Chapter 5
Learning Similarities from Examples Under the Evidence Accumulation Clustering Paradigm

Ana L.N. Fred, André Lourenço, Helena Aidos, Samuel Rota Bulò, Nicola Rebagliati, Mário A.T. Figueiredo, and Marcello Pelillo

Abstract The SIMBAD project puts forward a unified theory of data analysis under a (dis)similarity based object representation framework. Our work builds on the duality of probabilistic and similarity notions on pairwise object comparison. We address the Evidence Accumulation Clustering paradigm as a means of learning pairwise similarity between objects, summarized in a co-association matrix. We show the dual similarity/probabilistic interpretation of the co-association matrix and exploit these for coherent consensus clustering methods, either exploring embeddings over learned pairwise similarities, in an attempt to better highlight the clustering

A.L.N. Fred (✉) · H. Aidos · M.A.T. Figueiredo
Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
e-mail: afred@lx.it.pt

H. Aidos
e-mail: haidos@lx.it.pt

M.A.T. Figueiredo
e-mail: mtf@lx.it.pt

A. Lourenço
Instituto Superior de Engenharia de Lisboa, Lisbon, Portugal
e-mail: arlourenco@lx.it.pt

A. Lourenço
Instituto de Telecomunicações, Lisbon, Portugal

S. Rota Bulò
Fondazione Bruno Kessler, Povo, Trento, Italy
e-mail: samyrota@gmail.com

N. Rebagliati
VTT Technical Research Centre of Finland, Espoo, Finland
e-mail: nicola.rebagliati@gmail.com

M. Pelillo
DAIS, Università Ca' Foscari, Venezia, Italy
e-mail: pelillo@dais.unive.it

M. Pelillo (ed.), *Similarity-Based Pattern Analysis and Recognition*,
Advances in Computer Vision and Pattern Recognition,
DOI 10.1007/978-1-4471-5628-4_5, © Springer-Verlag London 2013

structure of the data, or by means of a unified probabilistic approach leading to soft assignments of objects to clusters.

5.1 Introduction

The goal of clustering algorithms is to organize a set of unlabeled objects into groups or clusters such that objects within a cluster are more similar than objects in distinct clusters. Clustering techniques require the definition of a similarity measure between patterns, geometrical or probabilistic, which is not easy to specify in the absence of any prior knowledge about cluster shapes and structure. On the other hand, clustering solutions unveil or induce pairwise similarity, when grouping objects in a same cluster. Given the diversity of clustering algorithms, each one with its own approach for estimating the number of clusters, imposing a structure on the data, and validating the resulting clusters, we are faced with a myriad of potential similarity learners.

Clustering ensemble methods obtain consensus solutions from a set of base clustering algorithms, thus constituting a step towards the goal of assumption-free clustering. Several authors have shown that these methods tend to reveal more robust and stable cluster structures than the individual clusterings in the Clustering Ensemble (CE) [9, 10, 39].

There is a close connection between the concepts of pairwise similarity and probability in the context of unsupervised learning. It is a common assumption that, if two objects are similar, it is very likely that they are grouped together by some clustering algorithm; the higher the similarity, the higher the probability of co-occurrence in a cluster. Conversely, if two objects co-occur very often in the same cluster (high co-occurrence probability), then it is very likely that they are very similar. This duality and correspondence between pairwise similarity and pairwise probability within clusters forms the core idea of the clustering ensemble approach known as *Evidence Accumulation Clustering* (EAC) [9, 11].

The idea of evidence accumulation clustering is to combine the results of multiple clusterings into a single data partition by viewing each clustering result as an independent evidence of data organization. This evidence is accumulated in a co-associations matrix, the intrinsic learned pairwise similarity, which constitutes the core of the method. A consensus solution is obtained by applying a clustering algorithm over this matrix.

In this chapter. we build on the EAC paradigm, exploring the duality of similarity-based and probabilistic interpretations of the learned co-association matrix in order to produce robust and informative consensus solutions. Interpreting co-associations as new data representations, we propose to use embeddings over this matrix, as an intermediate step in the consensus clustering process, in order to extract relevant information into lower dimensional spaces. Consensus (hard) data partitions are obtained from the later by applying hierarchical clustering algorithms. By assuming a probabilistic re-interpretation of the co-association matrix, we then propose a fully probabilistic formulation of the clustering problem, leading to soft

consensus solutions. The method, that we denote as PEACE (*Probabilistic Evidence Accumulation for Clustering Ensembles*), obtains probabilistic cluster assignments through an optimization process that maximizes the likelihood of observing the empirical co-associations given the underlying object to cluster probabilistic assignment model.

The chapter is organized as follows. We start we a brief review of related work on clustering ensemble methods in Sect. 5.2. The notation and basic definitions are provided in Sect. 5.3. The EAC paradigm is reviewed in Sect. 5.4, while the proposed methods based on embeddings and probabilistic modeling are presented in Sects. 5.5 and 5.6, respectively. Results of the application of these methods to real and synthetic benchmark data, in a comparative study with the baseline EAC method, is provided in Sect. 5.7. Conclusions are drawn in a final section.

5.2 Related Work

Clustering is one of the central problems in Pattern Recognition and Machine Learning. Hundreds of clustering algorithms exist, handling differently issues such as cluster shape, density, noise. k-means is one of the most studied and used algorithms [17, 18, 41].

Recently, taking advantage of the diversity of clustering solutions produced by clustering algorithms over the same dataset, an approach known as *Clustering Ensemble methods*, has been proposed and gained an increasing interest [2, 9, 22, 39]. Given a set of data partitions—a clustering ensemble (CE)—these methods propose a consensus partition based on a combination strategy, having in general a leveraging effect over the single data partitions in the CE.

The topic of clustering combination and consensus clustering are completing the first decade of research.

Different paradigms were followed in the literate: (i) similarity between objects, induced by the clustering ensemble [9, 11, 39]; (ii) similarity between partitions [2, 7, 33, 42–44]; (iii) combining similarity between objects and partitions [8]; (iv) probabilistic approaches to cluster ensembles [42, 45, 46].

Strehl and Ghosh [39] formulated the clustering ensemble problem as an optimization problem based on the maximal average mutual information between the optimal combined clustering and the clustering ensemble exploring graph theoretical concepts, and presenting three algorithms to solve it: Cluster-based Similarity Partitioning Algorithm (CSPA), Hyper Graph Partitioning Algorithm (HGPA) and Meta CLustering Algorithm (MCLA). CSPA, uses a graph partitioning algorithm, METIS [20], for extracting a consensus partition from the co-association matrix. In [33], this approach was extended to allow soft clusterings on the clustering ensemble. Hyper Graph Partitioning Algorithm (HGPA) and Meta CLustering Algorithm (MCLA) are based on hyper-graphs, where vertices correspond to objects, and the hyperedges correspond to the clusters of the clustering ensemble. HGPA obtains the consensus solution using an hyper-graph partitioning algorithm, HMETIS [21]; MCLA uses another heuristic which allows clustering clusters.

Topchy et al. [43, 44] proposed the Quadratic Mutual Information Algorithm (QMI) based on similarities between the partitions on the ensemble rather than similarities between objects. It is based on the notion of median partition defined as the partition that best summarizes the partitions of the ensemble and is optimized using an algorithm based on a squared error criterion.

Ayad and Kamel [2], following [7], proposed the idea of cumulative voting as a solution for the problem of aligning the cluster labels. Each clustering of the clustering ensemble is transformed into a probabilistic representation with respect to a common reference clustering. Three voting schemes are presented: Un-normalized fixed-Reference Cumulative Voting (URCV), fixed-Reference Cumulative Voting (RCV), and Adaptive Cumulative Voting (ACV).

Fern and Brodley [8] proposed the Hybrid Bipartite Graph Formulation (HBGF), where both data points and clusters of the ensemble are modeled as vertices retaining all of the information provided by the clustering ensemble, and allowing to consider the similarity among data points and clusters. The partitioning of this bipartite graph is produced using the multi-way spectral graph partitioning algorithm proposed by Ng et al. [32], which seeks to optimize the normalized cut criterion [37], or as alternative a graph partitioning algorithm, METIS [20].

In [42, 44], Topchy et al. proposed a probabilistic interpretation of the clustering combination problem, formulation the problem as a multinomial mixture model (MM) over the labels of the clustering ensembles. In Wang et al. [45], this idea was extended, introducing a Bayesian version of the multinomial mixture model, entitled Bayesian cluster ensembles (BCE). Using a strategy very similar to *Latent Dirichlet Allocation* (LDA) models [38], but applied to a different input space, features are now the labels of the ensembles, the posterior distribution being approximated using variational inference or Gibbs sampling. More recently, a nonparametric version of BCE was proposed [46].

5.3 Notation and Definitions

Sets are denoted by uppercase calligraphic letters (e.g., $\mathcal{O}$, $\mathcal{E}$, ...) except for $\mathbb{R}$ and $\mathbb{R}_+$ which represent the sets of real numbers and nonnegative real numbers, respectively. The *cardinality* of a set is written as $|\cdot|$. We denote *vectors* with lowercase boldface letters (e.g., $\mathbf{x}$, $\mathbf{y}$, ...) and *matrices* with uppercase boldface letters (e.g., $\mathbf{X}$, $\mathbf{Y}$, ...). The ith component of a vector $\mathbf{x}$ is denoted as x_i and the (i, j)th component of a matrix $\mathbf{Y}$ is written as y_{ij}. The *transposition* operator is given by the symbol $^\top$. The ℓ_p-norm of a vector $\mathbf{x}$ is written as $\|\mathbf{x}\|_p$ and we implicitly assume a ℓ_2 (or Euclidean) norm, where p is omitted. We denote by $\mathbf{1}_n$ an n-dimensional column vector of all 1's and by $\mathbf{e}_n^{(j)}$ the jth column of the n-dimensional identity matrix. The *trace* of matrix $\mathbf{M} \in \mathbb{R}^{n\times n}$ is given by $\mathrm{Tr}(\mathbf{M}) = \sum_{i=1}^{n} m_{ii}$.

A *probability distribution* over a discrete set $\{1, \ldots, K\}$ is an element of the *standard simplex* Δ_K, which is defined as

$$\Delta_K = \left\{\mathbf{x} \in \mathbb{R}_+^K : \|\mathbf{x}\|_1 = 1\right\}.$$

The *support* $\sigma(\mathbf{x})$ of a probability distribution $\mathbf{x} \in \Delta_K$ is the set of indices corresponding to positive components of $\mathbf{x}$, i.e.,

$$\sigma(\mathbf{x}) = \big\{i \in \{1, \ldots, K\} : x_i > 0\big\}.$$

Random variables (r.v.) are represented by uppercase letters (e.g., X), and realizations of the later by corresponding lowercase letters. The probability on an event is denoted as $\Pr(\cdot)$ The *expected value* of a random variable X is denoted by $E(X)$.

The *entropy* of a probability distribution $\mathbf{x} \in \Delta_K$ is given by

$$H(\mathbf{x}) = -\sum_{j=1}^{K} x_j \log(x_j)$$

and the *Kullback–Leibler divergence* between two distributions $\mathbf{x}, \mathbf{y} \in \Delta_K$ is given by

$$D_{\mathrm{KL}}(\mathbf{x} \parallel \mathbf{y}) = \sum_{j=1}^{K} x_j \log\left(\frac{x_j}{y_j}\right),$$

where we assume $\log 0 \equiv -\infty$ and $0 \log 0 \equiv 0$.

Let $\mathscr{S} = \{s_1, \ldots, s_n\}$ denote a data set with n objects or samples. Let $\mathscr{O} = \{1, \ldots, n\}$ be the indices of the set of n objects, and let $\mathscr{O}_u \subseteq \mathscr{O}$ represent a subsampling (without replacement) from $\mathscr{O}$, with $|\mathscr{O}_u| < n$. When objects are represented in vector form in a d-dimensional feature space, we denote by $\mathbf{O} = [\mathbf{o}_1, \ldots, \mathbf{o}_n]$ the $d \times n$ matrix of object vectors, column i corresponding to the vector representation, $\mathbf{o}_i$, of the ith object. An alternative to the feature representation is the (dis)similarity representation defined on direct pairwise object comparisons. We denote the dissimilarity representation by a $n \times n$ matrix $\mathbb{D}$, where $d_{ij} = d(s_i, s_j)$ is the dissimilarity value between samples i and j.

