)
Twopat	.038	.190	.252(.020)	.018(.003)	.144(.025)	.070(.002)
N30	.050	.072	.061(.012)	.012(.001)	.098(.005)	.108(.003)
D75	.021	.021	.016(.003)	.004(.001)	.033(.002)	.038(.001)
S2500	.072	.072	.063(.013)	.012(.001)	.116(.004)	.122(.005)
min	.019	.013	.010	.004	.033	.038
max	.574	.326	.481	.574	.649	.588
avg	.223	.151	.232	.202	.295	.280

(ii) the feature-to-cluster assignments $\mathbf{\Delta}$ are computed randomly, since no well-founded strategy other than PCE can be employed here. Regarding the $\mathbf{\Gamma}$ values, we considered a number of clustering ensemble methods, i.e., those proposed in [71, 64, 13] (cf. Sect. 2.2); for the sake of brevity of presentation, we only reported the results achieved by the clustering ensemble method which has been recognized as the best one for each dataset and assessment criterion. This baseline is hereinafter referred to as *MAX-CE*.

– Two standard projective clustering methods, namely PROCLUS [3] and LAC [23] (cf. Sect. 2.1).[2]

5.2 Results

5.2.1 Accuracy

We present accuracy results of the projective consensus clusterings computed by the proposed MOEA-PCE and EM-PCE algorithms, as well as by the AVG-ensemble, MAX-CE, PROCLUS, and LAC baselines. Tables 4–6 report on the external evaluation w.r.t. the reference classification (assessment criteria $F1_{of}$, $F1_o$, and $F1_f$, respectively), whereas Tables 7–9 report on the internal evaluation w.r.t. the projective ensemble solutions (assessment criteria $\overline{F1}_{of}$, $\overline{F1}_o$, and $\overline{F1}_f$, respectively). For the randomized algorithms (i.e., PROCLUS, LAC, MOEA-PCE, and EM-PCE), all

[2] We used the PROCLUS implementation from the publicly available OpenSubspace framework [61].

Table 5 Evaluation w.r.t. the reference classification ($F1_o$)

dataset	*AVG-ensemble*	*MAX-CE*	*PROCLUS*	*LAC*	**MOEA-PCE**	**EM-PCE**
Iris	.864	.879	.657(.109)	.847(.095)	.967(.026)	.880(.002)
Wine	.681	.835	.515(.064)	.758(.076)	.835(.037)	.731(.009)
Glass	.363	.425	.354(.048)	.376(.058)	.474(.053)	.509(.012)
Ecoli	.672	.636	.536(.090)	.675(.046)	.760(.014)	.667(.021)
Yeast	.369	.341	.251(.044)	.372(.035)	.417(.007)	.333(.007)
Mult.-Feat.	.212	.488	.298(.047)	.178(.013)	.319(.069)	.369(.020)
Segmentation	.323	.550	.359(.087)	.270(.050)	.443(.062)	.568(.014)
Abalone	.187	.167	.132(.012)	.187(.010)	.208(.006)	.169(.005)
Waveform	.375	.500	.500(.039)	.368(.012)	.515(.070)	.415(.001)
Letter	.256	.341	.283(.035)	.230(.036)	.331(.030)	.306(.005)
Isolet	.866	.844	.579(.103)	.828(.149)	.959(.031)	.978(.001)
Gisette	.618	.667	.578(.076)	.623(.042)	.728(.015)	.674(.007)
p53-Mutants	.596	.585	.422(.113)	.598(.084)	.728(.038)	.619(.028)
Amazon	.447	.504	.383(.065)	.455(.076)	.555(.036)	.488(.009)
Arcene	.604	.628	.602(.023)	.614(.037)	.705(.009)	.626(.001)
Shapes	.609	.625	.518(.063)	.620(.037)	.681(.017)	.693(.015)
Tracedata	.537	.539	.595(.078)	.540(.019)	.614(.002)	.628(.059)
ControlChart	.262	.286	.543(.069)	.260(.027)	.319(.020)	.332(.003)
Twopat	.297	.400	.306(.016)	.304(.011)	.355(.011)	.296(.002)
N30	.502	.777	.628(.138)	.220(.001)	.807(.219)	.884(.013)
D75	.595	.752	.536(.109)	.247(.001)	.857(.146)	.952(.018)
S2500	.611	.788	.602(.138)	.220(.001)	.880(.156)	.895(.031)
min	.187	.167	.132	.178	.208	.169
max	.866	.879	.657	.847	.967	.978
avg	.493	.571	.463	.445	.612	.591

tables contain average results over 50 different runs along with the corresponding standard deviation (under brackets).

In the following, we (*i*) discuss results obtained by the various methods over all datasets, (*ii*) analyze in detail the statistical significance of the differences in performance by each method, (*iii*) present a graphical summary of the results which illustrates some major findings of our experimental study, and (*iv*) investigate a bit deeper on how the effectiveness of the selected methods relates to some ensemble properties such as the algorithm used for generating the ensemble and/or the number of clusters in the ensemble members.

Average performance. The results obtained by each method averaged over all datasets are summarized in the last row of Tables 4–9.

The evaluation w.r.t. the reference classification (Tables 4–6) shows that MOEA-PCE and EM-PCE outperformed all baselines according to each of the selected criteria. The average and maximum improvements achieved by MOEA-PCE were 0.120 and 0.169 (w.r.t. AVG-ensemble), 0.137 and 0.226 (w.r.t. MAX-CE), 0.070 and 0.149 (w.r.t. PROCLUS), and 0.149 and 0.186 (w.r.t. LAC). The average and maximum improvements by EM-PCE were 0.102 and 0.150 (w.r.t. AVG-ensemble), 0.108 and 0.174 (w.r.t. MAX-CE), 0.041 and 0.128 (w.r.t. PROCLUS), and 0.119 and 0.146 (w.r.t. LAC). MOEA-PCE performed better than EM-PCE in terms of all $F1_{of}$, $F1_o$ and $F1_f$ (improvements of 0.015, 0.021, and 0.052, respectively), whereas the two methods achieved similar results on $F1_{of}$. Concerning the baselines, MAX-CE was the worst method w.r.t. $F1_{of}$ and $F1_f$, whereas it was the best baseline in terms of $F1_o$, even though still worse than both MOEA-PCE and EM-PCE. This behavior of MAX-CE can be explained as it employs accurate methods (i.e., standard clustering

Table 6 Evaluation w.r.t. the reference classification ($F1_f$)

dataset	*AVG-ensemble*	*MAX-CE*	*PROCLUS*	*LAC*	**MOEA-PCE**	**EM-PCE**
Iris	.679	.624	.667(.001)	.682(.040)	.974(.019)	.667(.001)
Wine	.397	.304	.778(.001)	.346(.051)	.643(.051)	.426(.006)
Glass	.553	.486	.819(.022)	.549(.090)	.804(.023)	.662(.029)
Ecoli	.744	.585	.993(.017)	.751(.060)	.906(.021)	.970(.013)
Yeast	.705	.619	.783(.029)	.711(.040)	.846(.011)	.774(.010)
Mult.-Feat.	.462	.204	.382(.089)	.361(.072)	.768(.013)	.795(.001)
Segmentation	.583	.384	.815(.093)	.539(.182)	.861(.026)	.747(.029)
Abalone	.703	.630	.764(.002)	.702(.021)	.822(.023)	.716(.006)
Waveform	.248	.628	.831(.025)	.183(.040)	.660(.067)	.792(.001)
Letter	.375	.699	.775(.023)	.372(.121)	.643(.037)	.595(.008)
Isolet	.141	.080	.585(.074)	.131(.004)	.171(.004)	.143(.001)
Gisette	.741	.223	.640(.091)	.754(.053)	.876(.016)	.797(.001)
p53-Mutants	.626	.358	.634(.045)	.604(.018)	.721(.009)	.671(.001)
Amazon	.803	.482	.722(.073)	.800(.029)	.890(.005)	.828(.001)
Arcene	.429	.025	.244(.082)	.427(.041)	.536(.014)	.264(.003)
Shapes	.359	.225	.436(.051)	.364(.024)	.428(.011)	.448(.002)
Tracedata	.677	.521	.770(.064)	.669(.030)	.787(.010)	.800(.006)
ControlChart	.081	.606	.736(.037)	.080(.005)	.322(.044)	.673(.002)
Twopat	.138	.533	.851(.029)	.070(.015)	.451(.079)	.233(.001)
N30	.105	.105	.093(.001)	.099(.001)	.131(.002)	.119(.001)
D75	.032	.029	.029(.001)	.027(.001)	.041(.001)	.039(.001)
S2500	.115	.108	.100(.003)	.104(.002)	.141(.002)	.124(.002)
min	.032	.025	.029	.027	.041	.039
max	.803	.699	.993	.800	.974	.970
avg	.441	.384	.611	.424	.610	.558

ensemble methods) only for computing the object-to-cluster assignments, whereas the feature-to-cluster assignments are computed in a naïve way (cf. Sect. 5.1.5).

The remarks drawn from the evaluation w.r.t. the reference classification were confirmed by the results in terms of internal assessment criteria (Tables 7–9). In particular, the gaps in performance between the MAX-CE and PROCLUS baselines and the proposed MOEA-PCE and EM-PCE were larger than those observed in the previous evaluation. Indeed, MOEA-PCE and EM-PCE achieved improvements w.r.t. MAX-CE equal to 0.172 and 0.136 (average), and 0.243 and 0.204 (maximum), respectively. As far as PROCLUS, the improvements obtained by MOEA-PCE and EM-PCE were 0.267 and 0.231 (average), and 0.307 and 0.268 (maximum), respectively. The better performance of MOEA-PCE w.r.t. EM-PCE was confirmed: MOEA-PCE reached an average improvement w.r.t. EM-PCE equal to 0.036 (in the evaluation w.r.t. the reference classification, this improvement was 0.029). Among the baselines, as expected, AVG-ensemble again outperformed MAX-CE in terms of the object/feature-based criterion $\overline{F1}_{of}$ and the feature-based criterion $\overline{F1}_f$, whereas MAX-CE performed better in terms of the object-based $\overline{F1}_o$.

Looking at the standard deviations reported in the tables, both our proposed methods (especially EM-PCE) revealed to be quite insensitive to randomization. Indeed, the standard deviations were in the order of 10^{-3} in most cases, and only occasionally in the order of 10^{-2}.

Statistical significance. We performed a statistical significance test to assess the relative performance means of the proposed and competing methods. More precisely, we adopted an unpaired, unequal-variance T-Test methodology, under the null hypothesis of no difference in the means between any two groups of performance scores of

Table 7 Evaluation w.r.t. the projective ensemble solutions ($\overline{F1}_{of}$)

dataset	*AVG-ensemble*	*MAX-CE*	*PROCLUS*	*LAC*	**MOEA-PCE**	**EM-PCE**
Iris	.813	.435	.612(.119)	.818(.074)	.858(.010)	.830(.002)
Wine	.533	.333	.281(.020)	.583(.043)	.603(.008)	.569(.002)
Glass	.373	.262	.203(.020)	.386(.059)	.430(.007)	.437(.009)
Ecoli	.614	.379	.414(.043)	.619(.034)	.630(.009)	.625(.004)
Yeast	.596	.341	.231(.028)	.595(.038)	.610(.008)	.586(.017)
Mult.-Feat.	.150	.079	.007(.002)	.052(.020)	.221(.008)	.233(.013)
Segmentation	.182	.143	.107(.014)	.116(.053)	.269(.009)	.308(.003)
Abalone	.686	.379	.191(.022)	.690(.021)	.675(.011)	.541(.011)
Waveform	.101	.079	.101(.004)	.110(.017)	.146(.004)	.110(.001)
Letter	.142	.088	.084(.007)	.128(.060)	.196(.005)	.146(.002)
Isolet	.704	.391	.066(.036)	.682(.120)	.751(.009)	.790(.001)
Gisette	.526	.120	.206(.132)	.521(.017)	.597(.003)	.505(.002)
p53-Mutants	.565	.345	.052(.048)	.563(.017)	.628(.005)	.632(.001)
Amazon	.417	.272	.235(.077)	.412(.030)	.468(.008)	.444(.015)
Arcene	.484	.047	.180(.051)	.493(.094)	.589(.009)	.421(.004)
Shapes	.404	.308	.163(.023)	.424(.026)	.444(.009)	.468(.009)
Tracedata	.667	.435	.416(.051)	.679(.069)	.723(.008)	.716(.001)
ControlChart	.137	.017	.013(.001)	.143(.029)	.186(.005)	.037(.002)
Twopat	.283	.026	.058(.004)	.317(.028)	.370(.012)	.144(.001)
N30	.201	.220	.076(.010)	.118(.001)	.323(.013)	.301(.003)
D75	.299	.274	.029(.004)	.139(.001)	.423(.014)	.429(.004)
S2500	.302	.271	.100(.018)	.123(.001)	.442(.013)	.449(.010)
min	.101	.017	.007	.052	.146	.037
max	.813	.435	.612	.818	.858	.830
avg	.417	.238	.174	.396	.481	.442

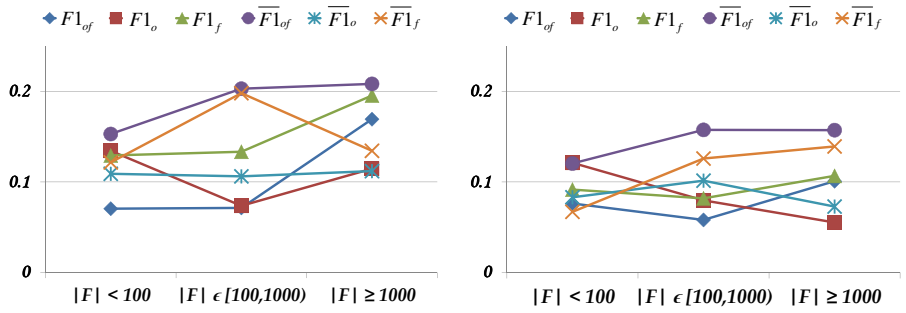
the selected methods. Tables 10–14 show the p-values corresponding to the statistical tests performed to compare MOEA-PCE vs. EM-PCE (Table 10), MOEA-PCE vs. the baselines (Tables 11–12), and EM-PCE vs. the baselines (Tables 13–14). Boldface and italic p-values corresponded to a fail in the test (i.e., null hypothesis not rejected) at 5% and 1% significance level (two-tail test), respectively.

