Expert Systems With Applications 51 (2016) 151–160

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Clustering using PK-D: A connectivity and density dissimilarity

Ariel E. Bayá*, Mónica G. Larese, Pablo M. Granitto

CIFASIS, French Argentine International Center for Information and Systems Sciences, UNR-CONICET (Argentina), Bv. 27 de Febrero 210 Bis, Rosario 2000, Argentina

ARTICLE INFO

Keywords:
Clustering
Dimensionality reduction

ABSTRACT

We present a new dissimilarity, which combines connectivity and density information. Usually, connectivity and density are conceived as mutually exclusive concepts; however, we discuss a novel procedure to merge both information sources. Once we have calculated the new dissimilarity, we apply MDS in order to find a low dimensional vector space representation. The new data representation can be used for clustering and data visualization, which is not pursued in this paper. Instead we use clustering to estimate the gain from our approach consisting of dissimilarity + MDS. Hence, we analyze the partitions' quality obtained by clustering high dimensional data with various well known clustering algorithms based on density, connectivity and message passing, as well as simple algorithms like *k*-means and Hierarchical Clustering (HC). The quality gap between the partitions found by *k*-means and HC alone compared to *k*-means and HC using our new low dimensional vector space representation is remarkable. Moreover, our tests using high dimensional gene expression and image data confirm these results and show a steady performance, which surpasses spectral clustering and other algorithms relevant to our work.

1. Introduction

Clustering algorithms can be used to uncover unknown relations existing in a set of unlabeled data. These algorithms can be divided into families according to their characteristics, for example there are partitional and hierarchical algorithms (Jain, Murty, & Flynn, 1999; Xu & Wunsch II, 2005). Similarly, hierarchical algorithms can be divided into agglomerative and divisive methods. Thus, we could make a taxonomy to categorize all clustering methods. This classification of algorithms into "families" shows a particular approach to clustering, one requiring many different algorithms for different kinds of data. On the contrary, we aspire to solve many clustering tasks using a reduced number of simple algorithms. Moreover, our main goal is to use the most simple algorithms available. As a result, we direct our interest into clustering methods involving kernels (Dhillon, Guan, & Kulis, 2004; Mika et al., 1999) or more general representations based on dissimilarity matrix (Pekalska & Duin, 2008; Pekalska, Paclik, & Duin, 2002; Schölkopf, 2001). These types of methods are able to simplify the clustering procedure by using a low dimensional vector representation derived from the kernel or dissimilarity matrix. There is a rich bibliography describing both groups of methods including also a crossover area discussing the relation between kernels and dissimilarities, i.e. a formal discussion explaining when a dissimilarity can be treated as a kernel and how to proceed when it cannot (Pekalska & Duin, 2008; Schölkopf, 2001; Williams, 2002). The dissimilarity proposed in this work does not qualify as a Mercer Kernel, hence, its decomposition leads to a non-Euclidean space. To find an Euclidean representation approximating the original data we only consider the positive spectra of the dissimilarity. Moreover, we only use a small subset of the eigenvectors of the centered dissimilarity. However, we can accurately represent datasets formed by arbitrary shaped clusters or high dimensional noisy data, even if the clusters do not have spherical shape or Gaussian distribution.

As we stated above, our motivation is to reduce the complexity of a clustering problem by improving the representation of the data. We have pursued this goal in a previous work and, as a result, we developed a penalized metric (Bayá & Granitto, 2011) that permitted us to cluster with a simple algorithm data having arbitrary shapes and high dimensionality. However, this metric could not overcome many of the limitations from methods based on connectivity. The solution proposed in the present paper aims at: (i) finding a lower dimensional representation of the original data and (ii) overcoming some of the limitations known to exist in connectivity approaches (Bayá & Granitto, 2011). Thus, we build a new dissimilarity combining connectivity and density information as an improvement to methods based solely on connectivity. Next, we apply MDS to the dissimilarity to find a new representation of the

* Corresponding author. Tel.: +54 341 4815569x326; fax: +54 341 4821771.
E-mail addresses: baya@cifasis-conicet.gov.ar (A.E. Bayá), larese@cifasis-conicet.gov.ar (M.G. Larese), granitto@cifasis-conicet.gov.ar (P.M. Granitto).

http://dx.doi.org/10.1016/j.eswa.2015.12.037
0957-4174/© 2016 Elsevier Ltd. All rights reserved.

original data. The representation found by MDS allows us to use any clustering algorithm without restricting us to those relying on dissimilarity matrix. Finally, we use a simple clustering algorithm to find groups in the new representation and compare the quality of them with other similar algorithms.

