A Study of Embedding Methods under the Evidence Accumulation Framework

Helena Aidos and Ana Fred

Instituto de Telecomunicações, Instituto Superior Técnico, Lisbon, Portugal
{haidos,afred}@lx.it.pt

Abstract. In this paper we address a voting mechanism to combine clustering ensembles leading to the so-called co-association matrix, under the Evidence Accumulation Clustering framework. Different clustering techniques can be applied to this matrix to obtain the combined data partition, and different clustering strategies may yield too different combination results. We propose to apply embedding methods over this matrix, in an attempt to reduce the sensitivity of the final partition to the clustering method, and still obtain competitive and consistent results. We present a study of several embedding methods over this matrix, 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. When applying several clustering techniques over these new representations, we evaluate the impact of these transformations in terms of performance and coherence of the obtained data partition. Experimental results, on synthetic and real benchmark datasets, show that extracting the relevant features through dimensionality reduction yields more consistent results than applying the clustering algorithms directly to the co-association matrix.

Keywords: clustering ensembles, co-association matrix, evidence accumulation clustering, embedding methods.

1 Introduction

Clustering is one of the central problems in Pattern Recognition and Machine Learning. Given a set of unlabeled data, its typical goal is to group objects into clusters, such that objects within a cluster are similar, and objects in distinct clusters are dissimilar. Assuming that clusters are disjoint, the clustering process leads to a data partition. 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 [9,18].

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 [4,16,10,1]. 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.

We can generate clustering ensembles following two approaches: choice of data representation or choice of clustering algorithms or algorithmic parameters. In the first

M. Pelillo and E.R. Hancock (Eds.): SIMBAD 2011, LNCS 7005, pp. 290–305, 2011.

case, we can get different representations of objects by applying different preprocessing mechanisms or feature extraction techniques, or just by sampling the data a number of times. We can also have clustering ensembles if we use several clustering algorithms or just the same algorithm with different parameter values.

Fred and Jain [5] proposed a clustering ensemble approach based on the combination of information provided by a set of different partitions of a given dataset, through the Evidence Accumulation method. To combine all the different partitions, Fred and Jain [5] proposed a voting scheme, which leads to a pairwise relationships matrix, called "co-association matrix". The final data partition is obtained by applying a clustering algorithm over the co-association matrix. One main advantage of this voting scheme is that it can deal with partitions having different number of clusters and different data representations.

The application of different clustering techniques to this matrix may yield different solutions. We propose to use embedding methods (also called dimensionality reduction (DR) methods) over this matrix, in an attempt to reduce the sensitivity of the combined data partition to the clustering method, and obtain better and more consensual results. We present a study of the performance and coherence of the solutions when different clustering techniques are applied to the resulting data representations. To obtain those representations we will follow two approaches: interpret the co-association matrix as a feature space, and as a similarity space.

The first approach is similar to the one proposed by Kuncheva *et al.* [11]: we will view the co-association matrix as a feature space, but instead of using the full feature space, we will reduce its dimension using several dimensionality reduction (DR) methods. These DR techniques aim to take a set of data points in a high-dimensional space and output a new set of data points in a lower-dimensional space, in a way that preserves the topology of the high-dimensional data. This new data representation is commonly called an *embedding* of the original dataset. We will empirically show that the use of DR methods to remove redundant features improves the quality and consistency of the final partition for different clustering techniques.

In the other approach we view the co-association matrix as a similarity space, as in [5]. However, instead of applying directly the clustering techniques to this matrix, we will first apply DR methods to it. Many DR methods take as input some distance measure between points (usually in a distance matrix whose (i, j) entry contains the distance between data points i and j). Therefore, if one converts the similarity measures in the co-association matrix into distance (or dissimilarity) measures, one can input this dissimilarity matrix into the DR methods directly. The resulting low-dimensional data points are then clustered with several clustering techniques. Again, we intend to study if there exists consistency and an improvement in the quality of the solutions.

The dimensionality reduction methods used have different characteristics such as: linear vs. nonlinear; preserving local structure vs. preserving global structure; preserve spatial distances vs. preserving graph distances. This means that different embedding strategies may influence differently the solutions; we intend to study if there exists a class of embedding methods suitable for certain types of datasets (well separate clusters, touching clusters).

This paper is organized as follows: Section 2 gives a brief explanation of the embedding algorithms used in the study. Section 3 explains the evidence accumulation approach, including the construction of the co-association matrix. Section 4 explains the new methodology proposed in this paper and the two interpretations we give to this matrix. Section 5 describes the datasets used in this study and the experimental results for the two interpretations of the co-association matrix: feature space (section 5.2) and similarity space (section 5.3). We summarize and discuss the main findings in Section 6. Conclusions are drawn in Section 7.

2 Embedding Methods

To perform embeddings we will use several unsupervised DR methods: Locality Preserving Projections (LPP) [7]. Neighborhood Preserving Projections (NPE) [6], Sammon's mapping [15], Curvilinear Component Analysis (CCA) [3], Isomap [17], Curvilinear Distance Analysis (CDA) [13], Locally Linear Embedding (LLE) [14] and Laplacian Eigenmap (LE) [2]. We now briefly introduce each of these algorithms.

2.1 Nonlinear Methods

The *Locally Linear Embedding* (LLE) [14] assumes that the data manifold is smooth and sampled densely enough such that each data point lies close to a locally linear subspace on the manifold. In other words, the manifold smoothness and sampling should be enough to locally approximate the manifold by a hyperplane. LLE makes a locally linear approximation of the whole data manifold; it first estimates a local coordinate system for each data point 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 $E(\mathbf{W}) = \sum_i \|\mathbf{x}_i - \sum_j W_{i,j}\mathbf{x}_j\|^2$ with respect to the coefficients $W_{i,j}$, under the constraints that $W_{i,j} = 0$ if i and j are not neighbors, and $\sum_j W_{i,j} = 1$. After finding these weights, the low-dimensional configuration of points is next found by minimizing $E(\mathbf{Y}) = \sum_i \|\mathbf{y}_i - \sum_j W_{i,j}\mathbf{y}_j\|^2$ with respect to the low-dimensional representation $\mathbf{y}_i$ of each data point.