The goal of clustering is to organize the objects into K groups or clusters. We distinguish between *hard* and *soft* clusterings. A *hard* clustering is a function $p_u : \mathscr{O}_u \to \{1, \ldots, K_u\}$ assigning a class label, out of K_u available ones, to data points in $\mathscr{O}_u \subseteq \mathscr{O}$. The result of this clustering is a data partition, written as a vector $\mathbf{p}^{(u)} = p_u(\mathscr{O}_u) = [p_i^{(u)}]_{i=1:n}$, $p_i^{(u)} = \mathbf{p}^{(u)}(i) \in \{1, \ldots, K_u\}$, or alternatively, on cluster sets representation: $\mathscr{P}_u = \{\mathscr{C}_1, \mathscr{C}_2, \ldots, \mathscr{C}_{K_u}\}$, where $\mathscr{C}_l$ denotes the lth cluster (the set of object indices composing cluster l), each object belonging to only one cluster. A *soft* clustering is a function s_u mapping each object $i \in \mathscr{O}_u$ into a probability distribution $\gamma_i^{(u)} \in \Delta_{K_u}$, $\gamma_i^{(u)}$ denoting the soft assignment or degree of membership of object i to each of the K_u clusters. The result of a soft clustering s_u is thus a matrix $\gamma^{(u)} = [\gamma_{kj}^{(u)}]_{k=1:K_u}^{j=1:n}$, $\gamma_{kj}^{(u)}$ denoting the degree of membership of object j to cluster k in clustering u.

In this chapter, pairwise similarities are to be learnt from clustering committees. Without loss of generality, we will consider committees of hard clusterings. We define $\mathscr{E} = \{p_u\}_{u=1}^{N} = \{\mathscr{P}_1, \mathscr{P}_2, \ldots, \mathscr{P}_N\}$ a clustering ensemble, i.e., a set of N clusterings (partitions) obtained by applying different algorithms (i.e., different

parameterizations and/or initializations) on (possibly) sub-sampled versions of the objects set.

Since each clustering in the ensemble may stem from a sub-sampled version of the original data set $\mathscr{O}$, some pairs of objects may not appear in all clusterings. Let $\Omega_{ij} \subseteq \{1, \ldots, N\}$ denote the set of clustering indices where both objects i and j have been clustered, i.e., $(u \in \Omega_{ij}) \Leftrightarrow ((i \in \mathscr{O}_u) \wedge (j \in \mathscr{O}_u))$, and let $N_{ij} = |\Omega_{ij}|$ denote its cardinality.

According to the EAC paradigm, and following the vector notation for the representation of partitions, the ensemble of clusterings is summarized in the $n \times n$ *co-association matrix* $\mathbb{C} = [c_{ij}]$, where

$$c_{ij} = \sum_{l \in \Omega_{ij}} \mathbf{1}_{p_i^{(l)} = p_j^{(l)}}, \quad c_{ij} \in \{0, \ldots, N_{ij}\} \tag{5.1}$$

is the number of times objects i and j are co-assigned the same cluster label over the ensemble $\mathscr{E}$ ($\mathbf{1}_p$ is the indicator function, giving 1 if p holds true, and 0 otherwise). An alternative summarization is the *normalized co-association matrix*, $\hat{\mathbb{C}} = [\hat{c}_{ij}]$, where

$$\hat{c}_{ij} = \frac{c_{ij}}{N_{ij}}, \quad \hat{c}_{ij} \in [0, 1] \tag{5.2}$$

represents the percentage of times objects i and j are gathered in a same cluster over the clustering ensemble.

5.4 The Evidence Accumulation Paradigm (EAC)

The EAC paradigm can be summarized in the following three steps method:

EAC

1. *Build a clustering ensemble $\mathscr{E}$*. A diversity of clustering solutions is achieved by running several algorithms, or the same algorithm with different parameter values and/or initializations, on possibly sub-sampled versions of the data set.
2. *Accumulate evidence from $\mathscr{E}$ in a pairwise co-association matrix*. Evidence on pairwise associations are accumulated from the individual clusterings in $\mathscr{E}$. The summary of these associations are given either by:
   - Computing $\mathbb{C}$ and $\{N_{ij}\}$, as given in Sect. 5.3, Eq. (5.1);
   - Determining $\hat{\mathbb{C}}$ using Eq. (5.2).

   This voting mechanism is the key issue of the method, subsuming the problem of class correspondence in consensus clustering.
3. *Extract the consensus clustering from the co-associations*. By applying a clustering algorithm over the learned pairwise associations between objects, a consensus clustering is obtained.

The object of the EAC method is the CE, on which it is built, not the actual objects. As such, it is a clustering method that intrinsically preserves data privacy: Individual descriptions of the underlying data are not required in order to produce a clustering combination solution. Furthermore, it effectively fuses information from multiple views of the data, exploring single or hybrid representations, either feature-based or similarity-based. Some of its steps and characteristics are detailed next.

5.4.1 Building Clustering Ensembles

Clustering ensembles can be generated by following two main approaches: (i) choice of data representation and (ii) choice of clustering algorithms or algorithmic parameters.

In the first approach, different partitions of the objects under analysis may be produced by (a) employing different preprocessing and/or feature extraction mechanisms, which ultimately lead to different pattern representations (vectors, strings, graphs, correlations, dissimilarities, etc.) in different feature spaces, or dissimilarity spaces, (b) exploring subspaces of the same data representation, such as using subsets of features, or embeddings, and (c) perturbing the data, such as in bootstrapping techniques (like bagging), or sampling approaches, as, for instance, using a set of prototype samples to represent huge data sets.

In the second approach, we can generate clustering ensembles by (i) applying different clustering algorithms, exploring different concepts of clustering structure, (ii) using the same clustering algorithm with different parameters or initializations, and (iii) exploring different dissimilarity measures for evaluating inter-pattern relationships, within a given clustering algorithm.

A combination of these two main mechanisms for producing clustering ensembles leads to exploration of distinct views of inter-pattern relationships. From a computational perspective, clustering results produced in an "independent way" facilitate efficient data analysis by utilizing distributed computing, and reuse of results obtained previously.

5.4.2 Properties of the Normalized Co-association Matrix $\hat{\mathbb{C}}$

Given the overall general formulation of the EAC paradigm, the method explicitly produces as intermediate result a matrix accumulating evidence on pairwise associations. The later can be given different interpretations, as presented next.

5.4.2.1 EAC as a Kernel Method

The most direct and intuitive interpretation of the normalized co-association matrix, $\hat{\mathbb{C}}$, is as a measure of pairwise similarity between objects, as put in evidence

in pairwise associations provided by the individual clusterings in the ensemble $\mathscr{E}$. In fact, it is expected that very similar objects are very often put in a same cluster by clustering algorithms. The use of different algorithms and/or parameter configurations for each clustering algorithm enables the derivation of similarity between patterns without the use of a priori information about the number of clusters or the tuning of parameter values. As such, the EAC method, mapping the individual evidence of pairwise similarity in the clustering ensemble into a learned similarity matrix, i.e., by computing a similarity between objects, further used within some consensus clustering algorithm, can be formalized as a kernel method in unsupervised learning.

5.4.2.2 Co-associations as Pairwise Stability Indices and Multi-EAC

Data subsampling has largely been explored in clustering ensemble methods with the purpose of increasing diversity in the CE, as well as a means to handle the problem of missing data; however, it can also be used as a mechanism for data perturbation in order to evaluate the stability of clustering solutions.

When a clustering ensemble is produced by applying the same clustering algorithm (with the same parameter(s) value(s)) over subsampled versions of the original data, the matrix $\hat{\mathbb{C}}$ summarizes the replicability of clustering solutions in terms of stability of pairwise associations, measured in the interval [0; 1].

Taking as basic premise that spurious clusters generated by a clustering algorithm are not likely to be stable, the pairwise stability interpretation of $\hat{\mathbb{C}}$, under these CE construction conditions, has been explored in an extension of the EAC methodology, known as *Multi-EAC*, that incorporates diverse criteria clustering ensembles in a selective combination strategy at the cluster level, as opposed to the overall partition level. This approach has proven to better unveil the intrinsic data organization in the learned pairwise similarity [12], leading to better consensus clustering solutions [27].

5.4.2.3 $\hat{\mathbb{C}}$ as a Pairwise Probability Estimator

Let us denote by X_{ij} a random variable indicating if objects i and j belong to the same cluster. X_{ij} is a Bernoulli distributed r.v. with parameter $\theta_{ij} = E(X_{ij})$:

$$X_{ij} = \begin{cases} 1 & \text{with probability } \theta_{ij}, \\ 0 & \text{with probability } (1 - \theta_{ij}). \end{cases} \tag{5.3}$$

For each pair of objects i and j, we collect from $\mathscr{E}$, the clustering ensemble, N_{ij} independent realizations $x_{ij}^{(u)}$ of X_{ij}, given by

$$x_{ij}^{(u)} = \begin{cases} 1 \quad \text{if } p_i^{(u)} = p_j^{(u)} \\ (\text{objects } i \text{ and } j \text{ have the same cluster label in partition } \mathscr{P}_u), \\ 0 \quad \text{otherwise}, \end{cases} \tag{5.4}$$

for $u \in \Omega_{ij}$. The maximum likelihood (ML) estimate $\hat{\theta}_{ij}$ of the parameter θ_{ij} of each r.v. X_{ij} is given by the empirical mean $\bar{x}_{ij}$, i.e.,

$$\hat{\theta}_{ij} = \bar{x}_{ij} = \frac{1}{N_{ij}} \sum_{u \in \Omega_{ij}} x_{ij}^{(u)} \equiv \frac{c_{ij}}{N_{ij}} \equiv \hat{c}_{ij}. \tag{5.5}$$

Thus, the normalized co-association matrix, $\hat{\mathbb{C}}$, corresponds to the maximum likelihood estimate of the probability of pairs of objects being in the same cluster, as assessed by the clustering committee $\mathscr{E}$.

5.4.3 From Co-associations to Consensus Clustering

As delineated in Sect. 5.4, consensus clustering solutions are obtained by applying a clustering algorithm over the (normalized) co-association matrix. Given the possible different interpretations of the normalized co-association matrix, $\hat{\mathbb{C}}$, as described in Sect. 5.4.2, different classes of algorithms can be explored for deriving the consensus solution. We categorize them according to the underlying assumption about data representation:

- *(Dis)similarity-based Data Representation* The interpretation of $\hat{\mathbb{C}}$ as a similarity representation of objects, where intrinsic structure is enhanced through the evidence accumulation process, enables the determination of consensus partitions through a variety of clustering algorithms that explicitly use similarities as input, such as in graph-based techniques (e.g., hierarchical linkage methods). Examples of these have largely been explored in the literature, as in the seminal work [9].
- *Vector-based Object Description* The consensus matrix $\hat{\mathbb{C}}$ can also be used as data, rather than as similarity, each line i in the matrix corresponding to a feature vector representation of object i, as its similarity to all objects in the data set. It has been noted [23] that consensus solutions based on this interpretation of $\hat{\mathbb{C}}$ often lead to better results, as compared to similarity-based counterparts.
- *Co-occurrence Probability* The probabilistic interpretation of matrix $\hat{\mathbb{C}}$ as a ML estimate of the probability of pairs of objects being in the same cluster forms the basis of a new class of probabilistic consensus clustering solutions. Starting from the observation that co-occurrences are a special type of dyads, the work in [29] proposes a generative aspect model for dyadic data, as for the normalized co-association matrix; building on the framework of learning from dyadic data by statistical mixture models [16], the authors further explore this generative model for devising consensus clustering solutions under the EAC paradigm. Assuming a multi-labeling framework, where each object has an (unknown) probability of being assigned to each cluster, and exploring $\hat{\mathbb{C}}$ as empirical co-association matrix, the work in [34] formalizes the problem of consensus clustering as an optimization in probability domain, thus obtaining the soft class assignments. The later basic probabilistic formulation is further explored in Sect. 5.6, proposing a new objective function and optimization mechanism.

The above similarity-based and vector-based data descriptors interpretations of co-associations can be explored as input for a clustering algorithm to extract the consensus solution. In addition, they can be seen as data representations in high dimensional spaces, the structure of interest possibly being better described on an embedded manifold. This leads to the application of embedding techniques over the matrix $\hat{\mathbb{C}}$, as an additional intermediate step in the process of deriving a consensus clustering. This approach was first put forward in [1], being further explored in Sect. 5.5.

5.5 Finding Consensus Data Partitions by Exploring Embeddings

We propose to apply embedding methods, also called dimensionality reduction (DR) methods, over the normalized co-association matrix, $\hat{\mathbb{C}}$, interpreting it in two ways: (i) as a feature space, and (ii) as a similarity space. In the first case, we reduce the dimensionality of the feature space; in the second case, we obtain a representation constrained to the similarity matrix $\hat{\mathbb{C}}$. The overall consensus clustering method, hereafter named as DR-EAC, produces consensus solutions by applying a clustering algorithm over the embedded space.