This manuscript has the following structure: Section 2 first considers previous works related to our ideas and goals. It also describes our two dissimilarity variants, the merging strategy and other related matters. Section 3 discusses first how to set up the parameters of our dissimilarity and then it shows the results of our experiments on real data. Finally, Section 4 presents some conclusions and considers ideas for future work.

2. Finding a new data representation

2.1. Related work

The idea of developing a method able to model the complex relations between the patterns of a dataset is not new. There are many dimensionality reduction methods (Belkin & Niyogi, 2003; Roweis & Saul, 2000; Tenenbaum, De Silva, & Langford, 2000), kernel methods (Mika et al., 1999) and spectral methods (Luxburg, 2007; Nadler & Galun, 2007) trying to accomplish this. Dimensionality reduction target is to find a new representation using fewer dimensions that preserves the relations existing in the original data. The outcome from these methods can be used for visualization, clustering or classification. However, applying clustering or classification after dimensionality reduction might not have always the desired effect. Preserving the ties between the new representation and the original data might not always be helpful to find “good” partitions. For example, Principal Components Analysis (James, Witten, Hastie, & Tibshirani, 2014) finds a representation preserving ties by retaining the components with highest standard deviation, however, the components dividing data into groups might not be those with highest standard deviation. Analogously, the cost function used by ISOMAP (Tenenbaum et al., 2000) or LLE (Roweis & Saul, 2000) to find a lower data representation does not emphasize in preserving the natural differences within the data. Our dissimilarity, on the contrary, is specially designed for clustering rather than visualization, hence, it emphasizes natural differences within the data. Therefore, after applying MDS we find a representation making the subsequent clustering step easier.

In a previous work we developed a distance called PKNNG (Bayá & Granitto, 2011), which we successfully used to cluster arbitrary shaped clusters, high dimensional noisy data and data embedded in a manifold. However, we were restricted to combine it with a small subset of clustering algorithms since PKNNG transformed the original data into a dissimilarity matrix. This dissimilarity is based on a graph of neighbors, hence, it relies only on connectivity. The components from the neighbor graph are connected by penalized edges joining the closest components with a single edge. Finally, the geodesic distance is calculated between all pairs using Dijkstra’s algorithm. Since PKNNG relies on connectivity there are cases that are too complicated or not possible to solve, for example, a pair of overlapped spherical clusters with Gaussian distribution. We explore the use of density information as a possible solution to some of the limitations of PKNNG. There is a similarity known as Evidence Accumulation by multiple Clustering (EAC) (Fred & Jain, 2005), which is used for clustering. However, there is a range of values for which EAC does not behave as a good similarity because it fragments the information to the point of rendering it useless. Fred and Jain (2005) discuss this issue in great detail. Yet, we have found out that fragmented information provides us with interesting insight about the density among neighbors. Our method aggregates this information to PKNNG in an effort to overcome previous limitations.

Our dissimilarity is used in combination with Classical MDS to find a more simple representation of the original data. There are several methods that have already explored this topic, for instance, Xu, Hancock, and Wilson (2014) used Ricci flows to remove artifacts rendering dissimilarity non-Euclidean. Later they tested their corrected dissimilarity in classification problems. Solving classification problems in vector spaces derived from dissimilarities has been properly introduced by Pękalska and Duin (Pekalska & Duin, 2008; Pekalska et al., 2002). There are some results under particular circumstances showing that classification using dissimilarity based feature spaces can be better than the ones obtained based on kernels (Kim & Duin, 2010). However, it should be noted that the data supporting this conclusion is reduced. There are other contributions related to our work pursuing visualization rather than clustering. Isomap (Tenenbaum et al., 2000) aims to find a lower dimensional representation from a dataset by using connectivity, connections through the shortest path and geodesic distance. Both Isomap and PKNNG share some features, however, the penalization scheme from PKNNG opposes to the idea of preserving geometrical relations. Instead, PKNNG is intended for clustering, hence it penalizes non-neighboring distances. There are methods pursuing the objective of ISOMAP but using different strategies to achieve it, among the most relevant visualization/dimensionality reduction methods we can name: Local Liner Embedding (LLE) (Roweis & Saul, 2000), Laplacian Eigenmaps (Belkin & Niyogi, 2003) and stochastic neighbor embedding (SNE) (Hinton & Roweis, 2003).