The *Laplacian Eigenmap* (LE) [2] uses a graph embedding approach. An undirected k-nearest neighbor graph is formed, where each data point is a vertex. Points i and j are connected by an edge with weight $W_{i,j} = 1$ 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 [2]. 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 what happens to the other points. Technically, it minimizes $\frac{1}{2}\sum_{i,j} \|\mathbf{y}_i - \mathbf{y}_j\|^2 W_{i,j} = \mathbf{y}^T\mathbf{L}\mathbf{y}$ with respect to the low-dimensional point locations $\mathbf{y}_i$, where $\mathbf{L} = \mathbf{D} - \mathbf{W}$ is the graph Laplacian and $\mathbf{D}$ is a diagonal matrix with elements $\mathrm{D}_{ii} = \sum_j W_{i,j}$. 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. In practice, the low-dimensional configuration is found by solving the generalized eigenvalue problem $\mathbf{L}\mathbf{y} = \lambda \mathbf{D}\mathbf{y}$ [2]. The smallest eigenvalue corresponds to the trivial solution, but the eigenvectors corresponding to the next smallest eigenvalues yield the desired LE solution.

Isomap [17] is a variant of Multidimensional Scaling (MDS) [12], which finds a configuration of output coordinates matching a given distance matrix. Isomap does not compute pairwise input-space distances as simple Euclidean distances but as *geodesic distances* along the manifold of the data (technically, along a graph formed by connecting all k-nearest neighbors). Given these geodesic distances the output coordinates are found by standard linear MDS. When output coordinates are found for such input distances, the manifold structure in the original data becomes unfolded; it has been shown that this algorithm is asymptotically able to recover certain types of manifolds.

Curvilinear component analysis (CCA) [3] is a variant of MDS [12] 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. In our case, 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 Curvilinear Distance Analysis (CDA) [13] is an extension of CCA. The idea is to replace in MDS the Euclidean distances in the original space with geodesic distances in the same manner as in the Isomap algorithm. Otherwise the algorithm is similar to CCA.

2.2 Linear Methods

Locality Preserving Projections (LPP) [7] is a linear dimensionality reduction method that preserves 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) [6] 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), which means that it has properties similar to that method.

3 Evidence Accumulation: The Co-association Matrix

Let $X = \{x_1, x_2, \ldots, x_n\}$ be a set of n objects or samples represented in a feature space or some other data representation. A clustering algorithm takes X as input and groups the n patterns into k clusters, forming a partition P. A *clustering ensemble*, $\mathbb{P}$, is a set of N different partitions of the data X:

$$\mathbb{P} = \{P^1, P^2, \ldots, P^N\} \tag{1}$$

$$\begin{aligned} P^1 &= \left\{C_1^1, C_2^1, \ldots, C_{k_1}^1\right\} \\ &\vdots \\ P^N &= \left\{C_1^N, C_2^N, \ldots, C_{k_N}^N\right\}, \end{aligned}$$

where C_j^i is the jth cluster in data partition P^i, which has k_i clusters and n_j^i is the cardinality of C_j^i, with $\sum_{j=1}^{k_i} n_j^i = n, i = 1, \ldots, N$.

The *evidence accumulation* approach, proposed by Fred and Jain [5], is a three-step cluster ensemble method: 1- build the clustering ensemble (CE); 2- combine evidence in the CE, mapping it into a co-association matrix; 3- extract the consensus partition by applying a clustering algorithm over the co-association matrix. The basic idea is that patterns belonging to a "natural" cluster are very likely to be assigned to the same cluster in different data partitions. Taking the co-occurrences of pairs of patterns in the same cluster as votes for their association, the N data partitions of n patterns yield a $n \times n$ co-association matrix:

$$\mathcal{C}(i,j) = \frac{n_{ij}}{N}, \tag{2}$$

where n_{ij} is the number of times the pattern pair (i,j) is assigned to the same cluster among the N partitions.

In its normalized form, as per expression (2), matrix $\mathcal{C}$ can be given different interpretations, either probabilistic or simply as pairwise similarity. Another issue is how to address and use this matrix for clustering purposes. In the following we propose a novel methodology by applying DR techniques.

4 Dimensionality Reduction in Evidence Accumulation Clustering

We propose a new methodology called Dimensionality Reduction in Evidence Accumulation Clustering (DR-EAC), which is based on the Evidence Accumulation Clustering (EAC) method described above. As said before, the evidence accumulation approach is a three-step cluster ensemble method; we now propose a four-step method. We build the clustering ensemble (step 1) and the co-association matrix (step 2) similarly to the evidence accumulation approach. However, instead of applying a clustering algorithm directly to the co-association matrix, we apply a DR technique to it (which is now step 3). 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 dataset, which is then fed into a clustering algorithm (which is now step 4). We now discuss each of these four steps in more detail.

1) Build the Clustering Ensemble. As referred before, there are several ways to produce a clustering ensemble. In this study we build a clustering ensemble by running the k-means algorithm to produce a total of $N = 200$ data partitions, each one with k clusters, k being an integer randomly drawn between $k_{min} = \max\{\sqrt{n}/2, n/50\}$ and $k_{max} = k_{min} + 20$, where n is the number of samples of the dataset.

2) Obtain the co-association matrix. We begin by computing the co-association matrix according to equation (2). Then, we interpret this matrix in one of two possible ways:

- *Co-associations viewed as Features:* One way to look at matrix $\mathcal{C}$ is to say that its i-th row represents a new set of features for the i-th data point, an idea originally proposed by Kuncheva *et al.* [11]. Thus, each pattern is now represented by how many times it was grouped together with all other patterns.

- *Co-association viewed as Similarities:* We can transform the co-association matrix $\mathcal{C}$, which is a similarity matrix, into a dissimilarity matrix (or distance matrix). Since many DR methods can take as input a matrix of pairwise distances (or dissimilarities), if we transform this matrix of similarities into a matrix of dissimilarities we can exploit this property. Since the elements of $\mathcal{C}$ lie between 0 and 1, we use a very simple transformation: the new dissimilarity matrix has the element (i, j) given by $1 - \mathcal{C}(i, j)$.

3) Apply Dimensionality Reduction techniques. We apply DR techniques to obtain a new representation of the data, preserving the topology of the original data. For the DR methods 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). In all cases we let each algorithm choose the most suitable parameter and dimension 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 [17]. It is beyond the scope of this paper to detail how these parameters should be chosen; the relevant information can be found in the references cited in Section 2.