5.5.1 Embedding Methods

In the following, we assume that objects are represented in d-dimensional feature spaces, a data set being represented by the matrix $\mathbf{O}$. The goal is to find a new data representation, $\mathbf{X}$, assuming that the data of interest lie on an embedded linear or nonlinear manifold within the higher-dimensional space. To perform embeddings we will use several unsupervised dimensionality reduction (DR) methods, namely Locality Preserving Projections (LPP) [14], Neighborhood Preserving Projections (NPE) [15], Sammon's mapping [36], Curvilinear Component Analysis (CCA) [6], Isomap [40], Curvilinear Distance Analysis (CDA) [25], Locally Linear Embedding (LLE) [35] and Laplacian Eigenmap (LE) [3] (see Chaps. 2, 6 and 7 for other approaches). We now briefly introduce each of these algorithms.

5.5.1.1 Nonlinear Methods

Locally Linear Embedding (LLE) The working hypothesis of LLE [35] is that the data manifold is smooth and sampled densely enough such that, in the neighborhood of each data point, the manifold can be well approximated by its tangent hyperplane. This hyperplane will usually be dependent of the point on which one is approximating the manifold, hence the word *Locally* Linear Embedding. It should be noted that the name can be misleading—this method is nonlinear.

LLE makes a locally linear approximation of the whole data manifold; it begins by estimating a local coordinate system for each object i, represented by the vector $\mathbf{o}_i$, from its k-nearest neighbors. To produce the embedding, LLE finds low-dimensional coordinates that preserve the previously estimated local coordinate systems as well as possible.

Technically, LLE first minimizes the reconstruction error $e(\mathbf{W}) = \sum_i \|\mathbf{o}_i - \sum_j w_{ij}\mathbf{o}_j\|^2$ with respect to the coefficients w_{ij}, under the constraints that $w_{ij} = 0$ if i and j are not neighbors, and $\sum_j w_{ij} = 1$. After finding these weights, the low-dimensional configuration of points is next found by minimizing $e(\mathbf{X}) = \sum_i \|\mathbf{x}_i - \sum_j w_{ij}\mathbf{x}_j\|^2$ with respect to the low-dimensional representation $\mathbf{x}_i$ of each object.

Laplacian Eigenmap (LE) The *Laplacian Eigenmap* [3] uses a graph embedding approach. It begins by constructing a graph where each data point is a node, and each node is connected to k other nodes corresponding to the k nearest neighbors of that point. Points i and j are connected by an edge with weight $w_{ij} = 1$ if j is among the k nearest neighbors of i, otherwise the edge weight is set to zero; this simple weighting method has been found to work well in practice [3].

To find a low-dimensional embedding of the graph, the algorithm tries to put points that are connected in the graph as close to each other as possible and does not care about what might happen to the other points.

Technically, LE minimizes $\sum_{i,j} \|\mathbf{x}_i - \mathbf{x}_j\|^2 w_{ij} = \mathrm{Tr}(\mathbf{X}^T\mathbf{L}\mathbf{X})$ with respect to the low-dimensional object representations $\mathbf{x}_i$, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian and $\mathbf{D}$ is a diagonal matrix with elements $\mathbf{d}_{ii} = \sum_j w_{ij}$. This cost function has an undesirable trivial solution: having all points in the same position would have a cost of zero, which would be a global minimum of the cost function. To avoid this problem, the low-dimensional configuration is found by solving the generalized eigenvalue problem $\mathbf{L}\mathbf{x}_i = \lambda_i\mathbf{D}\mathbf{x}_i$ [3]. The smallest eigenvalue corresponds to the trivial solution, but the next smallest eigenvalues yield the desired LE solution ($\mathbf{X}$ being the matrix with the corresponding eigenvectors).

Isomap *Isomap* [40] is a variant of Multidimensional Scaling (MDS) [24], which attempts to find output coordinates that match a given distance matrix. This distance matrix is not computed using simple Euclidean distances; instead, *geodesic distances* along the manifold of the data are used.[1]

Given these geodesic distances, the output coordinates are found by standard linear MDS.

Let $\mathbf{o}_i$ and $\mathbf{x}_i$ denote the coordinates of point i on the input (high-dimensional) space and output (low-dimensional) space, respectively. MDS attempts to find the $\mathbf{x}_i$ for all i which minimizes the squared difference between distances in the input space and output space: $\sum_{i,j}(d(\mathbf{o}_i, \mathbf{o}_j) - d(\mathbf{x}_i, \mathbf{x}_j))^2$. In simple terms, MDS is attempting to find the low-dimensional representation of the data which makes the distances between data points as close as possible to the distances in the original space.

[1] Technically, these distances are computed along a graph formed by connecting all k-nearest neighbors.

Curvilinear component analysis CCA [6] is a variant of MDS [24] that tries to preserve only distances between points that are near each other in the embedding. This is achieved by weighting each term in the MDS cost function by a coefficient that depends on the corresponding pairwise distance in the embedding; this coefficient is simply 1 if the distance is below a predetermined threshold and 0 if it is larger. This approach is similar to Isomap, but the determination of whether two points are neighbors is done in the output space in CCA, rather than in the input space as in Isomap.

Curvilinear distance analysis CDA [25] is a variant of CCA. Whereas MDS measures distances in the original space using the Euclidean distance, in CDA distances in the original space are measured with geodesic distances, like in Isomap. In all other aspects, CDA is similar to CCA.

5.5.1.2 Linear Methods

Locality Preserving Projections LPP [14] is a linear dimensionality reduction method which attempts to preserve local neighborhood information. It shares many properties of nonlinear techniques such as Laplacian Eigenmaps or Locally Linear Embedding, since it is a linear approximation of the nonlinear Laplacian Eigenmaps.

Neighborhood Preserving Projections NPE [15] is a linear dimensionality reduction method that preserves the local structure of the data. It has similar properties to LPP, but it is a linear approximation of Locally Linear Embedding (LLE).

5.5.2 The DR-EAC Method

We now present the proposed methodology called *Dimensionality Reduction in Evidence Accumulation Clustering* (DR-EAC). It extends the three step EAC method described previously (see Sect. 5.4) with an additional intermediate step: instead of applying a clustering algorithm directly to the normalized co-association matrix, we apply a DR technique to it. As detailed below, we propose two ways to do this, depending on how one interprets the co-association matrix. This DR technique outputs a low-dimensional data representation, which is then fed into a clustering algorithm, deriving the consensus partition. The DR-EAC method is thus summarized in the following four steps:

DR-EAC

1. *Build the clustering ensemble* $\mathscr{E}$. As discussed before (see Sect. 5.4.1), this can be accomplished in a variety of ways.
2. *Obtain the normalized co-association matrix,* $\hat{\mathbb{C}}$, as per expression (5.2)—see Sect. 5.3. Then, we interpret this matrix in one of two possible ways (see Sect. 5.4.2):

- *Co-associations viewed as Features:* the ith row of $\hat{\mathbb{C}}$ represents a new set of features for the ith object, an idea originally proposed by Kuncheva et al. [13]. Each object is now represented by the percentage of times it was grouped together with each of the other objects.
- *Co-associations viewed as Similarities.* Since many DR methods can take as input a matrix of pairwise distances (or dissimilarities), if we transform this similarity matrix $\hat{\mathbb{C}}$ into a matrix of dissimilarities $\mathbb{D}$, we can exploit this property. Since the elements of $\hat{\mathbb{C}}$ take values in the interval $[0, 1]$, we use a very simple transformation: the new dissimilarity matrix $\mathbb{D}$ has the element d_{ij} given by $1 - \hat{c}_{ij}$.

3. *Apply Dimensionality Reduction techniques.* We apply DR techniques, according to either of the interpretations above, to obtain a new representation of the data, preserving the topology of the original data.
4. *Extract the consensus partition.* After we get the embedded data, we apply a clustering algorithm to the later in order to extract the consensus solution.

For the DR methods, in step 3, we need to choose a target dimension to reduce the data to and, in some cases, we also have to choose a parameter of the method (usually the number of nearest neighbors to consider). The target dimension is chosen using a Maximum Likelihood Estimator [26]. This MLE assumes that the data points follow a Poisson process (i.e., they are drawn independently from a uniform distribution over the data manifold) and constructs hyperspheres of growing radii r. It then checks how quickly the number of neighbors inside that hypersphere grows with r; this dependence conveys information about the intrinsic dimension of the data.

For example, if the data lies on a 2-dimensional manifold, the number of neighbors inside a hypersphere of radius r should grow approximately with r^2, even if the input space has a higher dimension $d \gg 2$.

In all cases, we let each algorithm choose the most suitable parameter of the DR method by an intrinsic criterion. This intrinsic criterion can be the value of the cost function that each algorithm has to minimize, or the reconstruction error. For example, in Isomap we chose the parameter (which is the number of nearest neighbors used to construct a graph) which minimizes the residual variance [40]. It is beyond the scope of this chapter to detail how these parameters should be chosen; the relevant information can be found in the references cited in Sect. 5.5.1.

5.6 PEACE: Probabilistic Evidence Accumulation for Clustering Ensembles

In this section, we propose a probabilistic formulation and solution of the consensus clustering extraction that fully exploits the probabilistic interpretation of the normalized co-association matrix, $\hat{\mathbb{C}}$, presented in Sect. 5.4.2.3.

5.6.1 Problem Formulation

Consider a general probabilistic multi-labeling framework, where each object has an (unknown) probability of being assigned to each cluster. Define the vector

$$\mathbf{y}_i = [y_{1i}, \ldots, y_{Ki}]^T \in \Delta_K \tag{5.6}$$

representing the probability distribution over the set of class labels $\{1, \ldots, K\}$ which characterizes object $i \in \mathscr{O}$, that is, $y_{ki} = \Pr(i \in \mathscr{C}_k)$, where $\mathscr{C}_k$ denotes the kth cluster. Let $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_n] \in \Delta_K^n$ be a $K \times n$ matrix collecting all objects class labels probability distributions.

In our model, we assume that objects are assigned to clusters independently, i.e., $\Pr(i \in \mathscr{C}_k, j \in \mathscr{C}_k) = \Pr(i \in \mathscr{C}_k)\Pr(j \in \mathscr{C}_k)$. Following this independence assumption and definition (5.6), the probability of objects i and j being assigned to the same cluster is given by

$$\sum_{k=1}^{K} \Pr(i \in \mathscr{C}_k, j \in \mathscr{C}_k) = \sum_{k=1}^{K} y_{ki}\, y_{kj} = \mathbf{y}_i^\top \mathbf{y}_j. \tag{5.7}$$

Let C_{ij} be a binomial random variable (r.v.) representing the number of times that objects i and j are co-clustered; from the modeling assumptions above, we have that $C_{ij} \sim \text{Binomial}(N_{ij}, \mathbf{y}_i^\top \mathbf{y}_j)$, that is,

$$\Pr(C_{ij} = c \mid \mathbf{y}_i, \mathbf{y}_j) = \binom{N_{ij}}{c} \left(\mathbf{y}_i^\top \mathbf{y}_j\right)^c \left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right)^{N_{ij}-c}.$$

Each element c_{ij} of the co-association matrix $\mathbb{C}$ is interpreted as a sample of the r.v. C_{ij}, and the different C_{ij}'s are all assumed independent. Consequently, the probability of observing $\mathbb{C}$, given the class probabilities $\mathbf{Y}$, is given by

$$\Pr(\mathbb{C} \mid \mathbf{Y}) = \prod_{\substack{i,j \in \mathscr{O} \\ i \neq j}} \binom{N_{ij}}{c_{ij}} \left(\mathbf{y}_i^\top \mathbf{y}_j\right)^{c_{ij}} \left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right)^{N_{ij}-c_{ij}}.$$

We therefore formulate the probabilistic consensus clustering problem as an estimation of the unknown class assignments $\mathbf{Y}$, by maximizing the log-likelihood $\log \Pr(\mathbb{C} \mid \mathbf{Y})$ with respect to $\mathbf{Y}$. This yields the following maximization problem

$$\widehat{\mathbf{Y}} = \arg\max_{\mathbf{Y} \in \Delta_K^n} f(\mathbf{Y}), \tag{5.8}$$

where

$$f(\mathbf{Y}) = \sum_{\substack{i,j \in \mathscr{O} \\ i \neq j}} c_{ij} \log\left(\mathbf{y}_i^\top \mathbf{y}_j\right) + (N_{ij} - c_{ij}) \log\left(1 - \mathbf{y}_i^\top \mathbf{y}_j\right) \tag{5.9}$$

(constant terms have been dropped).

It's interesting to notice that $f(\mathbf{Y})$ can be written in terms of the Kullback–Leibler divergence $D_{\mathrm{KL}}(\cdot \parallel \cdot)$ as

$$f(\mathbf{Y}) = -\sum_{\substack{i,j \in \mathscr{O} \\ i \neq j}} N_{ij}\big[H(\mathbf{z}_{ij}) + D_{\mathrm{KL}}\big(\mathbf{z}_{ij} \parallel \mathbf{w}_{ij}(\mathbf{Y})\big)\big],$$

where $\mathbf{z}_{ij} = (c_{ij}/N_{ij}, 1 - (c_{ij}/N_{ij}))^\top \equiv (\hat{c}_{ij}, 1 - \hat{c}_{ij})^\top \in \Delta_2$, $\mathbf{w}_{ij}(\mathbf{Y}) = (\mathbf{y}_i^\top \mathbf{y}_j, 1 - \mathbf{y}_i^\top \mathbf{y}_j)^\top \in \Delta_2$, $\hat{c}_{ij}$ are elements of the normalized co-association matrix $\hat{\mathbb{C}}$, and $H(\cdot)$ is the entropy.