All tables show a strong evidence that the null hypothesis was rejected in nearly all cases. On a total of 1188 tests, the null hypothesis could not be rejected on 58 cases at 5% significance level, and just 12 cases at 1% significance level. Particularly robust appeared the results obtained by MOEA-PCE vs. the AVG-ensemble and MAX-CE baselines (Tables 11), and EM-PCE vs. the AVG-ensemble and MAX-CE baselines (Tables 13). Overall, by integrating the evidence of results from the previous evaluation (Tables 4–9), we can hence state the superiority of MOEA-PCE over EM-PCE as well as of both MOEA-PCE and EM-PCE over all baselines are actually statistically significant.

Summary of improvements over the baselines. We summarize in Fig. 2 the average improvements obtained by the proposed MOEA-PCE and EM-PCE upon the baselines. For each of the six assessment criteria, we report the average gains in accuracy of MOEA-PCE/EM-PCE w.r.t. the baselines, computed by averaging over the difference between the results obtained by MOEA-PCE/EM-PCE and each baseline. To relate the accuracy results to the dimensionality of the datasets involved into this comparison, we further aggregated (averaged) such gains by distinguishing among datasets having dimensionality $|\mathcal{F}| < 100$, $|\mathcal{F}| \in [100, 1000)$, and $|\mathcal{F}| \geq 1000$, which were represented by the first, second, and third point in each series in the charts, respectively.

Table 8 Evaluation w.r.t. the projective ensemble solutions ($\overline{F1}_o$)

dataset	*AVG-ensemble*	*MAX-CE*	*PROCLUS*	*LAC*	**MOEA-PCE**	**EM-PCE**
Iris	.836	.841	.640(.099)	.835(.064)	.915(.007)	.842(.002)
Wine	.615	.696	.477(.043)	.663(.050)	.723(.010)	.662(.005)
Glass	.416	.505	.323(.032)	.442(.065)	.513(.008)	.511(.010)
Ecoli	.758	.754	.581(.068)	.768(.036)	.810(.013)	.804(.005)
Yeast	.723	.699	.293(.039)	.721(.031)	.754(.011)	.681(.017)
Mult.-Feat.	.138	.225	.113(.010)	.115(.009)	.216(.009)	.262(.015)
Segmentation	.220	.335	.198(.028)	.190(.026)	.313(.008)	.343(.005)
Abalone	.732	.683	.294(.033)	.736(.020)	.749(.015)	.592(.011)
Waveform	.414	.497	.421(.018)	.419(.013)	.489(.008)	.473(.001)
Letter	.263	.358	.219(.015)	.244(.053)	.324(.005)	.309(.004)
Isolet	.805	.785	.559(.088)	.780(.096)	.876(.010)	.875(.001)
Gisette	.588	.622	.565(.037)	.586(.023)	.693(.004)	.615(.003)
p53-Mutants	.600	.645	.456(.049)	.584(.060)	.720(.006)	.679(.001)
Amazon	.437	.499	.292(.036)	.424(.038)	.519(.008)	.473(.017)
Arcene	.717	.791	.684(.062)	.703(.094)	.840(.007)	.800(.001)
Shapes	.629	.644	.503(.056)	.648(.029)	.692(.010)	.701(.012)
Tracedata	.737	.744	.590(.062)	.743(.064)	.829(.008)	.830(.001)
ControlChart	.291	.375	.237(.012)	.295(.017)	.347(.004)	.307(.004)
Twopat	.443	.533	.328(.010)	.459(.021)	.528(.006)	.463(.002)
N30	.286	.419	.370(.050)	.169(.001)	.435(.013)	.469(.004)
D75	.399	.515	.388(.052)	.205(.001)	.551(.015)	.580(.005)
S2500	.403	.517	.413(.070)	.183(.001)	.562(.015)	.575(.011)
min	.138	.225	.113	.115	.216	.262
max	.836	.841	.684	.835	.915	.875
avg	.520	.576	.407	.496	.609	.584



(a) MOEA-PCE (b) EM-PCE

Fig. 2 Summary of the average gains of MOEA-PCE and EM-PCE w.r.t. the baselines.

As illustrated by the plots in the figure, the average gains in accuracy of both MOEA-PCE and EM-PCE w.r.t. the baselines were always positive, ranging from 0.06 and 0.2, thus confirming evidence of superiority of both our proposed methods. An interesting remark concerns a relation between the performance trends and the dimensionality of the datasets. According to four out of six assessment criteria (i.e., $F1_{of}$, $F1_f$, $\overline{F1}_{of}$, and $\overline{F1}_f$), MOEA-PCE and EM-PCE tended to obtain larger improvements upon the baselines as the dimensionality of the dataset increased. This suggests that our PCE methods can handle high dimensionality better than traditional projective clustering algorithms, whose performance has shown to be in general inversely proportional to the dataset dimensionality, as discussed in some recent studies [60, 40].

Table 9 Evaluation w.r.t. the projective ensemble solutions ($\overline{F1}_f$)

dataset	*AVG-ensemble*	*MAX-CE*	*PROCLUS*	*LAC*	**MOEA-PCE**	**EM-PCE**
Iris	.978	.760	.967(.045)	.984(.014)	.964(.008)	.989(.001)
Wine	.838	.600	.577(.005)	.868(.024)	.861(.008)	.856(.002)
Glass	.823	.685	.628(.010)	.827(.022)	.833(.007)	.843(.003)
Ecoli	.831	.719	.800(.005)	.834(.026)	.848(.007)	.818(.006)
Yeast	.866	.743	.835(.017)	.873(.023)	.870(.005)	.890(.005)
Mult.-Feat.	.622	.578	.186(.062)	.654(.027)	.656(.008)	.662(.001)
Segmentation	.796	.664	.573(.056)	.814(.071)	.854(.014)	.815(.005)
Abalone	.960	.822	.672(.003)	.957(.012)	.944(.004)	.976(.002)
Waveform	.198	.204	.225(.007)	.243(.026)	.266(.013)	.279(.001)
Letter	.549	.382	.381(.009)	.553(.047)	.604(.011)	.453(.009)
Isolet	.855	.517	.145(.019)	.852(.048)	.838(.004)	.891(.001)
Gisette	.715	.284	.604(.084)	.719(.041)	.750(.005)	.771(.001)
p53-Mutants	.881	.551	.789(.052)	.889(.007)	.858(.002)	.920(.001)
Amazon	.897	.580	.793(.080)	.902(.015)	.874(.002)	.945(.001)
Arcene	.617	.077	.334(.071)	.629(.087)	.664(.010)	.546(.006)
Shapes	.629	.555	.364(.017)	.653(.012)	.641(.005)	.629(.004)
Tracedata	.878	.618	.671(.026)	.889(.019)	.853(.003)	.848(.003)
ControlChart	.446	.066	.062(.003)	.466(.080)	.510(.022)	.154(.001)
Twopat	.608	.091	.140(.004)	.654(.048)	.685(.024)	.342(.001)
N30	.831	.674	.262(.009)	.851(.008)	.810(.005)	.670(.010)
D75	.814	.649	.084(.003)	.787(.045)	.799(.008)	.741(.006)
S2500	.798	.669	.276(.006)	.780(.018)	.803(.007)	.772(.009)
min	.198	.066	.062	.243	.266	.154
max	.978	.822	.967	.984	.964	.989
avg	.747	.522	.471	.758	.763	.719

Table 10 P-values for unpaired T-Test (df: 98): MOEA-PCE vs. EM-PCE

dataset	$F1_{of}$	$F1_o$	$F1_f$	$\overline{F1}_{of}$	$\overline{F1}_o$	$\overline{F1}_f$
Iris	2.1E-08	1.0E-28	8.2E-61	5.2E-26	7.7E-58	8.5E-28
Wine	5.0E-51	1.1E-41	8.5E-30	5.0E-04	2.9E-03	1.2E-41
Glass	1.7E-52	3.3E-05	1.3E-45	3.7E-05	**2.9E-01**	1.2E-14
Ecoli	5.0E-51	1.1E-41	8.5E-30	5.0E-04	2.9E-03	1.2E-41
Yeast	6.5E-14	4.9E-77	2.6E-56	2.8E-13	1.0E-40	3.2E-38
M.-Feat.	1.6E-33	8.3E-06	6.9E-20	3.5E-07	1.0E-30	2.9E-06
Segm.	1.0E-49	2.1E-19	5.8E-37	3.5E-36	1.3E-36	2.8E-27
Abalone	2.2E-35	4.6E-56	7.1E-37	6.7E-79	9.9E-73	7.5E-53
Wave	2.6E-06	1.9E-13	1.5E-18	2.6E-53	7.3E-19	2.7E-09
Letter	**5.9E-01**	3.5E-07	6.3E-12	4.5E-60	5.8E-32	2.3E-86
Isolet	3.5E-33	8.4E-05	9.4E-49	9.1E-34	**5.0E-01**	1.0E-57
Gisette	1.6E-08	5.8E-35	3.0E-36	3.5E-107	1.3E-101	1.2E-37
p53-M.	**6.4E-02**	1.5E-28	4.6E-39	1.9E-06	5.6E-43	1.5E-75
Amazon	**1.0E-01**	3.8E-18	3.8E-56	7.0E-15	1.5E-26	1.7E-77
Arcene	7.5E-61	3.4E-48	4.3E-69	9.5E-82	7.7E-40	1.6E-75
Shapes	1.2E-67	4.0E-04	1.2E-16	1.4E-23	1.0E-04	1.1E-21
Trace	3.2E-41	**1.0E-01**	8.4E-12	2.1E-07	**4.0E-01**	6.3E-15
Control	1.3E-53	2.9E-05	6.1E-46	4.3E-81	4.7E-70	3.6E-61
Twopat	1.5E-20	2.4E-39	1.5E-24	3.6E-65	1.8E-55	1.7E-58
N30	1.0E-44	*1.8E-02*	3.9E-46	8.6E-17	4.8E-24	4.6E-72
D75	1.0E-42	3.6E-05	1.5E-22	4.3E-03	2.4E-18	8.9E-59
S2500	9.9E-41	**5.1E-01**	1.1E-63	4.8E-03	4.7E-06	9.5E-34

Varying the ensemble properties. Here we focus on analyzing the performance of the selected methods by varying some properties in the projective ensemble. Figures 3–5 show the average gains of the proposed MOEA-PCE/EM-PCE w.r.t. the baselines for each assessment criterion. In particular, we varied either the number of clusters in each member of the ensemble (Figs. 3–4) or the algorithm used for deriving the

Table 11 P-values for unpaired T-Test (df: 98): MOEA-PCE vs. AVG-ensemble and MAX-CE baselines