4) Extract the consensus partition. After we get the embedded data, we apply eight well-known clustering algorithms: k-means, single-link, complete-link, average-link, Ward-link, centroid-link, median-link and weighted-link [9].

4.1 Quality Measures

We use two quality measures to assess the results: consistency index (CI) and normalized mutual information (NMI).

The CI simply measures the fraction of patterns correctly grouped together compared to the ground-truth labeling. It takes values between 0 and 1, and it is a measure of the accuracy of the clustering.

The NMI [16] is a symmetric measure of the information shared between two partitions. Consider the partition P^a, which describes a labeling of the n patterns in the dataset X into k_a clusters. If one takes frequency counts as approximations for probabilities, the entropy of the data partition P^a is given by $H(P^a) = -\sum_{i=1}^{k_a} \frac{n_i^a}{n} \log\left(\frac{n_i^a}{n}\right)$, where n_i^a represents the number of patterns in cluster $C_i^a \in P^a$. The agreement between two partitions P^a and P^b is given by their mutual information:

$$I(P^a, P^b) = \sum_{i=1}^{k_a} \sum_{j=1}^{k_b} \frac{n_{ij}^{ab}}{n} \log\left(\frac{\frac{n_{ij}^{ab}}{n}}{\frac{n_i^a}{n} \cdot \frac{n_j^b}{n}}\right),$$

whith n_{ij}^{ab} the number of shared patterns between clusters $C_i^a \in P^a$ and $C_j^b \in P^b$.

The NMI is then defined by

$$NMI(P^a, P^b) = \frac{I(P^a, P^b)}{\sqrt{H(P^a)H(P^b)}}.$$

It is similar to the widely used mutual information, but normalized to be in the interval $[0, 1]$. For each DR method, we compute the NMI between all 28 pairs of clustering algorithms[1]. We then take the average of these 28 NMI values to obtain the average NMI for that DR method. This average NMI will measure how consistent the partitions are among the 8 clustering algorithms after applying that DR method.

5 Experimental Results

We will apply the new methodology described in section 4 to several datasets, in an attempt to improve the quality and robustness of the solutions, compared to the evidence accumulation approach. We will apply the clustering algorithms mentioned in section 4 to the co-association matrix directly (in both interpretations), an approach we will denote by EAC_F (Evidence Accumulation Clustering in the feature space) and EAC (Evidence Accumulation Clustering in the sense presented by [5]). The idea is to verify empirically whether the use of embedding methods and subsequent clustering algorithms is advantageous relative to the application of clustering algorithms on the co-association matrix directly. Also, we will try to find some correspondence between pairs of embedding and clustering methods suitable for some types of data. In that sense, we will study synthetic data and real data, with the synthetic data divided in two broad meta-sets: datasets with separate clusters and datasets with touching clusters.

5.1 Data

We used 18 datasets: 10 synthetic datasets (5 well-separated and 5 with touching clusters), and 8 real datasets from the UCI Machine Learning Repository[2]. The synthetic datasets were chosen to take into account a wide variety of situations: well-separated and touching clusters; gaussian and non-gaussian clusters; arbitrary shapes; and diverse cluster densities. These synthetic datasets are shown in figure 1. The *Iris* dataset consists of three species of Iris plants (Setosa, Versicolor and Virginica). This dataset is characterized by four features and 50 samples in each cluster. *Std Yeast* is composed of 384 samples (genes) over two cell cycles of yeast cell data. This dataset is characterized by 17 features and consisting of five clusters corresponding to the five phases of the cell cycle. The *Pima* dataset is composed of 768 samples (genes) from National Institute of Diabetes and Digestive and Kidney Diseases, it has 8 features and two clusters. *Wine* consists of the results of a chemical analysis of wines grown in the same region in Italy divided into three clusters with 59, 71 and 48 patterns described by 13 features. *Optdigits* is a subset of Handwritten Digits dataset containing only the first 100 patterns of each digit, from a total of 1000 data samples characterized by 64 attributes. The *Wisconsin Breast-Cancer* dataset consists of 683 patterns represented by nine features and has two clusters. The *House Votes* dataset consists of votes for each of the U.S. House of Representatives Congressmen on the 16 key votes identified by the Congressional Quarterly Almanac. It is composed by two clusters and only the patterns without missing values

[1] 28 is the number of off-diagonal elements in the upper triangular part of the matrix containing the NMI between pairs of clustering algorithms, which is an 8-by-8 matrix.

[2] `http://archive.ics.uci.edu/ml`

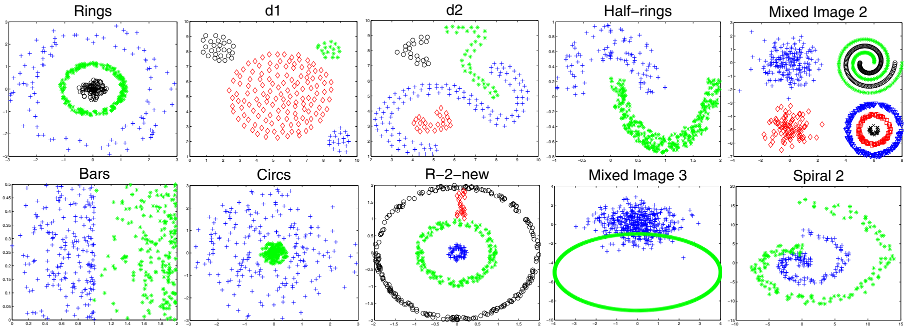

Fig. 1. Synthetic datasets

were considered, for a total of 232 samples (125 democrats and 107 republicans). The *Crabs* dataset consists of 200 patterns represented by 5 features and has two classes. Pima, House Votes, Crabs and Wine were normalized to have unit variance.

5.2 Experiment 1: Feature Space

In this section we interpret the co-association matrix as a new feature space, as described in Section 4. The application of clustering algorithms directly to the co-association matrix viewed as a feature space, is here denoted by EAC$_F$.