5.6.2 Optimization Algorithm

The optimization method described in this chapter belongs to the class of primal line-search procedures. This method iteratively finds a direction which is *feasible*, i.e., satisfying the constraints, and *ascending*, i.e., guaranteeing a (local) increase of the objective function, along which a better solution is sought. The procedure is iterated until it converges, or a maximum number of iterations is reached.

The first part of this section describes the procedure to determine the search direction in the optimization algorithm. The second part is devoted to determining an optimal step size to be taken in the direction found.

5.6.2.1 Computation of a Search Direction

Consider the Lagrangian of (5.8):

$$\mathscr{L}(\mathbf{Y}, \boldsymbol{\lambda}, \mathbf{M}) = f(\mathbf{Y}) + \mathrm{Tr}\big(\mathbf{M}^\top \mathbf{Y}\big) - \boldsymbol{\lambda}^\top \big(\mathbf{Y}^\top \mathbf{1}_K - \mathbf{1}_n\big),$$

where $\mathbf{M} = (\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_n) \in \mathbb{R}_+^{K \times n}$ and $\boldsymbol{\lambda} \in \mathbb{R}^n$ are the Lagrangian multipliers (related to positiveness and simplex constraints), and $\mathbf{Y} \in \mathrm{dom}(f)$. By differentiating $\mathscr{L}$ with respect to $\mathbf{y}_i$ and λ and considering the complementary slackness conditions, we obtain the first order Karush–Kuhn–Tucker (KKT) conditions [30] for local optimality:

$$\begin{cases} g_i(\mathbf{Y}) - \lambda_i \mathbf{1}_n + \boldsymbol{\mu}_i = \mathbf{0}, & \forall i \in \mathscr{O}, \\ \mathbf{Y}^\top \mathbf{1}_K - \mathbf{1}_n = \mathbf{0}, \\ \mathrm{Tr}(\mathbf{M}^\top \mathbf{Y}) = 0, \end{cases} \tag{5.10}$$

where $g_i(\mathbf{Y})$, the partial derivative of $f(\mathbf{Y})$ with respect to $\mathbf{y}_i$, is given by

$$g_i(\mathbf{Y}) = \sum_{j \in \mathscr{O} \setminus \{i\}} c_{ij} \frac{\mathbf{y}_j}{\mathbf{y}_i^\top \mathbf{y}_j} - (N_{ij} - c_{ij}) \frac{\mathbf{y}_j}{1 - \mathbf{y}_i^\top \mathbf{y}_j},$$

and $\mathbf{1}_n$ denotes a n-dimensional column vector of all 1's. We can express the Lagrange multipliers $\boldsymbol{\lambda}$ in terms of $\mathbf{Y}$ by noting that

$$\mathbf{y}_i^\top \left[g_i(\mathbf{Y}) - \lambda_i \mathbf{1}_n + \boldsymbol{\mu}_i \right] = 0,$$

yields $\lambda_i = \mathbf{y}_i^\top g_i(\mathbf{Y})$ for all $i \in \mathscr{O}$.

Let $r_i(\mathbf{Y})$ be given as

$$r_i(\mathbf{Y}) = g_i(\mathbf{Y}) - \lambda_i \mathbf{1}_K = g_i(\mathbf{Y}) - \mathbf{y}_i^\top g_i(\mathbf{Y}) \mathbf{1}_K,$$

and let $\sigma(\mathbf{y}_i)$ denote the support of $\mathbf{y}_i$, i.e., the set of indices corresponding to (strictly) positive entries of $\mathbf{y}_i$. An alternative characterization of the KKT conditions, where the Lagrange multipliers do not appear, is

$$\begin{cases} [r_i(\mathbf{Y})]_k = 0, & \forall i \in \mathscr{O}, \forall k \in \sigma(\mathbf{y}_i), \\ [r_i(\mathbf{Y})]_k \leq 0, & \forall i \in \mathscr{O}, \forall k \notin \sigma(\mathbf{y}_i), \\ \mathbf{Y}^\top \mathbf{1}_K - \mathbf{1}_n = \mathbf{0}. \end{cases} \tag{5.11}$$

The two characterizations (5.11) and (5.10) are equivalent. This can be verified by exploiting the non-negativity of both matrices $\mathbf{M}$ and $\mathbf{Y}$, and the complementary slackness conditions.

The following proposition plays an important role in the selection of the search direction. Hereafter, we denote by $(\mathbf{y}_j)_k$ the kth component of cluster assignment $\mathbf{y}_j$.

Proposition 5.1 *Assume* $\mathbf{Y} \in \mathrm{dom}(f)$ *to be feasible for* (5.8), *i.e.*, $\mathbf{Y} \in \Delta_K^n \cap \mathrm{dom}(f)$. *Consider*

$$j \in \arg\max_{i \in \mathscr{O}} \left\{ \left[g_i(\mathbf{Y}) \right]_{k_i^+} - \left[g_i(\mathbf{Y}) \right]_{k_i^-} \right\},$$

where

$$k_i^+ \in \arg\max_{k \in \{1 \cdots K\}} \left[g_i(\mathbf{Y}) \right]_k \quad \textit{and}$$

$$k_i^- \in \arg\min_{k \in \sigma(\mathbf{y}_j)} \left[g_i(\mathbf{Y}) \right]_k.$$

Then the following holds:

- $[g_j(\mathbf{Y})]_{k_j^+} \geq [g_j(\mathbf{Y})]_{k_j^-}$ *and*
- $\mathbf{Y}$ *satisfies the KKT conditions for* (5.8) *if and only if* $[g_j(\mathbf{Y})]_{k_j^+} = [g_j(\mathbf{Y})]_{k_j^-}$.

Proof We prove the first point by simple derivations as follows:

$$\left[g_j(\mathbf{Y}) \right]_{k_j^+} \geq \mathbf{y}_j^\top g_j(\mathbf{Y}) = \sum_{k \in \sigma(\mathbf{y}_j)} (\mathbf{y}_j)_k \left[g_j(\mathbf{Y}) \right]_k$$

$$\geq \sum_{k \in \sigma(\mathbf{y}_j)} (\mathbf{y}_i)_k \big[g_j(\mathbf{Y})\big]_{k_j^-} = \big[g_j(\mathbf{Y})\big]_{k_j^-}.$$

By subtracting $\mathbf{y}_j^\top g_j(\mathbf{Y})$, we obtain the equivalent relation

$$\big[r_j(\mathbf{Y})\big]_{k_j^+} \geq 0 \geq \big[r_j(\mathbf{Y})\big]_{k_j^-}, \tag{5.12}$$

where equality holds if and only if $[g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^+}$.

As for the second point, assume that $\mathbf{Y}$ satisfies the KKT conditions. Then $[r_j(\mathbf{Y})]_{k_j^-} = 0$ because $k_j^- \in \sigma(\mathbf{y}_j)$. It follows by (5.12) that also $[r_j(\mathbf{Y})]_{k_j^+} = 0$ and therefore $[g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^+}$. On the other hand, if we assume that $[g_j(\mathbf{Y})]_{k_j^-} = [g_j(\mathbf{Y})]_{k_j^+}$ then, by (5.12) and by definition of j, we have that $[r_i(\mathbf{Y})]_{k_i^+} = [r_i(\mathbf{Y})]_{k_i^+} = 0$ for all $i \in \mathscr{O}$. By exploiting the definition of k_i^+ and k_i^-, it is straightforward to verify that $\mathbf{Y}$ satisfies the KKT conditions. □

Given $\mathbf{Y}$ a non-optimal feasible solution of (5.8), we can determine the indices k_j^+, k_j^- and j as stated in Proposition 5.1. The next proposition shows how to build a feasible and ascending search direction by using these indices. Later on, we will point out some desired properties of this search direction. We denote by $\mathbf{e}_n^{(j)}$ the jth column of the n-dimensional identity matrix.

Proposition 5.2 *Let $\mathbf{Y} \in \Delta_K^n \cap \mathrm{dom}(f)$ and assume that the KKT conditions do not hold. Let $\mathbf{D} = (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})(\mathbf{e}_n^j)^\top$, where j, $k^+ = k_j^+$ and $k^- = k_j^-$ are computed as in Proposition 5.1. Then, for all $0 \leq \varepsilon \leq (\mathbf{y}_j)_{k^-}$, we have that $\mathbf{Z}_\varepsilon = \mathbf{Y} + \varepsilon\mathbf{D}$ belongs to Δ_K^n, and for all small enough, positive values of ε, we have $f(\mathbf{Z}_\varepsilon) > f(\mathbf{Y})$.*

Proof Let $\mathbf{Z}_\varepsilon = \mathbf{Y} + \varepsilon\mathbf{D}$. Then for any ε,

$$\mathbf{Z}_\varepsilon^\top \mathbf{1}_K = (\mathbf{Y} + \varepsilon\mathbf{D})^\top \mathbf{1}_K = \mathbf{Y}^\top \mathbf{1}_K + \varepsilon\mathbf{D}^\top \mathbf{1}_K = \mathbf{1}_n + \varepsilon\mathbf{e}_n^j \big(\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-}\big)^\top \mathbf{1}_K = \mathbf{1}_n.$$

As ε increases, only the (k^-, j)th entry of $\mathbf{Z}_\varepsilon$, which is given by $(\mathbf{y}_j)_{k^-} - \varepsilon$, decreases. This entry is nonnegative for all values of ε satisfying $\varepsilon \leq (\mathbf{y}_j)_{k^-}$. Hence, $\mathbf{Z}_\varepsilon \in \Delta_K^n$ for all positive values of ε not exceeding $(\mathbf{y}_j)_{k^-}$ as required.

As for the second point, the Taylor expansion of f at $\mathbf{Y}$ gives, for all small enough positive values of ε:

$$\begin{aligned} f(\mathbf{Z}_\varepsilon) - f(\mathbf{Y}) &= \varepsilon \left[\lim_{\varepsilon \to 0} \frac{d}{d\varepsilon} f(\mathbf{Z}_\varepsilon) \right] + O\big(\varepsilon^2\big) \\ &= \big(\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-}\big)^\top g_j(\mathbf{Y}) + O\big(\varepsilon^2\big) > 0 \\ &= \big[g_j(\mathbf{Y})\big]_{k^+} - \big[g_j(\mathbf{Y})\big]_{k^-} + O\big(\varepsilon^2\big) > 0. \end{aligned}$$

The last inequality comes from Proposition 5.1 because if Y does not satisfy the KKT conditions then $[g_j(\mathbf{Y})]_{k^+} - [g_j(\mathbf{Y})]_{k^-} > 0$. □

5.6.2.2 Computation of an Optimal Step Size

Proposition 5.2 provides a direction **D** that is both feasible and ascending for **Y** with respect to (5.8). We will now address the problem of determining an optimal step ε^* to be taken along the direction **D**. This optimal step is given by the following one dimensional optimization problem:

$$\varepsilon^* \in \underset{0 \leq \varepsilon \leq (\mathbf{y}_j)_{k^-}}{\arg\max} \; f(\mathbf{Z}_\varepsilon), \tag{5.13}$$

where $\mathbf{Z}_\varepsilon = \mathbf{Y} + \varepsilon \mathbf{D}$. This problem is concave as stated in the following proposition.

Proposition 5.3 *The optimization problem in* (5.13) *is concave.*

Proof The direction **D** is everywhere null except in the jth column. Since the sum in (5.9) is taken over all pairs (i, j) such that $i \neq j$ we have that the argument of every log function (which is a concave function) is linear in ε. Concavity is preserved by the composition of concave functions with linear ones and by the sum of concave functions [5]. Hence, the maximization problem is concave. □

Let $\rho(\varepsilon')$ denote the first order derivative of f with respect to ε evaluated at ε', i.e.,

$$\rho\left(\varepsilon'\right) = \lim_{\varepsilon \to \varepsilon'} \frac{d}{d\varepsilon} f(\mathbf{Z}_\varepsilon) = \left(\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-}\right)^\top g_j(\mathbf{Z}_{\varepsilon'}).$$

By the convexity of (5.13) and Kachurovskii's theorem [19], we have that ρ is non-increasing in the interval $0 \leq \varepsilon \leq (\mathbf{y}_j)_{k^-}$. Moreover, $\rho(0) > 0$ since **D** is an ascending direction as stated by Proposition 5.2. In order to compute the optimal step ε^* in (5.13), we distinguish 2 cases:

- If $\rho((\mathbf{y}_j)_{k^-}) \geq 0$ then $\varepsilon^* = (\mathbf{y}_j)_{k^-}$ for $f(\mathbf{Z}_\varepsilon)$ is non-decreasing in the feasible set of (5.13);
- If $\rho((\mathbf{y}_j)_{k^-}) < 0$ then ε^* is a zero of ρ that can be found by dichotomic search.