	MOEA-PCE vs. *AVG-ensemble*						MOEA-PCE vs. *MAX-CE*					
dataset	$F1_{of}$	$F1_o$	$F1_f$	$\overline{F1}_{of}$	$\overline{F1}_o$	$\overline{F1}_f$	$F1_{of}$	$F1_o$	$F1_f$	$\overline{F1}_{of}$	$\overline{F1}_o$	$\overline{F1}_f$
Iris	1.6E-54	3.3E-53	5.8E-60	1.8E-34	1.7E-52	3.0E-17	1.2E-49	9.6E-29	1.4E-63	1.2E-81	4.1E-51	1.5E-71
Wine	2.6E-53	2.0E-10	6.7E-45	4.8E-17	1.4E-32	1.3E-22	9.9E-52	3.5E-48	2.6E-59	7.6E-73	4.6E-34	3.3E-64
Glass	4.0E-56	3.8E-14	7.4E-53	1.2E-46	5.9E-54	1.8E-14	4.3E-54	3.7E-08	7.3E-58	1.6E-69	1.9E-08	3.9E-68
Ecoli	2.6E-53	2.0E-10	6.7E-45	4.8E-17	1.4E-32	1.3E-22	9.9E-52	3.5E-48	2.6E-59	7.6E-73	4.6E-34	3.3E-64
Yeast	4.4E-57	5.3E-80	2.2E-56	5.6E-17	1.2E-25	4.8E-07	5.5E-56	9.7E-52	1.7E-66	4.5E-77	4.5E-37	2.9E-71
M.-Feat.	3.8E-11	1.8E-19	1.7E-69	1.0E-47	2.6E-48	2.9E-33	4.8E-46	1.9E-22	1.7E-82	2.3E-62	3.4E-09	1.7E-50
Segm.	1.1E-17	5.7E-21	8.5E-52	3.1E-49	5.1E-54	4.1E-33	1.6E-48	2.2E-16	3.2E-63	4.6E-57	1.5E-24	7.3E-58
Abalone	3.5E-76	1.2E-95	4.3E-37	2.8E-09	3.7E-10	8.6E-30	8.0E-51	7.5E-43	4.8E-47	1.8E-72	2.0E-33	2.9E-72
Wave	2.6E-13	6.9E-31	1.2E-40	6.4E-55	2.8E-49	5.4E-38	**3.0E-02**	**1.4E-01**	1.6E-03	2.4E-63	7.8E-09	4.3E-36
Letter	1.7E-31	4.9E-14	9.7E-44	1.2E-52	1.1E-56	1.1E-36	2.8E-09	*2.2E-02*	4.4E-14	2.4E-67	2.2E-44	5.4E-66
Isolet	1.1E-115	2.4E-71	2.9E-47	1.5E-37	8.8E-43	7.9E-34	3.5E-63	2.6E-30	9.5E-71	1.6E-80	5.6E-48	7.7E-96
Gisette	1.9E-42	1.2E-07	2.2E-47	1.6E-70	3.6E-73	6.3E-45	1.7E-70	1.7E-32	8.6E-81	4.8E-111	7.3E-65	7.7E-100
p53-M.	3.2E-49	3.1E-24	2.3E-52	1.2E-53	4.6E-65	2.6E-54	6.5E-50	1.1E-30	9.5E-81	1.4E-85	4.3E-55	1.0E-109
Amazon	1.1E-28	3.0E-43	2.9E-63	6.4E-42	7.4E-52	3.0E-53	1.5E-45	1.8E-13	3.3E-96	1.7E-70	3.0E-23	1.3E-107
Arcene	5.0E-69	2.1E-74	6.3E-46	1.1E-54	1.3E-63	2.4E-35	1.5E-70	3.1E-47	5.3E-79	1.4E-89	2.6E-44	1.0E-88
Shapes	1.1E-81	2.9E-64	3.8E-40	2.1E-34	7.6E-41	1.6E-20	3.9E-47	6.2E-28	7.3E-63	5.4E-60	3.0E-35	1.6E-60
Trace	1.4E-59	7.3E-71	4.1E-53	5.6E-43	5.3E-53	4.3E-50	2.0E-57	1.4E-74	7.7E-72	1.3E-77	2.5E-51	1.1E-97
Control	1.5E-58	7.4E-55	5.4E-38	1.1E-48	3.8E-59	1.0E-25	2.3E-39	6.7E-16	2.2E-41	6.7E-75	1.6E-44	8.1E-66
Twopat	5.2E-43	2.3E-65	1.3E-31	8.4E-45	6.4E-57	2.0E-27	1.6E-23	1.9E-32	2.6E-09	6.9E-74	1.3E-06	5.9E-70
N30	6.7E-97	2.2E-27	4.5E-53	4.8E-50	5.2E-53	5.2E-34	2.1E-24	**3.4E-01**	4.5E-53	1.7E-46	5.4E-11	2.9E-73
D75	7.7E-126	6.7E-39	2.2E-51	7.9E-49	1.7E-50	4.7E-17	1.2E-35	6.6E-06	1.8E-57	1.1E-52	1.7E-21	2.6E-63
S2500	1.1E-103	6.0E-36	1.4E-61	1.3E-51	2.6E-52	1.0E-05	6.8E-43	1.4E-04	1.3E-66	7.8E-56	2.0E-26	3.4E-64

Table 12 P-values for unpaired T-Test (df: 98): MOEA-PCE vs. PROCLUS and LAC

	MOEA-PCE vs. *PROCLUS*						MOEA-PCE vs. *LAC*					
dataset	$F1_{of}$	$F1_o$	$F1_f$	$\overline{F1}_{of}$	$\overline{F1}_o$	$\overline{F1}_f$	$F1_{of}$	$F1_o$	$F1_f$	$\overline{F1}_{of}$	$\overline{F1}_o$	$\overline{F1}_f$
Iris	1.2E-14	4.8E-26	7.8E-61	2.5E-19	1.0E-24	**6.5E-01**	**2.1E-01**	9.7E-12	8.0E-54	4.9E-04	1.7E-11	8.6E-13
Wine	2.9E-03	4.3E-23	4.1E-39	3.1E-38	4.8E-29	1.7E-58	**8.8E-01**	8.9E-18	6.4E-25	**3.4E-02**	1.3E-10	7.3E-04
Glass	**1.7E-01**	2.3E-20	8.2E-04	1.6E-60	9.5E-43	1.7E-98	1.1E-05	6.1E-14	3.5E-26	3.8E-06	7.2E-10	**7.9E-02**
Ecoli	2.9E-03	4.3E-23	4.1E-39	3.1E-38	4.8E-29	1.7E-58	**8.8E-01**	8.9E-18	6.4E-25	**3.4E-02**	1.3E-10	7.3E-04
Yeast	6.0E-22	1.9E-31	2.7E-21	7.0E-63	3.5E-59	7.9E-20	**2.7E-01**	3.8E-12	6.4E-30	*7.0E-03*	3.4E-09	**3.3E-01**
M.-Feat.	1.6E-61	**7.4E-02**	3.4E-34	3.7E-79	4.1E-73	4.7E-46	1.0E-72	1.4E-19	3.8E-40	9.9E-57	7.4E-76	**5.7E-01**
Segm.	**3.1E-02**	3.5E-07	1.6E-03	8.2E-77	7.6E-35	2.3E-38	2.7E-19	3.8E-27	7.8E-17	7.4E-26	1.0E-38	2.5E-04
Abalone	**5.9E-01**	4.0E-50	5.5E-23	1.1E-87	2.9E-73	5.2E-128	1.7E-25	7.8E-20	3.3E-47	2.9E-05	4.6E-04	1.1E-09
Wave	6.3E-20	**1.8E-01**	8.5E-25	2.9E-74	8.1E-36	2.2E-32	6.2E-34	7.2E-20	6.6E-57	2.2E-20	3.0E-49	3.5E-07
Letter	1.8E-11	5.5E-11	1.2E-34	3.3E-88	8.7E-48	1.7E-102	7.7E-29	2.9E-27	1.0E-21	2.6E-10	3.2E-14	7.1E-10
Isolet	5.1E-15	1.9E-32	9.5E-39	4.3E-70	4.1E-30	1.7E-83	2.5E-04	1.5E-07	3.1E-73	2.1E-04	8.4E-09	**4.6E-02**
Gisette	5.6E-18	7.7E-19	5.9E-24	7.8E-26	4.6E-29	1.7E-16	**9.3E-02**	1.9E-24	3.2E-22	1.5E-35	6.0E-36	2.6E-06
p53-M.	1.1E-49	1.0E-25	1.8E-18	2.2E-55	2.3E-38	2.3E-12	2.3E-05	1.1E-14	6.1E-52	6.4E-33	5.6E-21	9.3E-35
Amazon	3.4E-12	2.3E-26	3.8E-21	1.3E-26	2.5E-43	4.8E-09	**9.0E-01**	5.1E-12	3.5E-27	3.5E-18	3.0E-23	4.8E-18
Arcene	1.8E-39	3.6E-39	4.0E-30	1.4E-47	4.8E-23	1.7E-35	2.2E-04	2.6E-23	1.2E-25	3.9E-09	1.2E-13	*7.5E-03*
Shapes	2.2E-13	2.1E-24	**3.0E-01**	1.3E-65	4.7E-29	7.2E-69	6.5E-04	4.2E-16	3.8E-26	6.1E-06	1.7E-14	4.1E-08
Trace	1.0E-08	**8.7E-02**	**7.4E-02**	4.4E-41	1.2E-31	2.1E-43	**1.3E-01**	2.8E-31	9.6E-35	5.0E-05	1.5E-12	3.8E-18
Control	4.6E-39	1.7E-29	4.7E-70	1.5E-76	2.6E-56	1.3E-67	5.6E-37	5.6E-21	1.1E-38	6.1E-14	5.1E-27	5.6E-04
Twopat	3.5E-47	1.4E-30	2.9E-41	2.5E-82	3.8E-91	1.2E-70	8.6E-34	2.8E-41	6.0E-37	6.3E-19	5.8E-30	1.6E-04
N30	2.1E-19	6.3E-06	4.3E-61	1.4E-98	4.1E-12	2.3E-126	9.8E-60	5.8E-24	1.0E-66	5.5E-61	2.6E-65	1.5E-45
D75	1.9E-39	3.8E-21	2.3E-84	1.5E-83	1.3E-28	3.0E-122	1.2E-59	9.7E-33	5.4E-79	2.2E-66	6.1E-68	**6.3E-02**
S2500	1.3E-27	3.5E-15	4.3E-83	4.3E-97	3.5E-20	5.2E-155	1.8E-66	6.2E-33	6.1E-96	4.6E-69	9.9E-71	4.1E-12

ensemble solutions (Fig. 5). For the sake of brevity of presentation, we present results for only one high-dimensional dataset, i.e., p53-Mutants.

Considering Figs. 3–4, for the majority of the assessment criteria, the performance of both MOEA-PCE and EM-PCE tended to decrease as the number of clusters K increased. For both algorithms, the trends of object & feature-based criteria (i.e., $F1_{of}$ and $\overline{F1}_{of}$) followed those of object-based only criteria (i.e., $F1_o$ and $\overline{F1}_o$),

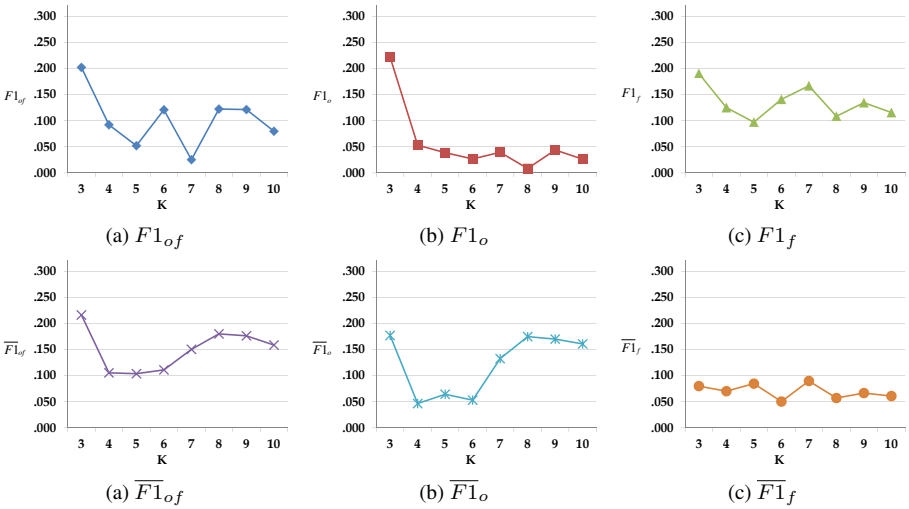
Table 13 P-values for unpaired T-Test (df: 98): EM-PCE vs. AVG-ensembles and MAX-CE baselines

	EM-PCE vs. *AVG-ensemble*						EM-PCE vs. *MAX-CE*					
dataset	$F1_{of}$	$F1_{o}$	$F1_{f}$	$\overline{F1}_{of}$	$\overline{F1}_{o}$	$\overline{F1}_{f}$	$F1_{of}$	$F1_{o}$	$F1_{f}$	$\overline{F1}_{of}$	$\overline{F1}_{o}$	$\overline{F1}_{f}$
Iris	3.6E-112	9.7E-97	2.8E-117	9.1E-45	2.2E-22	4.2E-108	4.6E-111	3.9E-03	2.0E-144	1.6E-111	*6.1E-03*	1.0E-172
Wine	6.2E-47	7.3E-30	1.4E-62	1.2E-26	2.8E-48	1.7E-21	6.4E-70	9.6E-14	6.7E-74	9.6E-92	5.0E-50	5.7E-63
Glass	3.4E-48	2.8E-29	1.6E-30	8.1E-44	2.2E-49	1.6E-43	8.8E-68	1.8E-42	2.7E-40	4.8E-65	1.4E-04	2.6E-87
Ecoli	6.2E-47	7.3E-30	1.4E-62	1.2E-26	2.8E-48	1.7E-21	6.4E-70	9.6E-14	6.7E-74	9.6E-92	5.0E-50	5.7E-63
Yeast	4.1E-76	2.5E-85	2.2E-43	1.4E-04	1.8E-22	1.6E-37	3.3E-70	6.1E-10	2.0E-60	1.3E-58	1.6E-09	9.7E-76
M.-Feat.	3.1E-40	1.0E-34	1.5E-186	9.3E-42	6.2E-47	4.1E-75	3.3E-59	8.8E-40	9.1E-199	9.7E-55	1.3E-22	6.8E-91
Segm.	6.0E-49	8.8E-10	4.7E-39	7.7E-79	3.3E-70	2.1E-31	2.3E-75	4.5E-12	1.0E-55	1.4E-84	2.2E-15	7.1E-75
Abalone	1.6E-70	7.2E-101	7.8E-21	7.8E-56	3.8E-56	1.5E-46	1.5E-55	*7.8E-03*	3.3E-59	3.5E-58	4.6E-47	1.4E-94
Wave	1.4E-39	7.7E-131	7.9E-139	6.4E-53	2.0E-82	9.3E-130	1.3E-31	1.8E-116	2.6E-113	3.7E-79	2.6E-63	4.0E-128
Letter	2.3E-58	1.4E-57	1.1E-71	1.3E-19	3.7E-56	7.2E-52	1.2E-31	2.8E-43	8.8E-56	2.1E-74	1.7E-57	1.6E-45
Isolet	1.7E-202	2.8E-191	2.5E-22	3.6E-124	1.9E-128	1.2E-115	5.6E-155	2.7E-152	3.8E-94	8.0E-157	8.4E-134	1.9E-165
Gisette	1.8E-57	2.1E-48	1.7E-145	1.2E-56	1.2E-50	6.5E-79	6.4E-90	3.5E-08	5.0E-195	1.7E-118	2.8E-23	6.2E-125
p53-M.	1.0E-48	**1.0E-01**	1.5E-154	6.2E-107	1.1E-105	2.3E-141	4.4E-51	3.7E-11	7.1E-196	7.5E-138	1.1E-87	3.5E-189
Amazon	4.6E-51	1.1E-76	8.2E-144	1.9E-16	1.0E-19	2.8E-182	8.9E-71	3.2E-16	5.3E-200	3.2E-53	2.4E-14	3.3E-225
Arcene	2.1E-119	1.6E-127	3.1E-90	9.4E-61	5.9E-133	6.9E-54	3.0E-92	4.0E-37	4.7E-98	1.2E-98	5.6E-87	5.0E-94
Shapes	1.6E-83	3.0E-67	1.4E-82	2.5E-43	1.6E-40	**1.0E+00**	3.0E-69	7.0E-34	4.0E-102	1.2E-62	1.0E-35	1.4E-65
Trace	7.4E-49	3.8E-07	2.3E-67	2.4E-106	9.0E-104	2.4E-52	1.3E-61	2.4E-14	8.9E-85	1.6E-143	4.2E-102	1.3E-95
Control	3.1E-26	9.1E-97	1.4E-121	4.1E-90	1.9E-30	1.1E-140	4.1E-23	1.1E-60	3.3E-75	6.3E-56	1.4E-60	3.7E-115
Twopat	1.5E-100	9.9E-95	3.5E-159	1.3E-119	2.7E-55	6.0E-117	5.3E-87	7.8E-86	1.2E-183	4.1E-116	6.7E-82	1.0E-115
N30	1.7E-105	4.2E-88	3.5E-51	3.5E-73	2.7E-84	2.5E-60	1.2E-54	5.1E-46	3.5E-51	1.0E-68	9.6E-57	*8.7E-03*
D75	9.7E-148	1.5E-85	9.9E-65	6.0E-77	6.1E-77	1.2E-53	1.7E-73	3.6E-53	2.6E-72	1.1E-80	3.3E-55	1.5E-58
S2500	6.6E-100	3.7E-70	5.4E-33	3.3E-58	2.2E-59	1.7E-25	1.8E-51	1.0E-28	7.4E-45	2.9E-62	1.3E-36	1.1E-53

