

Metacluster-based Projective Clustering Ensembles

Francesco Gullo · Carlotta Domeniconi ·
Andrea Tagarelli

Received: date / Accepted: date

Abstract The Projective Clustering Ensemble (PCE) problem is a recent clustering advance aimed at combining the two powerful tools of *clustering ensembles* and *projective clustering*. PCE has been formalized as either a two-objective or a single-objective optimization problem. Two-objective PCE has been recognized as more accurate than its single-objective counterpart, although it is unable to jointly handle the object-based and feature-based cluster representations.

In this paper, we push forward the current PCE research, aiming to overcome the limitations of all existing PCE formulations. We propose a novel single-objective PCE formulation so that *(i)* the object-based and feature-based cluster representations are jointly considered, and *(ii)* the resulting optimization strategy follows a metacluster-based methodology borrowed from traditional clustering ensembles. As a result, the proposed formulation features best suitability to the PCE problem, thus guaranteeing improved effectiveness. Experiments on benchmark datasets have shown how the proposed approach achieves better average accuracy than all existing PCE methods, as well as ef-

F. Gullo
Yahoo! Research
08018 Barcelona, Spain
E-mail: gullo@yahoo-inc.com

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

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

efficiency superior to the most accurate existing metacluster-based PCE method on larger datasets.

Keywords Clustering · Clustering Ensembles · Projective Clustering · Subspace Clustering · Dimensionality reduction · Optimization

1 Introduction

After more than four decades, a large number of algorithms has been developed for data clustering, focusing on different aspects such as data types, algorithmic features, and application targets [14]. In the last few years, there has been an increasing interest in developing advanced tools for data clustering. In this respect, *projective clustering* and *clustering ensembles* represent two of the most important directions of research.

The goal of projective clustering (or projected clustering) [32, 45, 1, 13, 29] is to discover *projective clusters*, i.e., subsets of the input data having different (possibly overlapping) subsets of features (subspaces) associated with them. Projective clustering is closely related to the *subspace clustering* problem [4, 34, 26, 30], as both detect clusters that exist in different subspaces; however, the goal of subspace clustering is to search for *all* clusters in *all* meaningful subspaces, whereas projective clustering methods output a single partition of the input dataset. Projective clustering aims to solve issues that typically arise in high-dimensional data, such as sparsity and concentration of distances [6, 21, 40]. Existing projective clustering methods [26, 30] can be classified into four main approaches: *bottom-up* (it finds subspaces recognized as “interesting” and assigns each data object to the most similar subspace [29, 37]), *top-down* (it finds the subspaces starting from the full feature space [32, 28, 43, 1, 3, 44, 42, 44, 7]), *soft* (it produces soft data clusterings [29, 10] and/or clusterings having differently weighted feature-to-cluster assignments [13, 10]), and *hybrid* (it combines elements of both projective and subspace clustering approaches [35, 2, 25]).

The problem of clustering ensembles [38, 41, 12, 15], also known as consensus clustering [33] or aggregation clustering [16], is stated as follows: given a set of clustering solutions, or *ensemble*, to derive a *consensus clustering* that properly summarizes the solutions in the ensemble. The input ensemble is typically generated by varying one or more aspects of the clustering process, such as the clustering algorithm, the parameter setting, and the number of features, objects or clusters. The majority of the existing clustering ensemble methods follow the *instance-based* approach or the *metacluster-based* approach, or a combination of both (i.e., *hybrid*). In particular, the metacluster-based approach lies in the principle “to cluster clusters” [9, 38, 8], whereby a new dataset is inferred whose objects are the clusters that belong to the clustering solutions in the ensemble. This dataset is then partitioned in order to produce a set of *metaclusters* (i.e., sets of clusters), and the consensus clustering is finally computed by assigning each data object to the metacluster that optimizes a specific criterion (e.g., majority voting).

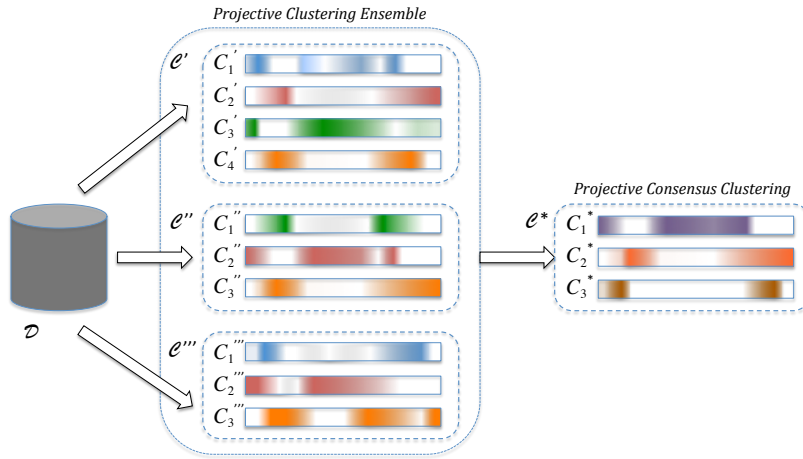


Fig. 1 Illustration of a projective clustering ensemble and derived consensus clustering. Each gradient refers to the cluster memberships over all objects. Colors denote different feature subspaces associated with the projective clusters. (The color version of this figure is available only in the electronic edition)

Projective clustering and clustering ensembles have been recently treated in a unified framework [17, 20]. The underlying motivation of that study is that many real-world application problems are high dimensional and lack a-priori knowledge. Examples are: clustering of multi-view data, privacy preserving clustering, news or document retrieval based on pre-defined categorizations, and distributed clustering of high-dimensional data. To address both issues simultaneously, the problem of *projective clustering ensembles* (PCE) is hence formalized, whose goal is to compute a *projective consensus clustering* from an ensemble of projective clustering solutions. 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 projective clustering solutions, which are obtained according to different views over the same dataset. A projective cluster is graphically represented as a rectangle filled with a color gradient, where higher intensities correspond to larger membership values 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. In particular, the first projective consensus cluster is derived by summarizing C_1' , C_2'' , and C_2''' , the second is derived from C_2' , C_3'' , and C_3''' , and the third 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.

Two formulations of PCE have been proposed in [20], namely *two-objective* PCE and *single-objective* PCE. The former consists in the optimization of two objective functions, which separately consider the data object clustering and the feature-to-cluster assignment. The latter embeds in one objective function the object-based and feature-based representations of the various clusters. A heuristic developed for two-objective PCE, called *MOEA-PCE*, has shown to be particularly accurate, although it has drawbacks concerning efficiency, parameter setting, and interpretability of results. In contrast, the heuristic developed for single-objective PCE, called *EM-PCE*, has shown better efficiency while being outperformed by two-objective PCE in terms of effectiveness. An attempt to improve the accuracy of single-objective PCE has been proposed in [18]; the approach is based on a corrective term designed to achieve a better balance between the object-to-cluster assignment and the feature-to-cluster assignment when measuring the error of a candidate projective consensus clustering. Despite the achieved improvement in accuracy, the heuristics in [18] are still outperformed by two-objective PCE, thus suggesting that the path indicated by the two-objective formulation is the one to be followed. Nevertheless, the two-objective PCE suffers from a major weakness: it does not take into consideration the interrelation between the object-based and the feature-based cluster representations. In a nutshell, according to the early two-objective PCE formulation, consensus clusterings whose clusters have both object-based and feature-based representations that well comply with the input ensemble but that are not correctly “coupled” with one another might mistakenly be recognized as ideal. This fact can lead to projective consensus clustering solutions that contain conceptual flaws in their cluster composition. By preventing this scenario, one can improve the two-objective PCE formulation and corresponding methods.

Contributions. In this paper, we pursue a new approach to the study of PCE, which is motivated by our insights on the theoretical foundations of two-objective and single-objective PCE formulations. Our aim is to provide a stronger tie between the PCE and the traditional clustering ensemble problem. By investigating the opportunity of adapting existing approaches for clustering ensembles to the PCE problem, we propose a new single-objective formulation of PCE which resembles a *metacluster-based* clustering ensemble approach and extends our first attempt in this regard [19]. The key idea underlying our proposal is to define a function that measures the distance of a projective clustering solution from a given ensemble, in such a way that the object-based and the feature-based cluster representations are considered as a whole. We show that the new PCE formulation is theoretically sound and has advantages over the ones proposed previously [20, 19]. It enables the development of heuristic algorithms that can exploit the results obtained by the majority of existing clustering ensemble methods. Specifically, we define a heuristic that follows a metacluster-based approach, called *Enhanced MetaCluster-Based Projective Clustering Ensembles* (E-CB-PCE), which computes the consensus clustering starting from a partition (i.e., metaclustering) inferred from the set of

clusters belonging to the ensemble components. Compared to the previously developed metacluster-based heuristics in [19], E-CB-PCE involves more constraints aimed to ensure a total coverage of the ensemble components in terms of clusters selected for deriving the projective consensus clustering solution. Experimental results have revealed that the proposed E-CB-PCE is on average more accurate than all previous PCE methods, according to each of the selected assessment criteria. Moreover, E-CB-PCE has shown to improve the efficiency of its most direct competitor CB-PCE [19] on larger datasets up to two orders of magnitude.

Organization of the paper. The remainder of this paper is organized as follows. Section 2 provides the background on the PCE problem and its early single-objective and two-objective formulations. Section 3 focuses on the proposed metacluster-based PCE approach. Section 4 presents the two developed heuristics for the metacluster-based PCE along with an analysis of their computational complexity. Section 5 describes the experimental evaluation and presents the results, and Section 6 concludes the paper. Finally, we give the proofs of all the theoretical results of the paper in the Appendix.

2 Early Projective Clustering Ensembles (PCE)

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} .¹ 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* assignments of the f -th feature to C , i.e., the probability $\Pr(f|C)$ that the feature f is *informative* for cluster C (f belongs to the subspace associated with C).

The object-based ($\mathbf{\Gamma}_C$) and the feature-based ($\mathbf{\Delta}_C$) representations of a projective cluster C implicitly define the *projective cluster representation matrix* (for short, *projective matrix*) \mathbf{X}_C of C . \mathbf{X}_C is a $|\mathcal{D}| \times |\mathcal{F}|$ matrix that stores, $\forall \mathbf{o} \in \mathcal{D}$, $f \in \mathcal{F}$, the probability of the intersection of the events “*object \mathbf{o} belongs to C* ” and “*feature f belongs to the subspace associated with C* ”. Under the assumption of independence between the two events, such a probability is equal to $\Pr(C|\mathbf{o}) = \Gamma_{C,\mathbf{o}}$ joint with $\Pr(f|C) = \Delta_{C,f}$. Hence, given $\mathcal{D} = \{\mathbf{o}_1, \dots, \mathbf{o}_{|\mathcal{D}|}\}$ and $\mathcal{F} = \{1, \dots, |\mathcal{F}|\}$, the matrix \mathbf{X}_C can formally be

¹ Vectorial notation here denotes row vectors.

defined as:

$$\mathbf{X}_C = \mathbf{\Gamma}_C^T \mathbf{\Delta}_C = \begin{pmatrix} \Gamma_{C,\mathbf{o}_1} \times \Delta_{C,1} & \dots & \Gamma_{C,\mathbf{o}_1} \times \Delta_{C,|\mathcal{F}|} \\ \vdots & & \vdots \\ \Gamma_{C,\mathbf{o}_{|\mathcal{D}|}} \times \Delta_{C,1} & \dots & \Gamma_{C,\mathbf{o}_{|\mathcal{D}|}} \times \Delta_{C,|\mathcal{F}|} \end{pmatrix}. \quad (1)$$

A couple of interesting results put in relation the projective matrix and the object/feature-based representation of a projective cluster. Such results are formalized in the following two propositions. We will exploit them in the remainder of the paper.

Proposition 1 *For any two projective clusters C, C' it holds that $\mathbf{X}_C = \mathbf{X}_{C'}$ if and only if $\mathbf{\Gamma}_C = \mathbf{\Gamma}_{C'}$ and $\mathbf{\Delta}_C = \mathbf{\Delta}_{C'}$. \square*

Proposition 2 *For any projective cluster C , its object-based representation $\mathbf{\Gamma}_C = (\Gamma_{C,\mathbf{o}_1}, \dots, \Gamma_{C,\mathbf{o}_{|\mathcal{D}|}})$ and feature-based representation $\mathbf{\Delta}_C = (\Delta_{C,1}, \dots, \Delta_{C,|\mathcal{F}|})$ can uniquely be derived from its projective matrix \mathbf{X}_C as follows:*

$$\Gamma_{C,\mathbf{o}_i} = \sum_{j=1}^{|\mathcal{F}|} X_C(i, j), \quad \Delta_{C,j} = \frac{X_C(1, j)}{\Gamma_{C,\mathbf{o}_1}} = \dots = \frac{X_C(|\mathcal{D}|, j)}{\Gamma_{C,\mathbf{o}_{|\mathcal{D}|}}},$$

where $X_C(i, j)$ denotes the element (i, j) of the matrix \mathbf{X}_C . \square

A *projective clustering solution*, denoted by C , is defined as a set of projective clusters that satisfy the following conditions:

$$\sum_{C \in \mathcal{C}} \Gamma_{C,\mathbf{o}} = 1, \quad \forall \mathbf{o} \in \mathcal{D}, \quad \text{and} \quad \sum_{f \in \mathcal{F}} \Delta_{C,f} = 1, \quad \forall C \in \mathcal{C}. \quad (2)$$

The semantics of any projective clustering \mathcal{C} is that for each projective cluster $C \in \mathcal{C}$, the objects belonging to C are close to each other if (and only if) they are projected onto the subspace associated with C .

A *projective ensemble* \mathcal{E} is defined as a set of projective clustering solutions. No information about the ensemble generation strategy (algorithms and/or setups), nor original feature values of the objects within \mathcal{D} are provided along with \mathcal{E} . Moreover, each projective clustering solution in \mathcal{E} may contain in general a different number of clusters.

The goal of PCE is to derive a *projective consensus clustering* that properly summarizes the projective clustering solutions within the input projective ensemble.

2.1 Single-objective PCE

A first PCE formulation proposed in [20] is based on a single-objective function:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \sum_{C \in \mathcal{C}} \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}}^\alpha X_{C,\mathbf{o}}, \quad (3)$$

where

$$X_{C,\mathbf{o}} = \sum_{f \in \mathcal{F}} (\Delta_{C,f} - \Lambda_{\mathbf{o},f})^2, \quad (4)$$

$$\Lambda_{\mathbf{o},f} = \frac{1}{|\mathcal{E}|} \sum_{\hat{c} \in \mathcal{E}} \sum_{\hat{c} \in \hat{c}} \Gamma_{\hat{c},\mathbf{o}} \Delta_{\hat{c},f}, \quad (5)$$

and $\alpha > 1$ is a positive integer that ensures non-linearity of the objective function w.r.t. $\Gamma_{C,\mathbf{o}}$. To solve the optimization problem based on the above function, the *EM-based Projective Clustering Ensembles (EM-PCE)* heuristic is defined. EM-PCE iteratively looks for the optimal values of $\Gamma_{C,\mathbf{o}}$ (resp. $\Delta_{C,f}$) while keeping $\Delta_{C,f}$ (resp. $\Gamma_{C,\mathbf{o}}$) fixed, until convergence.

Weaknesses of single-objective PCE. The objective function at the basis of the problem in (3) does not allow for a perfect balance between object- and feature-to-cluster assignments when measuring the error of a candidate projective consensus clustering solution. This weakness is formally shown in [18] and avoided by adjusting (3) with a corrective term. The final form of the problem based on the corrected objective function is the following:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \sum_{C \in \mathcal{C}} \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}}^\alpha \left(\frac{1}{2|\mathcal{E}|} X_{C,\mathbf{o}} + \frac{1}{|\mathcal{D}|-1} X'_{C,\mathbf{o}} \right), \quad (6)$$

where

$$X'_{C,\mathbf{o}} = \sum_{\mathbf{o}' \neq \mathbf{o}} \left(1 - \frac{\Gamma_{C,\mathbf{o}'}}{|\mathcal{E}|} \sum_{\hat{c} \in \mathcal{E}} \sum_{\hat{c} \in \hat{c}} \Gamma_{\hat{c},\mathbf{o}} \Gamma_{\hat{c},\mathbf{o}'} \right).$$

The above optimization problem is tackled in [18] by proposing two different heuristics. The first one, called *E-EM-PCE*, follows the same scheme as the EM-PCE algorithm for the early single-objective PCE formulation. The second heuristic, called *E-2S-PCE*, consists of two sequential steps that handle the object-to-cluster and the feature-to-cluster assignments separately.