Suppose the second case holds, i.e., assume $\rho((\mathbf{y}_j)_{k^-}) < 0$. Then ε^* can be found by iteratively updating the search interval as follows:

$$\begin{aligned} \left(\ell^{(0)}, r^{(0)}\right) &= \left(0, (\mathbf{y}_j)_{k^-}\right), \\ \left(\ell^{(t+1)}, r^{(t+1)}\right) &= \begin{cases} (\ell^{(t)}, m^{(t)}) & \text{if } \rho(m^{(t)}) < 0, \\ (m^{(t)}, r^{(t)}) & \text{if } \rho(m^{(t)}) > 0, \\ (m^{(t)}, m^{(t)}) & \text{if } \rho(m^{(t)}) = 0, \end{cases} \end{aligned} \tag{5.14}$$

for all $t > 0$, where $m^{(t)}$ denotes the center of segment $[\ell^{(t)}, r^{(t)}]$, i.e., $m^{(t)} = (\ell^{(t)} + r^{(t)})/2$.

We are not in general interested in determining a precise step size ε^* but an approximation is sufficient. Hence, the dichotomic search is carried out until the

interval size is below a given threshold. If δ is this threshold, the number of iterations required is expected to be $\log_2((\mathbf{y}_j)_{k^-}/\delta)$ in the worst case.

5.6.2.3 Algorithm and Computational Complexity

Consider a generic iteration t of our algorithm (shown in Algorithm 1) and assume $A^{(t)} = \mathbf{Y}^\top \mathbf{Y}$ and $g_i^{(t)} = g_i(\mathbf{Y})$ given for all $i \in \mathscr{O}$, where $\mathbf{Y} = \mathbf{Y}^{(t)}$.

The computation of ε^* requires the evaluation of function ρ at different values of ε. Each function evaluation can be carried out in $O(n)$ steps by exploiting $\mathbf{A}^{(t)}$ as follows:

$$\rho(\varepsilon) = \sum_{i \in \mathscr{O} \setminus \{j\}} c_{ji} \frac{\mathbf{d}_j^\top \mathbf{y}_i}{A_{ji}^{(t)} + \varepsilon \mathbf{d}_j^\top \mathbf{y}_i} + (N_{ji} - c_{ji}) \frac{\mathbf{d}_j^\top \mathbf{y}_i}{1 - A_{ji}^{(t)} - \varepsilon \mathbf{d}_j^\top \mathbf{y}_i}, \tag{5.15}$$

where $\mathbf{d}_j = (\mathbf{e}_K^{k^+} - \mathbf{e}_K^{k^-})$. The complexity of the computation of the optimal step size is thus $O(n\gamma)$ where γ is the average number of iterations needed by the dichotomic search.

Next, we can efficiently update $\mathbf{A}^{(t)}$ as follows:

$$\mathbf{A}^{(t+1)} = \left(\mathbf{Y}^{(t+1)}\right)^\top \mathbf{Y}^{(t+1)} = \mathbf{A}^{(t)} + \varepsilon^* \left(\mathbf{D}^\top \mathbf{Y}^{(t)} + {\mathbf{Y}^{(t)}}^\top \mathbf{D} + \varepsilon^* \mathbf{D}^\top \mathbf{D}\right). \tag{5.16}$$

Indeed, since $\mathbf{D}$ has only two nonzero entries, namely (k^-, j) and (k^+, j), the terms within parenthesis can be computed in $O(n)$.

The computation of $\mathbf{Y}^{(t+1)}$ can be performed in constant time by exploiting the sparsity of $\mathbf{D}$ as $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \varepsilon^* \mathbf{D}$.

The computation of $g_i^{(t+1)} = g_i(\mathbf{Y}^{(t+1)})$ for each $i \in \mathscr{O} \setminus \{j\}$ can be efficiently accomplished in constant time (it requires $O(nK)$ to update all of them) as follows:

$$\begin{aligned} g_i^{(t+1)} = g_i^{(t)} &+ c_{ij} \left(\frac{\mathbf{y}_j^{(t+1)}}{A_{ij}^{(t+1)}} - \frac{\mathbf{y}_j^{(t)}}{A_{ij}^{(t)}} \right) \\ &+ (N_{ij} - c_{ij}) \left(\frac{\mathbf{y}_j^{(t+1)}}{1 - A_{ij}^{(t+1)}} - \frac{\mathbf{y}_j^{(t)}}{1 - A_{ij}^{(t)}} \right). \end{aligned} \tag{5.17}$$

The complexity of the computation of $g_j^{(t+1)}$, on the other hand, requires $O(nK)$ steps:

$$g_j^{(t+1)} = \sum_{i \in \mathscr{O} \setminus \{j\}} c_{ji} \frac{\mathbf{y}_i^{(t+1)}}{A_{ji}^{(t+1)}} - (N_{ji} - c_{ji}) \frac{\mathbf{y}_i^{(t+1)}}{1 - A_{ji}^{(t+1)}}. \tag{5.18}$$

By iteratively updating the quantities $A^{(t)}$, $g_i^{(t)}$ and $Y^{(t)}$ according to the aforementioned procedures, we can keep a per-iteration complexity of $O(nK)$, that is linear in the number of variables in $\mathbf{Y}$.

Algorithm 1: PEACE

Require: $\mathscr{E}$: ensemble of clusterings
Require: $\mathbf{Y}^{(0)} \in \Delta_K^n \cap \text{dom}(f)$: starting distribution
  Compute $\mathbb{C}$ and $\{N_{ij}\}$ from $\mathscr{E}$
  Initialize $\mathbf{A}_i^{(0)} \leftarrow (\mathbf{Y}^{(0)})^\top \mathbf{Y}^{(0)}$
  Initialize $g_i^{(0)} \leftarrow g_i(\mathbf{Y}^{(0)})$ for all $i \in \mathscr{O}$, as per Eq. (5.17)
  $t \leftarrow 0$
  while termination-condition **do**
    Compute k^+, k^-, j as in Proposition 5.1
    Compute $\mathbf{D}$ as in Proposition 5.2
    Compute ε^* as described in Sect. 5.6.2.2/5.6.2.3
    Update $\mathbf{A}^{(t+1)}$ as per Eq. (5.16)
    Update $\mathbf{Y}^{(t+1)} = \mathbf{Y}^{(t)} + \varepsilon^* \mathbf{D}$
    Update $g_i^{(t+1)}$ as per Eq. (5.17)
    Update $g_j^{(t+1)}$ as per Eq. (5.18)
    $t \leftarrow t + 1$
  end while
  return $\mathbf{Y}^{(t)}$

Iterations stop when the KKT conditions of Proposition 5.1 are satisfied under a given tolerance τ, i.e., $([g_j(\mathbf{Y})]_{k_j^+} - [g_j(\mathbf{Y})]_{k_j^-}) < \tau$.

5.7 Results and Discussion

We evaluated the previous methods on both real and synthetic datasets, in a comparative study with the EAC method. In the later, we explored three hierarchical algorithms for the computation of the consensus solution from the normalized co-association matrix, namely *single-link* (SL), *average link* (AL), and *Wards link* (WL). In this study, we assume known the true number of clusters, K. In order to assess the quality of consensus results, we compute the *consistency index* (CI) between the consensus partition and the ground-truth labeling of the data. The consistency index, also called H index [31], gives the accuracy of the obtained partitions and is obtained by matching the clusters in the consensus partition $\mathscr{P}^i$ with the ground truth partition $\mathscr{P}^{\text{GT}}$:

$$\text{CI}\left(\mathscr{P}^i, \mathscr{P}^{\text{GT}}\right) = \frac{1}{n} \sum_{k'=\text{match}(k)} m_{k,k'}, \tag{5.19}$$

where $m_{k,k'}$ denotes the contingency table, i.e., $m_{k,k'} = |\mathscr{C}_k^{(i)} \cap \mathscr{C}_{k'}^{(\text{GT})}|$. It corresponds to the percentage of correct labelings when the number of clusters in $\mathscr{P}^i$ and $\mathscr{P}^{\text{GT}}$ is the same.

5.7.1 Experimental Setup

We conducted experiments on synthetic datasets (see Fig. 5.1), and on real-world datasets from the UCI Irvine and UCI KDD Machine Learning Repository: iris, wine, house-votes, ionosphere, std-yeast-cell, breast-cancer, and optidigits. Table 5.1 summarizes the experimental setting, indicating the number of clusters, K, and the size, n, of each data set.

Two different types of clustering ensembles were created, exploring different strategies:

- $\mathscr{E}$*-Split*—implementing a split strategy [28] (splitting "natural" clusters into small clusters), the K-means was used as base clustering algorithm, with K randomly chosen in an interval $\{K_{\min}, K_{\max}\} = \{\lceil\sqrt{n}/2\rceil, \lceil\sqrt{n}\rceil\}$. The size of each CE was $N = 100$.
- $\mathscr{E}$*-Hybrid*—a combination of multiple algorithms (agglomerative hierarchical algorithms: single, average, ward, centroid link; k-means; spectral clustering [32]) with different number of clusters K_i, as specified in Table 5.1 (last column). For each clustering approach and each parametrization of the same, we generated $N = 100$ different subsampled versions of the data-set (90 % resampling percentage).

5.7.2 Clustering Results Using Embeddings

We applied the DR-EAC method to the clustering ensembles $\mathscr{E}$-Split and $\mathscr{E}$-Hybrid, in the two interpretations of the normalized co-association matrix: as similarity, hereafter denoted as *Similarity Space*; and as features, denoted as *Feature Space*. This leads to four experimental scenarios. For each scenario, we applied each of the dimensionality reduction methods described in Sect. 5.5.1, namely LPP, NPE, LLE, LE, Sammon, CCA, Isomap, and CDA. For extracting the consensus partition, we used the same three hierarchical agglomerative methods used with EAC: single-link, average-link, and Wards-link.

Figure 5.2 summarizes the overall performance of the several variants of the method, in direct comparison with EAC. In this figure, each sub-figure plots the four scenario matrices for a given DR method, as indicated at the top. For each scenario, lines correspond to data sets, and columns to the consensus extraction algorithm, SL, AL, and WL. Within each cell, a color scheme is used to code the comparative performances of the DR-EAc vs. EAC methods, as measured by the consistency index, with white corresponding to equal performance, warm color meaning a superiority of DR-EAC over EAC (in a gradient where red corresponding to high/significantly increased performance values); and cool colors (in a gradient of blue) represent a decrease in performance of DR-EAC in comparison with EAC.

Figures 5.3 and 5.4 present the best consistency index obtained for each data set (indicated on the left of each plot), and each consensus clustering method (indicated at the bottom), for the four combinations of interpretations of the matrix $\hat{\mathbb{C}}$