Table 14 P-values for unpaired T-Test (df: 98): EM-PCE vs. PROCLUS and LAC

	EM-PCE vs. *PROCLUS*						EM-PCE vs. *LAC*					
dataset	$F1_{of}$	$F1_{o}$	$F1_{f}$	$\overline{F1}_{of}$	$\overline{F1}_{o}$	$\overline{F1}_{f}$	$F1_{of}$	$F1_{o}$	$F1_{f}$	$\overline{F1}_{of}$	$\overline{F1}_{o}$	$\overline{F1}_{f}$
Iris	3.6E-17	4.2E-19	**8.5E-01**	3.2E-17	4.8E-19	1.4E-03	**6.3E-02**	*1.9E-02*	*1.1E-02*	**2.7E-01**	**4.8E-01**	*1.9E-02*
Wine	1.2E-11	6.7E-14	3.3E-11	2.2E-36	7.0E-28	8.5E-32	1.1E-21	**2.5E-01**	3.4E-31	**2.3E-01**	6.7E-09	8.1E-05
Glass	1.9E-10	7.2E-29	2.4E-49	1.3E-65	9.3E-44	5.3E-76	3.1E-20	1.1E-21	1.3E-11	2.1E-07	1.6E-09	7.6E-06
Ecoli	1.2E-11	6.7E-14	3.3E-11	2.2E-36	7.0E-28	8.5E-32	1.1E-21	**2.5E-01**	3.4E-31	**2.3E-01**	6.7E-09	8.1E-05
Yeast	3.1E-17	8.3E-18	**5.0E-02**	2.0E-76	3.0E-61	6.4E-29	6.7E-04	3.7E-10	5.4E-15	**1.6E-01**	1.4E-11	5.5E-06
M.-Feat.	8.7E-81	1.5E-14	6.7E-35	9.6E-66	6.8E-71	2.0E-45	4.6E-82	1.8E-67	3.6E-40	6.4E-67	1.5E-68	**3.9E-02**
Segm.	1.3E-11	2.2E-22	8.8E-06	1.5E-63	2.0E-38	2.0E-33	4.7E-27	4.5E-43	2.0E-10	8.4E-30	1.4E-41	**8.9E-01**
Abalone	1.1E-05	4.3E-29	3.6E-52	8.0E-80	1.7E-55	2.3E-172	**4.4E-02**	2.4E-17	2.7E-05	1.4E-55	7.3E-57	2.1E-15
Wave	5.0E-24	4.3E-20	9.4E-15	2.8E-20	5.3E-26	2.1E-45	8.3E-104	1.0E-30	7.1E-60	**8.8E-01**	1.8E-33	5.0E-13
Letter	2.3E-14	2.6E-05	1.6E-51	3.8E-52	1.7E-42	1.4E-60	2.8E-34	5.0E-20	1.9E-17	**4.5E-02**	2.5E-11	4.2E-20
Isolet	1.4E-13	3.5E-31	5.2E-40	1.1E-65	1.1E-29	5.0E-80	1.1E-12	5.2E-09	4.3E-27	8.2E-08	1.1E-08	6.0E-07
Gisette	8.6E-19	9.5E-12	2.2E-16	7.5E-21	1.4E-12	1.0E-18	4.0E-05	2.2E-11	7.8E-07	1.5E-08	1.2E-11	8.2E-12
p53-M.	4.7E-50	9.7E-17	7.6E-07	1.3E-54	2.7E-34	1.0E-22	1.2E-07	**1.1E-01**	1.7E-30	9.8E-32	7.6E-15	6.7E-33
Amazon	9.0E-13	2.7E-15	1.4E-13	5.5E-25	2.2E-43	6.2E-18	**4.9E-01**	4.0E-03	1.4E-08	2.9E-09	8.4E-12	3.8E-26
Arcene	8.2E-04	1.9E-09	**9.6E-02**	3.2E-35	1.2E-17	3.1E-26	3.2E-24	**3.6E-02**	5.7E-32	2.9E-06	3.4E-09	1.9E-08
Shapes	1.6E-30	1.1E-25	**1.0E-01**	1.4E-68	2.8E-30	1.4E-63	4.7E-35	1.2E-19	3.9E-29	5.4E-16	7.5E-18	3.2E-19
Trace	**2.6E-02**	*1.8E-02*	2.0E-03	1.7E-39	3.2E-31	6.7E-43	1.1E-20	2.5E-14	4.4E-35	4.6E-04	1.1E-12	1.7E-20
Control	4.0E-27	1.1E-26	3.4E-16	2.6E-69	4.7E-46	1.1E-78	2.6E-50	5.1E-24	3.0E-128	9.2E-30	1.4E-05	3.2E-31
Twopat	1.9E-49	6.8E-05	1.3E-66	3.9E-71	6.0E-58	8.9E-99	6.8E-85	1.2E-05	5.1E-52	7.9E-41	**1.9E-01**	9.1E-42
N30	6.2E-33	1.5E-17	2.5E-64	2.5E-81	1.2E-18	1.4E-129	2.3E-75	1.1E-84	3.1E-81	5.3E-86	7.6E-95	5.3E-95
D75	8.3E-46	1.1E-31	3.0E-65	1.0E-166	2.1E-30	9.1E-143	2.3E-88	6.6E-80	6.7E-111	5.0E-94	1.8E-92	5.1E-09
S2500	7.4E-39	2.9E-20	6.9E-69	9.8E-90	1.4E-21	2.4E-133	4.2E-68	4.5E-67	3.1E-69	4.0E-75	6.8E-77	*8.3E-03*

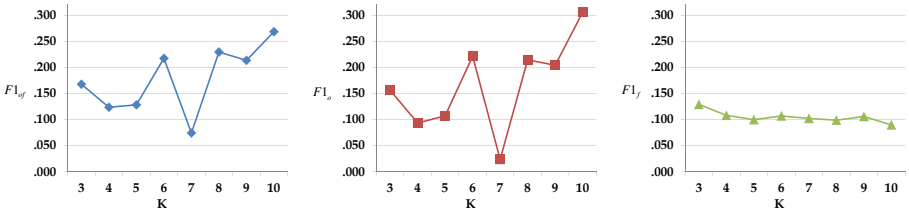
and were more largely subject to fluctuations. Moreover, while the average gains measured by the feature-based only criteria showed relatively small variations, the performance difference especially between MOEA-PCE and the baselines showed a drastic reduction from $K = 3$ to $K = 4$, and generally tended to decrease as the number of clusters increased.



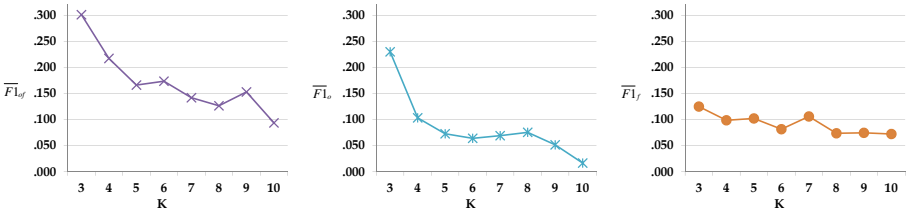
(a) $F1_{of}$ (b) $F1_o$ (c) $F1_f$

(a) $\overline{F1}_{of}$ (b) $\overline{F1}_o$ (c) $\overline{F1}_f$

Fig. 3 Average gains of MOEA-PCE w.r.t. the baselines by varying the number of clusters in the ensemble members (p53-Mutants dataset).



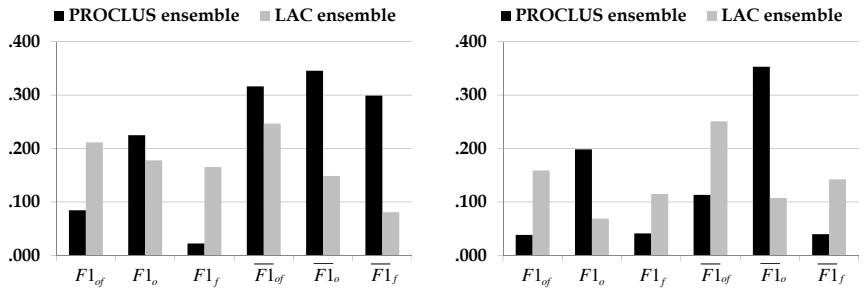
(a) $F1_{of}$ (b) $F1_o$ (c) $F1_f$



(a) $\overline{F1}_{of}$ (b) $\overline{F1}_o$ (c) $\overline{F1}_f$

Fig. 4 Average gains of EM-PCE w.r.t. the baselines by varying the number of clusters in the ensemble members (p53-Mutants dataset).

As concerns the choice of the method for generating the ensemble (Fig. 5), we observed that the average gains obtained by MOEA-PCE and EM-PCE w.r.t. the baselines were similarly affected according to $F1_{of}$, $F1_f$, $F1_o$, and $\overline{F1}_o$. Specifically, considering $F1_{of}$ and $F1_f$ (resp. $F1_o$ and $\overline{F1}_o$), both MOEA-PCE and EM-PCE gains were lower (resp. higher) on ensembles generated by the PROCLUS projective clustering algorithm. Moreover, in terms of $\overline{F1}_{of}$ and $\overline{F1}_f$, the relative performance



(a) MOEA-PCE (b) EM-PCE

Fig. 5 Average gains of MOEA-PCE and EM-PCE w.r.t. the baselines by varying the ensemble generation method (p53-Mutants dataset).

Table 15 Execution times (milliseconds)

	TOTAL		ONLINE		OFFLINE	
dataset	*MOEA-PCE*	*EM-PCE*	*MOEA-PCE*	*EM-PCE*	*MOEA-PCE*	*EM-PCE*
Iris	2,056	33	2,056	28	—	5
Wine	2,558	101	2,558	94	—	7
Glass	7,712	201	7,712	190	—	11
Ecoli	14,401	240	14,401	226	—	15
Yeast	227,878	1,259	227,878	1,067	—	193
Mult.-Feat.	490,602	56,655	490,602	13,852	—	42,803
Segmentation	233,951	4,931	233,951	4,361	—	570
Abalone	3,411,116	8,354	3,411,116	7,240	—	1,114
Waveform	125,247	4,005	125,247	2,730	—	1,276
Letter	2,248,695	27,566	2,248,695	25,069	—	2,497
Isolet	20,676,754	666,809	20,676,754	154,468	—	512,341
Gisette	966,108	804,676	966,108	243,839	—	560,837
p53-Mutants	58,695	16,492	58,695	3,955	—	12,537
Amazon	395,988	24,684	395,988	5,797	—	18,887
Arcene	120,537	20,961	120,537	4,903	—	16,058
Shapes	211,654	10,800	211,654	2,659	—	8,141
Tracedata	12,777	1,120	12,777	776	—	343
ControlChart	50,798	750	50,798	450	—	300
Twopat	31,850	8,576	31,850	7,928	—	647
N30	164,969	3,916	164,969	3,552	—	364
D75	135,297	2,592	135,297	1,359	—	1,234
S2500	290,408	3,036	290,408	2,372	—	664

variations MOEA-PCE and EM-PCE were quite similar on ensembles generated by LAC.