2.2 Two-objective PCE

PCE is also formulated in [20] as a two-objective optimization problem, whose functions take into account the object-based (function Ψ_o) and the feature-based (function Ψ_f) cluster representations of a given projective ensemble \mathcal{E} , respectively:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \{ \Psi_o(\mathcal{C}, \mathcal{E}), \Psi_f(\mathcal{C}, \mathcal{E}) \}, \quad (7)$$

where the argmin function is over all possible projective clustering solutions \mathcal{C} that satisfy the conditions reported in (2) (this makes the searching space in principle infinite), and

$$\Psi_o(\mathcal{C}, \mathcal{E}) = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \bar{\psi}_o(\mathcal{C}, \hat{\mathcal{C}}), \quad \Psi_f(\mathcal{C}, \mathcal{E}) = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \bar{\psi}_f(\mathcal{C}, \hat{\mathcal{C}}). \quad (8)$$

Functions $\bar{\psi}_o$ and $\bar{\psi}_f$ are defined as $\bar{\psi}_o(\mathcal{C}', \mathcal{C}'') = \frac{1}{2}(\psi_o(\mathcal{C}', \mathcal{C}'') + \psi_o(\mathcal{C}'', \mathcal{C}'))$ and $\bar{\psi}_f(\mathcal{C}', \mathcal{C}'') = \frac{1}{2}(\psi_f(\mathcal{C}', \mathcal{C}'') + \psi_f(\mathcal{C}'', \mathcal{C}'))$, respectively, where

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

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

and $J(\mathbf{u}, \mathbf{v}) = (\mathbf{u} \mathbf{v}^T) / (\|\mathbf{u}\|_2^2 + \|\mathbf{v}\|_2^2 - \mathbf{u} \mathbf{v}^T) \in [0, 1]$ denotes the extended Jaccard similarity coefficient (also known as Tanimoto coefficient) between any two real-valued vectors \mathbf{u} and \mathbf{v} [39].

The problem defined in (7) is solved by a heuristic, called *MOEA-PCE*, in which a *Pareto*-based *Multi-Objective Evolutionary Algorithm* is exploited to avoid combining the two objective functions into a single one.

Weaknesses of two-objective PCE. Experimental evidence in [20] has shown that the two-objective PCE formulation is much more accurate than the single-objective counterpart. Nevertheless, the original two-objective PCE also suffers from an important conceptual issue which has been firstly identified in [19], and we further investigate in this paper. The existence of this issue proves that the accuracy of two-objective PCE can still be improved, which is a major goal of this work. We unveil this issue in the following example.

Example 1 Let \mathcal{E} be a projective ensemble defined over a set \mathcal{D} of data objects and a set \mathcal{F} of features. Suppose that \mathcal{E} contains only one projective clustering solution \mathcal{C} and that \mathcal{C} in turn contains two projective clusters \mathcal{C}' and \mathcal{C}'' , whose object- and feature-based representations are different from one another, i.e., $\exists \mathbf{o} \in \mathcal{D}$ s.t. $\Gamma_{\mathcal{C}', \mathbf{o}} \neq \Gamma_{\mathcal{C}'', \mathbf{o}}$, and $\exists f \in \mathcal{F}$ s.t. $\Delta_{\mathcal{C}', f} \neq \Delta_{\mathcal{C}'', f}$.

Let us consider two candidate projective consensus clusterings $\mathcal{C}_1 = \{\mathcal{C}'_1, \mathcal{C}''_1\}$ and $\mathcal{C}_2 = \{\mathcal{C}'_2, \mathcal{C}''_2\}$. We assume that $\mathcal{C}_1 = \mathcal{C}$, whereas \mathcal{C}_2 is defined as follows. Cluster \mathcal{C}'_2 has object- and feature-based representations given by $\mathbf{\Gamma}_{\mathcal{C}'}$ (i.e., the object-based representation of the first cluster \mathcal{C}' within \mathcal{C}) and $\mathbf{\Delta}_{\mathcal{C}''}$ (i.e., the feature-based representation of the second cluster \mathcal{C}'' within \mathcal{C}), respectively; cluster \mathcal{C}''_2 has object- and feature-based representations given by $\mathbf{\Gamma}_{\mathcal{C}''}$ (i.e., the object-based representation of the second cluster \mathcal{C}'' within \mathcal{C}) and $\mathbf{\Delta}_{\mathcal{C}'}$ (i.e., the feature-based representation of the first cluster \mathcal{C}' within \mathcal{C}), respectively. According to (8), it is easy to see that:

$$\Psi_o(\mathcal{C}_1, \mathcal{E}) = \Psi_o(\mathcal{C}_2, \mathcal{E}) = 0, \quad \text{and} \quad \Psi_f(\mathcal{C}_1, \mathcal{E}) = \Psi_f(\mathcal{C}_2, \mathcal{E}) = 0.$$

Thus, both candidates \mathcal{C}_1 and \mathcal{C}_2 minimize the objectives of the early two-objective PCE formulation reported in (7), and hence, they are both recognized as optimal solutions. This conclusion is conceptually wrong, because only \mathcal{C}_1 should be recognized as an optimal solution, since only \mathcal{C}_1 exactly corresponds to the unique solution of the ensemble. Conversely, \mathcal{C}_2 is not well-representative of the ensemble \mathcal{E} , as the object- and the feature-based representations of its clusters are inversely associated to each other w.r.t. the associations in \mathcal{C} . Indeed, in \mathcal{C}_2 , $\mathcal{C}'_1 = \langle \Gamma_{\mathcal{C}'}, \Delta_{\mathcal{C}''} \rangle$ and $\mathcal{C}''_1 = \langle \Gamma_{\mathcal{C}''}, \Delta_{\mathcal{C}'}, \Delta_{\mathcal{C}''} \rangle$, whereas, the solution $\mathcal{C} \in \mathcal{E}$ is such that $\mathcal{C}' = \langle \Gamma_{\mathcal{C}'}, \Delta_{\mathcal{C}'}, \Delta_{\mathcal{C}''} \rangle$ and $\mathcal{C}'' = \langle \Gamma_{\mathcal{C}''}, \Delta_{\mathcal{C}''} \rangle$. \square

The issue described in the above example arises because the two-objective PCE formulation ignores that the object-based and the feature-based representations of a projective cluster are strictly coupled to one another and, therefore, they need to be considered as a whole. In other words, in order to effectively evaluate the quality of a candidate projective consensus clustering, detecting the correct object-based and feature-based representations of the various clusters in a standalone fashion is not enough; instead, a mapping between the two (i.e., which object-based representation a feature-based representation should be coupled with and vice versa) should be properly discovered as well. For this purpose, the objective functions Ψ_o and Ψ_f should not be kept separated, but they should somehow be put in relation to one another. We show next how this can be overcome by employing a *metacluster-based* PCE formulation.

3 Metacluster-based PCE

3.1 Early metacluster-based PCE formulation

In our previous work [19], we attempted to solve the main drawback of two-objective PCE shown in Example 1 by proposing the following alternative formulation based on a single-objective function:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \Psi_{of}(\mathcal{C}, \mathcal{E}), \quad (9)$$

where Ψ_{of} is a function designed to measure the “distance” of any well-defined projective clustering solution \mathcal{C} from \mathcal{E} in terms of both data clustering and feature-to-cluster assignment. To define Ψ_{of} , we resorted to the early two-objective PCE formulation and adapted the (asymmetric) measure therein involved to the new setting:

$$\Psi_{of}(\mathcal{C}, \mathcal{E}) = \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \bar{\psi}_{of}(\mathcal{C}, \hat{\mathcal{C}}), \quad (10)$$

where

$$\bar{\psi}_{of}(\mathcal{C}', \mathcal{C}'') = \frac{1}{2} \left(\psi_{of}(\mathcal{C}', \mathcal{C}'') + \psi_{of}(\mathcal{C}'', \mathcal{C}') \right), \quad (11)$$

and

$$\psi_{of}(C', C'') = \frac{1}{|C'|} \sum_{C'' \in C'} \left(1 - \max_{C''' \in C''} \hat{J}(\mathbf{X}_{C'}, \mathbf{X}_{C''}) \right). \quad (12)$$

In order to measure the similarity between any pair C', C'' of projective clusters jointly in terms of object-based representation and feature-based representation, the corresponding projective matrices $\mathbf{X}_{C'}$ and $\mathbf{X}_{C''}$ are compared to each other. To accomplish this, the two matrices are first linearized as monodimensional vectors, and then compared by means of some distance measure between real-valued vectors. We resorted to the Tanimoto similarity coefficient (also known as extended Jaccard coefficient), as it represents a trade-off solution between Euclidean and Cosine measures in terms of scale/translation invariance [39]. Moreover, it has a fixed-range codomain $([0, 1])$, which is a desirable property in the design of the proposed objective function. More precisely, the generalized definition of the Tanimoto coefficient operating on real-valued matrices is as follows:

$$\hat{J}(\mathbf{X}, \hat{\mathbf{X}}) = \frac{\sum_{i=1}^{|\text{rows}(\mathbf{X})|} \mathbf{X}_i \hat{\mathbf{X}}_i^T}{\|\mathbf{X}\|_2^2 + \|\hat{\mathbf{X}}\|_2^2 - \sum_{i=1}^{|\text{rows}(\mathbf{X})|} \mathbf{X}_i \hat{\mathbf{X}}_i^T}, \quad (13)$$

where $\mathbf{X}_i \hat{\mathbf{X}}_i^T$ denotes the scalar product between the i -th rows of matrices \mathbf{X} and $\hat{\mathbf{X}}$.

It is easy to note that the PCE formulation reported in (9) is well-suited to measure the quality of a candidate consensus clustering in terms of both object-to-cluster and feature-to-cluster assignments as a whole. As shown in [19], this allows for overcoming the conceptual disadvantages of both early single-objective and two-objective PCE. Particularly, the issue described in Example 1 does not arise anymore with the PCE formulation in (9). Indeed, considering again the two candidate projective consensus clusterings \mathcal{C}_1 and \mathcal{C}_2 of Example 1, it is straightforward to see that:

$$\Psi_{of}(\mathcal{C}_1, \mathcal{E}) = 0, \quad \text{and} \quad \Psi_{of}(\mathcal{C}_2, \mathcal{E}) > 0.$$

and hence, \mathcal{C}_1 would be correctly recognized as an optimal solution, whereas \mathcal{C}_2 would not.

Weaknesses of early metacluster-based PCE formulation. Although the early metacluster-based PCE formulation in (9) mitigates the issues of two-objective PCE, we will show how to further improve that formulation. For this purpose, next we provide a detailed analysis of the theoretical properties of the formulation in (9), with the ultimate goal of defining an *enhanced* metacluster-based PCE formulation that discards the controversial aspects of the earlier formulation.

Let us denote the expression $1 - \hat{J}(\mathbf{X}_{C'}, \mathbf{X}_{C''})$ with $T(\mathbf{X}_{C'}, \mathbf{X}_{C''})$ or, more simply, $T(C', C'')$, for any two clusters C', C'' . Since $\hat{J} \in [0, 1]$, $T(\cdot, \cdot)$ is

regarded as Tanimoto distance [19]. Combining (10), (11), and (12), we have that:

$$\begin{aligned}\Psi_{of}(\mathcal{C}, \mathcal{E}) &= \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \bar{\psi}_{of}(\mathcal{C}, \hat{\mathcal{C}}) = \\ &= \sum_{\hat{\mathcal{C}} \in \mathcal{E}} \frac{1}{2} \left(\frac{1}{|\mathcal{C}|} \sum_{C \in \mathcal{C}} \left(1 - \max_{\hat{C} \in \hat{\mathcal{C}}} \hat{J}(\mathbf{X}_C, \mathbf{X}_{\hat{C}}) \right) + \frac{1}{|\hat{\mathcal{C}}|} \sum_{\hat{C} \in \hat{\mathcal{C}}} \left(1 - \max_{C \in \mathcal{C}} \hat{J}(\mathbf{X}_{\hat{C}}, \mathbf{X}_C) \right) \right).\end{aligned}$$

By discarding the constant terms and observing that

$$1 - \max_{C \in \mathcal{C}} \hat{J}(\mathbf{X}_C, \mathbf{X}_{\hat{C}}) = \min_{C \in \mathcal{C}} (1 - \hat{J}(\mathbf{X}_C, \mathbf{X}_{\hat{C}})) = \min_{C \in \mathcal{C}} T(C, \hat{C}),$$

we can rewrite Ψ_{of} as:

$$\Psi_{of}(\mathcal{C}, \mathcal{E}) = \underbrace{\sum_{\hat{\mathcal{C}} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} \min_{C \in \mathcal{C}} T(C, \hat{C})}_{\Psi'_{of}(\mathcal{C}, \mathcal{E})} + \underbrace{\sum_{C \in \mathcal{C}} \sum_{\hat{C} \in \mathcal{E}} \min_{\hat{C} \in \hat{\mathcal{C}}} T(C, \hat{C})}_{\Psi''_{of}(\mathcal{C}, \mathcal{E})}. \quad (14)$$

Thus, the objective function Ψ_{of} corresponds to a sum of two objective functions Ψ'_{of} and Ψ''_{of} , which are the focus of the following discussion. Let us introduce the variable $x(\hat{C}, C) \in \{0, 1\}$, which is 1 if cluster \hat{C} , belonging to any projective clustering solution within the ensemble \mathcal{E} , is “mapped” to the cluster C of the candidate projective consensus clustering \mathcal{C} , and 0 otherwise. The meaning of the “mapping” between clusters is clarified next.

EXPLAINING THE Ψ'_{of} FUNCTION. The Ψ'_{of} function models a modified version of the K -Means problem [22], where (i) the input dataset for the clustering task corresponds to the set of projective clusters belonging to all solutions in the ensemble \mathcal{E} , i.e., the set $\{\hat{C} \mid \hat{C} \in \hat{\mathcal{C}}, \hat{\mathcal{C}} \in \mathcal{E}\}$, (ii) each “object” in such a dataset is hence a projective cluster represented by its projective matrix, (iii) the centers have the form of projective matrices satisfying the constraints defined in (2), and (iv) the distance between objects and centers is computed according to the Tanimoto distance $T(\cdot, \cdot)$, rather than the classic squared Euclidean distance. Formally, the Ψ'_{of} function models the following optimization problem:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \sum_{C \in \mathcal{C}} \sum_{\hat{C} \in \mathcal{E}} \sum_{\hat{C} \in \hat{\mathcal{C}}} x(\hat{C}, C) T(\hat{C}, C) \quad (15)$$

s.t.

$$\begin{aligned}\mathcal{C} &\text{ satisfies (2),} \\ x(\hat{C}, C) &\in \{0, 1\}, \quad \forall \hat{C} \in \mathcal{E}, \forall \hat{C} \in \hat{\mathcal{C}}, \forall C \in \mathcal{C}, \\ \sum_{C \in \mathcal{C}} x(\hat{C}, C) &= 1, \quad \forall \hat{C} \in \mathcal{E}, \forall \hat{C} \in \hat{\mathcal{C}}.\end{aligned} \quad (16)$$

Due to the analogies with K -Means, it is easy to see that the solution for the above problem is a set of K “center” projective matrices (representing

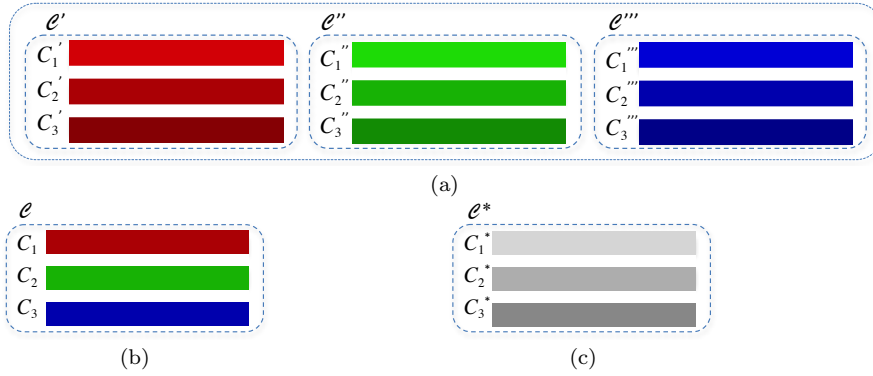


Fig. 2 (a) a projective ensemble where, for each projective cluster, different projective matrix representations are depicted with different color shades, (b) optimal projective consensus clustering according to the Ψ'_{of} objective function, (c) expected projective consensus clustering. (The color version of this figure is available only in the electronic edition)

the various clusters in the optimal C^*) that minimize the sum of the Tanimoto distances between each projective cluster in the ensemble and its closest center. In the following example we show that such a problem might not be particularly appropriate in the context of PCE.