Our objective is to construct a dissimilarity with discriminative properties by aggregating two sources of information: density and connectivity. The discriminative properties emphasize the dissimilarity between non-neighboring samples in order to simplify the search for clusters. We use three methods as basic blocks to build our function. The first two blocks are used to measure density by one of two methods: (1) EAC, which is a method based on ensembles of *k*-means (Forgy, 1965) and (2) a method based on k nearest neighbors (Mitchell, 1997) (k-nn). The k-nn ensembles mimics the behavior of EAC but using an unsupervised k-nn algorithm instead. As a result, the first and second block originate each a dissimilarity variant, which estimates density in a different way. The third building block is the PKNNG distance. Finally, after having our dissimilarity we apply classical MDS (Cox & Cox, 2000) to find a lower dimensional data representation, which we use to find clusters. Fig. 1 shows a diagram of the proposed pipeline and Section 2.3 provides a thorough description of our approach.

2.2. Finding a low dimensional data representation

Assuming there is a generic dissimilarity ($D \in \mathcal{R}^{n \times n}$) we would like to find a new vector space representation based on D. We define $S = D^2$ and $H = I - \frac{1}{n}\mathbf{1}\mathbf{1}^T$, where S is a squared dissimilarity matrix ($s_{ij} = d_{ij}^2$), I is the identity matrix, $\mathbf{1}$ is a $n \times 1$ vector of ones and H is the centering matrix. We use these elements to find a new vector representation X:

$$\begin{aligned} B &= \frac{-HSH}{2} \\ &= V\Lambda V^T = XX^T, \end{aligned} \tag{1}$$

where Λ is a diagonal matrix containing the eigenvalues of B and V is an orthogonal eigenvectors matrix. When B is not a semidefinite matrix there will be negative eigenvalues in Λ. A discussion about this issue and the full derivation of the previous equation can be found in Pekalska et al. (2002), Williams (2002) and Schölkopf (2001). Mercer’s Theorem (Cristianini and Shawe-Taylor, 2000, Section 3.3.1) relates the eigenvalues from Λ to squared norms in the new space representation having V as a base. Hence, the existence of negative eigenvalues amounts to negative squared distances, which contradicts Euclidean geometry. We solve this problem by

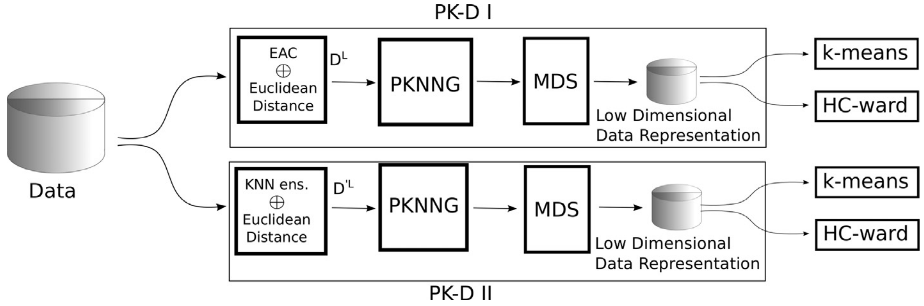


Fig. 1. The diagram depicts the PK-D pipeline. A full description of PK-D can be found in Section 2.3. The low dimensional data representation found at the end of the pipeline is used for clustering. The results section compares the clustering quality of PK-D + MDS and clustering vs. the quality of other clustering methods.