5.2.2 Efficiency

Table 15 shows the runtimes of the proposed algorithms MOEA-PCE and EM-PCE. The reported times (expressed in milliseconds) are organized to distinguish between the online and offline phases. This evaluation was mainly aimed to assess that EM-PCE always achieved a large efficiency gain w.r.t. MOEA-PCE, confirming the complexity analysis reported in Sect. 4.3. Looking at the total runtimes, EM-PCE outperformed MOEA-PCE by 2 orders of magnitude on 13 out of 22 datasets, while being 1 order of magnitude faster on other 6 datasets, and 3 orders faster on Abalone. The

two algorithms performed on the same order of magnitude only on Gisette and p53-Mutants, which could be ascribed to the high dimensionality and minimal number of classes (2) of these datasets; on the other hand, EM-PCE still outperformed MOEA-PCE in runtime on the two datasets with higher dimensionality (i.e., Amazon and Arcene), as we observed that EM-PCE converged with fewer iterations than MOEA-PCE. It is also interesting to observe the contribution of the online phase to the total runtime by EM-PCE: the online-to-total ratio was 64% on average, up to a maximum of 94% on Glass, and above 50% on 15 out of 22 datasets.

5.3 Application: PCE for News Stories

In the Introduction, we depicted an example scenario for clustering news summaries and supporting cluster-based indexing. Here we use a similar scenario to demonstrate the applicability of PCE to a real-world large document collection.

Reuters Corpus Volume 1 (RCV1) [51] is a major benchmark for text classification/clustering research, which consists of over 800,000 newswire stories in XML format. RCV1 lends itself particularly well for our case study since every news is originally provided with possibly multiple categorizations according to three different category fields: TOPICS (i.e., major subjects of a news), INDUSTRIES (i.e., types of businesses discussed), and REGIONS (i.e., geographic locations as well as economic/political information about a news). Each of these category fields corresponds to a different value for the `class` attribute of the `metadata.codes` element, and the category values are alphanumerical codes contained in the relative `code` child element. Moreover, every news has three main textual elements: `title`, `headline`, and `text`; in most cases, the `headline` text is the same as, or is contained in, the `title` text. For example, a news with "USA: Netscape unveils new products, embraces Microsoft" as `title` is assigned with two TOPICS codes ("C22", "CCAT"), three INDUSTRIES codes ("I33020", "I3302020", "I3302021"), and one REGIONS code ("USA").

For our experiments, we chose to ignore the body of the news (i.e., the `text` element), and exploited only the content within titles and headlines to generate the feature (term) space. This setting resembles the scenario discussed in the Introduction, where the full-text content of the news is not made freely available to the users, which can only directly access summaries of the news (i.e., titles and headlines).

We built an evaluation collection, hereafter denoted as RCV1-ensemble. Figure 6 shows an overview of the process of generation of RCV1-ensemble, which is described next in detail. From the whole RCV1 collection, we filtered out very short news (i.e., XML documents with size less than 6KB), and any news that did not have at least one value for each of the three category fields. Next, since there is a one-to-many relationship between labels and codes in the RCV1 categorization systems [51], we mapped each category code value to its corresponding label (code-to-label mapping module in Figure 6); for example, the INDUSTRIES codes "I3302" and "I33020" are both mapped to the "Computer Systems and Software" label. Finally, we selected all news with labels having document-frequency above the following thresholds: 1000 for the TOPICS field (which corresponded to the 16 most

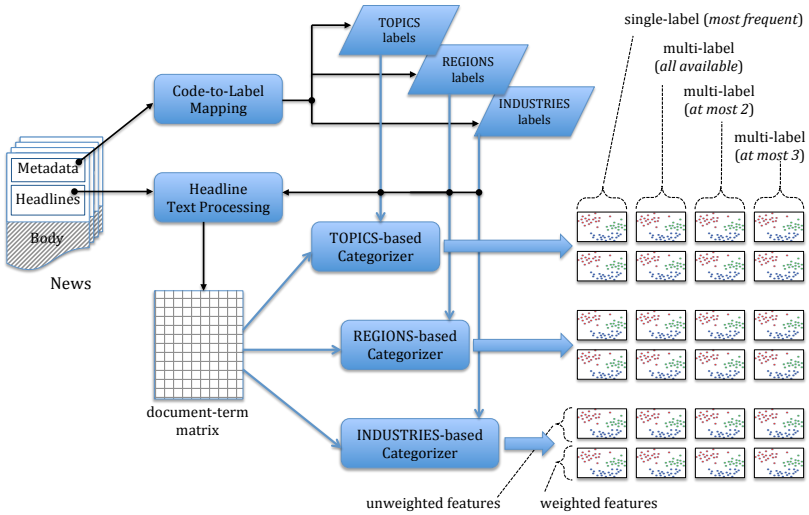


Fig. 6 Illustration of RCV1-ensemble generation

frequent labels for the TOPICS field), 250 for the REGIONS field (19 most frequent REGIONS labels), and 300 for the INDUSTRIES field (24 most frequent INDUSTRIES labels). This resulted in a collection of 7,891 news. The feature space of the news in RCV1-ensemble was created by processing the text of the `title` and of the `headline`. We discarded strings of digits, retained alphanumerical terms, and performed removal of stop-words and word stemming (based on Porter's algorithm[3]). Moreover, the terms with a document frequency greater than 50% were filtered out. This finally led to 5,460 features.

We generated a reference projective ensemble of 24 members based on the three category fields as follows:

- 3 multi-label categorizations, one for each of the three category fields. Each news was assigned with as many categories as *all* of its labels that belong to a given field;
- 3 single-label categorizations, one for each of the three category fields. Each news was assigned with one category corresponding to the news label (of a given field) having the highest document-frequency in the news collection;
- 3 multi-label categorizations, one for each of the three category fields. Each news was assigned with at most two categories corresponding to its two labels (of a given field) having the highest document-frequency in the news collection;
- 3 multi-label categorizations, one for each of the three category fields. Each news was assigned with at most three categories corresponding to its three labels (of a given field) having the highest document-frequency in the news collection;
- For each of the above categorizations, one ensemble member was derived with unweighted features, and another one was derived with weighted features. For the latter case, the weight of a given feature assigned to a group of news was com-

[3] http://www.tartarus.org/∼martin/PorterStemmer/.

Table 16 RCV1-ensemble: Evaluation w.r.t. the projective ensemble solutions

assessment criterion	*AVG-ensemble*	*MAX-CE*	*MOEA-PCE*	*EM-PCE*
$\overline{F1}_{of}$	.227	.096	.261	.229
$\overline{F1}_{o}$	.319	.336	.348	.324
$\overline{F1}_{f}$	.500	.354	.515	.579

Table 17 RCV1-ensemble: Execution times (milliseconds)

	MOEA-PCE	*EM-PCE*
TOTAL	5,211,395	2,476,814
ONLINE	5,211,395	1,101,451
OFFLINE	—	1,375,363

puted as directly proportional to the corresponding cumulated term-frequency over all news in that group.

Suppose there are three news n_1, n_2, n_3, which are originally labeled as follows: $T_2, T_4, R_1, I_5, I_3, I_2$ (for n_1), $T_1, T_6, T_3, R_1, R_2, R_4, I_1, I_2, I_4$ (for n_2), $T_4, T_3, T_5, T_7, R_3, R_2, I_5, I_3, I_7$ (for n_3), where Ts, Rs and Is denote TOPICS, REGIONS, and INDUSTRIES labels, respectively, and the subscript index denotes the rank of a label w.r.t. its document-frequency. A single-label TOPICS-based categorization is: n_1 assigned to T_2, n_2 assigned to T_1, and n_3 assigned to T_3. A multi-label (at-most-2) INDUSTRIES-based categorization is: n_1 assigned to categories I_2, I_3, n_2 assigned to I_1, I_2, and n_3 assigned to I_3, I_5. Moreover, suppose that a group of news contains the following terms (in parentheses, the total count of occurrences over all news in the group): t_4 (6), t_2 (12), t_7 (18), t_5 (8); in the case of weighting, for instance t_4 is weighted with 6/44.

Table 16 shows the accuracy values based on internal validity criteria obtained by the proposed PCE methods as well as the two baselines, while Table 17 summarizes the time performances of the PCE methods. MOEA-PCE outperformed both the baselines, with maximum gains of 0.034 ($\overline{F1}_{of}$), 0.029 ($\overline{F1}_{o}$), and 0.161 ($\overline{F1}_{f}$). While requiring nearly half of the total running time by MOEA-PCE, EM-PCE gave comparable results to AVG-ensemble according to $\overline{F1}_{of}$ and $\overline{F1}_{o}$, and better than MAX-CE based on $\overline{F1}_{of}$; however, EM-PCE still significantly outperformed both baselines according to $\overline{F1}_{f}$, with maximum gain of 0.225.

In order to qualitatively compare our best method w.r.t. the best baseline method (i.e., MOEA-PCE w.r.t. AVG-ensemble), we also analyzed the consensus cluster descriptions, focusing on each cluster's top ranked terms according to their feature-to-cluster assignment values. We randomly selected a (Pareto optimal) consensus clustering obtained by MOEA-PCE and compared it to AVG-ensemble. The goal was to gain some knowledge about the characteristics and the differences in the respective cluster descriptions. A first finding concerns an evident good separation of the MOEA-PCE cluster descriptions, while overlapping descriptors were found in some of the baseline clusters. For example, more baseline clusters were described by the same terms concerning 'oil' and 'petroleum', or 'technology' and 'computers', or 'tobacco'. The baseline cluster descriptions tended to contain a larger number of

broad topic-terms than MOEA-PCE, like ‘stock’, ‘bank’, ‘oil’, ‘airline’; however, this also resulted in a higher sensitivity to the presence of extremely popular terms (i.e., implicit stopwords); for example, the term ‘usa’ was present in nearly all baseline cluster descriptions. Overall, MOEA-PCE descriptions typically covered more topics than those obtained by the baseline method. Moreover, the MOEA-PCE descriptions spanned over terms that, within the same cluster, usually corresponded to labels from the TOPICS, REGIONS and INDUSTRIES fields, whereas the baseline descriptions more often corresponded to INDUSTRIES labels only, and sometimes to REGIONS labels as well. This explains why the MOEA-PCE consensus clustering integrated the different perspectives of the data (i.e., TOPICS, REGIONS, INDUSTRIES) better than the baseline method.

6 Conclusion

The projective clustering ensembles (PCE) problem fills the gap between projective clustering and clustering ensembles, which were originally conceived as separate problems to handle high dimensionality and multi-view data issues, respectively. PCE is formally defined as an optimization problem, with a two-objective formulation and a single-objective formulation, which mainly differ in the way object- and feature-based cluster representations are treated. For solving either PCE formulation, we have provided well-founded heuristics: the two-objective PCE is developed in the MOEA-PCE algorithm, which resorts to the domain of multi-objective evolutionary algorithms, and the single-objective PCE is implemented in an EM-like algorithm, called EM-PCE. As shown in the experimental evaluation, MOEA-PCE generally produces higher-quality projective consensus clusterings, but pays in efficiency w.r.t. EM-PCE. Both algorithms improve upon baseline methods in terms of external as well as internal evaluation criteria. We have also illustrated how the PCE problem is applied to a real-life case study, hence assessed how PCE methods can be used to address the two issues of high dimensionality and multi-view data in clustering applications.

A major goal of this paper was to define a complete specification of the PCE problem originally proposed in [35], with theoretical insights as well extensive experimental evaluation. While maintaining our approach to solving PCE as an optimization problem, we are certainly aware that improved or alternative formulations can be defined according to a number of aspects. The two-objective PCE treats separately the object-based and feature-based representation of any projective cluster in the optimization of the objective functions. By considering the two representations as interrelated, the resulting formulation would likely improve the significance of the detected clusters. In this respect, solutions have been identified in [36], where enhancements to the single-objective PCE are introduced to reduce the accuracy gap from the two-objective PCE, and in [37], where the two cluster representations are kept together in a suitably defined notion of distance for projective clustering solutions. Another issue in the current PCE formulations concerns a lack of versatility with respect to standard approaches to the clustering ensembles problem. As theoretically demonstrated in [37], cluster-based approaches to clustering ensembles are

well-suited to PCE. The development of methods for PCE that can profitably exploit existing clustering ensemble schemes is hence a further point of investigation which mainly concerns improvement in applicability of PCE.

References

1. E. Achtert, C. Böhm, H. -P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek. Finding Hierarchies of Subspace Clusters. In *Proc. European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD)*, pages 446–453, 2006.
2. E. Achtert, C. Böhm, H. -P. Kriegel, P. Kröger, I. Müller-Gorman, and A. Zimek. Detection and Visualization of Subspace Cluster Hierarchies. In *Proc. Int. Conf. on Database Systems for Advanced Applications (DASFAA)*, pages 152–163, 2007.
3. C. C. Aggarwal, C. M. Procopiuc, J. L. Wolf, P. S. Yu, and J. S. Park. Fast Algorithms for Projected Clustering. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 61–72, 1999.
4. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In *Proc. ACM SIGMOD Int. Conf. on Management Of Data*, pages 94–105, 1998.
5. M. Ankerst, M. M. Breunig, H. -P. Kriegel, and J. Sander. OPTICS: Ordering Points To Identify the Clustering Structure. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 49–60, 1999.
6. I. Assent, R. Krieger, E. Müller, and T. Seidl. EDSC: efficient density-based subspace clustering. In *Proc. ACM Conf. on Information and Knowledge Management (CIKM)*, pages 1093–1102, 2008.
7. A. Asuncion and D.J. Newman. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/, 2010.
8. 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.
9. J. P. Barthélemy and B. Leclerc. The Median Procedure for Partitions. *Partitioning Data Sets*, 19:3–33, 1995.
10. R. Bellman. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.
11. K. S. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft. When Is "Nearest Neighbor" Meaningful? In *Proc. Int. Conf. on Database Theory (ICDT)*, pages 217–235, 1999.
12. C. Böhm, K. Kailing, H. P. Kriegel, and P. Kröger. Density Connected Clustering with Local Subspace Preferences. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 27–34, 2004.
13. 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.