Analyzing the average NMI in figure 2 over all clustering algorithms used to obtain the final partition, we notice that LE and LPP are the ones that produce more coherent solutions for the synthetic datasets with separate clusters (figure 2 top), which indicates that they are robust to the extraction algorithm. CCA and CDA are the algorithms with the most dispersion in the solutions for all datasets. Unlike for separate clusters, the NMI for datasets with touching clusters (figure 2 middle) shows that no DR algorithm is robust to the choice of the clustering algorithm. In the real datasets, LE is the most consistent DR algorithm in half of the datasets (Wine, Std Yeast, Optdigits and Iris).

Even if the NMI is high, it is not necessarily true that we have a high CI (i.e. that the results of the clustering algorithms are good), it only means that the clustering algorithms obtained similar final partitions. However, the use of that measure is a good indicator that the embedded space yields good clustering results regardless of the clustering algorithm. This is an advantage, since we do not know *a priori* which is the most suitable clustering algorithm for a certain kind of data.

Table 1 contains the best CI values (first row of each dataset) and the corresponding clustering algorithm used for that solution; it also presents the average CI over all the clustering algorithms (second row of each dataset). Based on figure 2 we have claimed that LE and LPP are the ones that produce the most coherent solutions for the synthetic datasets with separate clusters; Table 1 corroborates these findings, since LE and LPP usually yield maximum CI for several clustering algorithms.

In synthetic datasets with separate clusters, LE and LPP, which are local algorithms, combine well with multiple hierarchical clustering algorithms. Isomap and Sammon, which are global and nonlinear, combine well with single-link, which is also the best clustering algorithm for EAC$_F$.

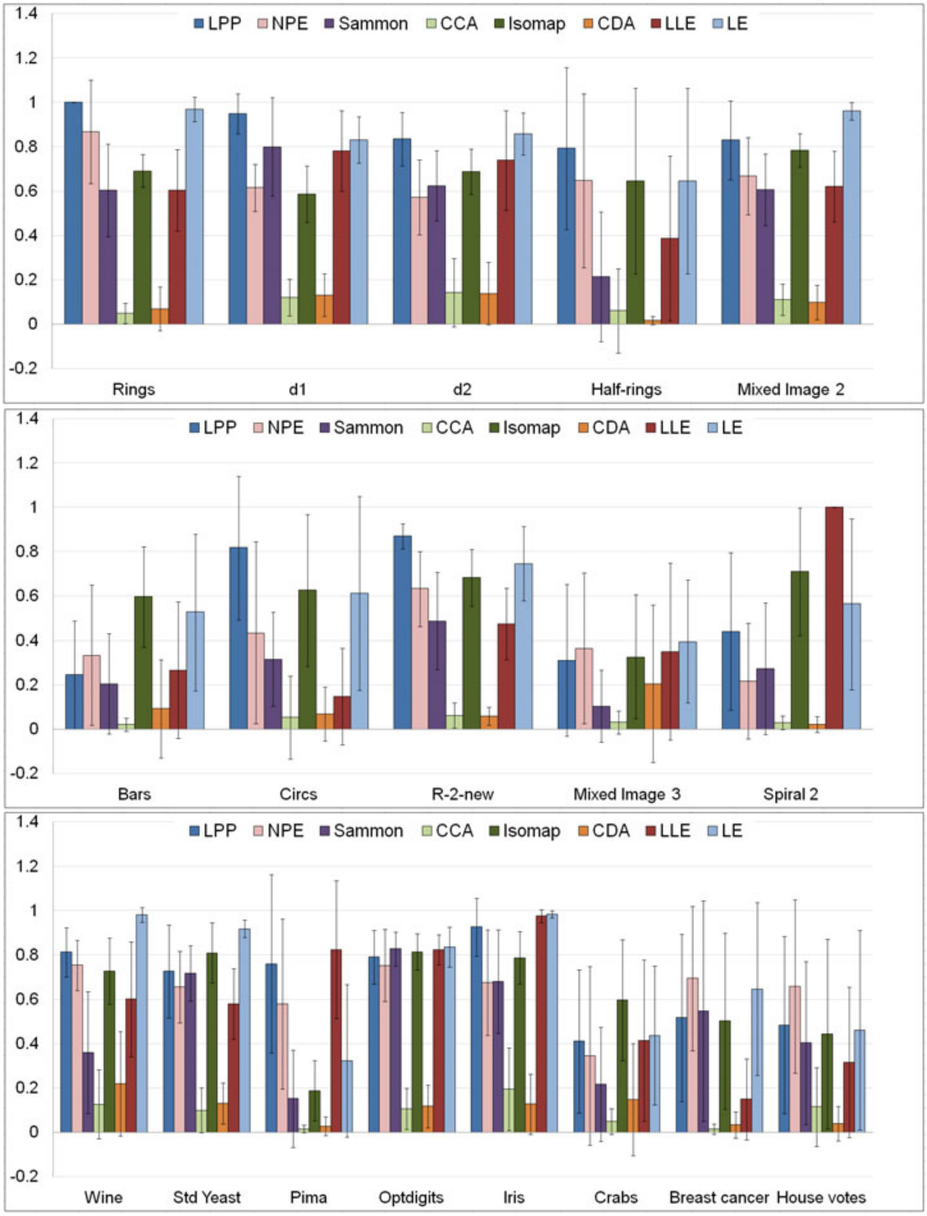

Fig. 2. Mean and standard deviation of Normalized Mutual Information over the clustering algorithms for each dataset and each embedding method. The co-association matrix was interpreted as features. *Top:* Synthetic datasets with separate clusters. *Middle:* Synthetic datasets with touching clusters. *Bottom:* Real datasets.

The analysis of the CI values for the synthetic datasets with touching clusters, shown in Table 1, shows that LPP, Isomap and LE are, on average values, better than EAC_F. In terms of maximum values, EAC_F outperforms the DR-based methods only in one dataset (R-2-new), and still by a very small margin; while it is outperformed in all remaining datasets.

The best DR-clustering algorithm pairs, for synthetic datasets with touching clusters, are LPP with k-means, Sammon with Ward-link and CDA with k-means. The overall best DR is Isomap, which is in first place in maximum CI for 4 out of 5 datasets.

The analysis of CI values for real datasets (see Table 1), shows that all DR methods do relatively well when compared to EAC_F, except for CCA and CDA. Isomap and Sammon are the two best DR algorithms when compared to the remaining DR techniques, especially in the Optdigits dataset. CCA and CDA are the worst overall methods, especially in the Std Yeast and Optdigits datasets.