Example 2 Figure 2-(a) shows an ensemble composed by three projective clustering solutions. Projective clusters in each solution are depicted as colored rectangles whose color shade corresponds to a certain representation provided by its corresponding projective matrix. Similar shades denote similar projective matrices, and, therefore, similar clusters.² In this example, the clusters within the same projective clustering solution are highly similar to each other, and are highly dissimilar from the clusters of the other solutions. Assuming $K = 3$ clusters in the output projective consensus clustering, the optimal partition of the ensemble of Fig. 2-(a) according to the problem defined in (15)–(16) would correspond to the one identified by the solutions in the ensemble themselves. This leads to the optimal consensus clustering reported in Fig. 2-(b), whose clusters C_1 , C_2 , and C_3 take the colors *red*, *green*, and *blue* as a result of the summarization of the sets $\{C'_1, C'_2, C'_3\}$, $\{C''_1, C''_2, C''_3\}$, $\{C'''_1, C'''_2, C'''_3\}$, respectively. This solution is actually not very intuitive for PCE, since a good projective consensus clustering is supposed to summarize the information available from the ensemble by putting in relation clusters that belong to different solutions of the ensemble. Therefore, it would be much more meaningful if a consensus clustering would have the form reported in Fig. 2-(c), whose *grey*-shaded clusters C_1^* , C_2^* , and C_3^* derive from mixing the *red*,

² The meaning of the colors here is different from Figure 1, where different colors and gradients refer to the distinct parts of the two-fold projective cluster representation, i.e., the feature-to-cluster assignment and the object-to-cluster assignment, respectively. Here, instead, we focus on the unified view of the representation provided by the projective matrix.

green, and blue colors of the sets $\{C'_1, C''_1, C'''_1\}$, $\{C'_2, C''_2, C'''_2\}$, $\{C'_3, C''_3, C'''_3\}$, respectively. \square

EXPLAINING THE Ψ''_{of} FUNCTION. The optimization problem defined by the Ψ''_{of} function in (14) can be rewritten as follows:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \sum_{C \in \mathcal{C}} \sum_{\hat{C} \in \mathcal{E}} \sum_{C \in \hat{C}} x(\hat{C}, C) T(\hat{C}, C) \quad (17)$$

s.t.

$$\begin{aligned} &\mathcal{C} \text{ satisfies (2),} \\ &x(\hat{C}, C) \in \{0, 1\}, \quad \forall \hat{C} \in \mathcal{E}, \forall C \in \hat{C}, \\ &\sum_{\hat{C} \in \mathcal{E}} x(\hat{C}, C) = 1, \quad \forall C \in \mathcal{C}. \end{aligned} \quad (18)$$

Let us informally explain what is given above. Consider first the case where the number K of clusters in the output consensus clustering is equal to 1 and consider all subsets \mathbf{C} of $\{\hat{C} \mid \hat{C} \in \mathcal{C}, \mathcal{C} \in \mathcal{E}\}$ that satisfy the following condition: each solution within the ensemble \mathcal{E} has exactly one cluster in \mathbf{C} ; also, for each \mathbf{C} , let $\mathbf{X}_{\mathbf{C}}$ denote the projective matrix that (i) satisfies (2), and (ii) minimizes the sum $d_{\mathbf{C}}$ of the Tanimoto distances between itself and (the projective matrices of) all members of \mathbf{C} . The problem defined by Ψ''_{of} (for $K = 1$) aims to find a matrix $\mathbf{X}_{\mathbf{C}^*}$ such that $d_{\mathbf{C}}$ is minimum; the final output would be a projective clustering \mathcal{C}^* composed by only one cluster whose projective matrix corresponds to the matrix $\mathbf{X}_{\mathbf{C}^*}$. This interpretation can easily be generalized to the case $K > 1$; indeed, in this case, all K optimal solution matrices (i.e., projective clusters in \mathcal{C}^*) would necessarily have the same form, as formally stated in the next proposition.

Proposition 3 *Given a projective ensemble \mathcal{E} , let \mathcal{Y} denote the set of all clusterings obtained by taking exactly one cluster from each ensemble member, i.e., $\mathcal{Y} = \{\mathbf{C} \mid \mathbf{C} \subseteq \{\hat{C} \mid \hat{C} \in \mathcal{C} \wedge \hat{C} \in \mathcal{E}\} \wedge |\hat{C} \cap \mathbf{C}| = 1, \forall \hat{C} \in \mathcal{E}\}$. Moreover, let $\mathbf{X}^* = \arg \min_{\hat{\mathbf{X}}} \min_{\mathbf{C} \in \mathcal{Y}} \sum_{C \in \mathbf{C}} T(\hat{\mathbf{X}}, \mathbf{X}_C)$ s.t. $\hat{\mathbf{X}}$ satisfies (2). Given an integer $K \geq 1$, it holds that the optimal projective consensus clustering for the optimization problem defined in (17)–(18) is $\mathcal{C}^* = \{C^*_1, \dots, C^*_K\}$ s.t. $\mathbf{X}_{C^*_1} = \dots = \mathbf{X}_{C^*_K} = \mathbf{X}^*$. \square*

The above result provides a clearer explanation of the optimization problem based on the objective function Ψ''_{of} . Some issues that may arise with Ψ''_{of} are discussed in the following example.

Example 3 The ensemble illustrated in Fig. 3-(a) contains three projective clustering solutions whose clusters are very different from each other (different color shades), but are similar to some clusters from other solutions in the ensemble. In particular, C'_1, C''_1, C'''_1 are exactly the same (red shade), and the other groups of similar clusters are C'_2, C''_2, C'''_2 (green shade), and C'_3, C''_3, C'''_3 (blue shade). According to Proposition 3, the optimal solution for the

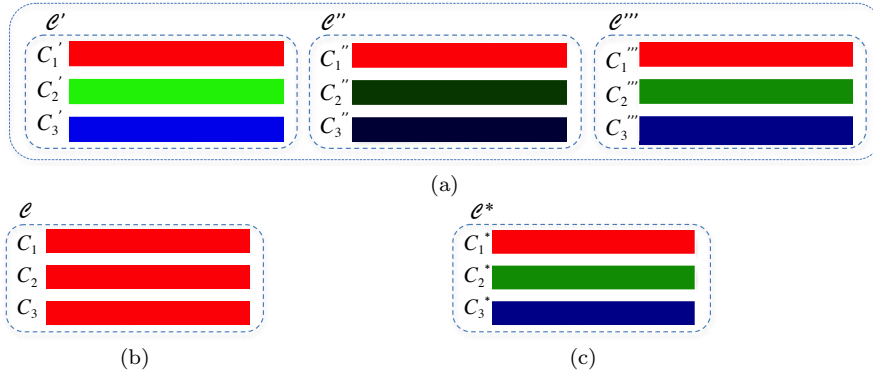


Fig. 3 (a) a projective ensemble where, for each projective cluster, different projective matrix representations are depicted with different color shades, (b) optimal projective consensus clustering according to the Ψ''_{of} objective function, (c) expected projective consensus clustering. (The color version of this figure is available only in the electronic edition)

optimization problem based on Ψ''_{of} with $K = 3$ is the one depicted in Fig. 3-(b), as the clusters in the *red*-shaded group are more similar to each other than the *green*-shaded and *blue*-shaded groups. This solution is conceptually far away from the ideal one reported in Fig. 3-(c), which is expected to be composed by three different clusters, each one summarizing a group of clusters with different shades. \square

3.2 Enhanced metacluster-based PCE formulation

We have discussed above the limitations due to the treatment of the functions Ψ'_{of} and Ψ''_{of} as separate [19]. Combining Ψ'_{of} and Ψ''_{of} into a single function somehow mitigates their respective undesired effects. Nevertheless, it is hard to understand to which extent each one of the two objective functions really contributes to hide the weaknesses of the other. Hence, it is more appropriate to have a problem formulation based on a single objective function which does not suffer from either of the drawbacks illustrated in Examples 2 and 3. We achieve this goal by proposing a modified metacluster-based PCE formulation, whose details are discussed next.

Let us first consider again the issue due to the function Ψ'_{of} described in Example 2. We recall that a major problem in this case is that the summarization provided by the clusters in the optimal consensus clustering might put in relation clusters from the same solutions in the ensemble, rather than coupling clusters from different ensemble components. It can be noted that this arises because the mappings of the various clusters in the ensemble to the center matrices are unconstrained (cf. the formal definition of the problem in (15)–(16)). The issue can be overcome by defining such mappings so to guarantee that each center is associated with (at least) one cluster from each different solution in the ensemble. These constraints actually correspond

to those defined in (18), which is indeed the optimization problem based on function Ψ''_{of} .

Focusing now on the function Ψ''_{of} , a possible solution to the issue described in Example 3 is to constrain the clusters forming the output projective consensus clustering to be different from each other. Alternatively, one can observe that the issue of Example 3 is mainly due to the fact that not all clusters in the ensemble are required to be mapped to a cluster of the projective consensus clustering. Based on this consideration, we can fix such an issue by resorting to some constraints of the problem based on the other function Ψ'_{of} , particularly the constraints listed in (16). Thus, in conclusion, the constraints reported in (16) and (18) represent a solution to the issues pertaining the functions Ψ'_{of} and Ψ''_{of} , respectively. As such, we define our enhanced metacluster-based PCE formulation by involving both these constraints. We hereinafter refer to this problem as **CB-PCE ENHANCED**.

Problem 1 (CB-PCE enhanced) Given a projective ensemble \mathcal{E} defined over a set \mathcal{D} of objects and a set \mathcal{F} of features, and an integer $K > 0$, find a projective clustering solution \mathcal{C}^* such that $|\mathcal{C}^*| = K$ and:

$$\mathcal{C}^* = \arg \min_{\mathcal{C}} \sum_{C \in \mathcal{C}} \sum_{\hat{C} \in \mathcal{E}} \sum_{\tilde{C} \in \hat{\mathcal{C}}} x(\hat{C}, C) T(\hat{C}, C) \quad (19)$$

s.t.

\mathcal{C} satisfies (2),

$$x(\hat{C}, C) \in \{0, 1\}, \quad \forall \hat{C} \in \mathcal{E}, \forall \tilde{C} \in \hat{\mathcal{C}}, \forall C \in \mathcal{C},$$

$$\sum_{C \in \mathcal{C}} x(\hat{C}, C) \geq 1, \quad \forall \hat{C} \in \mathcal{E}, \forall \tilde{C} \in \hat{\mathcal{C}}, \quad (20)$$

$$\sum_{\tilde{C} \in \hat{\mathcal{C}}} x(\hat{C}, C) \geq 1, \quad \forall \hat{C} \in \mathcal{E}, \forall C \in \mathcal{C}. \quad (21)$$

□

Note that the constraints (20) and (21) have been slightly modified w.r.t. the original ones in (16) and (18), in order to handle the cases where the number of clusters of some ensemble solutions is smaller or larger than the number of clusters in the output projective consensus clustering. In summary, the constraints in (20) force each cluster \hat{C} of the ensemble to be mapped to at least one cluster in the candidate projective consensus clustering \mathcal{C} , while the constraints in (21) ensure that each cluster $C \in \mathcal{C}$ is coupled with at least one cluster from each ensemble solution $\hat{\mathcal{C}}$.

4 Heuristics for CB-PCE enhanced

The **CB-PCE ENHANCED** problem can formally be shown to be **NP-hard**. The detailed proof, which is reported in the appendix, is based on a reduction from the **JACCARD MEDIAN** problem [11]. This reduction shows that **CB-PCE ENHANCED** remains hard even on a restricted version of the problem

where the input dataset is a singleton and the number of clusters in both the output projective consensus clustering and each ensemble solution is 1.

Lemma 1 *Let CB-PCE RESTRICTED be a special version of the CB-PCE ENHANCED problem where (i) $|\mathcal{D}| = 1$, (ii) $K = 1$ (K denotes the number of clusters in the output projective consensus clustering), (iii) $|\hat{\mathcal{C}}| = 1, \forall \hat{\mathcal{C}} \in \mathcal{E}$, (iv) $n(\Delta_{\hat{\mathcal{C}}}) = n(\Delta_{\hat{\mathcal{C}}'}) = n_{\mathcal{E}}, \forall \hat{\mathcal{C}}, \hat{\mathcal{C}}'$, where $n(\Delta_{\hat{\mathcal{C}}}) = \sum_{f \in \mathcal{F}} \mathbb{I}[\Delta_{\hat{\mathcal{C}},f} > 0]$, and (v) $\Delta_{\hat{\mathcal{C}},f} = \frac{1}{n_{\mathcal{E}}}, \forall f \in \mathcal{F}, \forall \hat{\mathcal{C}} \in \hat{\mathcal{C}}, \forall \hat{\mathcal{C}} \in \mathcal{E}$. CB-PCE RESTRICTED is **NP-hard**. \square*

Theorem 1 *CB-PCE ENHANCED is **NP-hard**. \square*

The above results prompted us to develop heuristics for approximating CB-PCE ENHANCED. In this respect, consider the formulation of CB-PCE ENHANCED reported in (19)–(21) and suppose that, for any input ensemble \mathcal{E} , the optimal mappings between the clusters \hat{C} within \mathcal{E} and the clusters C of the output projective consensus partition are available, i.e., suppose we know in advance the optimal $x(\hat{C}, C)$ values, $\forall \hat{C} \in \mathcal{E}, \forall \hat{C} \in \hat{\mathcal{C}}, \forall C \in \mathcal{C}$. Within this view, a metacluster might be derived for each $C \in \mathcal{C}$ that contains all clusters \hat{C} in the ensemble that are mapped to C (i.e., such that $x(\hat{C}, C) = 1$). In this way, the optimum for CB-PCE ENHANCED would be found by computing, for each metacluster, the projective matrix that minimizes the sum of the distances from all members of that metacluster under the constraints in (2). Unfortunately, neither the optimal $x(\hat{C}, C)$ mappings can be known in advance as they are part of the optimization process, nor the computation of the optimal projective matrices given an optimal mapping is feasible, as it is easy to observe from Lemma 1 that even this subproblem is **NP-hard**.

Nevertheless, the above reasoning interestingly reveals that CB-PCE ENHANCED can conceptually be split into two different sub-problems: the first one concerning the mapping of the clusters within the ensemble to the clusters in the output projective consensus clustering, and the second one consisting in finding optimal projective matrices given that mapping. To tackle these sub-problems, we improve upon the CB-PCE heuristic originally introduced in [19], as it still complies with the two-step nature of CB-PCE ENHANCED.