14. 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.
15. L. Breiman. Bagging predictors . *Machine Learning*, 24(2):123–140, 1996.
16. 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.
17. L. Chen, Q. Jiang, and S. Wang. A Probability Model for Projective Clustering on High Dimensional Data. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 755–760, 2008.
18. K. Deb. *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.
19. K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. *IEEE Trans. on Evolutionary Computation*, 6(2):182–197, 2002.
20. A. P. Dempster, N. M. Laird, and D. B. Rdin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society*, 39:1–38, 1977.
21. E. Dimitriadou, A. Weingesse, and K. Hornik. Voting-Merging: An Ensemble Method for Clustering. In *Proc. Int. Conf. on Artificial Neural Networks (ICANN)*, pages 217–224, 2001.
22. C. Domeniconi and M. Al-Razgan. Weighted Cluster Ensembles: Methods and Analysis. *ACM Trans. on Knowledge Discovery from Data (TKDD)*, 2(4), 2009.
23. 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.
24. S. Dudoit and J. Fridlyand. Bagging to Improve the Accuracy of a Clustering Procedure. *Bioinformatics*, 19(9):1090–1099, 2003.
25. M. Ester, H. -P. Kriegel, J. Sander, and X. Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proc. Int. Conf. on Knowledge Discovery and Data Mining (KDD)*, pages 226–231, 1996.
26. 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.
27. X. Z. Fern and W. Lin. Cluster Ensemble Selection. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 787–797, 2008.
28. B. Fischer and J. M. Buhmann. Bagging for Path-Based Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 25(11):1411–1415, 2003.
29. A. L. N. Fred. Finding Consistent Clusters in Data Partitions. In *Proc. Int. Workshop on Multiple Classifier Systems (MCS)*, pages 309–318, 2001.
30. 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.
31. G. Gan, C. Ma, and J. Wu. *Data Clustering: Theory, Algorithms, and Applications*. ASA-SIAM Series on Statistics and Applied Probability, 2007.
32. R. Ghaemi, N. bin Sulaiman, H. Ibrahim, and N. Mustapha. A review: accuracy optimization in clustering ensembles using genetic algorithms. *Artif. Intell. Rev.*, 35(4):287–318, 2011.

33. J. Ghosh and A. Acharya. Cluster ensembles. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 1(4):305–315, 2011.
34. A. Gionis, H. Mannila, and P. Tsaparas. Clustering Aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1), 2007.
35. F. Gullo, C. Domeniconi, and A. Tagarelli. Projective Clustering Ensembles. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 794–799, 2009.
36. F. Gullo, C. Domeniconi, and A. Tagarelli. Enhancing Single-Objective Projective Clustering Ensembles. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 833–838, 2010.
37. F. Gullo, C. Domeniconi, and A. Tagarelli. Advancing data clustering via projective clustering ensembles. In *SIGMOD Conference*, pages 733–744, 2011.
38. F. Gullo, A. Tagarelli, and S. Greco. Diversity-Based Weighting Schemes for Clustering Ensembles. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 437–448, 2009.
39. S. Günnemann, B. Boden, and T. Seidl. DB-CSC: A Density-Based Approach for Subspace Clustering in Graphs with Feature Vectors. In *Proc. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 565–580, 2011.
40. S. Günnemann, I. Färber, E. Müller, I. Assent, and T. Seidl. External evaluation measures for subspace clustering. In *Proc. ACM Conf. on Information and Knowledge Management (CIKM)*, pages 1363–1372, 2011.
41. A. Hinneburg, C. C. Aggarwal, and D. A. Keim. What Is the Nearest Neighbor in High Dimensional Spaces? In *Proc. Int. Conf. on Very Large Data Bases (VLDB)*, pages 506–515, 2000.
42. A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, 1988.
43. 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.
44. 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.
45. E. Keogh, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR Time Series Classification/Clustering Page. http://www.cs.ucr.edu/∼eamonn/time_series_data/, 2003.
46. H. -P. Kriegel, P. Kroger, M. Renz, and S. Wurst. A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 250–257, 2005.
47. H. -P. Kriegel, P. Kröger, and A. Zimek. Clustering High-Dimensional Data: A Survey on Subspace Clustering, Pattern-Based Clustering, and Correlation Clustering. *ACM Trans. on Knowledge Discovery from Data (TKDD)*, 3(1):1–58, 2009.
48. M. Kriv´anek and J. Morávek. NP-hard problems in hierarchical-tree clustering. *Acta Informatica*, 23(3):311–323, 1986.
49. H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.

50. L. I. Kuncheva, S. T. Hadjitodorov, and L. P. Todorova. Experimental Comparison of Cluster Ensemble Methods. In *Proc. Int. Conf. on Information Fusion*, pages 1–7, 2006.
51. D. D. Lewis, Y. Yang, T. Rose, and F. Li. RCV1: A New Benchmark Collection for Text Categorization Research. *Journal of Machine Learning Research*, 5:361–397, 2004.
52. T. Li and C. Ding. Weighted Consensus Clustering. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 798–809, 2008.
53. 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.
54. B. Liu, Y. Xia, and P. S. Yu. Clustering Through Decision Tree Construction. In *Proc. Int. Conf. on Information and Knowledge Management (CIKM)*, pages 20–29, 2000.
55. M. Meila. Comparing clusterings: an axiomatic view. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 577–584, 2005.
56. G. Moise, J. Sander, and M. Ester. Robust projected clustering. *Knowledge and Information Systems*, 14(3):273–298, 2008.
57. G. Moise, A. Zimek, P. Kröger, H.-P. Kriegel, and J. Sander. Subspace and projected clustering: experimental evaluation and analysis. *Knowledge and Information Systems*, 21(3):299–326, 2009.
58. E. Müller, I. Assent, S. Günnemann, R. Krieger, and T. Seidl. Relevant Subspace Clustering: Mining the Most Interesting Non-redundant Concepts in High Dimensional Data. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 377–386, 2009.
59. E. Müller, I. Assent, S. Günnemann, and T. Seidl. Scalable density-based subspace clustering. In *Proc. ACM Conf. on Information and Knowledge Management (CIKM)*, pages 1077–1086, 2011.
60. E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating clustering in subspace projections of high dimensional data. *Proceedings of the VLDB Endowment (PVLDB)*, 2(1):1270–1281, 2009.
61. E. Müller, S. Günnemann, I. Assent, and T. Seidl. Evaluating Clustering in Subspace Projections of High Dimensional Data. http://dme.rwth-aachen.de/en/OpenSubspace/evaluation, 2009.
62. 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.
63. E. Ka Ka Ng, A. Wai-Chee Fu, and R. Chi-Wing Wong. Projective Clustering by Histograms. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 17(3):369–383, 2005.
64. N. Nguyen and R. Caruana. Consensus Clustering. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 607–612, 2007.
65. L. Parsons, E. Haque, and H. Liu. Subspace Clustering for High Dimensional Data: A Review. *SIGKDD Explorations*, 6(1):90–105, 2004.
66. A. Patrikainen and M. Meila. Comparing subspace clusterings. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 18(7):902–916, 2006.

67. C. M. Procopiuc, M. Jones, P. K. Agarwal, and T. M. Murali. A Monte Carlo algorithm for fast projective clustering. In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 418–427, 2002.
68. R. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.
69. K. Sequeira and M. Zaki. SCHISM: A New Approach for Interesting Subspace Mining. In *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pages 186–193, 2004.
70. N. Srinivas and K. Deb. Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. *Evolutionary Computation*, 2(3):221–248, 1994.
71. A. Strehl and J. Ghosh. Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research*, 3:583–617, 2002.
72. A. Strehl, J. Ghosh, and R. Mooney. Impact of Similarity Measures on Web-Page Clustering. In *Proc. of the AAAI Workshop on Artificial Intelligence for Web Search*, pages 58–64, 2000.
73. N. Tomasev, M. Radovanovic, D. Mladenic, and M. Ivanovic. The Role of Hubness in Clustering High-Dimensional Data. In *Proc. Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining (PAKDD)*, pages 183–195, 2011.
74. A. P. Topchy, A. K. Jain, and W. F. Punch. A Mixture Model for Clustering Ensembles. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 379–390, 2004.
75. A. P. Topchy, A. K. Jain, and W. F. Punch. Clustering Ensembles: Models of Consensus and Weak Partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 27(12):1866–1881, 2005.
76. C.J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.
77. H. Wang, H. Shan, and A. Banerjee. Bayesian Cluster Ensembles. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 209–220, 2009.
78. H. Wang, H. Shan, and A. Banerjee. Bayesian cluster ensembles. *Statistical Analysis and Data Mining*, 4(1):54–70, 2011.
79. P. Wang, C. Domeniconi, and K. B. Laskey. Nonparametric Bayesian Clustering Ensembles. In *Proc. Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*, pages 435–450, 2010.
80. P. Wang, K. B. Laskey, C. Domeniconi, and M. Jordan. Nonparametric Bayesian Co-clustering Ensembles. In *Proc. SIAM Int. Conf. on Data Mining (SDM)*, pages 331–342, 2011.
81. K. -G. Woo, J. -H. Lee, M. -H. Kim, and Y. -J. Lee. FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. *Information and Software Technology*, 46(4):255–271, 2004.
82. Y. Yang and M. S. Kamel. An aggregated clustering approach using multi-ant colonies algorithms. *Pattern Recognition*, 39(7):1278–1289, 2006.
83. K. Y. Yip, D. W. Cheung, and M. K. Ng. HARP: A Practical Projected Clustering Algorithm. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 16(11):1387–1397, 2004.
84. K. Y. Yip, D. W. Cheung, and M. K. Ng. On Discovery of Extremely Low-Dimensional Clusters using Semi-Supervised Projected Clustering. In *Proc.*

IEEE Int. Conf. on Data Engineering (ICDE), pages 329–340, 2005.
85. M. L. Yiu and N. Mamoulis. Iterative Projected Clustering by Subspace Mining. *IEEE Trans. on Knowledge and Data Engineering (TKDE)*, 17(2):176–189, 2005.
86. 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.

A Proofs

A.1 Proofs of Section 4.1

Lemma 1 *Given two projective clustering solutions $\mathcal{C}', \mathcal{C}''$, it holds that $\overline{\psi}_o(\mathcal{C}', \mathcal{C}'') = 0$ if and only if:*

1) *For each cluster $C' \in \mathcal{C}'$, there exists a cluster $C'' \in \mathcal{C}''$ such that $\Gamma_{C',\mathbf{o}} = \Gamma_{C'',\mathbf{o}}$, $\forall \mathbf{o}$*
2) *For each cluster $C'' \in \mathcal{C}''$, there exists a cluster $C' \in \mathcal{C}'$ such that $\Gamma_{C'',\mathbf{o}} = \Gamma_{C',\mathbf{o}}$, $\forall \mathbf{o}$*

Proof Since $\overline{\psi}_o(\mathcal{C}', \mathcal{C}'')$ is defined in (5) as the average between $\psi_o(\mathcal{C}', \mathcal{C}'')$ and $\psi_o(\mathcal{C}'', \mathcal{C}')$, and $\psi_o(\cdot, \cdot) \geq 0$, the Lemma is proved if and only if both $\psi_o(\mathcal{C}', \mathcal{C}'') = 0$ and $\psi_o(\mathcal{C}'', \mathcal{C}') = 0$. Regarding $\psi_o(\mathcal{C}', \mathcal{C}'')$, we note that:

$$\psi_o(\mathcal{C}', \mathcal{C}'') = \frac{1}{|\mathcal{C}'|} \sum_{C' \in \mathcal{C}'} \left(1 - \max_{C'' \in \mathcal{C}''} J\left(\mathbf{\Gamma}_{C'}, \mathbf{\Gamma}_{C''}\right)\right) = 1 - \frac{1}{|\mathcal{C}'|} \sum_{C' \in \mathcal{C}'} \max_{C'' \in \mathcal{C}''} J\left(\mathbf{\Gamma}_{C'}, \mathbf{\Gamma}_{C''}\right)$$

Thus, $\psi_o(\mathcal{C}', \mathcal{C}'') = 0$ if and only if $\sum_{C' \in \mathcal{C}'} \max_{C'' \in \mathcal{C}''} J\left(\mathbf{\Gamma}_{C'}, \mathbf{\Gamma}_{C''}\right) = |\mathcal{C}'|$, i.e., as $J \in [0, 1]$, if and only if, for each $C' \in \mathcal{C}'$, there exists a cluster $C'' \in \mathcal{C}''$ such that $J\left(\mathbf{\Gamma}_{C'}, \mathbf{\Gamma}_{C''}\right)$ is maximum (equal to 1). As the extended Jaccard coefficient J between any two real-valued vectors $\mathbf{u}$ and $\mathbf{v}$ is maximum if and only if $\mathbf{u}$ and $\mathbf{v}$ are exactly the same, the condition for having $\psi_o(\mathcal{C}'', \mathcal{C}') = 0$ corresponds to Condition 1) of the Lemma. An analogous reasoning applies to prove that Condition 2) is instead required for having $\psi_o(\mathcal{C}'', \mathcal{C}') = 0$.