These results show the advantage of performing DR over using EAC_F. In fact, from Table 1, using DR gives in general the best CI in all datasets, both in terms of maximum CI and of average CI.

Overall, for both synthetic and real datasets, there is no DR algorithm which is always robust in terms of NMI. However, LE and LPP (which is a linear version of LE), seem to have this property, especially in synthetic datasets with separate clusters. For the real datasets, LPP and LE present the best results, except in the Optdigits dataset, which yields better results with a global DR method (like Isomap and Sammon), instead of a local method.

5.3 Experiment 2: Similarity Space

In this section we interpret the entries of the co-association matrix as similarity values. We transform these into dissimilarity values, as described in Section 4. We plug-in this dissimilarity matrix into the embedding methods and will add “EA-” (from “Evidence Accumulation”) before the acronyms of the DR methods to emphasize the dependency of this matrix.

The analysis of NMI values for the synthetic datasets with separate clusters, shown in Figure 3, shows that EA-LE and EA-LLE yield the most coherent clustering results, except for the Half-rings dataset. For the Mixed Image 2 dataset, local algorithms (EA-LPP, EA-NPE, EA-LLE and EA-LE) and global algorithms that preserve “geodesic” distances (EA-Isomap, EA-CDA) have very coherent results. However, the analysis of the CI values (Table 2) immediately shows that results are not good for that dataset. This suggest that the co-association matrix might not be the best clustering ensemble approach for this dataset.

Similar to the feature space, the analysis of NMI values for synthetic datasets with touching clusters (figure 3 middle) suggests that no DR algorithm is robust to the choice of clustering algorithm; except the EA-Sammon in the Mixed Image 3. For the real datasets (figure 3 bottom) EA-LE is the DR algorithm with the most consistent results, except for the Pima, Crabs and Breast cancer datasets.

The best overall DR methods, for the synthetic datasets with separate clusters, are EA-LE and EA-LLE. EA-Isomap, EA-CCA, EA-CDA and EA-LE yield the best results

Table 1. Consistency index (%) for co-association matrix interpreted as features. *(First row)* Best CI and clustering algorithm(s) which yield that CI value. Legend: (1) k-means, (2) single-link, (3) complete-link, (4) average-link, (5) Ward-link, (6) centroid-link, (7) median-link, (8) weighted-link. *(Second row)* Average CI (%) over all clustering methods. The gray cells correspond to the best NMI presented in figure 2 and the best average CI are shown in bold.

		EAC$_F$	LPP	NPE	Sammon	CCA	Isomap	CDA	LLE	LE
Synthetic data with separate clusters	Rings	100	100	61.25	100	50.00	100	52.00	85.50	100
		(2)	(1-8)	(1)	(2)	(2)	(2)	(4)	(5)	(2-8)
		65.28	**100**	55.13	64.78	43.00	73.56	45.75	66.06	99.47
	d1	100	100	82.00	100	70.00	100	70.50	72.50	100
		(2-8)	(2-8)	(6)	(2,4-8)	(2,6)	(2)	(2)	(2)	(2,8)
		98.44	**98.19**	65.25	91.75	54.75	65.31	50.13	69.69	85.88
	d2	100	93.50	51.00	100	59.00	69.00	60.50	50.50	67.00
		(2)	(7)	(7)	(2)	(6)	(2)	(4)	(2)	(3,4,6)
		76.31	**73.87**	42.56	73.00	49.87	59.56	49.25	44.88	61.69
	Half-rings	100	100	69.75	100	81.75	100	74.75	100	100
		(2)	(1,2,4-8)	(2,4-8)	(2)	(5)	(1,2)	(2,6,7)	(2)	(2,4-8)
		72.19	**93.31**	65.81	72.09	66.19	59.87	63.56	88.28	86.63
	Mixed Image 2	65.70	71.80	36.60	71.60	22.90	71.40	23.70	47.00	71.60
		(2)	(2)	(5)	(2)	(6)	(2)	(1)	(5)	(3)
		54.44	63.69	32.50	51.90	21.10	57.45	22.61	38.52	**70.66**
Synthetic data with touching clusters	Bars	99.25	99.25	79.25	99.25	59.50	99.25	73.75	76.00	96.00
		(5)	(4,5)	(1)	(5)	(1)	(2,3)	(1)	(7)	(1)
		68.19	75.25	64.78	68.19	53.09	**90.16**	58.78	62.84	76.84
	Circs	99.50	100	58.75	99.50	63.00	100	84.50	59.00	99.50
		(2,5,8)	(1-6,8)	(1)	(2,8)	(5)	(1,8)	(1)	(8)	(5)
		80.00	**96.16**	54.94	80.56	55.62	91.37	62.31	55.31	66.91
	R-2-new	90.20	77.40	44.40	89.20	50.40	82.80	51.20	57.20	78.60
		(4)	(1)	(2)	(4,5)	(6)	(4)	(7)	(5)	(1)
		66.60	**73.95**	40.55	70.52	39.52	71.22	42.57	51.67	71.52
	Mixed Image 3	84.90	71.90	66.80	74.60	54.80	89.50	74.80	55.30	83.80
		(5)	(1)	(3)	(5)	(8)	(3)	(1)	(3,5)	(4)
		61.52	67.10	58.16	61.59	52.15	73.42	55.59	53.17	**74.92**
	Spiral 2	77.67	77.67	64.33	77.67	58.67	85.00	51.67	85.00	85.00
		(2)	(2)	(8)	(2)	(1)	(2)	(1)	(1-8)	(2,5,7,8)
		63.50	70.96	56.54	61.12	52.50	82.54	50.75	**85.00**	81.33
Real data	Wine	96.07	98.31	90.45	96.07	72.47	96.63	84.27	61.24	96.63
		(8)	(3)	(3)	(5)	(1)	(5,6)	(1)	(5)	(1)
		75.91	94.03	77.18	71.07	46.49	88.48	58.43	47.68	**94.66**
	Std Yeast	60.94	63.80	58.07	61.20	37.24	61.20	35.94	60.16	71.35
		(4)	(1)	(7)	(8)	(4)	(3)	(6)	(5)	(3,5,7)
		54.88	58.36	50.10	54.10	33.33	57.19	32.49	51.14	**66.83**
	Pima	64.71	64.71	65.36	66.02	65.10	64.71	65.23	64.71	64.58
		(2,7)	(1,2,4,6-8)	(2)	(7)	(4)	(2)	(2,7)	(5)	(2)
		56.95	64.34	63.95	57.86	60.90	60.12	60.16	**64.49**	57.03
	Optdigits	87.90	49.60	52.00	85.40	22.50	84.10	17.60	46.30	55.90
		(8)	(5)	(5)	(1)	(5)	(3)	(1)	(3)	(5)
		69.75	31.06	39.34	**74.42**	17.46	71.18	14.53	43.91	38.61
	Iris	84.00	90.67	70.67	90.67	58.67	94.00	49.33	53.33	90.67
		(5,8)	(3)	(3)	(2,8)	(1)	(1)	(1)	(2,4-8)	(1-3,7,8)
		63.17	84.83	62.08	68.42	45.17	86.58	39.75	53.00	**90.42**
	Crabs	65.00	56.00	58.00	65.00	57.00	70.50	54.00	67.00	70.50
		(2)	(3)	(1,5)	(2)	(5)	(2)	(3)	(7)	(4,6)
		59.94	53.12	53.31	57.37	52.50	55.87	51.56	58.00	**62.81**
	Breast Cancer	62.96	68.81	58.13	64.86	64.86	94.58	74.23	75.55	68.67
		(2)	(5)	(2,3,7,8)	(2-4,6-8)	(2,6,7)	(4-8)	(1)	(8)	(1)
		56.81	61.11	56.44	61.68	60.65	**86.09**	64.81	67.84	60.45
	House Votes	89.22	88.36	81.90	87.93	81.47	87.07	61.21	64.66	74.14
		(1)	(1)	(5)	(1)	(1)	(3)	(5)	(1)	(1)
		74.52	71.28	63.31	**73.81**	59.37	69.34	54.69	57.81	62.88