4.1 The CB-PCE heuristic

Algorithm 1 sketches the CB-PCE heuristic defined in [19]. For clarity of presentation, the following symbols are used in addition to the notation provided in Section 2: \mathbf{M} denotes a set of metaclusters (i.e., a set of sets of clusters), $\mathcal{M} \in \mathbf{M}$ denotes a metacluster (i.e., a set of clusters), and $M \in \mathcal{M}$ denotes a cluster (i.e., a set of data objects). The mapping of clusters in the ensemble to clusters in the output projective consensus clustering is approximated in CB-PCE by exploiting a basic idea in traditional metacluster-based clustering ensembles, namely grouping all clusters in the ensemble in order to form a set

Algorithm 1 CB-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:  $\mathbf{P} \leftarrow \text{pairwiseClusterDistances}(\mathcal{E})$ 
2:  $\mathbf{M} \leftarrow \text{metaclusters}(\mathcal{E}, \mathbf{P}, K)$ 
3:  $\mathcal{C}^* \leftarrow \emptyset$ 
4: for all  $\mathcal{M} \in \mathbf{M}$  do
5:   compute  $\mathbf{\Gamma}_{\mathcal{M}}^*$  according to Theorem 2
6:   compute  $\mathbf{\Delta}_{\mathcal{M}}^*$  according to Theorem 3
7:    $\mathcal{C}^* \leftarrow \mathcal{C}^* \cup \{(\mathbf{\Gamma}_{\mathcal{M}}, \mathbf{\Delta}_{\mathcal{M}}^*)\}$ 
8: end for

```

of metaclusters (cf. Section 2). This is achieved by invoking the function *metaclusters* (Line 2), which aims to cluster the set of clusters from all solutions within the input ensemble \mathcal{E} . This function exploits the matrix \mathbf{P} of pairwise distances between all clusters in \mathcal{E} (Line 1), which are computed using the Tanimoto distance defined in (13).

The second conceptual step of the CB-PCE ENHANCED problem consists in finding a suitable projective matrix representation for each cluster in the output consensus clustering, given the various metaclusters (Lines 4–8). As shown above, the ideal solution concerning the direct optimization of the sum of the Tanimoto distances is **NP-hard**. Therefore the idea underlying CB-PCE is to derive projective matrices by focusing on the optimization of a different criterion which is easier to solve and well-suited to find reasonable and effective approximations. Specifically, the solution provided by CB-PCE is to adapt the widely used *majority voting* [38] to the context at hand.

Deriving projective matrices from metaclusters. For each metacluster we derive a projective matrix that optimizes the majority voting criterion. This is sufficient to obtain an approximation of the solution of the problem, since deriving the corresponding object- and feature-based cluster representations given a projective matrix can be performed easily by using Proposition 2. However, due to the use of the majority voting criterion, it is possible to derive object- and feature-based representation vectors directly, without requiring the computation of the projective matrix.

The object- and feature-based representations of each projective cluster to be included into the output consensus clustering \mathcal{C}^* are denoted as $\mathbf{\Gamma}_{\mathcal{M}}^*$ and $\mathbf{\Delta}_{\mathcal{M}}^*$, $\forall \mathcal{M} \in \mathbf{M}$, respectively. More precisely, $\mathbf{\Gamma}_{\mathcal{M}}^*$ (resp. $\mathbf{\Delta}_{\mathcal{M}}^*$) is the object-based (resp. feature-based) representation of the projective cluster within \mathcal{C}^* corresponding to the metacluster \mathcal{M} . Let us derive the values of $\mathbf{\Gamma}_{\mathcal{M}}^*$ first. Since the ensemble can in principle contain projective clusterings that are soft at the clustering level, the majority voting criterion leads to the definition of

the following optimization problem:

$$\{\mathbf{\Gamma}_{\mathcal{M}}^* | \mathcal{M} \in \mathbf{M}\} = \underset{\{\mathbf{\Gamma}_{\mathcal{M}} | \mathcal{M} \in \mathbf{M}\}}{\operatorname{argmin}} Q \quad (22)$$

s.t.

$$\sum_{\mathcal{M} \in \mathbf{M}} \Gamma_{\mathcal{M}, \mathbf{o}} = 1, \quad \forall \mathbf{o} \in \mathcal{D}, \quad (23)$$

$$\Gamma_{\mathcal{M}, \mathbf{o}} \geq 0, \quad \forall \mathcal{M} \in \mathbf{M}, \forall \mathbf{o} \in \mathcal{D}, \quad (24)$$

where

$$Q = \sum_{\mathcal{M} \in \mathbf{M}} \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{\mathcal{M}, \mathbf{o}}^\alpha A_{\mathcal{M}, \mathbf{o}}, \quad A_{\mathcal{M}, \mathbf{o}} = \frac{1}{|\mathcal{M}|} \sum_{M \in \mathcal{M}} (1 - \Gamma_{M, \mathbf{o}}),$$

and $\alpha > 1$ is an integer that guarantees the non-linearity of the objective function Q w.r.t. $\Gamma_{\mathcal{M}, \mathbf{o}}$, needed to ensure $\Gamma_{\mathcal{M}, \mathbf{o}}^* \in [0, 1]$ (rather than $\{0, 1\}$).³ The final solution for such a problem is stated in the next theorem. The details of its derivation are in the appendix.

Theorem 2 *The optimum of the problem defined in (22)-(24) is $(\forall \mathcal{M}, \forall \mathbf{o})$:*

$$\Gamma_{\mathcal{M}, \mathbf{o}}^* = \left[\sum_{\mathcal{M}' \in \mathbf{M}} \left(\frac{A_{\mathcal{M}, \mathbf{o}}}{A_{\mathcal{M}', \mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-1}.$$

□

A similar argument applies to $\Delta_{\mathcal{M}}^*$. In this case, the problem to be solved is as follows:

$$\{\Delta_{\mathcal{M}}^* | \mathcal{M} \in \mathbf{M}\} = \underset{\{\Delta_{\mathcal{M}} | \mathcal{M} \in \mathbf{M}\}}{\operatorname{argmin}} \sum_{\mathcal{M} \in \mathbf{M}} \sum_{f \in \mathcal{F}} \Delta_{\mathcal{M}, f}^\beta B_{\mathcal{M}, f} \quad (25)$$

s.t.

$$\sum_{f \in \mathcal{F}} \Delta_{\mathcal{M}, f} = 1, \quad \forall \mathcal{M} \in \mathbf{M}, \quad (26)$$

$$\Delta_{\mathcal{M}, f} \geq 0, \quad \forall \mathcal{M} \in \mathbf{M}, \forall f \in \mathcal{F}, \quad (27)$$

where $B_{\mathcal{M}, f} = |\mathcal{M}|^{-1} \sum_{M \in \mathcal{M}} 1 - \Delta_{M, f}$ and β plays the same role as α in the function Q . The solution of such a problem is similar to that derived for $\Gamma_{\mathcal{M}, \mathbf{o}}^*$.

Theorem 3 *The optimum of the problem defined in (25)-(27) is $(\forall \mathcal{M}, \forall f)$:*

$$\Delta_{\mathcal{M}, f}^* = \left[\sum_{f' \in \mathcal{F}} \left(\frac{B_{\mathcal{M}, f}}{B_{\mathcal{M}, f'}} \right)^{\frac{1}{\beta-1}} \right]^{-1}.$$

□

Function 2 `metaclusters`(\mathcal{E} , \mathbf{P} , K)

Input: a projective ensemble \mathcal{E} ; a matrix \mathbf{P} storing the pairwise distances between clusters in $\bigcup_{\hat{\mathcal{C}} \in \mathcal{E}} \hat{\mathcal{C}}$; the number K of output metaclusters.

Output: a set \mathbf{M} of K metaclusters.

```

1: initialize  $\mathbf{M} = \{\mathcal{M}_1, \dots, \mathcal{M}_K\}$  so to satisfy the constraints in (20)–(21)
2: compute initial  $V$  from  $\mathbf{P}$  as  $V \leftarrow \sum_{\mathcal{M} \in \mathbf{M}} \sum_{M_1, M_2 \in \mathcal{M}} T(M_1, M_2)$ 
3:  $V' \leftarrow V$ 
4: repeat
5:    $V \leftarrow V'$ 
6:    $M_{move} \leftarrow \mathbf{nil}$ ,  $M_{swap} \leftarrow \mathbf{nil}$ ,  $\mathcal{M}_{old} \leftarrow \mathbf{nil}$ ,  $\mathcal{M}_{new} \leftarrow \mathbf{nil}$ 
7:   for all  $\mathcal{M}_s \in \mathbf{M}$ ,  $M \in \mathcal{M}_s$  do
8:     for all  $\mathcal{M}_t \in \mathbf{M} \setminus \{\mathcal{M}_s\}$  do
9:        $\langle \hat{V}, \hat{M} \rangle \leftarrow \text{evaluateMove}(M, \mathcal{M}_s, \mathcal{M}_t, \mathbf{P})$ 
10:      if  $V + \hat{V} < V'$  then
11:         $V' \leftarrow V + \hat{V}$ 
12:         $M_{move} \leftarrow M$ ,  $M_{swap} \leftarrow \hat{M}$ ,  $\mathcal{M}_{old} \leftarrow \mathcal{M}_s$ ,  $\mathcal{M}_{new} \leftarrow \mathcal{M}_t$ 
13:      end if
14:    end for
15:  end for
16:  if  $V' < V$  then
17:    move  $M_{move}$  from  $\mathcal{M}_{old}$  to  $\mathcal{M}_{new}$ 
18:    if  $M_{swap} \neq \mathbf{nil}$ , move  $M_{swap}$  from  $\mathcal{M}_{new}$  to  $\mathcal{M}_{old}$ 
19:  end if
20: until no  $M$  changes metacluster
21: return  $\mathbf{M}$ 

```

4.2 The E-CB-PCE heuristic

Although CB-PCE provides an approximation to CB-PCE ENHANCED that reasonably exploits its intrinsic two-step nature, it does not take into account the new findings of this work. In particular, a major issue of CB-PCE is due to the approximation of the optimal mappings between clusters in the ensembles and clusters in the projective consensus clustering: it satisfies only the constraints listed in (20), and it ignores the constraints in (21). Indeed, the clustering of all the clusters in the ensemble in CB-PCE is carried out by using a standard clustering algorithm, and this is clearly not sufficient to enforce that each output metacluster must contain at least one cluster from each different projective clustering solution of the ensemble, which is what the constraints in (21) require. For this purpose, we describe next a modified version of CB-PCE, called *Enhanced CB-PCE (E-CB-PCE)*. E-CB-PCE follows the overall scheme of CB-PCE reported in Algorithm 1. The only difference is that it incorporates a well-suited (local-search) procedure to be used as the *metaclusters* subroutine in Line 2, whose main goal is to produce metaclusters satisfying the constraints in (21) along with those in (20).

The outline of the proposed method is given in Function 2. The set of metaclusters is initialized in such a way that the constraints in (20)–(21) are satisfied (Line 1). A score V for the set of metaclusters is computed, corre-

³ An alternative way of obtaining $\Gamma_{\mathcal{M}, \mathbf{o}}^* \in [0, 1]$ is to employ regularization terms [27].

Function 3 `evaluateMove`($M, \mathcal{M}_s, \mathcal{M}_t, \mathbf{P}$)

Input: a projective cluster M ; a source metacluster \mathcal{M}_s ; a target metacluster \mathcal{M}_t ; a matrix \mathbf{P} storing the pairwise distances between clusters in $\bigcup_{\hat{c} \in \mathcal{E}} \hat{\mathcal{C}}$.

Output: a pair $\langle \hat{V}, \hat{M} \rangle$, where \hat{V} is a real value and \hat{M} is a projective cluster.