In summary, Condition 1) of the Lemma is necessary and sufficient for $\psi_o(\mathcal{C}', \mathcal{C}'') = 0$, while Condition 2) is necessary and sufficient to have $\psi_o(\mathcal{C}'', \mathcal{C}') = 0$. This proves the Lemma. □

Lemma 2 *Given two projective clustering solutions $\mathcal{C}', \mathcal{C}''$, it holds that $\overline{\psi}_f(\mathcal{C}', \mathcal{C}'') = 0$ if and only if:*

1. *For each cluster $C' \in \mathcal{C}'$ there exists a cluster $C'' \in \mathcal{C}''$ such that $\Delta_{C',f} = \Delta_{C'',f}$, $\forall f$*
2. *For each cluster $C'' \in \mathcal{C}''$ there exists a cluster $C' \in \mathcal{C}'$ such that $\Delta_{C'',f} = \Delta_{C',f}$, $\forall f$*

Proof Analogous to Lemma 1. □

Proposition 1 *The two objective functions Ψ_o and Ψ_f of the problem defined in (2) are conflicting w.r.t. one another.*

Proof To prove the proposition, it is sufficient to find a projective ensemble $\mathcal{E}$, and any two candidate projective clustering solutions $\mathcal{C}', \mathcal{C}''$, such that the objective functions Ψ_o and Ψ_f disagree when applied to them. Formally, it should be proved that $\exists\, \mathcal{E}, \mathcal{C}', \mathcal{C}''$ such that:

$$(\Psi_o(\mathcal{C}', \mathcal{E}) - \Psi_o(\mathcal{C}'', \mathcal{E})) \times (\Psi_f(\mathcal{C}', \mathcal{E}) - \Psi_f(\mathcal{C}'', \mathcal{E})) < 0$$

which corresponds to either $\Psi_o(\mathcal{C}', \mathcal{E}) > \Psi_o(\mathcal{C}'', \mathcal{E}) \ \wedge \ \Psi_f(\mathcal{C}', \mathcal{E}) < \Psi_f(\mathcal{C}'', \mathcal{E})$ or $\Psi_o(\mathcal{C}', \mathcal{E}) < \Psi_o(\mathcal{C}'', \mathcal{E}) \ \wedge \ \Psi_f(\mathcal{C}', \mathcal{E}) > \Psi_f(\mathcal{C}'', \mathcal{E})$.

A choice for $\mathcal{E}$, $\mathcal{C}'$ and $\mathcal{C}''$ that satisfies the above requirement is as follows. Suppose $\mathcal{E}$ is composed by only one projective clustering solution $\hat{\mathcal{C}}$, and is defined over a set of three 2-dimensional objects (i.e., $|\mathcal{D}| = 3$ and $|\mathcal{F}| = 2$). $\hat{\mathcal{C}}$ has two projective clusters $\hat{C}_1$ and $\hat{C}_2$, whose object- and feature-based representations are as follows:

$$\mathbf{\Gamma}_{\hat{C}_1} = (1, 0, 0) \qquad \mathbf{\Delta}_{\hat{C}_1} = (1, 1) \qquad\qquad \mathbf{\Gamma}_{\hat{C}_2} = (0, 1, 1) \qquad \mathbf{\Delta}_{\hat{C}_2} = (1, 0)$$

The first cluster of $\hat{\mathcal{C}}$ contains the object $\mathbf{o}_1$ and is described by features 1 and 2, whereas the second cluster of $\hat{\mathcal{C}}$ contains the objects $\mathbf{o}_2$ and $\mathbf{o}_3$ and is described by feature 1 only.

Moreover, let $\mathcal{C}' = \{C'_1, C'_2\}$ and $\mathcal{C}'' = \{C''_1, C''_2\}$. The clusters within $\mathcal{C}'$ have the following object- and feature-based representations:

$$\mathbf{\Gamma}_{C'_1} = (1, 0, 0) \qquad \mathbf{\Delta}_{C'_1} = (0, 1) \qquad\qquad \mathbf{\Gamma}_{C'_2} = (0, 1, 1) \qquad \mathbf{\Delta}_{C'_2} = (1, 1)$$

That is, one cluster of $\mathcal{C}'$ (i.e., C'_1) contains the object $\mathbf{o}_1$ and is described by feature 2, and the other cluster (i.e., C'_2) contains the objects $\mathbf{o}_2$ and $\mathbf{o}_3$, and is described by all features. The clusters within $\mathcal{C}_2$ are represented as follows:

$$\mathbf{\Gamma}_{C''_1} = (1, 1, 0) \qquad \mathbf{\Delta}_{C''_1} = (1, 1) \qquad\qquad \mathbf{\Gamma}_{C''_2} = (0, 0, 1) \qquad \mathbf{\Delta}_{C''_2} = (1, 0)$$

That is, in $\mathcal{C}''$, one cluster (i.e., C''_1) contains the objects $\mathbf{o}_1$ and $\mathbf{o}_2$ and is described by features 1 and 2, and the other cluster (i.e., C''_2) contains the object $\mathbf{o}_3$ and is described by feature 1.

The pair $\langle \mathcal{C}', \hat{\mathcal{C}} \rangle$ satisfies the conditions of Lemma 1, but does not comply with Lemma 2: it holds that $\overline{\psi}_o(\mathcal{C}', \hat{\mathcal{C}}) = 0$ and $\overline{\psi}_f(\mathcal{C}', \hat{\mathcal{C}}) > 0$. As $\hat{\mathcal{C}}$ is the only solution in the projective ensemble $\mathcal{E}$, this implies that:

$$\Psi_o(\mathcal{C}', \mathcal{E}) = 0 \qquad \text{and} \qquad \Psi_f(\mathcal{C}', \mathcal{E}) > 0$$

Conversely, regarding the pair $\langle \mathcal{C}'', \hat{\mathcal{C}} \rangle$, Lemma 2 applies, whereas Lemma 1 does not. Thus, we have:

$$\Psi_o(\mathcal{C}'', \mathcal{E}) > 0 \qquad \text{and} \qquad \Psi_f(\mathcal{C}'', \mathcal{E}) = 0$$

From the latter statements, it is easy to verify that $(\Psi_o(\mathcal{C}', \mathcal{E}) - \Psi_o(\mathcal{C}'', \mathcal{E})) \times (\Psi_f(\mathcal{C}', \mathcal{E}) - \Psi_f(\mathcal{C}'', \mathcal{E})) < 0$, which proves the Proposition. □

A.2 Proofs of Section 4.2.1

Proposition 2 *In reference to the expression $\Lambda_{\mathbf{o},f}$ and the event $A_{\mathbf{o},f}$ introduced in Def. 7, it holds that:*

$$\Lambda_{\mathbf{o},f} = \frac{1}{|\mathcal{E}|} \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} \Gamma_{\hat{C},\mathbf{o}} \, \Delta_{\hat{C},f} \tag{7}$$

Proof According to the law of total probability, it results that:

$$\Lambda_{\mathbf{o},f} = \Pr(A_{\mathbf{o},f}|\mathcal{E}) = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \Pr(A_{\mathbf{o},f}|\hat{\mathcal{C}}) \; \Pr(\hat{\mathcal{C}}) = \frac{1}{|\mathcal{E}|} \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \Pr(A_{\mathbf{o},f}|\hat{\mathcal{C}}) \tag{17}$$

since the probability $\Pr(\hat{\mathcal{C}})$ of selecting the clustering solution $\hat{\mathcal{C}}$ is assumed to be the same for all the solutions within $\mathcal{E}$, as no further information is coupled with $\mathcal{E}$ (cf. Sect. 3); thus, $\Pr(\hat{\mathcal{C}}) = |\mathcal{E}|^{-1}$, $\forall \hat{\mathcal{C}} \in \mathcal{E}$.

The event $A_{\mathbf{o},f}|\hat{\mathcal{C}}$ is dependent from the set of events $\{\text{“}\mathbf{o} \in \hat{C}\text{”} \mid \hat{C} \in \hat{\mathcal{C}}\}$, which represent the assignments of object $\mathbf{o}$ to the various clusters within $\hat{\mathcal{C}}$. Such a set is a partition of the event space; thus, it can be exploited for computing $\Pr(A_{\mathbf{o},f}|\hat{\mathcal{C}})$ according to the law of total probability:

$$\Pr(A_{\mathbf{o},f}|\hat{\mathcal{C}}) = \sum_{\hat{C} \in \hat{\mathcal{C}}} \Pr(A_{\mathbf{o},f}|\text{“}\mathbf{o} \in \hat{C}\text{”}) \; \Pr(\text{“}\mathbf{o} \in \hat{C}\text{”}) \tag{18}$$

As $\Pr(\text{“}\mathbf{o} \in \hat{C}\text{”}) = \Pr(\mathbf{o}|\hat{C}) = \Gamma_{\hat{C},\mathbf{o}}$ according to Def. 2 and $\Pr(A_{\mathbf{o},f}|\text{“}\mathbf{o} \in \hat{C}\text{”}) = \Pr(f|\hat{C}) = \Delta_{\hat{C},f}$, since the probability that a feature f is informative for any object in a given projective cluster $\hat{C}$ can reasonably be assumed equal to the probability $\Pr(f|\hat{C}) = \Delta_{\hat{C},f}$ that f is informative for $\hat{C}$, any object of a given projective cluster, (17) can be rewritten as:

$$\Pr(A_{\mathbf{o},f}|\hat{\mathcal{C}}) = \sum_{\hat{C} \in \hat{\mathcal{C}}} \Gamma_{\hat{C},\mathbf{o}} \, \Delta_{\hat{C},f}$$

Substituting the latter expression into (18), we obtain

$$\Lambda_{\mathbf{o},f} = \frac{1}{|\mathcal{E}|} \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} \Gamma_{\hat{C},\mathbf{o}} \, \Delta_{\hat{C},f}$$

which proves the proposition. □

Proposition 3 *Let $\mathcal{E}$ be a projective ensemble defined over a set $\mathcal{D}$ of data objects, where each $\mathbf{o} \in \mathcal{D}$ is described by a set $\mathcal{F}$ of features. Given any two objects $\mathbf{o}, \mathbf{o}' \in \mathcal{D}$, let $d_{\mathbf{o},\mathbf{o}'}$ be the squared Euclidean distance between the object-to-cluster assignments of $\mathbf{o}$ and $\mathbf{o}'$ to the various clusters of all the solutions in $\mathcal{E}$, i.e., $d_{\mathbf{o},\mathbf{o}'} = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} (\Gamma_{\hat{C},\mathbf{o}} - \Gamma_{\hat{C},\mathbf{o}'})^2$. It holds that the squared Euclidean distance $\|\mathbf{\Lambda}_{\mathbf{o}} - \mathbf{\Lambda}_{\mathbf{o}'}\|^2$ between the feature-based representations of $\mathbf{o}$ and $\mathbf{o}'$ is directly proportional to $d_{\mathbf{o},\mathbf{o}'}$.*

Proof Let us denote by $\Phi = \{C_1, \ldots, C_H\}$ the global set $\bigcup_{\hat{\mathcal{C}} \in \mathcal{E}} \hat{\mathcal{C}}$ of clusters contained within all the solutions in the projective ensemble, where $H = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} |\hat{\mathcal{C}}|$. According to (7), it holds that:

$$\begin{aligned}\|\mathbf{\Lambda_o} - \mathbf{\Lambda_{o'}}\|^2 &= \sum_{f \in \mathcal{F}} \left(\Lambda_{\mathbf{o},f} - \Lambda_{\mathbf{o}',f}\right)^2 = \sum_{f \in \mathcal{F}} \left(\frac{1}{|\mathcal{E}|} \sum_{h=1}^{H} \Gamma_{C_h,\mathbf{o}} \Delta_{C_h,f} - \frac{1}{|\mathcal{E}|} \sum_{h=1}^{H} \Gamma_{C_h,\mathbf{o}'} \Delta_{C_h,f} \right)^2 = \\ &= \frac{1}{|\mathcal{E}|^2} \sum_{f \in \mathcal{F}} \left(\sum_{h=1}^{H} \Delta_{C_h,f} \ \left(\Gamma_{C_h,\mathbf{o}} - \Gamma_{C_h,\mathbf{o}'}\right) \right)^2 = \\ &= \frac{1}{|\mathcal{E}|^2} \sum_{f \in \mathcal{F}} \left(\sum_{h=1}^{H} \Delta_{C_h,f}^2 (\Gamma_{C_h,\mathbf{o}} - \Gamma_{C_h,\mathbf{o}'})^2 + W \right) \end{aligned} \tag{19}$$

where

$$W = 2 \sum_{h=1}^{H-1} \sum_{h'=h+1}^{H} \Delta_{C_h,f} \ \Delta_{C_{h'},f} (\Gamma_{C_h,\mathbf{o}} - \Gamma_{C_h,\mathbf{o}'}) \left(\Gamma_{C_{h'},\mathbf{o}} - \Gamma_{C_{h'},\mathbf{o}'}\right)$$

(19) clearly shows that $\|\mathbf{\Lambda_o} - \mathbf{\Lambda_{o'}}\|^2$ is directly proportional to $\sum_{h=1}^{H} \Delta_{C_h,f}^2 \ \ (\Gamma_{C_h,\mathbf{o}} - \Gamma_{C_h,\mathbf{o}'})^2 = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} \Delta_{\hat{C},f}^2 \ \left(\Gamma_{\hat{C},\mathbf{o}} - \Gamma_{\hat{C},\mathbf{o}'}\right)^2$, and hence to $d_{\mathbf{o},\mathbf{o}'}$, as $d_{\mathbf{o},\mathbf{o}'} = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} (\Gamma_{\hat{C},\mathbf{o}} - \Gamma_{\hat{C},\mathbf{o}'})^2$. □

A.3 Proofs of Section 4.2.2

Lemma 3 *It holds that* $Z_{C,f}/Y_C \leq 1$, $\forall C$, $\forall f$.