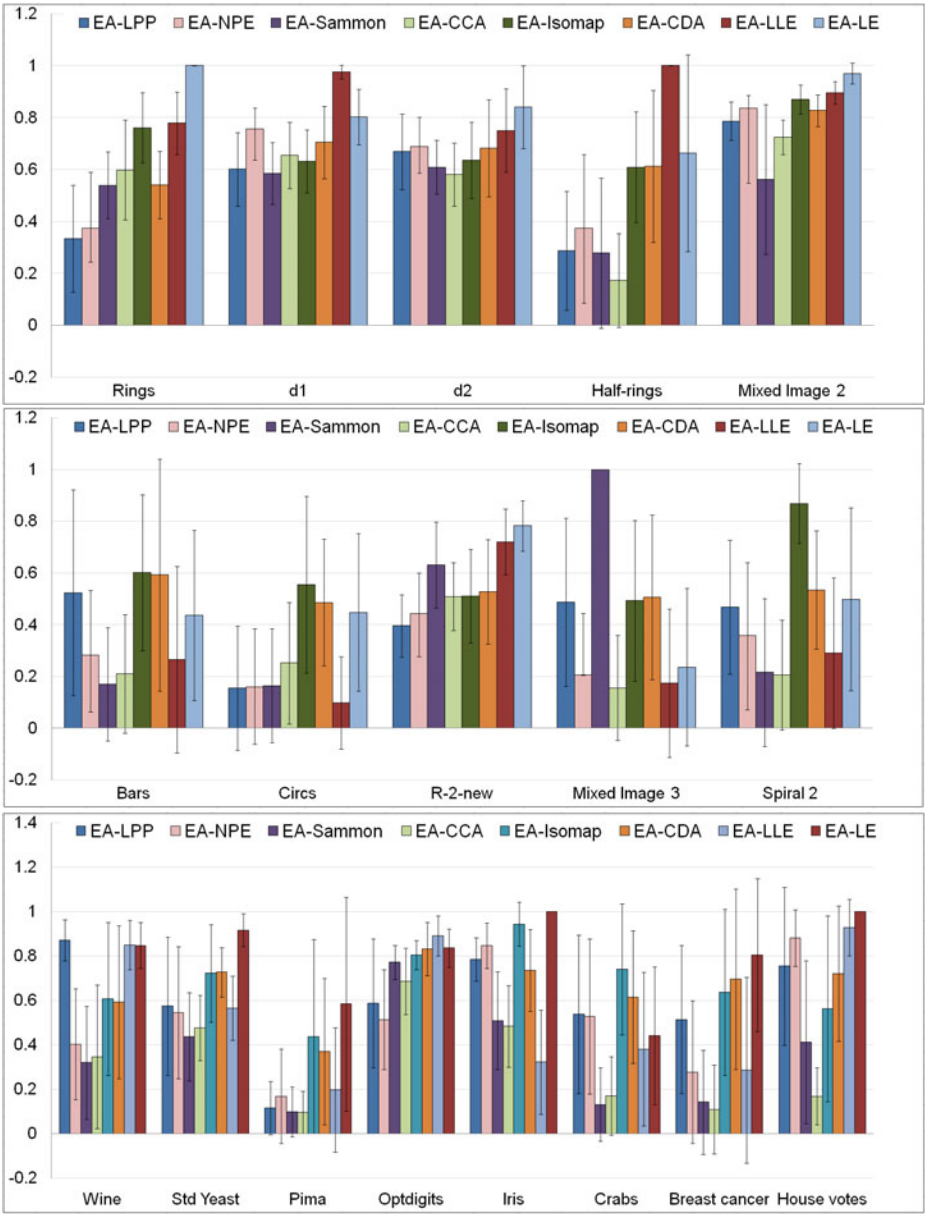

Fig. 3. Mean and standard deviation of Normalized Mutual Information over the clustering algorithms for each dataset and each embedding method. The co-association matrix was interpreted as similarities. *Top:* Synthetic datasets with separate clusters. *Middle:* Synthetic datasets with touching clusters. *Bottom:* Real datasets.

Table 2. Consistency index (%) for co-association matrix interpreted as similarities. *(First row)* Best CI and clustering algorithm(s) which yield that CI value. Legend: (1) k-means, (2) single-link, (3) complete-link, (4) average-link, (5) Ward-link, (6) centroid-link, (7) median-link, (8) weighted-link. *(Second row)* Average CI (%) over all clustering methods. The gray cells correspond to the best NMI presented in figure 3 and the best average CI are shown in bold.