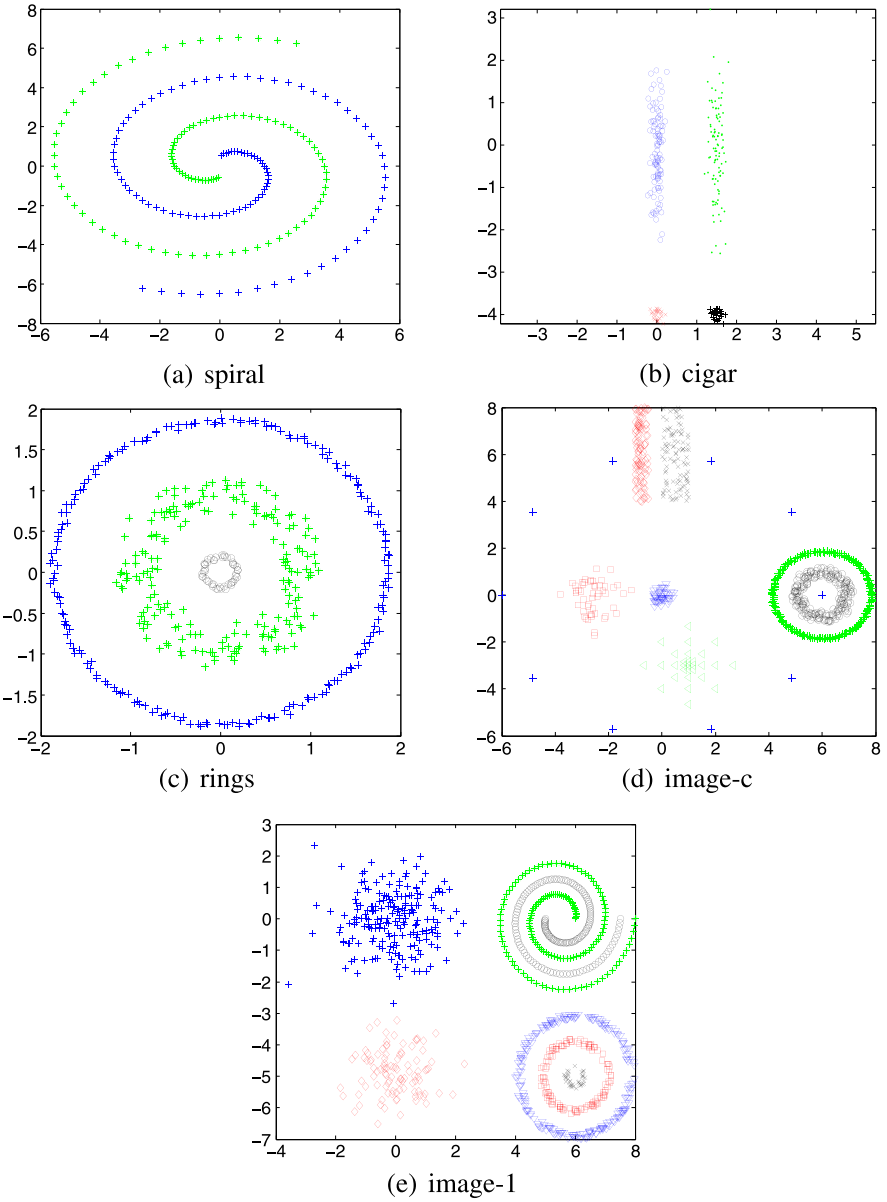


Fig. 5.1 Sketch of the synthetic data sets

and clustering ensemble types. For the DR-EAC method, the variant associated with the DR method is indicated by the corresponding DR designation. On each cell, the best consistency index value obtained by comparing results from the three clustering extraction algorithms is shown over a background color that reveals the wining

Table 5.1 Benchmark datasets and K_i parameter values for the clustering ensembles $\mathscr{E}$-Hybrid

Data-Sets	K	n	K_i—Ensemble
spiral	2	200	2–9
cigar	4	250	4–9
rings	3	450	2–6
image-c	7	739	8–15,20
image-1	8	1000	7–15,20
iris	3	150	3–10
wine	3	178	4–10,15,20
house-votes	2	232	4–8
ionsphere	2	351	4–10
std-yeast-cell	5	384	5–10
breast-cancer	2	683	2–10
optdigits	10	1000	10, 12, 15, 20

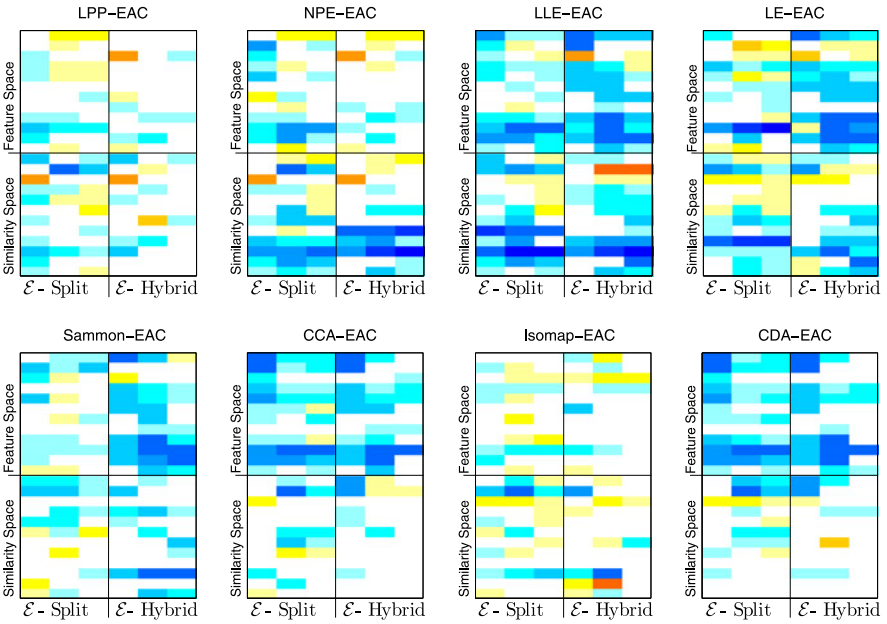
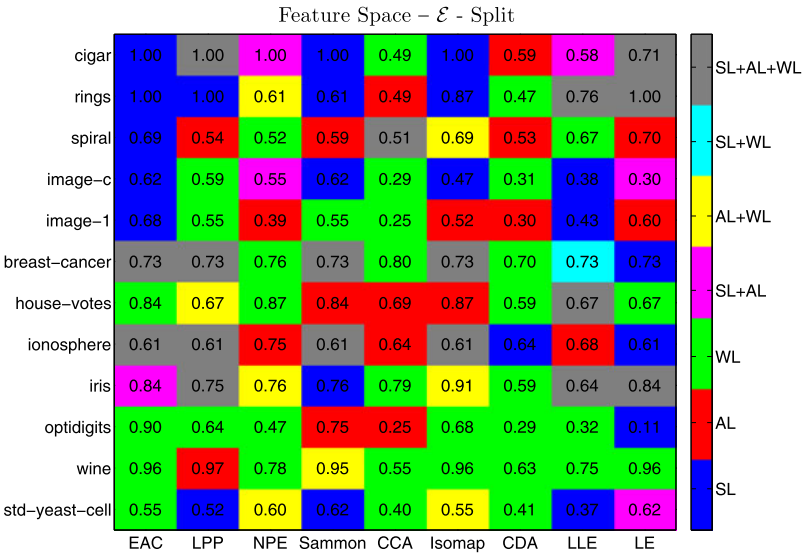
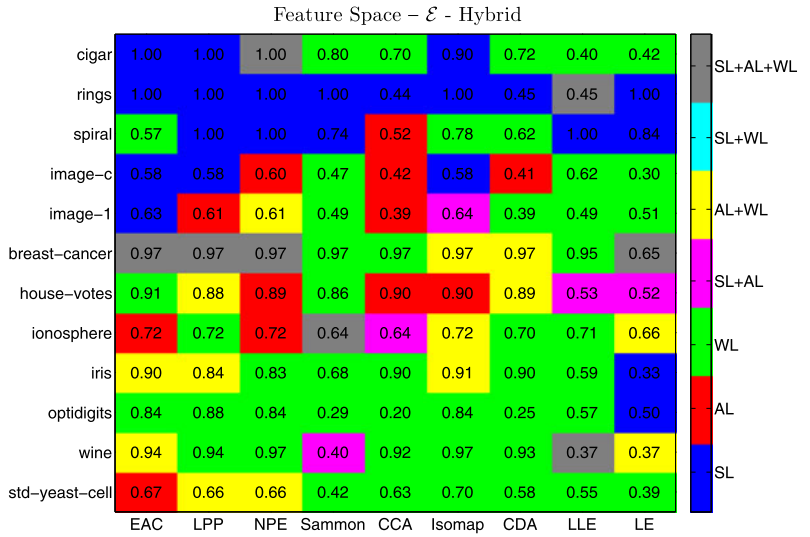


Fig. 5.2 Comparison of various DR methods with EAC using the consistency index. The *top-left sub-figure*, labeled "LPP-EAC", compares the DR method LPP with the EAC baseline. Four scenarios are depicted in this sub-figure: feature space vs. similarity space and $\mathscr{E}$-Split vs. $\mathscr{E}$-Hybrid. Each of the four scenarios presents a 12×3 matrix, corresponding to the 12 datasets and the 3 clustering methods in the following order: SL, AL, and WL. A white cell means that LPP and EAC yielded roughly the same performance. *Warm colors* mean that LPP yielded better performance, whereas *cool colors* mean that it yielded worse performance. Darker tones mean that the difference between the two methods was larger in absolute value. The other *seven sub-figures* show similar information for the seven remaining DR methods

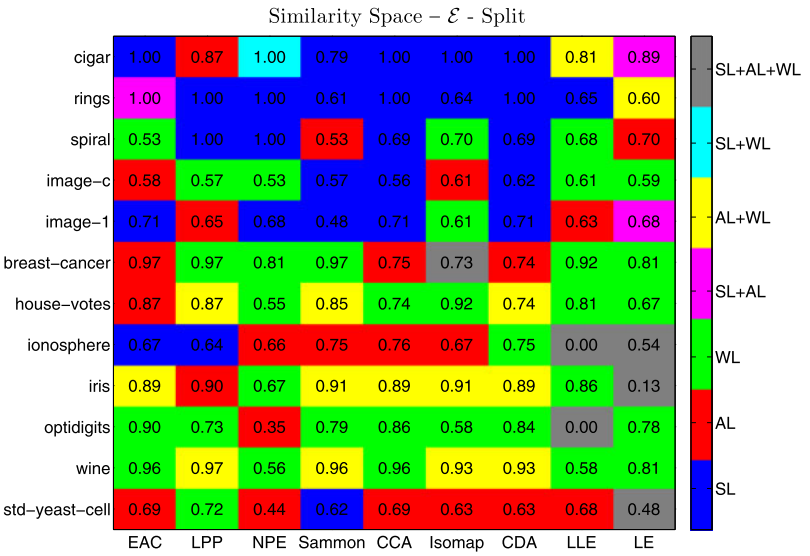


(a) Results for CE $\mathcal{E}$-Split

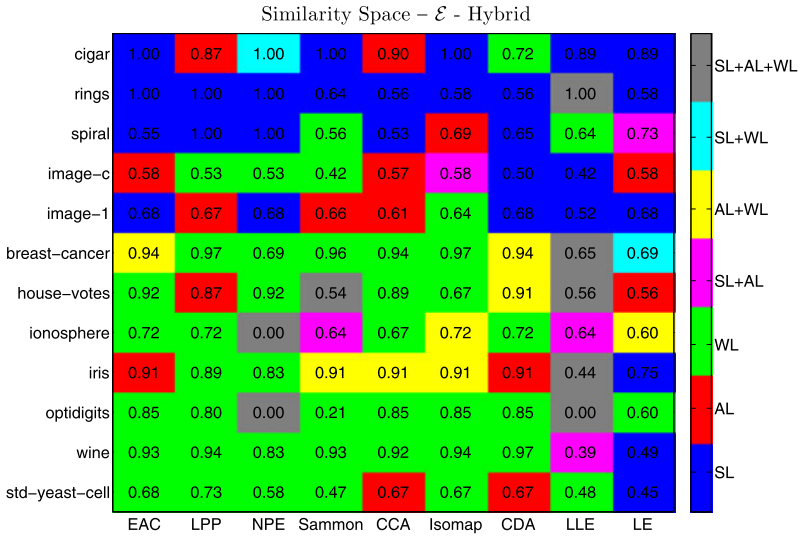


(b) Results for CE $\mathcal{E}$-Hybrid

Fig. 5.3 Results on the feature spaces. (*Top*) Consistency index for $\mathcal{E}$-Split for each dataset (*vertical axis*), DR method (*horizontal axis*), for the best clustering method (*color*). Each *cell* shows the value of the best consistency index obtained for the corresponding dataset and DR method out of the three clustering algorithms tested. A *blue cell* indicates that the best value came from using single-link, a *red cell* corresponds to average-link, and a *green cell* to Ward-link. Color addition is used to present ties: if both single-link and average-link yielded the maximum value, that cell is shown in magenta, etc. (*Bottom*) Same as before, but for $\mathcal{E}$-Hybrid



(a) Results for CE $\mathscr{E}$-Split



(b) Results for CE $\mathscr{E}$-Hybrid

Fig. 5.4 Results on the similarity spaces. The meaning of plots are as in Fig. 5.3

algorithm, according to the color scheme presented on the right of each figure. In addition, for each data set, we circle the best consensus clustering result obtained over the four combinations of spaces interpretations and CEs, as plotted in Figs. 5.3(a), 5.3(b), 5.4(b), and 5.4(a).

Figure 5.2 yields some interesting conclusions. The most immediate one is that blindly performing DR is a bad idea since there are many more blue–cyan cells than orange–yellow ones, randomly choosing a DR method for a certain dataset and clustering method is likely to decrease the performance. However, this should not discourage us from using DR. In fact, for some cases the improvement in the results is considerable, such as for certain cells of LPP in $\mathscr{E}$-Split.

Overall, Isomap is the method that more consistently produced better results than EAC (notice the high percentage of positive colors), with rare situations of (mild) decreased performance; however, improvements are also in general moderate to low (there are large white areas). LPP is a method that in general leads to good results; improvements are in some instances quite significant, as indicated in reddish tones. This is further corroborated by the analysis of Figs. 5.3 and 5.4, where we can notice the high number of best CI scores obtained for instance in the combination of the Similarity Space with the $\mathscr{E}$-Split CE (see Fig. 5.4(a)).