Proof The Lemma can be proved by noting that:

$$\frac{Z_{C,f}}{Y_C} = \frac{\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}}^{\alpha} \ \Lambda_{\mathbf{o},f}}{\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}}^{\alpha}} \leq 1 \quad \Leftarrow \quad \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}}^{\alpha} \Lambda_{\mathbf{o},f} \leq \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}}^{\alpha}$$

The latter inequality holds as its right hand side is an upper bound for the left side, since $\Lambda_{\mathbf{o},f} \leq 1$, $\forall f$, $\forall \mathbf{o}$ according to its probabilistic meaning explained in Proposition 2. □

Lemma 4 *It holds that* $X_{C,\mathbf{o}} \geq 0$, $Y_C \geq 0$, *and* $Z_{C,f} \geq 0$, $\forall C$, $\forall \mathbf{o}$, $\forall f$.

Proof Straightforward since $X_{C,\mathbf{o}}$, Y_C, and $Z_{C,f}$ are defined in (14), (15), and (16), respectively, as sums of terms greater than or equal to zero. Indeed, $\Gamma_{C,\mathbf{o}} \geq 0$, $\Lambda_{\mathbf{o},f} \geq 0$, $\forall C$, $\forall \mathbf{o}$, $\forall f$ due to their probabilistic meaning (Def. 2 and Proposition 2, respectively), and $(\Delta_{C,f} - \Lambda_{\mathbf{o},f})^2 \geq 0$, $\forall C$, $\forall \mathbf{o}$, $\forall f$. □

Lemma 5 *The feasible region defined by the constraints in (9)-(10) is a convex set.*

Proof Immediate as both equality (cf. (9)) and inequality (cf. (10)) constraints are linear w.r.t. the unknown quantities $\Gamma_{C,\mathbf{o}}$ and $\Delta_{C,f}$ of the problem. □

Lemma 6 *The function Q defined in (11) is convex w.r.t. $\Gamma_{C,\mathbf{o}}$ and $\Delta_{C,f}$.*

Proof Since Q is twice differentiable w.r.t. both $\Gamma_{C,\mathbf{o}}$ and $\Delta_{C,f}$, to prove the Lemma it is sufficient to show that:

$$\frac{\partial^2 Q}{\partial \Gamma_{C,\mathbf{o}}^2} \geq 0, \quad \forall \Gamma_{C,\mathbf{o}} \qquad \text{and} \qquad \frac{\partial^2 Q}{\partial \Delta_{C,f}^2} \geq 0, \quad \forall \Delta_{C,f}$$

It results that:

$$\frac{\partial^2 Q}{\partial \Gamma_{C,\mathbf{o}}^2} = \alpha\,(\alpha-1)\,(\Gamma_{C,\mathbf{o}})^{\alpha-2}\,X_{C,\mathbf{o}} \geq 0$$

since $\alpha > 1$ by definition, $\Gamma_{C,\mathbf{o}} \geq 0$ according to Def. 2, and $X_{C,\mathbf{o}} \geq 0$ according to Lemma 4. Similarly,

$$\frac{\partial^2 Q}{\partial \Delta_{C,f}^2} = 2\,Y_C \geq 0$$

since $Y_C \geq 0$ according to Lemma 4. □

Theorem 1 *For the problem P defined in (8)-(10), it holds that:*

1) Given the current values for $\Delta_{C,f}$, (12) computes the optimal $\Gamma^*_{C,\mathbf{o}}$, $\forall C$, $\forall \mathbf{o}$
2) Given the current values for $\Gamma_{C,\mathbf{o}}$, (13) computes the optimal $\Delta^*_{C,f}$, $\forall C$, $\forall f$

Proof The optimal $\Gamma^*_{C,\mathbf{o}}$ and $\Delta^*_{C,\mathbf{o}}$ can be found by means of the conventional *Lagrange multipliers* method. To this end, we first consider the relaxed problem P' obtained by temporarily discarding the inequality constraints from the constraint set of P (i.e., the constraints defined in (10)). Then, we define the new (unconstrained) objective function Q_λ for P' as follows:

$$Q_\lambda(\mathcal{C},\mathcal{E}) = Q(\mathcal{C},\mathcal{E}) \sum_{\mathbf{o}\in\mathcal{D}} \lambda'_{\mathbf{o}} \left(\sum_{C'\in\mathcal{C}} \Gamma_{C',\mathbf{o}} - 1 \right) + \sum_{C\in\mathcal{C}} \lambda''_C \left(\sum_{f'\in\mathcal{F}} \Delta_{C,f'} - 1 \right) \quad (20)$$

To prove Statement 1) of the theorem, for a fixed assignment of $\Delta_{C,f}$, we compute the optimal $\Gamma^*_{C,\mathbf{o}}$ by first retrieving the stationary points of Q_λ, i.e., the points for which

$$\nabla Q_\lambda = \left(\frac{\partial\, Q_\lambda}{\partial\, \Gamma_{C,\mathbf{o}}}, \frac{\partial\, Q_\lambda}{\partial\, \lambda'_{\mathbf{o}}} \right) = 0$$

Thus, we solve the following system of equations:

$$\frac{\partial\, Q_\lambda}{\partial\, \Gamma_{C,\mathbf{o}}} = \alpha\, (\Gamma_{C,\mathbf{o}})^{\alpha-1}\, X_{C,\mathbf{o}} + \lambda'_{\mathbf{o}} = 0 \tag{21}$$

$$\frac{\partial\, Q_\lambda}{\partial\, \lambda'_{\mathbf{o}}} = \sum_{C' \in \mathcal{C}} \Gamma_{C',\mathbf{o}} - 1 = 0 \tag{22}$$

Solving (21) w.r.t. $\Gamma_{C,\mathbf{o}}$ and substituting the solution in (22), we obtain:

$$\sum_{C' \in \mathcal{C}} \left(\frac{-\lambda'_{\mathbf{o}}}{\alpha\, X_{C',\mathbf{o}}} \right)^{\frac{1}{\alpha-1}} = 1 \tag{23}$$

Solving (23) w.r.t. $\lambda'_{\mathbf{o}}$ and substituting the solution in (21), we obtain:

$$\alpha\, (\Gamma_{C,\mathbf{o}})^{\alpha-1}\, X_{C,\mathbf{o}} - \left[\sum_{C' \in \mathcal{C}} \left(\frac{1}{\alpha\, X_{C',\mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-(\alpha-1)} = 0 \tag{24}$$

Finally, solving (24) w.r.t. $\Gamma_{C,\mathbf{o}}$, we obtain a stationary point whose expression is equal to that given in (12):

$$\Gamma^*_{C,\mathbf{o}} = \left[\sum_{C' \in \mathcal{C}} \left(\frac{X_{C,\mathbf{o}}}{X_{C',\mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-1} \tag{25}$$

Since it holds that *(i)* the stationary points of the Lagrangian function Q_λ are also stationary points of the original function Q, *(ii)* according to Lemma 5, the feasible region of P and, hence, the feasible region of P' is a convex set, and *(iii)* according to Lemma 6, Q is convex w.r.t. $\Gamma_{C,\mathbf{o}}$, it follows that such a stationary point represents a global minimum of Q, and accordingly the optimal solution of P' with $\Delta_{C,f}$ fixed.

Statement 2) of the theorem is easier to prove, as $\Delta_{C,f}$ are unconstrained in P', and hence Lagrange multipliers are not needed. Thus, for a fixed assignment of $\Gamma_{C,\mathbf{o}}$, the stationary points of the original function Q can be computed as follows:

$$\frac{\partial\, Q}{\partial\, \Delta_{C,f}} = 2 \sum_{\mathbf{o} \in \mathcal{D}} \Gamma^{\alpha}_{C,\mathbf{o}} (\Delta_{C,f} - \Lambda_{\mathbf{o},f}) = 2(\Delta_{C,f}\, Y_C - Z_{C,f}) = 0 \tag{26}$$

which can be easily solved to obtain the same expression given in (13):

$$\Delta^*_{C,f} = \frac{Z_{C,f}}{Y_C} \tag{27}$$

Similarly to the case of $\Gamma^*_{C,\mathbf{o}}$, such a stationary point represents a global minimum of Q, and therefore it represents the optimal solution of P' with $\Gamma_{C,\mathbf{o}}$ fixed.

According to the solutions of P' reported in (25) and (27), it holds that $\Gamma^*_{C,\mathbf{o}} \geq 0$ and $\Delta^*_{C,f} \geq 0$, $\forall C$, $\forall \mathbf{o}$, $\forall f$, as $X_{C,\mathbf{o}} \geq 0$, $Y_C \geq 0$, and $Z_{C,f} \geq 0$, $\forall C$, $\forall \mathbf{o}$, $\forall f$ (Lemma 4); also, $\Delta^*_{C,f} = Z_{C,f}/Y_C \leq 1$ according to Lemma 3. Therefore, such solutions satisfy the inequality constraints in (10) that were temporarily discarded in order to define the relaxed problem P'. Thus, they represent the optimal solutions of the original problem P, which proves the theorem. □

Theorem 2 *The EM-PCE algorithm (Alg. 2) converges to a local minimum of the function Q defined in (11) in a finite number of steps.*

Proof According to (11), the value of the function Q at the i-th iteration of Alg. 2 (for short, $Q^{(i)}$) can be expressed as a function of three terms:

$$Q^{(i)} = f(G^{(i)}, D^{(i)}, \mathcal{E})$$

where $\mathcal{E}$ is the input projective ensemble, and $G^{(i)} = \{\mathbf{\Gamma}_C \mid C \in \mathcal{C}^{(i)}\}$ (resp. $D^{(i)} = \{\mathbf{\Delta}_C \mid C \in \mathcal{C}^{(i)}\}$) is the set of the object-based (resp. feature-based) representation vectors of the clusters within the projective clustering solution $\mathcal{C}^{(i)}$ recognized as optimal at the i-th iteration.

According to the derivation of (12) in Theorem 1, the first step of the main cycle of the algorithm (Line 4) computes the set $G^{(i+1)}$ of object-based representation vectors at the $(i+1)$-th iteration as follows:

$$G^{(i+1)} = \arg\min_{\hat{G}} f(\hat{G}, D^{(i)}, \mathcal{E})$$

where the domain $\hat{G}$ of the $\arg\min$ function is a short-form denoting all sets of object-based representation vectors that do not violate the constraints given by the feasible region of the problem P defined in (8)-(10). Thus, it holds that $f(G^{(i+1)}, D^{(i)}, \mathcal{E}) \leq f(\hat{G}, D^{(i)}, \mathcal{E}),\ \forall\, \hat{G}$. In particular:

$$f(G^{(i+1)}, D^{(i)}, \mathcal{E}) \leq f(G^{(i)}, D^{(i)}, \mathcal{E}) \tag{28}$$

Similarly, according to the derivation of (13) in Theorem 1, the second step of the main cycle of the algorithm (Line 5) computes the set $D^{(i+1)}$ at the $(i+1)$-th iteration as follows:

$$D^{(i+1)} = \arg\min_{\hat{D}} f(G^{(i+1)}, \hat{D}, \mathcal{E})$$

Thus, it holds that $f(G^{(i+1)}, D^{(i+1)}, \mathcal{E}) \leq f(G^{(i+1)}, \hat{D}, \mathcal{E}),\ \forall\, \hat{D}$; in particular:

$$f(G^{(i+1)}, D^{(i+1)}, \mathcal{E}) \leq f(G^{(i+1)}, D^{(i)}, \mathcal{E}) \tag{29}$$

Combining (28) and (29), we obtain:

$$f(G^{(i+1)}, D^{(i+1)}, \mathcal{E}) \ \leq\ f(G^{(i+1)}, D^{(i)}, \mathcal{E}) \ \leq\ f(G^{(i)}, D^{(i)}, \mathcal{E})$$

Since $f(G^{(i+1)}, D^{(i+1)}, \mathcal{E}) = Q^{(i+1)}$ and $f(G^{(i)}, D^{(i)}, \mathcal{E}) = Q^{(i)}$, we also have:

$$Q^{(i+1)} \leq Q^{(i)} \tag{30}$$

(30) proves that Alg. 2 performs a gradient descent over the function Q. Furthermore, since Q is bounded below by 0 (indeed $Q \geq 0$), the execution of the algorithm necessarily terminates after a finite number of steps, when a fixed point (i.e., a local minimum of Q) is reached, i.e., when $Q^{(i^*)} = Q^{(i^*-1)}$ holds at the i^*-th iteration.

□

A.4 Proofs of Section 4.3

Proposition 4 *It holds that* $r(|\mathcal{D}|, |\mathcal{F}|) > 1$ *if* $(|\mathcal{D}| + |\mathcal{F}|)/K < 4\ I\ t$.

Proof Firstly, it can be noted that:

$$r(|\mathcal{D}|, |\mathcal{F}|, K) = \frac{I\ t\ K\ (|\mathcal{D}|+|\mathcal{F}|)}{|\mathcal{D}|\ |\mathcal{F}|} > 1 \Leftarrow I\ t\ K\ (|\mathcal{D}| + |\mathcal{F}|) > |\mathcal{D}|\ |\mathcal{F}| \Leftarrow$$
$$\Leftarrow \frac{2\ |\mathcal{D}|\ |\mathcal{F}|}{|\mathcal{D}|+|\mathcal{F}|} < 2\ I\ t\ K$$

that is $r(|\mathcal{D}|, |\mathcal{F}|, K) > 1$ when the harmonic mean between $|\mathcal{D}|$ and $|\mathcal{F}|$ is lower than $2\ I\ t\ K$. As the harmonic mean is never greater than the arithmetic mean, it holds that:

$$\frac{2\ |\mathcal{D}|\ |\mathcal{F}|}{|\mathcal{D}|+|\mathcal{F}|} < 2\ I\ t\ K \Leftarrow \frac{|\mathcal{D}|+|\mathcal{F}|}{2} < 2\ I\ t\ K \Leftarrow \frac{|\mathcal{D}| + |\mathcal{F}|}{K} < 4\ I\ t$$

which proves the Proposition. □