		EAC	EA-LPP	EA-NPE	EA-Sammon	EA-CCA	EA-Isomap	EA-CDA	EA-LLE	EA-LE
Synthetic data with separate clusters	Rings	100 (2,4,8)	74.00 (2)	74.00 (2)	77.50 (1)	77.50 (2)	63.25 (7,8)	79.00 (1)	81.00 (7,8)	100 (1-8)
		74.79	58.69	56.59	70.41	68.44	61.47	72.53	73.50	**100**
	d1	100 (2,4-8)	100 (2,4)	90.50 (2)	100 (2)	100 (2)	100 (2)	100 (2,7)	90.00 (2)	100 (2)
		94.07	74.31	69.06	59.62	61.31	67.62	77.06	**87.81**	71.75
	d2	100 (2)	100 (2)	61.50 (2)	66.50 (4)	100 (2)	88.50 (2)	100 (2)	100 (2)	79.00 (2)
		70.21	59.87	48.56	59.31	60.75	60.06	59.94	56.50	**64.50**
	Half-rings	100 (2,4,8)	94.75 (4)	88.00 (8)	81.75 (2)	93.25 (6)	100 (2)	100 (2)	100 (1-8)	100 (2,4-8)
		82.86	81.06	80.12	64.59	68.69	79.62	72.41	**100**	90.84
	Mixed Image 2	72.40 (8)	67.50 (6)	67.70 (2)	60.00 (1)	70.80 (2)	71.00 (2)	70.80 (2)	66.90 (2)	68.10 (2-4,6-8)
		53.34	60.10	61.46	50.72	60.31	63.45	62.81	64.09	**67.05**
Synthetic data with touching clusters	Bars	99.25 (4)	100 (5,8)	75.25 (1)	69.75 (4,6)	99.50 (6)	99.50 (3,5)	74.00 (1)	99.00 (4)	99.25 (5)
		74.25	88.69	65.53	61.00	77.53	**90.28**	61.84	77.66	69.84
	Circs	99.50 (2,4,5)	81.00 (3)	78.75 (5)	82.25 (1)	71.00 (3)	99.50 (1,4-6)	99.50 (2)	78.75 (5)	99.50 (2,5)
		76.54	63.50	62.47	66.22	61.37	**88.97**	73.78	63.37	76.47
	R-2-new	89.20 (5)	58.80 (2)	58.80 (2)	65.80 (2)	60.60 (2)	63.20 (2)	79.80 (2)	59.80 (2)	80.60 (8)
		65.77	44.32	47.55	60.32	45.62	45.12	44.92	53.77	**67.80**
	Mixed Image 3	88.70 (5)	92.40 (5)	75.00 (5)	50.10 (1-8)	85.10 (5)	89.60 (4)	91.90 (3)	82.60 (1)	76.10 (5)
		67.14	**82.00**	66.34	50.10	68.42	79.95	**82.00**	60.31	68.12
	Spiral 2	85.00 (2)	56.33 (4,5,7)	55.67 (5,8)	65.33 (1)	77.67 (2)	84.00 (1,5)	91.33 (7,8)	60.33 (7)	85.00 (2,5,7,8)
		63.43	54.29	53.67	58.54	61.00	79.25	**81.92**	54.75	78.79
Real data	Wine	93.82 (8)	98.31 (3)	73.03 (5)	97.75 (1)	97.19 (1)	94.94 (5)	94.94 (4,6)	91.01 (2)	91.57 (3-6)
		72.12	**92.84**	61.45	70.86	68.40	82.80	82.94	85.74	86.24
	Std Yeast	67.71 (4)	72.14 (8)	72.14 (5)	72.92 (4)	72.40 (4)	67.45 (7)	72.40 (3)	51.04 (5)	63.28 (4-6,8)
		51.79	**63.38**	59.89	50.13	52.11	60.03	60.87	41.89	61.36
	Pima	65.10 (6,7)	71.35 (7)	65.63 (6)	64.71 (2,4)	68.49 (4)	64.71 (2,3,6-8)	64.71 (2)	65.76 (7)	64.71 (2-4,6-8)
		62.91	**65.74**	61.95	60.81	60.03	62.04	58.41	64.13	63.49
	Optdigits	80.70 (5)	56.60 (5)	23.60 (1)	81.90 (5)	82.70 (5)	82.60 (5)	80.90 (5)	47.10 (5)	72.00 (5)
		55.41	43.86	20.92	70.91	64.61	70.74	**72.30**	36.24	60.35
	Iris	90.67 (2,4,5,8)	90.00 (4,6)	95.33 (1,3,8)	89.33 (5)	90.67 (2)	94.67 (1)	90.67 (2)	79.33 (1)	90.67 (1-8)
		75,62	83.92	88.75	70.75	67.75	**91.17**	71.83	57.25	90.67
	Crabs	71.00 (2)	54.00 (1)	88.00 (1,4-6)	70.50 (5)	71.00 (2)	71.00 (2)	71.00 (2)	66.00 (3)	74.50 (5)
		57.56	52.06	**78.31**	56.13	56.87	56.87	56.44	62.12	63.44
	Breast Cancer	69.84 (3)	95.75 (1,4)	81.41 (1)	94.29 (1)	85.65 (4)	97.07 (1)	97.22 (1)	88.43 (5)	96.05 (1)
		62.12	88.54	71.34	75.35	65.96	92.22	**92.90**	72.29	64.79
	House Votes	88.36 (4)	90.09 (1)	90.09 (4-6)	89.22 (3,4)	94.40 (4)	88.36 (3)	89.22 (1)	59.91 (3)	66.81 (1-8)
		68.53	84.80	**88.79**	72.90	70.53	81.14	85.67	59.54	66.81

with single-link. For the synthetic datasets with touching clusters, the best DR methods are EA-Isomap and EA-LE, when used with the appropriate clustering algorithm.

For the Std Yeast dataset the worst results correspond to nonlinear local DR methods (EA-LLE and EA-LE). For the Optdigits dataset, the worst results correspond to local methods (EA-LPP, EA-NPE, EA-LLE and EA-LE), while nonlinear global methods perform very well. In the House votes dataset, the best DR algorithms in average CI are linear methods (EA-LPP and EA-NPE) and nonlinear global methods that preserves "geodesic" distances (EA-Isomap and EA-CDA). These last two algorithms also have very good results for the Breast cancer dataset.