LLE, except for point-wise situations, is the method that overall performed worse, immediately followed by LE, with many dark blue areas.

CCA and CDA perform poorly on the feature space, having a more adequate behavior on the similarity space, in particular with the $\mathscr{E}$-Split CEs. This can be further observed in Fig. 5.4(a).

NPE is better suited for data with complex structure, namely the synthetic data sets; it nevertheless performs reasonably well on real data, in particular on $\mathscr{E}$-Hybrid CEs. Sammon mapping, on the other hand, performs better with $\mathscr{E}$-Split CEs, achieving moderate improvements.

Concerning best obtained results per data set and embedding method (Figs. 5.3 and 5.4), it is clear the overall better performance of the single-link algorithm for the extraction of the combined partition over the synthetic data sets (see the large areas of blue, pink and brown on all maps, in particular on the similarity space).

On the other hand, the Wards-link was the best performing method on the real data (green, yellow and brown areas).

5.7.3 Probabilistic Clustering

For each data set, the PEACE algorithm was applied to the clustering ensembles $\mathscr{E}$-Split and $\mathscr{E}$-Hybrid, leading to corresponding probabilistic cluster assignments.

Figure 5.5 illustrates the empirical co-association matrices, $\hat{\mathbb{C}}$, and corresponding estimated co-occurrences probabilities, $\mathbf{Y}^\top\mathbf{Y}$, on both clustering ensembles, for the iris dataset. In these images, $\hat{c}_{ij}$ values are represented in a gradient of colors from dark blue (corresponding to 0) to red (corresponding to 1). While a block structure of three clusters is apparent in all figures, it is more clear and less noisy in the true co-association $\mathbf{Y}^\top\mathbf{Y}$. The corresponding soft cluster assignments, $\mathbf{Y}$, are plotted in Fig. 5.6, were object indices are on the x-axis, and probabilities for each cluster assignment (on the y-axis) are given in color, in a gradient from dark blue to red.

For the direct comparison with the ground-truth hard-partition, $\mathscr{P}^{\mathrm{GT}}$, the probabilistic consensus clusterings are converted into hard-partitions by assigning each

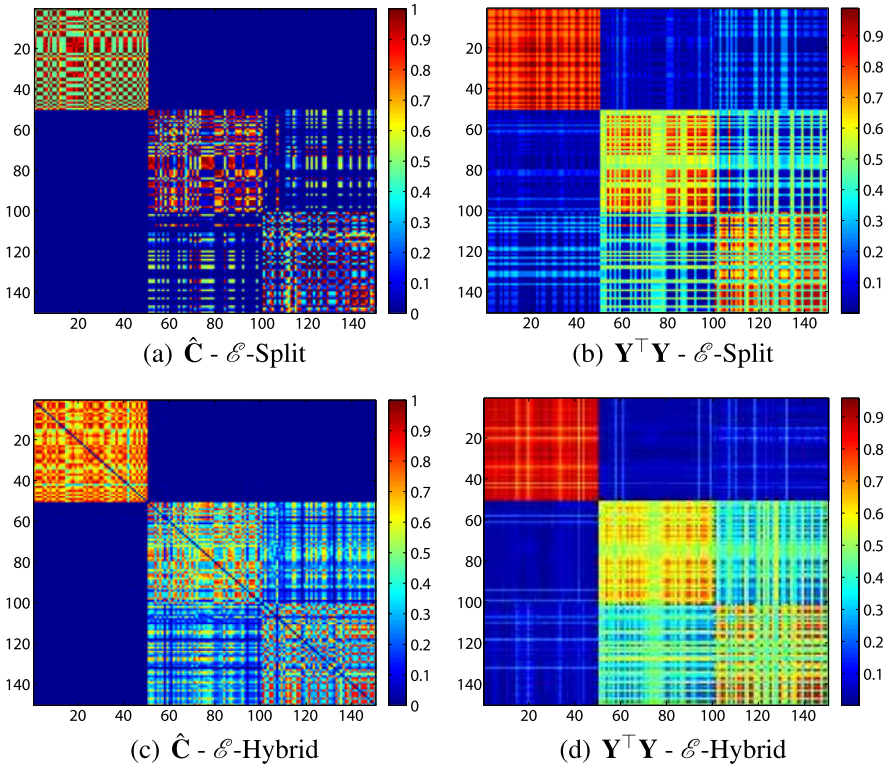
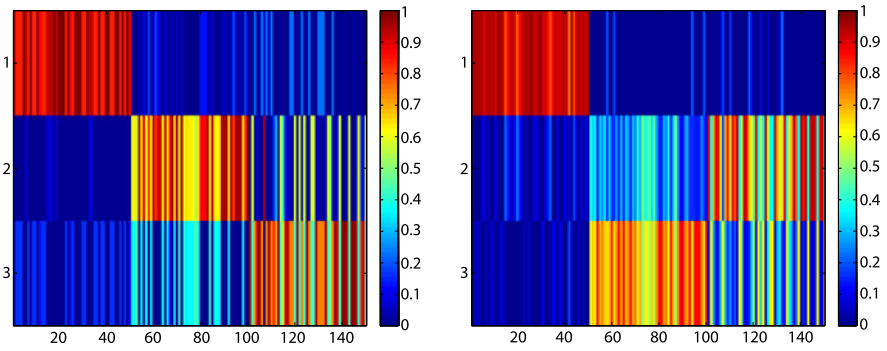


Fig. 5.5 Iris data set. Co-association matrices and corresponding estimated co-occurrences probabilities, as given by the PEACE algorithm. The *top row* corresponds to the clustering ensemble $\mathscr{E}$-split, while the *bottom row* corresponds to $\mathscr{E}$-Hybrid

object $\mathbf{o}_i$ to the class with the highest estimated probability in $\mathbf{y}_i$, i.e., according to the ML rule: $i \in \mathscr{C}_j : j = \arg\max_k y_{ik}$. Given different initializations in the optimization process, it is possible to obtain different consensus solutions with the proposed algorithm. We thus performed several runs of the algorithm, and evaluated the performances in terms of the consistency index, $\mathrm{CI}(\mathscr{P}^i, \mathscr{P}^{\mathrm{GT}})$. Tables 5.2 and 5.3 summarize the obtained results, indicating minimum, maximum, average, and standard deviation of the CIs for each data set. In addition, the first column ("selected") refers to the CI of the selected consensus solution over the several runs, according to the intrinsic optimization criterion, i.e., highest value of $\Pr(\mathbb{C} \mid \mathbf{Y})$. The last three columns in these tables register the results with the EAC method with three consensus extraction clustering algorithms: single-, average-, and Wards-link. Highest CI values for each data set are highlighted in bold.

From the analysis of Tables 5.2 and 5.3, it is apparent that the PEACE algorithm performs poorly in data sets exhibiting complex structure, where clusters are defined by connectedness as opposed to compactness properties, such as in most of the synthetic data sets. For these, the EAC method, in combination with the SL algorithm,



(a) **Y** - $\mathscr{E}$-Split (b) **Y** - $\mathscr{E}$-Hybrid

Fig. 5.6 Iris data set—probabilistic cluster assignments given by the PEACE algorithm on the clustering ensembles $\mathscr{E}$-split and $\mathscr{E}$-Hybrid

Table 5.2 Consistency indices of consensus solutions for the clustering ensemble $\mathscr{E}$-Split

Data set	PEACE					EAC		
	selected	av	std	max	min	SL	AL	WL
cigar	0.636	0.628	0.020	0.640	0.592	**1.000**	0.816	0.708
rings	0.509	0.526	0.018	0.551	0.509	**1.000**	**1.000**	0.729
spiral	0.505	0.505	0.000	0.505	0.505	0.505	0.500	**0.525**
image-c	0.499	0.503	0.002	0.505	0.499	**0.582**	**0.583**	0.433
image-1	0.555	0.570	0.025	0.613	0.550	**0.666**	0.590	0.465
breast-cancer	0.734	0.923	0.106	**0.971**	0.734	0.657	**0.971**	0.734
house-votes	**0.901**	0.892	0.012	**0.901**	0.879	0.668	0.871	0.853
ionosphere	0.632	0.632	0.000	0.632	0.632	**0.667**	0.541	0.613
iris	**0.907**	0.864	0.095	**0.907**	0.693	0.747	0.893	0.893
optdigits	0.894	0.871	0.042	**0.898**	0.798	0.618	0.798	**0.899**
std-yeast-cell	0.544	0.543	0.001	0.544	0.542	0.526	**0.688**	0.542
wine	**0.961**	**0.961**	0.000	**0.961**	**0.961**	0.674	0.927	**0.961**

performs the best, in particular when adopting the split strategy for building the CE, i.e., in $\mathscr{E}$-Split.

On the real data sets, however, the proposed algorithm shows an overall superior performance, positively correlated with the more dense block diagonal structure of the empirical co-association matrices. Corroborating this conclusion, we can notice the increased number of best results (as compared with EAC) in the $\mathscr{E}$-Hybrid CEs (Table 5.3), were this block structure is promoted by the use of lower K values for building the CEs. A notable exception to this conclusion on real data sets is the case of the optidigits, for which much better results are obtained by both PEACE and EAC methods when using the split strategy on the CE. It should be noted, however,

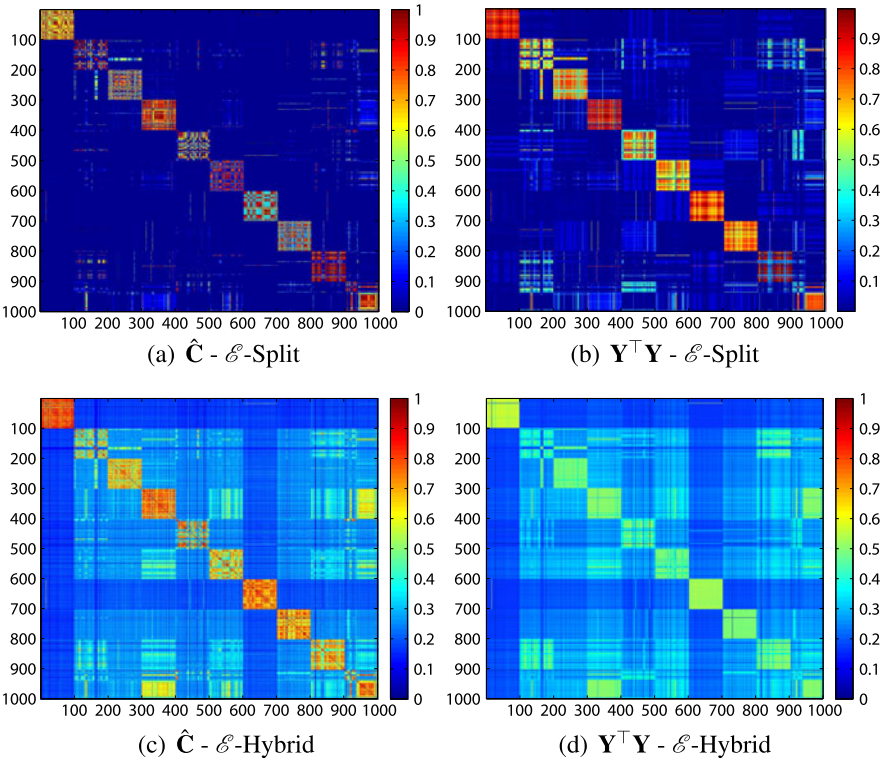
Table 5.3 Consistency indices of consensus solutions for the clustering ensemble $\mathscr{E}$-Hybrid

Data Set	PEACE					EAC		
	selected	av	std	max	min	SL	AL	WL
cigar	0.688	0.688	0.000	0.688	0.688	**1.000**	0.820	0.708
rings	0.318	0.320	0.006	0.331	0.318	**1.000**	0.349	0.351
spiral	0.510	0.510	0.000	0.510	0.510	**0.550**	0.505	0.515
image-c	**0.593**	0.533	0.034	**0.593**	0.517	0.514	0.583	0.559
image-1	0.625	0.625	0.001	0.626	0.625	**0.677**	0.620	0.606
breast-cancer	**0.968**	**0.968**	0.000	**0.968**	**0.968**	0.652	0.944	0.944
house-votes	0.901	0.901	0.000	0.901	0.901	0.530	0.530	**0.918**
ionosphere	**0.718**	**0.718**	0.000	**0.718**	**0.718**	0.644	0.658	0.715
iris	**0.913**	**0.913**	0.000	**0.913**	**0.913**	0.747	0.907	0.900
optdigits	0.497	0.419	0.072	0.499	0.366	0.499	0.716	**0.855**
std-yeast-cell	0.677	0.677	0.000	0.677	0.677	0.359	0.672	**0.680**
wine	**0.944**	0.939	0.003	**0.944**	0.938	0.393	0.371	0.927

that for this dataset the $\mathscr{E}$-Split CE does not explore a severe splitting strategy: as indicated in Table 5.1, this data set has 10 classes and 1000 samples, leading to and interval $\{K_{\min}, K_{\max}\} = \{15, 31\}$ for $\mathscr{E}$-Split, while the $\mathscr{E}$-Hybrid uses the values $\{10, 12, 15, 20\}$ for K. This suggests that the "mild" split strategy favors the revelation of the intrinsic organization structure of the dataset. This is apparent when we compare the empirical and "true" co-associations in the $\mathscr{E}$-Split with the ones in the $\mathscr{E}$-Hybrid in Fig. 5.7, where the intrinsic 10-class structure is more clear in $\mathscr{E}$-Split. This leads to considerably better probabilistic cluster assignments from the $\mathscr{E}$-Split CE, as seen in Fig. 5.8. If we reorder samples within each "natural" cluster in the co-association matrix, based on pairwise similarities, using for instance the VAT algorithm [4], we obtain the matrix in Fig. 5.9. In this figure, we can observe "microstructure" within each cluster, supposedly associated with writing styles; this can justify the better adequacy of the split strategy for this data set.