```

1: for any projective cluster  $M$  and metacluster  $\mathcal{M}$ , let  $g(M, \mathcal{M})$  denote  $\sum_{M' \in \mathcal{M}} T(M, M')$ 
2:  $\hat{M} \leftarrow \mathbf{nil}$ 
3: let  $\mathcal{C}_M$  the projective clustering in  $\mathcal{E}$  s.t.  $M \in \mathcal{C}_M$ 
4: if  $\mathcal{M}_s \cap \mathcal{C}_M \setminus \{M\} \neq \emptyset$  then
5:    $\hat{V} \leftarrow g(M, \mathcal{M}_t) - g(M, \mathcal{M}_s)$ 
6: else
7:    $\hat{M} \leftarrow \arg \min_{M' \in \mathcal{M}_t \cap \mathcal{C}_M} [(g(M, \mathcal{M}_t \setminus \{M'\}) - g(M, \mathcal{M}_s)) + (g(M', \mathcal{M}_s \setminus \{M\}) - g(M', \mathcal{M}_t))]$ 
8:    $\hat{V} \leftarrow [(g(M, \mathcal{M}_t \setminus \{\hat{M}\}) - g(M, \mathcal{M}_s)) + (g(\hat{M}, \mathcal{M}_s \setminus \{M\}) - g(\hat{M}, \mathcal{M}_t))]$ 
9: end if
10: return  $\langle \hat{V}, \hat{M} \rangle$ 

```

sponding to the sum of all pairwise Tanimoto distances between the clusters in the same metacluster (Line 2). The procedure performs an iterative step aimed at improving the score V (Lines 3-20). In particular, we employ a local search that moves a projective cluster M from its source metacluster \mathcal{M}_s to a target metacluster $\mathcal{M}_t \neq \mathcal{M}_s$ (Lines 7-15). The move that causes the largest decrease in the score V is performed (Lines 16-19). The method terminates when no valid move that improves the current score V is available, i.e., a local minimum of V has been reached.

The evaluation of the move of a cluster M from its source metacluster \mathcal{M}_s to a target metacluster $\mathcal{M}_t \neq \mathcal{M}_s$ is performed by the method *evaluateMove* presented in Function 3. This method returns the relative score \hat{V} of the move along with a “swap” projective cluster \hat{M} within \mathcal{M}_t that aims to replace M in \mathcal{M}_s . If no swapping is required, no swap cluster is returned (i.e., $\hat{M} = \mathbf{nil}$). The move is evaluated by distinguishing between two cases. Let us denote with \mathcal{C}_M the projective clustering solution in the input ensemble which contains M . If the source metacluster \mathcal{M}_s contains one or more clusters of \mathcal{C}_M besides M , removing M from \mathcal{M}_s does not violate the constraints in (21). Therefore, in this case, the score \hat{V} is computed as the sum of the Tanimoto distances of M from all the clusters in the new metacluster \mathcal{M}_t minus the distances between M and all the members of the old metacluster \mathcal{M}_s (Lines 4-5). Otherwise, if \mathcal{M}_s does not contain any other cluster of \mathcal{C}_M , the move evaluation also takes into account that a “swap” is needed to ensure that the constraints in (21) are satisfied (Lines 7-8). More precisely, M needs to be replaced with another cluster from \mathcal{C}_M that currently belongs to the target metacluster \mathcal{M}_t . Such a cluster from \mathcal{M}_t is chosen in such a way that the resulting score \hat{V} is minimized.

Table 1 Computational complexities of PCE methods

<i>method</i>	<i>total</i>	<i>online</i>	<i>offline</i>
<i>MOEA-PCE</i> [17, 20]	$\mathcal{O}(ItK^2 \mathcal{E} (\mathcal{D} + \mathcal{F}))$	$\mathcal{O}(ItK^2 \mathcal{E} (\mathcal{D} + \mathcal{F}))$	—
<i>EM-PCE</i> [17, 20]	$\mathcal{O}(K \mathcal{E} \mathcal{D} \mathcal{F})$	$\mathcal{O}(IK \mathcal{D} \mathcal{F})$	$\mathcal{O}(K \mathcal{E} \mathcal{D} \mathcal{F})$
<i>E-EM-PCE</i> [18]	$\mathcal{O}(K \mathcal{E} \mathcal{D} (IK + \mathcal{F}))$	$\mathcal{O}(IK \mathcal{D} (K \mathcal{E} + \mathcal{F}))$	$\mathcal{O}(K \mathcal{E} \mathcal{D} \mathcal{F})$
<i>E-2S-PCE</i> [18]	$\mathcal{O}(K \mathcal{E} \mathcal{D} (\mathcal{D} + \mathcal{F}))$	$\mathcal{O}(K \mathcal{D} (\mathcal{D} + \mathcal{E} \mathcal{F}))$	$\mathcal{O}(K \mathcal{E} \mathcal{D} (\mathcal{D} + \mathcal{F}))$
<i>CB-PCE</i> [19]	$\mathcal{O}(K^2 \mathcal{E} ^2 \mathcal{D} \mathcal{F})$	$\mathcal{O}(K \mathcal{E} (K \mathcal{E} + \mathcal{D} + \mathcal{F}))$	$\mathcal{O}(K^2 \mathcal{E} ^2 \mathcal{D} \mathcal{F})$
<i>FCB-PCE</i> [19]	$\mathcal{O}(K^2 \mathcal{E} ^2(\mathcal{D} + \mathcal{F}))$	$\mathcal{O}(K \mathcal{E} (K \mathcal{E} + \mathcal{D} + \mathcal{F}))$	$\mathcal{O}(K^2 \mathcal{E} ^2(\mathcal{D} + \mathcal{F}))$
<i>E-CB-PCE</i>	$\mathcal{O}(K^2 \mathcal{E} ^2(I + \mathcal{D} + \mathcal{F}))$	$\mathcal{O}(K \mathcal{E} (IK \mathcal{E} + \mathcal{D} + \mathcal{F}))$	$\mathcal{O}(K^2 \mathcal{E} ^2(\mathcal{D} + \mathcal{F}))$

4.3 Computational analysis

The bottleneck of the early CB-PCE heuristic is the computation of the pairwise Tanimoto distances between the clusters in the ensemble (Line 1 in Algorithm 1). For any single pair of clusters, the Tanimoto distance in principle takes $\mathcal{O}(|\mathcal{D}||\mathcal{F}|)$ time, as it requires to consider all the elements of the $|\mathcal{D}| \times |\mathcal{F}|$ projective matrices of those clusters. Nevertheless, we show in Proposition 4 that the Tanimoto distance formula can be rewritten in such a way that its computation takes instead $\mathcal{O}(|\mathcal{D}| + |\mathcal{F}|)$ time.

Proposition 4 *For any two projective clusters C, C' it holds that:*

$$T(\mathbf{X}_C, \mathbf{X}_{C'}) = 1 - \frac{p(C, C')}{q(C) + q(C') - p(C, C')}$$

where

$$p(C, C') = (\mathbf{\Gamma}_C \mathbf{\Gamma}_{C'}^T) \times (\mathbf{\Delta}_C \mathbf{\Delta}_{C'}^T), \quad q(C) = (\|\mathbf{\Gamma}_C\|_2^2) \times (\|\mathbf{\Delta}_C\|_2^2).$$

□

Thanks to the above result, the proposed E-CB-PCE improves upon the efficiency of the early CB-PCE. In particular, E-CB-PCE can now achieve a better time complexity for the Tanimoto distance of a single pair of clusters without introducing any approximation, which was the solution exploited in our previous work to obtain a speed-up (cf. FCB-PCE heuristic in [19]).

Time complexity of E-CB-PCE. Next we discuss in detail the computational complexity of the proposed E-CB-PCE. 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} and \mathcal{F} , and a positive integer K representing the number of clusters in the output projective consensus clustering. It is also reasonable to assume that the size $|\mathcal{C}|$ of each solution \mathcal{C} in \mathcal{E} is $\mathcal{O}(K)$. The complexity of E-CB-PCE can be broken down into three stages:

1. *Pre-processing*: it concerns the computation of the pairwise distances between clusters, by applying the Tanimoto distance $T(\cdot, \cdot)$ to projective matrices. According to Proposition 4, any single distance computation can be performed in $\mathcal{O}(|\mathcal{D}| + |\mathcal{F}|)$ time. Thus, computing the overall pairwise matrix takes $\mathcal{O}(K^2 |\mathcal{E}|^2 (|\mathcal{D}| + |\mathcal{F}|))$ time;

2. *Meta-clustering*: it concerns the clustering of the $\mathcal{O}(K|\mathcal{E}|)$ clusters of all the solutions in the ensemble according to the procedure described in Function 2. This procedure is based on a local-search optimization strategy of an objective function that is quadratic in the number of all clusters within the input ensemble. Thus, its time complexity is $\mathcal{O}(I K^2 |\mathcal{E}|^2)$, where I is the number of iterations to convergence.
3. *Post-processing*: it concerns the assignment of objects and features to the metaclusters based on Theorems 2 and 3. According to those theorems, both the object and the feature assignments need to look up all the clusters in each metacluster only once; thus, for each object and for each feature, it takes $\mathcal{O}(K|\mathcal{E}|)$ time. Performing this step for all objects and features leads to a total cost of $\mathcal{O}(K|\mathcal{E}|(|\mathcal{D}| + |\mathcal{F}|))$ for the entire post-processing step.

It can be noted that the first step is an *offline* phase, i.e., a phase to be performed only once in case of a multi-run execution, whereas the second and third are *online* steps. Thus, as summarized in Table 1 (where we also report the complexities of the earlier MOEA-PCE, EM-PCE [20], E-EM-PCE, E-2S-PCE [18], and CB-PCE, FCB-PCE [19] methods⁴), we can conclude that the *offline*, *online*, and *total* (i.e., offline + online) complexities of E-CB-PCE are $\mathcal{O}(K^2 |\mathcal{E}|^2 (|\mathcal{D}| + |\mathcal{F}|))$, $\mathcal{O}(K|\mathcal{E}|(IK|\mathcal{E}| + |\mathcal{D}| + |\mathcal{F}|))$, and $\mathcal{O}(K^2 |\mathcal{E}|^2 (I + |\mathcal{D}| + |\mathcal{F}|))$, respectively.

5 Experimental Evaluation

We evaluated accuracy and efficiency of the proposed E-CB-PCE algorithm and compared it with the early PCE methods, i.e., MOEA-PCE, EM-PCE [17, 20], E-EM-PCE, E-2S-PCE [18], and CB-PCE [19]. In the following, we introduce our evaluation methodology which includes the selected datasets, the strategy used for generating the projective ensembles, the setup of the proposed algorithms, and the assessment criteria for the projective consensus clusterings. Finally, we discuss obtained experimental results.

5.1 Evaluation methodology

5.1.1 Datasets

We selected 22 publicly available datasets with different characteristics in terms of number of objects, features, and classes. In the summary provided in Table 2, the first fifteen datasets are from the UCI Machine Learning Repository [5], the next four datasets are from the UCR Time Series Classification/Clustering Page [23], whereas the last three datasets are synthetically generated and selected from [31]. Note that the synthetic datasets originally had overlapping clusters. We selected for each dataset the maximal subset of

⁴ In Table 1, t denotes the population size for the genetic algorithm at the basis of MOEA-PCE.

Table 2 Datasets used in the experiments

<i>dataset</i>	<i>objects</i>	<i>features</i>	<i>classes</i>
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

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 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, whose effectiveness in the context of projective clustering has been already proven [13]. The diversity of the projective clustering solutions was ensured by randomly choosing the initial centroids and varying the LAC’s parameter h .

Note that 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 algorithm 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 contains solutions that have hard object-to-cluster assignments and weighted feature-to-cluster assignments, i.e., solutions as provided by standard LAC;
- \mathcal{E}_2 contains 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 $\Delta_{\mathcal{C}',f} = \mathbf{I} \left[\Delta_{\mathcal{C}',f} \geq |\mathcal{F}|^{-1} \sum_{f' \in \mathcal{F}} \Delta_{\mathcal{C}',f'} \right]$, $\forall \mathcal{C}' \in \mathcal{C}'$, $\forall f \in \mathcal{F}$, where $\mathbf{I}[A]$ is the *indicator function*, which is equal to 1 when the event A is true, and 0 otherwise;

- \mathcal{E}_3 contains solutions that have soft object-to-cluster assignments and weighted feature-to-cluster assignments. Starting from a LAC solution \mathcal{C} , a soft projective clustering \mathcal{C}'' is derived by computing the $\Gamma_{\mathcal{C}'',\mathbf{o}}$ values ($\forall \mathcal{C}'' \in \mathcal{C}'', \forall \mathbf{o} \in \mathcal{D}$), proportionally to the distance of \mathbf{o} from the centroids $\bar{\mathcal{C}}''$ of the clusters \mathcal{C}'' :

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

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

- \mathcal{E}_4 contains 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 correspond to averages over these projective ensembles.

5.1.3 Assessment criteria

We assessed the quality of a projective consensus clustering \mathcal{C} using both external and internal cluster validity criteria: 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, hereinafter denoted as $\tilde{\mathcal{C}}$, for any given dataset \mathcal{D} . Note that all the selected datasets are coupled with a reference classification that provides information about the ideal object-to-cluster assignments $\Gamma_{\tilde{\mathcal{C}},\mathbf{o}}$ ($\forall \tilde{\mathcal{C}} \in \tilde{\mathcal{C}}, \forall \mathbf{o} \in \mathcal{D}$), which are hard assignments. The $\Delta_{\tilde{\mathcal{C}},f}$ 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 ideal subspaces assigned to each group of objects identified by the reference classification, these subspaces are directly used to define unweighted $\Delta_{\tilde{\mathcal{C}},f}$ feature-to-cluster assignments in $\tilde{\mathcal{C}}$.
- For the remaining datasets, the $\Delta_{\tilde{\mathcal{C}},f}$ values are derived by applying the procedure suggested in [13] to the reference classification $\tilde{\mathcal{C}}$: given the $\Gamma_{\tilde{\mathcal{C}},\mathbf{o}}$

values ($\forall \tilde{C} \in \tilde{\mathcal{C}}, \forall \mathbf{o} \in \mathcal{D}$) originally provided along with $\tilde{\mathcal{C}}$, the $\Delta_{\tilde{C},f}$ values are computed as:

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

where the LAC parameter h is set to 0.2 and:

$$U(\tilde{C}, f) = \left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{\tilde{C},\mathbf{o}} \right)^{-1} \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{\tilde{C},\mathbf{o}} (\bar{C}_f - o_f)^2, \quad \bar{C}_f = \left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{\tilde{C},\mathbf{o}} \right)^{-1} \sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{\tilde{C},\mathbf{o}} \times o_f.$$

In order to compute the similarity between a projective consensus clustering \mathcal{C} and a reference classification $\tilde{\mathcal{C}}$, we resort to the popular *F1-measure* [36]. Particularly, 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:

$$P(C) = \frac{\max_{\tilde{C} \in \tilde{\mathcal{C}}} \text{overlap}(\tilde{C}, C)}{\text{size}(C)}, \quad R(C) = \frac{\max_{\tilde{C} \in \tilde{\mathcal{C}}} \text{overlap}(\tilde{C}, C)}{\text{size}(\arg \max_{\tilde{C} \in \tilde{\mathcal{C}}} \text{overlap}(\tilde{C}, C))},$$

and the *F1-measure* is defined as:

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

The values of the *F1-measure* belong to the interval $[0, 1]$, where larger values indicate more accurate projective consensus clusterings. The $\text{overlap}(\cdot, \cdot)$ and $\text{size}(\cdot)$ functions quantify the degree of overlap between two projective clusters and the size of a projective cluster, respectively. Such functions are defined based on three ways of comparing the various projective clusters, namely *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 hence define variants of the $\text{overlap}(\cdot, \cdot)$ and the $\text{size}(\cdot)$ functions to handle each of the three cases:

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