From Table 2, we notice that there exists at least one DR method that outperforms or equals EAC for each dataset, showing that there is an advantage in performing DR.

Like in the feature space, single-link is the best extraction method, except for real datasets. In real datasets, k-means and Ward link work better.

Overall, nonlinear methods are more suitable for this space, with local methods working better in synthetic data with separate clusters.

6 Discussion

There are some interesting findings to draw from all the above data. First, there is an advantage in using DR techniques on the co-association matrix to improve clustering results. However, care must be taken in choosing the right DR technique for each dataset.

Second, the use of DR techniques usually improves the average consistency index (CI) values over the co-association matrix. This suggests that using DR makes the clustering results less dependent on the choice of the specific clustering algorithm.

Although no DR algorithm consistently outperforms all the others, some algorithms do well in specific circumstances. Good results are obtained from datasets with separate clusters using LPP and LE (local DR methods). For datasets with touching clusters, Isomap and LE (nonlinear DR methods) yield the overall best results. Importantly, in real datasets no DR algorithm stood out from the others, and considerable variability was detected from dataset to dataset, again stressing out that the choice of the appropriate DR technique is crucial.

To further investigate this aspect, we have computed the measures N1[3] and silhouette for the real datasets studied in this paper. Those values are presented in table 3. Datasets Std Yeast and Pima stand out for having high values of N1, and in those datasets local DR methods yield the best clustering results in terms of average CI. On the other hand, datasets Optdigits and Breast Cancer stand out for having low values of N1 and the best results in those datasets come from global DR methods. Also, Crabs and Std Yeast have low values of the silhouette index and local DR methods perform well with these datasets. Given the relatively small number of datasets and DR methods used in this paper, we present these associations not as proven rules, but rather as temporary guidelines. We will actively research these types of associations using more datasets and more DR methods in the future.

[3] As explained in [8] "This method constructs a class-blind minimum spanning tree over the entire dataset, and counts the number of points incident to an edge going across the two classes. The fraction of such points over all points in the dataset is used as the N1 measure."

Table 3. N1 and Silhouette measures for the real datasets studied in this paper, and type of DR method that yields the best average CI for both types of spaces (feature and similarity spaces). The question mark (?) indicates datasets where the best DR type is different in the two spaces.

Real Datasets	N1	Silhouette	Best DR type
Wine	0.118	0.4368	local
Std Yeast	0.388	0.2274	local
Pima	0.438	0.1524	local
Optdigits	0.059	0.2892	global
Iris	0.100	0.6565	?
Crabs	0.160	0.0442	local
Breast Cancer	0.057	0.7178	global
House Votes	0.159	0.4471	?

There are some differences between using the co-association matrix as features or as similarities. For example, CCA and CDA perform poorly in the former case but considerably better in the latter. On the other hand, Sammon performs better in the feature space relative to the similarity space.

It is interesting to note that the DR algorithms which have the highest NMI values for each dataset are very often the ones which have also the highest average CI values. In other words, it seems that the DR algorithms which yield the most consistent partitions also yield the best partitions. Furthermore, for each dataset, the highest NMI between the feature space and the similarity space very often corresponds to the highest average CI as well. This suggests that NMI (a measure which does not need to know the true partition) can help predict the CI (which does use the true partition).

7 Conclusions

This study shows that the use of dimensionality reduction (DR) 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 the influence of parameter choice for the DR and clustering algorithms.

We also reported some interesting associations between types of datasets and appropriate DR methods; however, further work is needed to draw conclusive information.

Acknowledgments. We acknowledge financial support from the FET programme within the EU FP7, under the SIMBAD project (contract 213250). This work was partially supported by the Portuguese Foundation for Science and Technology (FCT), scholarship number SFRH/BD/39642/2007, and grant PTDC/EIA-CCO/103230/2008.

References

1. Ayad, H.G., Kamel, M.S.: Cluster-based cumulative ensembles. In: Oza, N.C., Polikar, R., Kittler, J., Roli, F. (eds.) MCS 2005. LNCS, vol. 3541, pp. 236–245. Springer, Heidelberg (2005)

2. 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)
3. Demartines, P., Hérault, J.: Curvilinear component analysis: a self-organizing neural network for nonlinear mapping of data sets. IEEE Trans. on Neural Networks 8(1), 148–154 (1997)
4. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 309–318. Springer, Heidelberg (2001)
5. Fred, A., Jain, A.: Combining multiple clusterings using evidence accumulation. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(6), 835–850 (2005)
6. 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)
7. He, X., Niyogi, P.: Locality preserving projections. In: Advances in Neural Information Processing Systems (NIPS 2003), vol. 16 (2004)
8. Ho, T.K., Basu, M., Law, M.H.C.: Measures of Geometrical Complexity in Classification Problems. In: Data Complexity in Pattern Recognition, Advanced Information and Knowledge Processing, 1st edn., vol. 16, pp. 3–23. Springer, Heidelberg (2006)
9. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. ACM Computing Surveys 31(3), 264–323 (1999)
10. Kuncheva, L.I., Hadjitodorov, S.T.: Using diversity in cluster ensembles. In: Proc. of the Int. Conf. on Systems, Man and Cybernetics, vol. 2, pp. 1214–1219 (2004)
11. Kuncheva, L.I., Hadjitodorov, S.T., Todorova, L.P.: Experimental comparison of cluster ensemble methods. In: Proc. of the 9th Int. Conf. on Information Fusion, FUSION 2006 (2006)
12. Lee, J.A., Verleysen, M.: Nonlinear Dimensionality Reduction. Information Science and Statistics. Springer, Heidelberg (2007)
13. Lee, J.A., Lendasse, A., Verleysen, M.: Nonlinear projection with curvilinear distances: Isomap versus curvilinear distance analysis. Neurocomputing 57, 49–76 (2004)
14. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. Science 290, 2323–2326 (2000)
15. Sammon, J.W.: A nonlinear mapping for data structure analysis. IEEE Trans. on Computers 18(5), 401–409 (1969)
16. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. Journal of Machine Learning Research 3, 583–617 (2002)
17. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
18. Theodoridis, S., Koutroumbas, K.: Pattern Recognition. Elsevier Academic Press (2003)