5.8 Conclusions

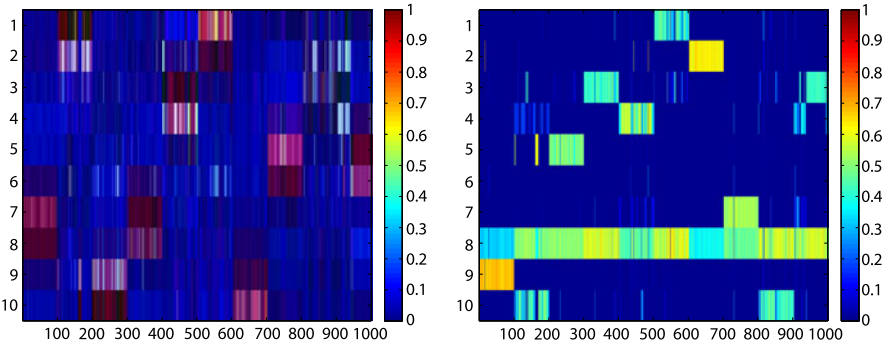
In this chapter, we addressed the Evidence Accumulation Clustering paradigm as a means of learning pairwise similarity between objects, summarized in a co-association matrix. We revised the EAC as a kernel method for extracting relations between objects. We discussed several possible interpretations for the learned co-associations, in particular the duality between similarity/data representation and probabilistic interpretations, and exploited these in two consensus clustering methods: DR-EAC, a hard clustering method exploring embeddings over learned pairwise associations; and PEACE, a unified probabilistic approach leading to soft assignments of objects to clusters.



(a) $\hat{\mathbf{C}}$ - $\mathscr{E}$-Split (b) $\mathbf{Y}^\top\mathbf{Y}$ - $\mathscr{E}$-Split

(c) $\hat{\mathbf{C}}$ - $\mathscr{E}$-Hybrid (d) $\mathbf{Y}^\top\mathbf{Y}$ - $\mathscr{E}$-Hybrid

Fig. 5.7 Optidigits data set. Co-association matrices and corresponding estimated co-occurrences probabilities, as given by the PEACE algorithm



(a) $\mathbf{Y}$ - $\mathscr{E}$-Split (b) $\mathbf{Y}$ - $\mathscr{E}$-Hybrid

Fig. 5.8 Optidigits data set—probabilistic cluster assignments given by the PEACE algorithm on the clustering ensembles $\mathscr{E}$-Split and $\mathscr{E}$-Hybrid

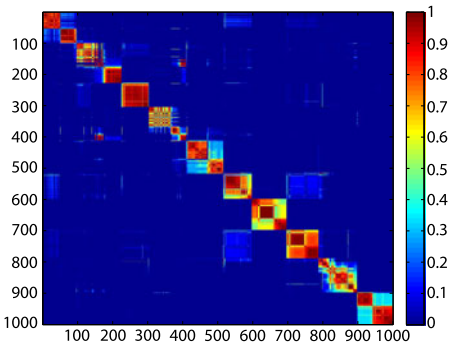


Fig. 5.9 Optidigits data set—reordered empirical co-association matrix $\hat{\mathbb{C}}$ for the $\mathscr{E}$-Split clustering ensemble, evincing micro-structure within each digit class

The DR-EAC method was evaluated in comparison with the EAC, several dimensionality reduction techniques being studied. Although no DR algorithm consistently outperformed all the others, this study showed that the use of dimensionality reduction techniques in clustering ensembles presents interesting advantages in accuracy and robustness. Future work is needed to study the influence of different strategies to construct the clustering ensemble, and criteria for the choice of DR and clustering algorithms.

PEACE obtains probabilistic cluster assignments through an optimization process that maximizes the likelihood of observing the empirical co-associations given the underlying object to cluster assignment model, which was shown to be equivalent to minimizing the Kullback–Leibler divergence between the empirical co-associations and the estimated "real" co-association distribution. When converting soft assignments to hard clusterings, the method performed favorably as compared with the EAC method for handling real data sets, and data with homogeneous clusters. In addition, PEACE, by providing probabilistic cluster assignments to objects, yields a richer level of information about cluster structure. Its poor performance on complex structure data sets is the object of current investigation.

References

1. Aidos, H., Fred, A.: A study of embedding methods under the evidence accumulation framework. In: Pelillo, M., Hancock, E. (eds.) Similarity-Based Pattern Recognition. Lecture Notes in Computer Science, vol. 7005, pp. 290–305. Springer, Berlin (2011). http://link.springer.com/chapter/10.1007/978-3-642-24471-1_21
2. Ayad, H., Kamel, M.S.: Cumulative voting consensus method for partitions with variable number of clusters. IEEE Trans. Pattern Anal. Mach. Intell. **30**(1), 160–173 (2008)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps and spectral techniques for embedding and clustering. In: Advances in Neural Information Processing Systems (NIPS 2001), vol. 14, pp. 585–591 (2002)
4. Bezdek, J., Hathaway, R.: Vat: a tool for visual assessment of (cluster) tendency. In: Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02, vol. 3, pp. 2225–2230 (2002)
5. Boyd, S., Vandenberghe, L.: Convex Optimization, 1st edn. Cambridge University Press, Cambridge (2004)

6. Demartines, P., Hérault, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. IEEE Trans. Neural Netw. **8**(1), 148–154 (1997)
7. Dimitriadou, E., Weingessel, A., Hornik, K.: A combination scheme for fuzzy clustering. In: AFSS'02, 332–338 (2002)
8. Fern, X.Z., Brodley, C.E.: Solving cluster ensemble problems by bipartite graph partitioning. In: Proc. ICML'04 (2004)
9. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) Multiple Classifier Systems, vol. 2096, pp. 309–318. Springer, Berlin (2001)
10. Fred, A., Jain, A.: Data clustering using evidence accumulation. In: Proc. of the 16th Int'l Conference on Pattern Recognition, pp. 276–280 (2002)
11. Fred, A., Jain, A.: Combining multiple clustering using evidence accumulation. IEEE Trans. Pattern Anal. Mach. Intell. **27**(6), 835–850 (2005)
12. Fred, A.L., Jain, A.K.: Learning pairwise similarity for data clustering. In: Proc. of the 18th Int'l Conference on Pattern Recognition (ICPR 2006), pp. 925–928. IEEE Comput. Soc., Washington (2006). doi:10.1109/ICPR.2006.754
13. Hadjitodorov, S.T., Kuncheva, L.I., Todorova, L.P.: Moderate diversity for better cluster ensembles. Inf. Fusion **7**(3), 264–275 (2006)
14. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems (NIPS 2003), vol. 16 (2004)
15. He, X., Cai, D., Yan, S., Zhang, H.J.: Neighborhood preserving embedding. In: Proc. of the 10th Int. Conf. on Computer Vision (ICCV 2005), vol. 2, pp. 1208–1213 (2005)
16. Hofmann, T., Puzicha, J., Jordan, M.I.: Learning from Dyadic Data. Advances in Neural Information Processing Systems (NIPS), vol. 11. MIT Press, Cambridge (1999)
17. Jain, A.K.: Data clustering: 50 years beyond k-means. Pattern Recognit. Lett. **31**(8), 651–666 (2010)
18. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Comput. Surv. **31**, 264–323 (1999)
19. Kachurovskii, I.R.: On monotone operators and convex functionals. Usp. Mat. Nauk **15**(4), 213–215 (1960)
20. Karypis, G., Kumar, V.: Multilevel algorithms for multi-constraint graph partitioning. In: Proceedings of the 10th Supercomputing Conference (1998)
21. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: applications in vlsi domain. In: Proc. Design Automation Conf. (1997)
22. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: Proc. of the IEEE International Conference on Systems, Man & Cybernetics, Hague, Netherlands, pp. 1214–1219 (2004)
23. Kuncheva, L., Hadjitodorov, S., Todorova, L.: Experimental comparison of cluster ensemble methods. In: 9th International Conference on Information Fusion, pp. 1–7 (2006). doi:10.1109/ICIF.2006.301614
24. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Information Science and Statistics. Springer, Berlin (2007)
25. Lee, J.A., Lendasse, A., Verleysen, M.: Nonlinear projection with curvilinear distances: isomap versus curvilinear distance analysis. Neurocomputing **57**, 49–76 (2004)
26. Levina, E., Bickel, P.J.: Maximum likelihood estimation of intrinsic dimension. In: Advances in Neural Information Processing Systems (NIPS 2004), vol. 17 (2004)
27. Lourenço, A., Fred, A.: Selectively learning clusters in multi-EAC. In: International Conference on Knowledge Discovery and Information Retrieval (KDIR 2010), Valencia, Spain (2010)
28. Lourenço, A., Fred, A., Jain, A.K.: On the scalability of evidence accumulation clustering. In: ICPR. Istanbul Turkey (2010)
29. Lourenço, A., Fred, A., Figueiredo, M.: A generative dyadic aspect model for evidence accumulation clustering. In: Pelillo, M., Hancock, E. (eds.) Similarity-Based Pattern Recognition. Lecture Notes in Computer Science, vol. 7005, pp. 104–116. Springer, Berlin (2011). http://link.springer.com/chapter/10.1007/978-3-642-24471-1_8

30. Luenberger, D.G., Ye, Y.: Linear and Nonlinear Programming, 3rd edn. Springer, Berlin (2008)
31. Meila, M.: Comparing clusterings by the variation of information. In: Proc. of the Sixteenth Annual Conf. of Computational Learning Theory (COLT). Springer, Berlin (2003)
32. Ng, A.Y., Jordan, M.I., Weiss, Y.: On spectral clustering: analysis and an algorithm. In: NIPS, pp. 849–856. MIT Press, Cambridge (2001)
33. Punera, K., Ghosh, J.: Advances in Fuzzy Clustering and Its Applications, Chap. Soft Consensus Clustering. Wiley, New York (2007)
34. Rota Bulò, S., Lourenço, A., Fred, A., Pelillo, M.: Pairwise probabilistic clustering using evidence accumulation. In: Proc. 2010 Int. Conf. on Structural, Syntactic, and Statistical Pattern Recognition, SSPR&SPR'10, pp. 395–404 (2010)
35. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science **290**, 2323–2326 (2000)
36. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. Comput. **18**(5), 401–409 (1969)
37. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. **22**(8), 888–905 (2000)
38. Steyvers, M., Griffiths, T.: Probabilistic Topic Models, Chap. Latent Semantic Analysis: a Road to Meaning. Laurence Erlbaum, Hillsdale (2007)
39. Strehl, A., Ghosh, J.: Cluster ensembles—a knowledge reuse framework for combining multiple partitions. J. Mach. Learn. Res. **3**, 583–617 (2002)
40. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290**, 2319–2323 (2000)
41. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier, Amsterdam (2003)
42. Topchy, A., Jain, A., Punch, W.: Combining multiple weak clusterings. In: IEEE Intl. Conf. on Data Mining, Melbourne, FL, pp. 331–338 (2003)
43. Topchy, A., Jain, A., Punch, W.: A mixture model of clustering ensembles. In: Proc. of the SIAM Conf. on Data Mining (2004)
44. Topchy, A., Jain, A.K., Punch, W.: Clustering ensembles: models of consensus and weak partitions. IEEE Trans. Pattern Anal. Mach. Intell. **27**(12), 1866–1881 (2005)
45. Wang, H., Shan, H., Banerjee, A.: Bayesian cluster ensembles. In: 9th SIAM Int. Conf. on Data Mining (2009)
46. Wang, P., Domeniconi, C., Laskey, K.B.: Nonparametric Bayesian clustering ensembles. In: ECML PKDD'10, pp. 435–450 (2010)