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 a projective consensus clustering and the solutions within \mathcal{E} . We define the object & feature-based measure $\overline{F1}_{of}$ (object-based $\overline{F1}_o$ and feature-based $\overline{F1}_f$ are defined similarly) as follows:

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

All these measures range within $[0, 1]$; moreover, 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} .

5.1.4 Parameter setting

To set the parameters α and β of the proposed E-CB-PCE, we performed a *leave-one-dataset-out* approach: for each dataset the performance of E-CB-PCE on the other datasets was assessed for different values of the parameter(s), and the value(s) 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 the settings were scarcely influenced by any specific dataset, which indicates that a relatively easy setup can be performed on new datasets for which a reference classification or other a-priori knowledge is not available. Particularly, the best-performance scores were mostly reached by setting both α and β to 2. For the competing methods, we set the parameters as suggested in their respective papers.

5.2 Results

Accuracy. Tables 3–5 give the results of the external evaluation w.r.t. the reference classification (assessment criteria $F1_{of}$, $F1_o$, and $F1_f$, respectively), and Tables 6–8 give the results of the internal evaluation w.r.t. the projective ensemble solutions (assessment criteria $\overline{F1}_{of}$, $\overline{F1}_o$, and $\overline{F1}_f$, respectively). All algorithms involved in the comparison are nondeterministic, thus all tables contain average results over 50 different runs along with the corresponding standard deviations (in parentheses). Moreover, to improve the readability of the results, for each competitor and assessment criterion, we summarize the average gain of E-CB-PCE w.r.t. the competing method in Table 9.

As observed in the summary reported in Table 9, E-CB-PCE achieved better average accuracy performance w.r.t. both MOEA-PCE and CB-PCE, thus showing that the most recent findings of this work incorporated into the proposed E-CB-PCE actually give the expected outcome. E-CB-PCE was more

Table 3 Evaluation w.r.t. the reference classification ($F1_{of}$)

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	<i>E-CB-PCE</i>
Iris	.649(.025)	.588(.002)	.532(.046)	.543(.074)	.640(.071)	.673(.017)
Wine	.345(.025)	.300(.003)	.303(.001)	.266(.047)	.327(.028)	.356(.001)
Glass	.279(.009)	.298(.007)	.307(.009)	.253(.035)	.311(.042)	.420(.032)
Ecoli	.518(.020)	.564(.013)	.573(.007)	.540(.104)	.599(.063)	.550(.035)
Yeast	.288(.009)	.237(.004)	.232(.005)	.202(.025)	.317(.025)	.308(.008)
Mult.-Feat.	.270(.019)	.300(.015)	.290(.048)	.405(.041)	.174(.021)	.180(.001)
Segmentation	.334(.018)	.400(.008)	.401(.001)	.382(.026)	.364(.054)	.452(.029)
Abalone	.116(.003)	.112(.003)	.111(.004)	.080(.006)	.116(.004)	.122(.002)
Waveform	.339(.056)	.338(.006)	.267(.058)	.296(.012)	.263(.026)	.253(.012)
Letter	.181(.025)	.155(.007)	.155(.006)	.141(.021)	.120(.023)	.185(.012)
Isolet	.141(.004)	.138(.001)	.138(.001)	.106(.025)	.140(.017)	.148(.001)
Gisette	.595(.015)	.532(.006)	.515(.022)	.541(.002)	.499(.004)	.545(.001)
p53-Mutants	.464(.021)	.411(.020)	.411(.002)	.397(.001)	.412(.011)	.418(.001)
Amazon	.441(.019)	.388(.006)	.388(.019)	.446(.067)	.437(.011)	.442(.022)
Arcene	.367(.012)	.142(.002)	.142(.001)	.153(.007)	.148(.016)	.165(.002)
Shapes	.243(.009)	.294(.007)	.297(.005)	.208(.031)	.281(.025)	.314(.006)
Tracedata	.438(.010)	.432(.012)	.432(.001)	.437(.019)	.491(.031)	.470(.014)
ControlChart	.092(.013)	.203(.020)	.195(.008)	.250(.021)	.063(.008)	.122(.007)
Twopat	.144(.025)	.070(.002)	.070(.001)	.071(.005)	.070(.007)	.073(.002)
N30	.098(.005)	.108(.003)	.107(.005)	.091(.013)	.110(.015)	.124(.003)
D75	.033(.002)	.038(.001)	.038(.001)	.032(.003)	.038(.005)	.039(.002)
S2500	.116(.004)	.122(.005)	.121(.007)	.098(.010)	.131(.009)	.127(.004)
<i>min</i>	.033	.038	.038	.032	.038	.039
<i>max</i>	.649	.588	.573	.543	.640	.673
<i>avg</i>	.295	.280	.274	.270	.275	.295

accurate than both MOEA-PCE and CB-PCE on 16 out of 22 datasets on average, while achieving average gains up to 0.147 ($\overline{F1}_{of}$ assessment criterion) and 0.078 ($F1_f$ assessment criterion) w.r.t. MOEA-PCE and CB-PCE, respectively.

Larger improvements were produced by E-CB-PCE w.r.t. the early single-objective PCE methods, i.e., EM-PCE, E-EM-PCE, and E-2S-PCE. The average gains achieved by E-CB-PCE w.r.t. EM-PCE, E-EM-PCE, and E-2S-PCE reported in Table 9 were in general larger than those observed w.r.t. the remaining competing methods MOEA-PCE and CB-PCE.

Looking at the standard deviations reported in Tables 3–8, it can be observed that the proposed E-CB-PCE was quite insensitive to its random component. The standard deviations were in the order of 10^{-3} in most cases, while being in the order of 10^{-2} in the remaining cases.

Efficiency. Table 10 shows the runtimes (in milliseconds) of the various algorithms involved in the comparison. As expected, the proposed E-CB-PCE was slower than EM-PCE on most datasets, while clearly outperforming MOEA-PCE. The runtimes of E-CB-PCE were one or two orders of magnitude smaller than those of MOEA-PCE on average, up to four orders on *Isolet*. Only on one dataset, MOEA-PCE was more efficient than E-CB-PCE (*Glass*), even though the runtimes of the two methods remained of the same order of magnitude.

Compared to CB-PCE, the proposed E-CB-PCE was faster on 11 datasets, resulting in one order faster on 3 of them (i.e., *Multiple-Features*, *Waveform*, *Amazon*), and two orders faster on 4 datasets (i.e., *Isolet*, *Gisette*, *p53-Mutants*, *Arcene*). In general, we observed that the performance of E-CB-PCE mainly

Table 4 Evaluation w.r.t. the reference classification ($F1_o$)

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	<i>E-CB-PCE</i>
Iris	.967(.026)	.880(.002)	.784(.076)	.733(.158)	.948(.137)	1(.027)
Wine	.835(.037)	.731(.009)	.735(.001)	.639(.152)	.855(.082)	.957(.003)
Glass	.474(.053)	.509(.012)	.537(.021)	.433(.052)	.507(.068)	.571(.038)
Ecoli	.760(.014)	.667(.021)	.665(.021)	.641(.127)	.789(.063)	.761(.007)
Yeast	.417(.007)	.333(.007)	.328(.014)	.298(.030)	.424(.034)	.395(.014)
Mult.-Feat.	.319(.069)	.369(.020)	.363(.053)	.493(.063)	.326(.035)	.269(.007)
Segmentation	.443(.062)	.568(.014)	.567(.007)	.554(.029)	.523(.082)	.659(.047)
Abalone	.208(.006)	.169(.005)	.164(.006)	.152(.014)	.198(.014)	.210(.004)
Waveform	.515(.070)	.415(.001)	.387(.056)	.382(.018)	.437(.027)	.437(.009)
Letter	.331(.030)	.306(.005)	.308(.002)	.299(.037)	.307(.031)	.305(.017)
Isolet	.959(.031)	.978(.001)	.978(.001)	.722(.208)	.977(.080)	1(.002)
Gisette	.728(.015)	.674(.007)	.658(.030)	.581(.140)	.667(.003)	.669(.001)
p53-Mutants	.728(.038)	.619(.028)	.642(.031)	.596(.002)	.650(.054)	.733(.001)
Amazon	.555(.036)	.488(.009)	.480(.047)	.553(.071)	.525(.011)	.518(.029)
Arcene	.705(.009)	.626(.001)	.614(.001)	.618(.071)	.647(.021)	.644(.019)
Shapes	.681(.017)	.693(.015)	.704(.013)	.531(.052)	.716(.041)	.786(.017)
Tracedata	.614(.002)	.628(.059)	.628(.001)	.619(.032)	.629(.029)	.617(.010)
ControlChart	.319(.020)	.332(.003)	.342(.026)	.345(.040)	.311(.014)	.321(.019)
Twopat	.355(.011)	.296(.002)	.296(.002)	.302(.021)	.313(.007)	.316(.001)
N30	.807(.219)	.884(.013)	.881(.017)	.752(.117)	.915(.096)	.997(.028)
D75	.857(.146)	.952(.018)	.955(.022)	.717(.118)	.931(.103)	.981(.046)
S2500	.880(.156)	.895(.031)	.889(.056)	.747(.093)	.983(.070)	.941(.014)
<i>min</i>	.208	.169	.164	.152	.198	.210
<i>max</i>	.967	.978	.978	.752	.983	1
<i>avg</i>	.612	.591	.587	.532	.617	.640

depended on the number of iterations needed for Function 2 to converge. In particular, due to the nature of the local-search moves performed at each iteration, the runtime of Function 2 could negatively compensate the performance gain achieved by E-CB-PCE w.r.t. CB-PCE in the other steps of the heuristic. However, as observed in the measurements reported, this mostly affected the smaller datasets.

6 Conclusion

In our previous work [17] we introduced a framework in which projective clustering and clustering ensembles are addressed simultaneously. This resulted in the formulation of a new problem called projective clustering ensembles (PCE). Since the original formulation, research efforts have been made to design a single-objective function that keeps the object-based and the feature-based cluster representations joined together, and at the same time facilitates the adaptation of a conventional clustering ensemble approach to the PCE problem. In this paper, we presented the latest advance of PCE by proposing a metacluster-based formulation and related heuristics, which are theoretically and experimentally proven to best fit the PCE problem.

We expect that alternative approaches to the PCE problem will be developed in the next few years. One general direction we envision is moving the burden of the computation from the clustering side to a proper representation model. In this respect, we argue that an approach worth exploring is tensorial models and related tensor decomposition methods [24]. In a nutshell, a

Table 5 Evaluation w.r.t. the reference classification ($F1_f$)

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	E-CB-PCE
Iris	.974(.019)	.667(.001)	.667(.001)	.667(.001)	.829(.044)	.807(.001)
Wine	.643(.051)	.426(.006)	.432(.022)	.437(.017)	.472(.058)	.476(.091)
Glass	.804(.023)	.662(.029)	.671(.037)	.662(.028)	.748(.086)	.734(.032)
Ecoli	.906(.021)	.970(.013)	.983(.027)	1(.001)	.880(.035)	1(.047)
Yeast	.846(.011)	.774(.010)	.772(.018)	.726(.025)	.827(.026)	.956(.018)
Mult.-Feat.	.768(.013)	.795(.001)	.802(.027)	.887(.003)	.502(.034)	.567(.026)
Segmentation	.861(.026)	.747(.029)	.737(.047)	.706(.001)	.745(.115)	.971(.056)
Abalone	.822(.023)	.716(.006)	.713(.007)	.670(.001)	.750(.008)	.754(.006)
Waveform	.660(.067)	.792(.001)	.792(.001)	.792(.001)	.722(.043)	.700(.036)
Letter	.643(.037)	.595(.008)	.598(.009)	.580(.001)	.593(.031)	.686(.027)
Isollet	.171(.004)	.143(.001)	.144(.005)	.142(.005)	.149(.005)	.150(.005)
Gisette	.876(.016)	.797(.001)	.794(.005)	.797(.001)	.739(.041)	.818(.001)
p53-Mutants	.721(.009)	.671(.001)	.671(.001)	.673(.001)	.673(.015)	.692(.001)
Amazon	.890(.005)	.828(.001)	.828(.001)	.829(.001)	.840(.013)	.849(.003)
Arcene	.536(.014)	.264(.003)	.262(.001)	.260(.001)	.268(.013)	.483(.010)
Shapes	.428(.011)	.448(.002)	.449(.001)	.405(.016)	.451(.018)	.444(.005)
Tracedata	.787(.010)	.800(.006)	.811(.015)	.796(.029)	.795(.027)	.894(.034)
ControlChart	.322(.044)	.673(.002)	.673(.003)	.673(.003)	.263(.027)	.473(.035)
Twopat	.451(.079)	.233(.001)	.233(.001)	.233(.001)	.281(.039)	.339(.021)
N30	.131(.002)	.119(.001)	.120(.002)	.113(.004)	.119(.003)	.338(.005)
D75	.041(.001)	.039(.001)	.039(.001)	.036(.001)	.040(.003)	.045(.001)
S2500	.141(.002)	.124(.002)	.127(.003)	.125(.005)	.130(.003)	.343(.001)
<i>min</i>	.041	.039	.039	.036	.040	.045
<i>max</i>	.974	.970	.983	1	.880	1
<i>avg</i>	.610	.558	.560	.555	.537	.615

tensor model (e.g., a third-order tensor) can provide an integrated representation of the relevant dimensions in the input ensemble (i.e., the objects, the features, and the clusters); in addition, during the tensor decomposition, a consensus clustering, or even multiple consensus clusterings, can be induced in a straightforward manner. Interestingly, the ability of some existing tensor decomposition methods to generate an unfolding of the tensor in which the correlations among all dimensions (i.e., aspects in the ensemble) are preserved, might play an important role in establishing natural mappings between the clusters in the input ensemble and the clusters of the consensus clustering. Moreover, the common problem due to the unavailability of feature relevance values could be relaxed in a tensor modeling, thus enabling a new generation of clustering ensemble methods.

References

1. Achtert, E., Böhm, C., Kriegel, H., Kröger, P., Müller-Gorman, I., Zimek, A.: Finding Hierarchies of Subspace Clusters. In: Proc. European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 446–453 (2006)
2. Achtert, E., Böhm, C., Kriegel, H., Kröger, P., Müller-Gorman, I., Zimek, A.: Detection and Visualization of Subspace Cluster Hierarchies. In: Proc. Int. Conf. on Database Systems for Advanced Applications (DASFAA), pp. 152–163 (2007)
3. Aggarwal, C.C., Procopiuc, C.M., Wolf, J.L., Yu, P.S., Park, J.S.: Fast Algorithms for Projected Clustering. In: Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 61–72 (1999)
4. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: Proc. ACM SIGMOD Int. Conf. on Management Of Data, pp. 94–105 (1998)

Table 6 Evaluation w.r.t. the projective ensemble solutions ($\overline{F1}_{of}$)

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	E-CB-PCE
Iris	.858(.010)	.830(.002)	.768(.049)	.638(.153)	.886(.107)	.961(.004)
Wine	.603(.008)	.569(.002)	.571(.001)	.498(.085)	.716(.035)	.765(.001)
Glass	.430(.007)	.437(.009)	.413(.018)	.438(.035)	.557(.044)	.613(.020)
Ecoli	.630(.009)	.625(.004)	.623(.019)	.496(.052)	.819(.033)	.846(.009)
Yeast	.610(.008)	.586(.017)	.589(.024)	.525(.046)	.768(.017)	.789(.007)
Mult.-Feat.	.221(.008)	.233(.013)	.221(.033)	.180(.016)	.237(.014)	.200(.005)
Segmentation	.269(.009)	.308(.003)	.308(.003)	.309(.020)	.320(.026)	.326(.008)
Abalone	.675(.011)	.541(.011)	.551(.008)	.571(.034)	.741(.022)	.772(.013)
Waveform	.146(.004)	.110(.001)	.105(.005)	.119(.005)	.462(.003)	.470(.003)
Letter	.196(.005)	.146(.002)	.146(.002)	.108(.009)	.337(.011)	.356(.006)
Isollet	.751(.009)	.790(.001)	.790(.001)	.593(.155)	.876(.075)	.914(.001)
Gisette	.597(.003)	.505(.002)	.483(.022)	.481(.018)	.622(.003)	.669(.004)
p53-Mutants	.628(.005)	.632(.001)	.588(.060)	.634(.001)	.671(.014)	.627(.036)
Amazon	.468(.008)	.444(.015)	.435(.039)	.520(.023)	.539(.024)	.518(.022)
Arcene	.589(.009)	.421(.004)	.419(.001)	.397(.043)	.800(.027)	.836(.001)
Shapes	.444(.009)	.468(.009)	.475(.009)	.307(.045)	.713(.026)	.764(.002)
Tracedata	.723(.008)	.716(.001)	.716(.001)	.603(.079)	.830(.074)	.909(.008)
ControlChart	.186(.005)	.037(.002)	.037(.001)	.049(.001)	.334(.015)	.270(.006)
Twopat	.370(.012)	.144(.001)	.144(.001)	.154(.007)	.526(.045)	.501(.001)
N30	.323(.013)	.301(.003)	.302(.004)	.276(.025)	.462(.029)	.510(.014)
D75	.423(.014)	.429(.004)	.429(.004)	.362(.034)	.573(.039)	.596(.014)
S2500	.442(.013)	.449(.010)	.439(.020)	.345(.036)	.604(.015)	.605(.004)
<i>min</i>	.146	.037	.037	.049	.237	.200
<i>max</i>	.858	.830	.790	.638	.886	.961
<i>avg</i>	.481	.442	.434	.391	.609	.628

5. Asuncion, A., Newman, D.: UCI Machine Learning Repository. <http://archive.ics.uci.edu/ml/> (2010)
6. Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U.: When Is “Nearest Neighbor” Meaningful? In: Proc. Int. Conf. on Database Theory (ICDT), pp. 217–235 (1999)
7. Böhm, C., Kailing, K., Kriegel, H.P., Kröger, P.: Density Connected Clustering with Local Subspace Preferences. In: Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 27–34 (2004)
8. Boulis, C., Ostendorf, M.: Combining Multiple Clustering Systems. In: Proc. European Conf. on Principles and Practice of Knowledge Discovery in Databases (PKDD), pp. 63–74 (2004)
9. Bradley, P.S., Fayyad, U.M.: Refining Initial Points for K-Means Clustering. In: Proc. Int. Conf. on Machine Learning (ICML), pp. 91–99 (1998)
10. Chen, L., Jiang, Q., Wang, S.: A Probability Model for Projective Clustering on High Dimensional Data. In: Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 755–760 (2008)
11. Chierichetti, F., Kumar, R., Pandey, S., Vassilvitskii, S.: Finding the Jaccard Median. In: Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 293–311 (2010)
12. Domeniconi, C., Al-Razgan, M.: Weighted Cluster Ensembles: Methods and Analysis. ACM Trans. on Knowledge Discovery from Data (TKDD) **2**(4) (2009)
13. Domeniconi, C., Gunopoulos, D., Ma, S., Yan, B., Al-Razgan, M., Papadopoulos, D.: Locally Adaptive Metrics for Clustering High Dimensional Data. Data Mining and Knowledge Discovery **14**(1), 63–97 (2007)
14. Gan, G., Ma, C., Wu, J.: Data Clustering: Theory, Algorithms, and Applications. ASA-SIAM Series on Statistics and Applied Probability (2007)
15. Ghosh, J., Acharya, A.: Cluster ensembles. Wiley Interdisc. Rev.: Data Mining and Knowledge Discovery **1**(4), 305–315 (2011)
16. Gionis, A., Mannila, H., Tsaparas, P.: Clustering Aggregation. ACM Transactions on Knowledge Discovery from Data (TKDD) **1**(1) (2007)
17. Gullo, F., Domeniconi, C., Tagarelli, A.: Projective Clustering Ensembles. In: Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 794–799 (2009)

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

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	E-CB-PCE
Iris	.915(.007)	.842(.002)	.785(.044)	.684(.142)	.886(.093)	.954(.004)
Wine	.723(.010)	.662(.005)	.667(.003)	.575(.105)	.716(.067)	.766(.004)
Glass	.513(.008)	.511(.010)	.482(.021)	.505(.046)	.557(.055)	.600(.029)
Ecoli	.810(.013)	.804(.005)	.816(.034)	.697(.093)	.819(.029)	.852(.006)
Yeast	.754(.011)	.681(.017)	.682(.026)	.609(.059)	.768(.023)	.790(.008)
Mult.-Feat.	.216(.009)	.262(.015)	.249(.040)	.186(.020)	.237(.015)	.176(.008)
Segmentation	.313(.008)	.343(.005)	.345(.002)	.346(.019)	.320(.039)	.356(.015)
Abalone	.749(.015)	.592(.011)	.601(.009)	.628(.036)	.741(.022)	.771(.012)
Waveform	.489(.008)	.473(.001)	.444(.028)	.433(.041)	.462(.008)	.477(.004)
Letter	.324(.005)	.309(.004)	.308(.002)	.274(.025)	.337(.011)	.364(.005)
Isolet	.876(.010)	.875(.001)	.875(.001)	.669(.176)	.876(.064)	.908(.001)
Gisette	.693(.004)	.615(.003)	.595(.020)	.516(.133)	.622(.018)	.603(.031)
p53-Mutants	.720(.006)	.679(.001)	.611(.106)	.677(.001)	.671(.033)	.649(.045)
Amazon	.519(.008)	.473(.017)	.459(.049)	.551(.026)	.539(.033)	.514(.030)
Arcene	.840(.007)	.800(.001)	.799(.002)	.754(.142)	.800(.008)	.805(.002)
Shapes	.692(.010)	.701(.012)	.708(.009)	.547(.056)	.713(.028)	.762(.002)
Tracedata	.829(.008)	.830(.001)	.831(.001)	.732(.093)	.830(.063)	.890(.008)
ControlChart	.347(.004)	.307(.004)	.311(.011)	.342(.007)	.334(.012)	.367(.006)
Twopat	.528(.006)	.463(.002)	.463(.001)	.488(.030)	.526(.047)	.573(.001)
N30	.435(.013)	.469(.004)	.470(.006)	.430(.039)	.462(.042)	.501(.009)
D75	.551(.015)	.580(.005)	.579(.006)	.507(.043)	.573(.039)	.595(.016)
S2500	.562(.015)	.575(.011)	.567(.019)	.504(.043)	.604(.029)	.594(.006)
<i>min</i>	.216	.262	.249	.186	.237	.176
<i>max</i>	.915	.875	.875	.754	.886	.954
<i>avg</i>	.609	.584	.575	.530	.609	.630

18. Gullo, F., Domeniconi, C., Tagarelli, A.: Enhancing Single-Objective Projective Clustering Ensembles. In: Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 833–838 (2010)
19. Gullo, F., Domeniconi, C., Tagarelli, A.: Advancing data clustering via projective clustering ensembles. In: Proc. ACM SIGMOD Int. Conf. on Management of Data, pp. 733–744 (2011)
20. Gullo, F., Domeniconi, C., Tagarelli, A.: Projective Clustering Ensembles. Data Mining and Knowledge Discovery (DAMI) **26**(3), 452–511 (2013)
21. Hinneburg, A., Aggarwal, C.C., Keim, D.A.: What Is the Nearest Neighbor in High Dimensional Spaces? In: Proc. Int. Conf. on Very Large Data Bases (VLDB), pp. 506–515 (2000)
22. Jain, A., Dubes, R.: Algorithms for Clustering Data. Prentice-Hall (1988)
23. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The UCR Time Series Classification/Clustering Page. http://www.cs.ucr.edu/~eamonn/time_series_data/ (2003)
24. Kolda, T.G., Bader, B.W.: Tensor Decompositions and Applications. SIAM Review **51**, 455–500 (2009)
25. Kriegel, H., Kroger, P., Renz, M., Wurst, S.: A Generic Framework for Efficient Subspace Clustering of High-Dimensional Data. In: Proc. IEEE Int. Conf. on Data Mining (ICDM), pp. 250–257 (2005)
26. Kriegel, H., Kröger, P., Zimek, A.: 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)
27. Li, R.P., Mukaidono, M.: Gaussian clustering method based on maximum-fuzzy-entropy interpretation. Fuzzy Sets and Systems **102**(2), 253–258 (1999)
28. Liu, B., Xia, Y., Yu, P.S.: Clustering Through Decision Tree Construction. In: Proc. Int. Conf. on Information and Knowledge Management (CIKM), pp. 20–29 (2000)
29. Moise, G., Sander, J., Ester, M.: Robust projected clustering. Knowledge and Information Systems **14**(3), 273–298 (2008)
30. Moise, G., Zimek, A., Kröger, P., Kriegel, H.P., Sander, J.: Subspace and projected clustering: experimental evaluation and analysis. Knowledge and Information Systems (KAIS) **21**(3), 299–326 (2009)

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

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	E-CB-PCE
Iris	.964(.008)	.989(.001)	.989(.001)	.986(.002)	.990(.004)	.992(.001)
Wine	.861(.008)	.856(.002)	.857(.003)	.852(.006)	.910(.023)	.905(.034)
Glass	.833(.007)	.843(.003)	.844(.006)	.849(.013)	.871(.030)	.846(.016)
Ecoli	.848(.007)	.818(.006)	.814(.004)	.804(.001)	.884(.010)	.893(.021)
Yeast	.870(.005)	.890(.005)	.888(.006)	.907(.011)	.909(.018)	.912(.010)
Mult.-Feat.	.656(.008)	.662(.001)	.665(.014)	.697(.004)	.715(.046)	.695(.010)
Segmentation	.854(.014)	.815(.005)	.812(.013)	.824(.001)	.942(.071)	.935(.023)
Abalone	.944(.004)	.976(.002)	.979(.004)	.994(.001)	.977(.010)	.984(.007)
Waveform	.266(.013)	.279(.001)	.279(.001)	.279(.001)	.323(.004)	.341(.008)
Letter	.604(.011)	.453(.009)	.448(.009)	.401(.001)	.686(.061)	.644(.019)
Isolet	.838(.004)	.891(.001)	.891(.001)	.861(.019)	.897(.012)	.903(.001)
Gisette	.750(.005)	.771(.001)	.770(.002)	.771(.001)	.751(.010)	.771(.010)
p53-Mutants	.858(.002)	.920(.001)	.920(.001)	.918(.001)	.908(.007)	.916(.001)
Amazon	.874(.002)	.945(.001)	.945(.001)	.943(.001)	.932(.005)	.939(.001)
Arcene	.664(.010)	.546(.006)	.544(.003)	.534(.006)	.558(.029)	.592(.001)
Shapes	.641(.005)	.629(.004)	.631(.002)	.585(.009)	.679(.011)	.693(.002)
Tracedata	.853(.003)	.848(.003)	.849(.001)	.820(.011)	.894(.028)	.930(.007)
ControlChart	.510(.022)	.154(.001)	.154(.001)	.154(.001)	.556(.038)	.396(.022)
Twopat	.685(.024)	.342(.001)	.343(.005)	.343(.001)	.599(.105)	.472(.073)
N30	.810(.005)	.670(.010)	.668(.019)	.676(.015)	.842(.055)	.808(.052)
D75	.799(.008)	.741(.006)	.743(.004)	.717(.017)	.832(.040)	.832(.009)
S2500	.803(.007)	.772(.009)	.770(.028)	.673(.028)	.818(.049)	.864(.012)
<i>min</i>	.266	.154	.154	.154	.323	.341
<i>max</i>	.964	.989	.989	.994	.990	.992
<i>avg</i>	.763	.719	.718	.709	.794	.785

Table 9 Average gains of E-CB-PCE w.r.t. the competing methods

<i>criterion</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>
$F1_{of}$	0	.014	.021	.025	.020
$F1_o$.029	.049	.054	.108	.023
$F1_f$.005	.056	.055	.060	.078
$\overline{F1}_{of}$.147	.186	.194	.237	.019
$\overline{F1}_o$.021	.046	.055	.100	.021
$\overline{F1}_f$.022	.066	.067	.076	-.010
<i>avg</i>	.037	.070	.074	.101	.025

31. Müller, E., Günnemann, S., Assent, I., Seidl, T.: Evaluating Clustering in Subspace Projections of High Dimensional Data. <http://dme.rwth-aachen.de/en/OpenSubspace/evaluation> (2009)
32. Ng, E.K.K., Fu, A.W.C., Wong, R.C.W.: Projective Clustering by Histograms. *IEEE Trans. on Knowledge and Data Engineering (TKDE)* **17**(3), 369–383 (2005)
33. Nguyen, N., Caruana, R.: Consensus Clustering. In: *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pp. 607–612 (2007)
34. Parsons, L., Haque, E., Liu, H.: Subspace Clustering for High Dimensional Data: A Review. *SIGKDD Explorations* **6**(1), 90–105 (2004)
35. Procopiuc, C.M., Jones, M., Agarwal, P.K., Murali, T.M.: A Monte Carlo algorithm for fast projective clustering. In: *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pp. 418–427 (2002)
36. van Rijsbergen, C.: *Information Retrieval*. Butterworths (1979)
37. Sequeira, K., Zaki, M.: SCHISM: A New Approach for Interesting Subspace Mining. In: *Proc. IEEE Int. Conf. on Data Mining (ICDM)*, pp. 186–193 (2004)
38. Strehl, A., Ghosh, J.: Cluster Ensembles — A Knowledge Reuse Framework for Combining Multiple Partitions. *Journal of Machine Learning Research* **3**, 583–617 (2002)
39. Strehl, A., Ghosh, J., Mooney, R.: Impact of Similarity Measures on Web-Page Clustering. In: *Proc. of the AAAI Workshop on Artificial Intelligence for Web Search*, pp. 58–64 (2000)

Table 10 Execution times (milliseconds)

<i>dataset</i>	<i>MOEA-PCE</i>	<i>EM-PCE</i>	<i>E-EM-PCE</i>	<i>E-2S-PCE</i>	<i>CB-PCE</i>	E-CB-PCE
Iris	2,056	37	109	253	74	1,492
Wine	2,558	29	88	163	144	1,223
Glass	7,712	56	615	248	500	9,177
Ecoli	14,401	59	685	625	1,147	9,337
Yeast	227,878	757	20,438	21,560	30,384	83,008
Multi.-Feat.	490,602	116,334	151,582	88,713	1,562,139	114,368
Segmentation	233,951	1,854	17,385	40,014	25,692	31,780
Abalone	3,411,116	4,105	156,959	430,974	275,053	857,164
Waveform	125,247	4,912	15,905	91,844	10,487	2,179
Letter	2,248,695	9,591	126,222	772,883	70,458	135,832
Isolet	20,676,754	10,100	10,000	8,488	66,136	1,447
Gisette	966,108	34,260	38,216	29,700	93,148	1,450
p53-Mutants	58,695	22,209	22,168	19,347	65,501	1,490
Amazon	395,988	21,556	21,619	20,914	135,446	12,737
Arcene	120,537	21,557	21,499	17,405	81,433	2,413
Shapes	211,654	17,752	19,152	12,473	282,857	106,180
Tracedata	12,777	1,062	1,108	960	9,000	4,716
ControlChart	50,798	1,522	5,397	2,801	13,900	20,708
TwoPAT	31,850	2,946	5,706	4,606	9,788	5,344
N30	164,969	1,340	13,781	16,243	22,904	44,517
D75	135,297	4,558	13,414	15,477	32,938	30,631
S2500	290,408	2,717	29,223	44,008	39,039	55,607

40. Tomasev, N., Radovanovic, M., Mladenic, D., Ivanovic, M.: The Role of Hubness in Clustering High-Dimensional Data. In: Proc. Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining (PAKDD), pp. 183–195 (2011)
41. Topchy, A.P., Jain, A.K., Punch, W.F.: Clustering Ensembles: Models of Consensus and Weak Partitions. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) **27**(12), 1866–1881 (2005)
42. Woo, K., Lee, J., Kim, M., Lee, Y.: FINDIT: a fast and intelligent subspace clustering algorithm using dimension voting. Information and Software Technology **46**(4), 255–271 (2004)
43. Yip, K.Y., Cheung, D.W., Ng, M.K.: HARP: A Practical Projected Clustering Algorithm. IEEE Trans. on Knowledge and Data Engineering (TKDE) **16**(11), 1387–1397 (2004)
44. Yip, K.Y., Cheung, D.W., Ng, M.K.: On Discovery of Extremely Low-Dimensional Clusters using Semi-Supervised Projected Clustering. In: Proc. IEEE Int. Conf. on Data Engineering (ICDE), pp. 329–340 (2005)
45. Yiu, M.L., Mamouli, N.: Iterative Projected Clustering by Subspace Mining. IEEE Trans. on Knowledge and Data Engineering (TKDE) **17**(2), 176–189 (2005)

A Proofs

A.1 Proofs of Section 2

Proposition 1 For any two projective clusters C, C' it holds that $\mathbf{X}_C = \mathbf{X}_{C'}$ if and only if $\Gamma_C = \Gamma_{C'}$ and $\Delta_C = \Delta_{C'}$.

Proof Let us prove both the directions of the implication. Proving $\mathbf{X}_C = \mathbf{X}_{C'} \Leftarrow \Gamma_C = \Gamma_{C'} \wedge \Delta_C = \Delta_{C'}$ is immediate: by definition, $\mathbf{X}_C = \Gamma_C^T \Delta_C$ and $\mathbf{X}_{C'} = \Gamma_{C'}^T \Delta_{C'}$. Regarding $\mathbf{X}_C = \mathbf{X}_{C'} \Rightarrow \Gamma_C = \Gamma_{C'} \wedge \Delta_C = \Delta_{C'}$, we

first note that:

$$\begin{aligned} \mathbf{X}_C = \mathbf{X}_{C'} &\Rightarrow X_C(i, j) = X_{C'}(i, j), \forall i, \forall j \Rightarrow \\ &\Rightarrow \Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j} = \Gamma_{C', \mathbf{o}_i} \times \Delta_{C', j}, \forall i, \forall j. \end{aligned} \quad (28)$$

By summing the statements in (28) over j , we obtain:

$$\begin{aligned} \Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j} &= \Gamma_{C', \mathbf{o}_i} \times \Delta_{C', j}, \forall i, \forall j \Rightarrow \\ &\Rightarrow \sum_{j=1}^{|\mathcal{F}|} \Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j} = \sum_{j=1}^{|\mathcal{F}|} \Gamma_{C', \mathbf{o}_i} \times \Delta_{C', j}, \forall i \Rightarrow \\ &\Rightarrow \Gamma_{C, \mathbf{o}_i} \sum_{j=1}^{|\mathcal{F}|} \Delta_{C, j} = \Gamma_{C', \mathbf{o}_i} \sum_{j=1}^{|\mathcal{F}|} \Delta_{C', j}, \forall i, \end{aligned}$$

which implies $\Gamma_{C, \mathbf{o}_i} = \Gamma_{C', \mathbf{o}_i}, \forall i$, since $\sum_{j=1}^{|\mathcal{F}|} \Delta_{C, j} = \sum_{j=1}^{|\mathcal{F}|} \Delta_{C', j} = 1$ by definition. Similarly, by summing the statements in (28) over i leads to:

$$\begin{aligned} \Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j} &= \Gamma_{C', \mathbf{o}_i} \times \Delta_{C', j}, \forall i, \forall j \Rightarrow \\ &\Rightarrow \sum_{i=1}^{|\mathcal{D}|} \Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j} = \sum_{i=1}^{|\mathcal{D}|} \Gamma_{C', \mathbf{o}_i} \times \Delta_{C', j}, \forall j \Rightarrow \\ &\Rightarrow \Delta_{C, j} \sum_{i=1}^{|\mathcal{D}|} \Gamma_{C, \mathbf{o}_i} = \Delta_{C', j} \sum_{i=1}^{|\mathcal{D}|} \Gamma_{C', \mathbf{o}_i}, \forall j \Rightarrow \\ &\Rightarrow \Delta_{C, j} = \Delta_{C', j}, \forall j, \end{aligned}$$

where the last derivation holds since the result $\Gamma_{C, \mathbf{o}_i} = \Gamma_{C', \mathbf{o}_i}, \forall i$, shown above clearly implies $\sum_{i=1}^{|\mathcal{D}|} \Gamma_{C, \mathbf{o}_i} = \sum_{i=1}^{|\mathcal{D}|} \Gamma_{C', \mathbf{o}_i}$. \square

Proposition 2 For any projective cluster C , its object-based representation $\Gamma_C = (\Gamma_{C, \mathbf{o}_1}, \dots, \Gamma_{C, \mathbf{o}_{|\mathcal{D}|}})$ and feature-based representation $\Delta_C = (\Delta_{C, 1}, \dots, \Delta_{C, |\mathcal{F}|})$ can uniquely be derived from its projective matrix \mathbf{X}_C as follows:

$$\Gamma_{C, \mathbf{o}_i} = \sum_{j=1}^{|\mathcal{F}|} X_C(i, j), \quad \Delta_{C, j} = \frac{X_C(1, j)}{\sum_{j=1}^{|\mathcal{F}|} X_C(1, j)} = \dots = \frac{X_C(|\mathcal{D}|, j)}{\sum_{j=1}^{|\mathcal{F}|} X_C(|\mathcal{D}|, j)},$$

where $X_C(i, j)$ denotes the element (i, j) of the matrix \mathbf{X}_C .

Proof The first statement holds since:

$$\sum_{j=1}^{|\mathcal{F}|} X_C(i, j) = \sum_{j=1}^{|\mathcal{F}|} \Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j} = \Gamma_{C, \mathbf{o}_i} \sum_{j=1}^{|\mathcal{F}|} \Delta_{C, j} = \Gamma_{C, \mathbf{o}_i}.$$

Once derived the expression for Γ_{C, \mathbf{o}_i} , the second statement is immediate, because each element $X_C(i, j)$ is by definition equal to $\Gamma_{C, \mathbf{o}_i} \times \Delta_{C, j}$. \square

A.2 Proofs of Section 3

Proposition 3 *Given a projective ensemble \mathcal{E} , let \mathcal{Y} denote the set of all clusterings obtained by taking exactly one cluster from each ensemble member, i.e., $\mathcal{Y} = \{\mathbf{C} \mid \mathbf{C} \subseteq \{\hat{C} \mid \hat{C} \in \hat{\mathcal{C}} \wedge \hat{C} \in \mathcal{E}\} \wedge |\hat{C} \cap \mathbf{C}| = 1, \forall \hat{C} \in \mathcal{E}\}$. Moreover, let $\mathbf{X}^* = \operatorname{argmin}_{\hat{\mathbf{X}}} \min_{\mathbf{C} \in \mathcal{Y}} \sum_{C \in \mathbf{C}} T(\hat{\mathbf{X}}, \mathbf{X}_C)$ s.t. $\hat{\mathbf{X}}$ satisfies (2). Given an integer $K \geq 1$, it holds that the optimal projective consensus clustering for the optimization problem defined in (17)–(18) is $\mathcal{C}^* = \{C_1^*, \dots, C_K^*\}$ s.t. $\mathbf{X}_{C_1^*} = \dots = \mathbf{X}_{C_K^*} = \mathbf{X}^*$.*

Proof The constraints in (18) imply that each cluster C of the optimal projective consensus clustering should be associated with exactly one cluster \hat{C} of each ensemble solution, but no limitations are imposed about associations of \hat{C} to multiple C , which are therefore allowed. Within this view, the set \mathcal{Y} identifies a sound and complete searching space for the assignments \implies **diventa mappings** $\longleftarrow x(\hat{C}, C)$. This implies that the matrix \mathbf{X}^* , which is the matrix that minimizes the sum of the Tanimoto distances between itself and all clusters within a set of clusters \mathbf{C} , over all matrices satisfying (2) and all $\mathbf{C} \in \mathcal{Y}$, would represent an optimal solution for the case $K = 1$. The generalization to the case $K > 1$ is straightforward, as $\mathcal{C}^* = \{C_1^*, \dots, C_K^*\}$ s.t. $\mathbf{X}_{C_1^*} = \dots = \mathbf{X}_{C_K^*} = \mathbf{X}^*$ represents a feasible solution to the problem and any other solution where $\mathbf{X}_{C_i^*} \neq \mathbf{X}^*$ for some $i \in [1..K]$ would have a non-smaller objective function value due to the optimality of \mathbf{X}^* . \square

A.3 Proofs of Section 4

Lemma 2 *Given an m -dimensional, binary vector \mathbf{u} , let \mathbf{u}' denote the vector $\frac{1}{n(\mathbf{u})} \mathbf{u}$, where $n(\mathbf{u}) = \sum_{j=1}^m u_j$. For any two m -dimensional binary vectors \mathbf{u} and \mathbf{v} , it holds that $J(\mathbf{u}', \mathbf{v}') = J(\mathbf{u}, \mathbf{v})$.*

Proof

$$\begin{aligned} J(\mathbf{u}', \mathbf{v}') &= \frac{\sum_{j=1}^m u'_j v'_j}{\sum_{j=1}^m (u'_j)^2 + \sum_{j=1}^m (v'_j)^2 - \sum_{j=1}^m u'_j v'_j} = \\ &= \frac{\sum_{j=1}^m \frac{u_j}{n(\mathbf{u})} \frac{v_j}{n(\mathbf{v})}}{\sum_{j=1}^m \left(\frac{u_j}{n(\mathbf{u})}\right)^2 + \sum_{j=1}^m \left(\frac{v_j}{n(\mathbf{v})}\right)^2 - \sum_{j=1}^m \frac{u_j}{n(\mathbf{u})} \frac{v_j}{n(\mathbf{v})}} = \\ &= \frac{\frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j}{\frac{1}{n(\mathbf{u})^2} \sum_{j=1}^m u_j^2 + \frac{1}{n(\mathbf{v})^2} \sum_{j=1}^m v_j^2 - \frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j}. \end{aligned}$$

Since \mathbf{u} and \mathbf{v} are binary vectors, it holds that $\sum_{j=1}^m u_j^2 = \sum_{j=1}^m u_j = n(\mathbf{u})$ and $\sum_{j=1}^m v_j^2 = \sum_{j=1}^m v_j = n(\mathbf{v})$; thus, it results that:

$$\begin{aligned}
J(\mathbf{u}', \mathbf{v}') &= \frac{\frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j}{\frac{1}{n(\mathbf{u})^2} \sum_{j=1}^m u_j^2 + \frac{1}{n(\mathbf{v})^2} \sum_{j=1}^m v_j^2 - \frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j} = \\
&= \frac{\frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j}{\frac{1}{n(\mathbf{u})} + \frac{1}{n(\mathbf{v})} - \frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j} = \\
&= \frac{\frac{1}{n(\mathbf{u})n(\mathbf{v})} \sum_{j=1}^m u_j v_j}{\frac{1}{n(\mathbf{u})n(\mathbf{v})} \left(n(\mathbf{u}) + n(\mathbf{v}) - \sum_{j=1}^m u_j v_j \right)} = \\
&= \frac{\sum_{j=1}^m u_j v_j}{\sum_{j=1}^m u_j^2 + \sum_{j=1}^m v_j^2 - \sum_{j=1}^m u_j v_j} = J(\mathbf{u}, \mathbf{v}).
\end{aligned}$$

□

Lemma 1 *Let CB-PCE RESTRICTED be a special version of the CB-PCE ENHANCED problem where (i) $|\mathcal{D}| = 1$, (ii) $K = 1$ (K denotes the number of clusters in the output projective consensus clustering), (iii) $|\hat{\mathcal{C}}| = 1$, $\forall \hat{\mathcal{C}} \in \mathcal{E}$, (iv) $n(\Delta_{\hat{\mathcal{C}}}) = n(\Delta_{\hat{\mathcal{C}}'}) = n_{\mathcal{E}}$, $\forall \hat{\mathcal{C}}, \hat{\mathcal{C}}'$, where $n(\Delta_{\hat{\mathcal{C}}}) = \sum_{f \in \mathcal{F}} \mathbb{I}[\Delta_{\hat{\mathcal{C}}, f} > 0]$, and (v) $\Delta_{\hat{\mathcal{C}}, f} = \frac{1}{n_{\mathcal{E}}}$, $\forall f \in \mathcal{F}$, $\forall \hat{\mathcal{C}} \in \hat{\mathcal{C}}, \forall \hat{\mathcal{C}} \in \mathcal{E}$. CB-PCE RESTRICTED is **NP-hard**.* □

Proof First, it is easy to note that, due to the restrictions w.r.t. the general CB-PCE ENHANCED, the projective matrix of each cluster $\hat{\mathcal{C}}$ within a valid input ensemble for CB-PCE RESTRICTED corresponds to its (mono-dimensional) feature-based representation $\Delta_{\hat{\mathcal{C}}}$. To show that CB-PCE RESTRICTED is **NP-hard**, we define a reduction from the JACCARD MEDIAN problem, which has been proven to be **NP-hard** in [11]. JACCARD MEDIAN is defined as follows: given a set U of m -dimensional, binary vectors, find a (possibly new) binary vector \mathbf{x} such that $\sum_{\mathbf{u} \in U} (1 - J(\mathbf{u}, \mathbf{x}))$ is minimized. The corresponding decision problem is: given a set U of m -dimensional binary vectors and a real value r , check whether a binary vector \mathbf{x} exists such that $\sum_{\mathbf{u} \in U} (1 - J(\mathbf{u}, \mathbf{x})) \leq r$. Given a set U of vectors in input to JACCARD MEDIAN, we define a valid instance of our problem by building a new set $U' = \{\mathbf{u}' \mid \mathbf{u}' = \frac{1}{n(\mathbf{u})} \mathbf{u}, \mathbf{u} \in U\}$. It is easy to see that U' is a valid input for CB-PCE RESTRICTED, as all vectors in U' represent valid projective matrices as their values sum to one, and this is the only condition to satisfy for an ensemble \mathcal{E} having $|\hat{\mathcal{C}}| = 1$, $\forall \hat{\mathcal{C}} \in \mathcal{E}$ and $|\mathcal{D}| = 1$. The key point is to show that, for each r , U has a solution for the decision version of JACCARD MEDIAN if and only if U admits a solution for the decision version of CB-PCE RESTRICTED. Let us prove the first side of the implication, i.e., if U is a *yes* instance for JACCARD MEDIAN then U' is a *yes* instance for CB-PCE RESTRICTED, for all r . To this end, let \mathbf{x} denote the binary vector that makes U a *yes* instance for JACCARD MEDIAN, i.e., the binary vector such that $\sum_{\mathbf{u} \in U} (1 - J(\mathbf{u}, \mathbf{x})) \leq r$.

According to Lemma 2, the Jaccard similarities, and therefore the Jaccard distances defined as $1 - J$, are preserved if each input vector \mathbf{u} is scaled up by a factor $n(\mathbf{u})$. This implies that $\sum_{\mathbf{u} \in U} (1 - J(\mathbf{u}, \mathbf{x})) = \sum_{\mathbf{u}' \in U'} (1 - J(\mathbf{u}', \mathbf{x}')) \leq r$. Thus, $\mathbf{x}' = \frac{1}{n(\mathbf{x})} \mathbf{x}$ makes U' a *yes* instance for CB-PCE RESTRICTED (note that, like each vector $\mathbf{u}' \in U'$, \mathbf{x}' satisfies the constraints of CB-PCE RESTRICTED on the output format because it represents a valid projective matrix). The second side of the implication, i.e., if U is a *no* instance for JACCARD MEDIAN then U' is a *no* instance for CB-PCE RESTRICTED, for all r , can be proved similarly. Indeed, according to Lemma 2 again, if no binary vector is within a distance r from the vectors in U , then no vector can exist with such a property for U' either. \square

Theorem 1 CB-PCE ENHANCED is **NP-hard**.

Proof Straightforward, since CB-PCE ENHANCED is a more general version of CB-PCE RESTRICTED, which has been proven to be **NP-hard** in Lemma 1. \square

Theorem 2 The optimum of the problem P defined in (22)-(24) is $(\forall \mathcal{M}, \forall \mathbf{o})$:

$$\Gamma_{\mathcal{M}, \mathbf{o}}^* = \left[\sum_{\mathcal{M}' \in \mathbf{M}} \left(\frac{A_{\mathcal{M}, \mathbf{o}}}{A_{\mathcal{M}', \mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-1}.$$

Proof The optimal $\Gamma_{\mathcal{M}, \mathbf{o}}^*$ can be found by means of the conventional *Lagrange multipliers* method. We first consider the relaxed problem P' of P obtained by temporarily discarding the inequality constraints from the constraint set of P (i.e., the constraints defined in (24)).

We define the new unconstrained objective function Q' for P' as follows:

$$Q' = Q + \sum_{\mathbf{o} \in \mathcal{D}} \lambda_{\mathbf{o}} \left(\sum_{\mathcal{M}' \in \mathbf{M}} \Gamma_{\mathcal{M}', \mathbf{o}} - 1 \right). \quad (29)$$

The optimal $\Gamma_{\mathcal{M}, \mathbf{o}}^*$ is computed by first retrieving the stationary points of Q' , i.e., the points for which

$$\nabla Q' = \left(\frac{\partial Q'}{\partial \Gamma_{\mathcal{M}, \mathbf{o}}}, \frac{\partial Q'}{\partial \lambda_{\mathbf{o}}} \right) = 0.$$

Thus, we solve the following system of equations:

$$\frac{\partial Q'}{\partial \Gamma_{\mathcal{M}, \mathbf{o}}} = \alpha A_{\mathcal{M}, \mathbf{o}} (\Gamma_{\mathcal{M}, \mathbf{o}})^{\alpha-1} + \lambda_{\mathbf{o}} = 0, \quad (30)$$

$$\frac{\partial Q'}{\partial \lambda_{\mathbf{o}}} = \sum_{\mathcal{M}' \in \mathbf{M}} \Gamma_{\mathcal{M}', \mathbf{o}} - 1 = 0. \quad (31)$$

By solving (30) w.r.t. $\Gamma_{\mathcal{M}, \mathbf{o}}$ and substituting the solution in (31), we obtain:

$$\sum_{\mathcal{M}' \in \mathbf{M}} \left(\frac{-\lambda_{\mathbf{o}}}{\alpha A_{\mathcal{M}', \mathbf{o}}} \right)^{\frac{1}{\alpha-1}} = 1. \quad (32)$$

By solving (32) w.r.t. $\lambda_{\mathbf{o}}$ and substituting such a solution in (30), we obtain:

$$\alpha A_{\mathcal{M},\mathbf{o}} (\Gamma_{\mathcal{M},\mathbf{o}})^{\alpha-1} - \left[\sum_{\mathcal{M}' \in \mathbf{M}} \left(\frac{1}{\alpha A_{\mathcal{M}',\mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-(\alpha-1)} = 0. \quad (33)$$

Finally, by solving (33) w.r.t. $\Gamma_{\mathcal{M},\mathbf{o}}$, we obtain a stationary point whose expression is exactly equal to that reported in the claim of the theorem:

$$\Gamma_{\mathcal{M},\mathbf{o}}^* = \left[\sum_{\mathcal{M}' \in \mathbf{M}} \left(\frac{A_{\mathcal{M},\mathbf{o}}}{A_{\mathcal{M}',\mathbf{o}}} \right)^{\frac{1}{\alpha-1}} \right]^{-1}. \quad (34)$$

As it holds that (i) the stationary points of the Lagrangian function Q' are also stationary points of the original objective function Q , (ii) the feasible region of P , and hence the feasible region of P' , is a convex set, and (iii) Q is convex w.r.t. $\Gamma_{\mathcal{M},\mathbf{o}}$, it follows that such a stationary point represents a global minimum of Q , and, accordingly, the optimal solution of P' . Moreover, as $A_{\mathcal{M},\mathbf{o}} \geq 0, \forall \mathcal{M}, \forall \mathbf{o}$, it is trivial to observe that $\Gamma_{\mathcal{M},\mathbf{o}}^* \geq 0, \forall \mathcal{M}, \forall \mathbf{o}$. Therefore, the solution in (34) satisfies the inequality constraints that were temporarily discarded in order to define the relaxed problem P' (cf. (24)); thus, it represents the optimal solution of the original problem P , which proves the theorem. \square

Theorem 3 *The optimum of the problem defined in (25)-(27) is $(\forall \mathcal{M}, \forall f)$:*

$$\Delta_{\mathcal{M},f}^* = \left[\sum_{f' \in \mathcal{F}} \left(\frac{B_{\mathcal{M},f}}{B_{\mathcal{M},f'}} \right)^{\frac{1}{\beta-1}} \right]^{-1}.$$

Proof Analogous to Theorem 2. \square

Proposition 4 *For any two projective clusters C, C' it holds that:*

$$T(\mathbf{X}_C, \mathbf{X}_{C'}) = 1 - \frac{p(C, C')}{q(C) + q(C') - p(C, C')},$$

where

$$p(C, C') = (\mathbf{\Gamma}_C \mathbf{\Gamma}_{C'}^T) \times (\mathbf{\Delta}_C \mathbf{\Delta}_{C'}^T), \quad q(C) = (\|\mathbf{\Gamma}_C\|_2^2) \times (\|\mathbf{\Delta}_C\|_2^2).$$

Proof The Tanimoto distance $T(\mathbf{X}_C, \mathbf{X}_{C'})$ is equal to $1 - \hat{J}(\mathbf{X}, \hat{\mathbf{X}})$. Regarding \hat{J} , it holds that:

$$\begin{aligned}
\hat{J}(\mathbf{X}, \hat{\mathbf{X}}) &= \frac{\sum_{i=1}^{|\text{rows}(\mathbf{X})|} \mathbf{X}_i^\top \cdot \hat{\mathbf{X}}_i}{\|\mathbf{X}\|_2^2 + \|\hat{\mathbf{X}}\|_2^2 - \sum_{i=1}^{|\text{rows}(\mathbf{X})|} \mathbf{X}_i^\top \cdot \hat{\mathbf{X}}_i} = \\
&= \frac{\sum_{\mathbf{o} \in \mathcal{D}} \sum_{f \in \mathcal{F}} \Gamma_{C,\mathbf{o}} \Delta_{C,f} \Gamma_{C',\mathbf{o}} \Delta_{C',f}}{\sum_{\mathbf{o} \in \mathcal{D}} \sum_{f \in \mathcal{F}} \Gamma_{C,\mathbf{o}} \Delta_{C,f} + \sum_{\mathbf{o} \in \mathcal{D}} \sum_{f \in \mathcal{F}} \Gamma_{C',\mathbf{o}} \Delta_{C',f} - \sum_{\mathbf{o} \in \mathcal{D}} \sum_{f \in \mathcal{F}} \Gamma_{C,\mathbf{o}} \Delta_{C,f} \Gamma_{C',\mathbf{o}} \Delta_{C',f}} = \\
&= \frac{\left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}} \Gamma_{C',\mathbf{o}} \right) \times \left(\sum_{f \in \mathcal{F}} \Delta_{C,f} \Delta_{C',f} \right)}{\left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}} \right) \times \left(\sum_{f \in \mathcal{F}} \Delta_{C,f} \right) + \left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C',\mathbf{o}} \right) \times \left(\sum_{f \in \mathcal{F}} \Delta_{C',f} \right) - \left(\sum_{\mathbf{o} \in \mathcal{D}} \Gamma_{C,\mathbf{o}} \Gamma_{C',\mathbf{o}} \right) \times \left(\sum_{f \in \mathcal{F}} \Delta_{C,f} \Delta_{C',f} \right)} = \\
&= \frac{\left(\mathbf{\Gamma}_C \mathbf{\Gamma}_{C'}^\top \right) \times \left(\mathbf{\Delta}_C \mathbf{\Delta}_{C'}^\top \right)}{\left(\|\mathbf{\Gamma}_C\|_2^2 \right) \times \left(\|\mathbf{\Delta}_C\|_2^2 \right) + \left(\|\mathbf{\Gamma}_{C'}\|_2^2 \right) \times \left(\|\mathbf{\Delta}_{C'}\|_2^2 \right) - \left(\mathbf{\Gamma}_C^\top \cdot \mathbf{\Gamma}_{C'} \right) \times \left(\mathbf{\Gamma}_C \mathbf{\Gamma}_{C'}^\top \right)} = \\
&= \frac{p(C, C')}{q(C) + q(C') - p(C, C')}.
\end{aligned}$$

□