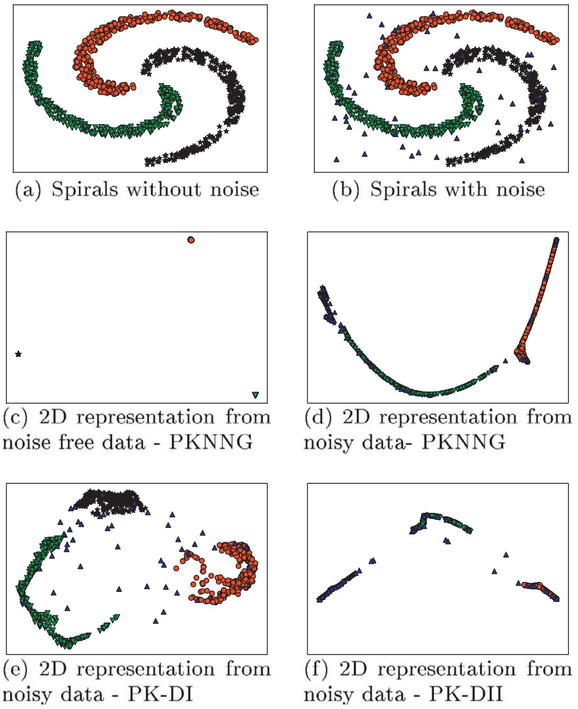

(a) Spirals without noise
(b) Spirals with noise
(c) 2D representation from noise free data - PKNNG
(d) 2D representation from noisy data- PKNNG
(e) 2D representation from noisy data - PK-DI
(f) 2D representation from noisy data - PK-DII

Fig. 2. Datasets: (a) Spirals without noise, (b) Spirals with noise. MDS representations of the Spirals: (c) 2D representation of the Spirals without noise using PKNNG. (d) 2D representation of the Spirals with noise using PKNNG. (e) 2D representation of the Spirals with noise using PKNNG + *k*-means ensembles. (f) 2D representation of the Spirals with noise using PKNNG + *k*-nn ensembles. The parameters used to find these representations are discussed in Section 3.1. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

restricting the eigenvectors from V to the k eigenvectors with positive eigenvalues. Then, we find our representation as:

$$X_k = V_k \Lambda_k^{\frac{1}{2}}, \tag{2}$$

where $V_k \in \mathcal{R}^{n \times k}$, $\Lambda_k \in \mathcal{R}^{k \times k}$ and all eigenvalues ($\lambda_i \in \Lambda_k$) are positive ($\lambda_i > 0$). The actual number (q) of eigenvectors required to find clusters with our dissimilarities is much smaller than the number of positive eigenvectors k ($q \ll k$). Section 3 tells how to set the number of eigenvectors (q).

The PKNNG dissimilarity has a penalization strategy to weight distances, hence, the dissimilarity between samples in the same neighborhood are small but this value increases dramatically for points laying far apart. This kind of response makes easy to find arbitrary shaped structures, where clusters are separated by a small distance. The key is that the area separating the clusters has no samples within, thus, there are no neighboring relation between clusters. For example, let us consider the two dimensional representation shown by Fig. 2c. This representation is based on the noise-free Spirals from Fig. 2a. This representation from Fig. 2c results from the use of PKNNG as dissimilarity D in Eq. (1), where $S = D^2$. Now, let us consider the spirals with noise shown by Fig. 2b. The analog representation corresponding to the use of PKNNG and MDS is illustrated by Fig. 2d. We can see that for a method solely based on connectivity it is much more difficult to provide a representation with good discriminative capability. Therefore, we introduce the use of density information to alleviate the influence of the noise but also as a general improvement to issues related to connectivity methods. Fig. 2e and f show each one a two dimensional representation of the noisy spirals. Each panel illustrates the response to a different density estimation method. By contrasting Fig. 2d–f we can say that adding density information to PKNNG reduces the damaging impact of noise. However, estimating the density of the data is not a trivial task nor it is to find a proper way to aggregate it to the pairwise distance.

2.3. Aggregating density and connectivity information

We have directed our attention first to EAC as a candidate to estimate the density of the data. This similarity can be used to cluster data using a simple hierarchical algorithm, because it produces a simple representation of any complicated cluster shape or non-Gaussian distribution. The idea proposed by Fred and Jain was to build an ensemble using many *k*-means partitions, which they used to derive a similarity. Clustering ensembles attracted much attention and a lot of work has been done on the subject (Vega-Pons & Ruiz-Shulcloper, 2011). We use the ensemble algorithm with a different setting in order to gather local information. We aim to find small neighborhoods of samples to estimate density. Below we present a brief description of EAC and k-nn ensembles.

EAC builds a similarity matrix T from a dataset $X \subset \mathcal{R}^n$ and a clustering algorithm. We follow Fred and Jain in the use of *k*-means as partitioning algorithm, hence, we cluster the dataset M times with it. Each clustering result tells us which samples of X are grouped together. After the clustering step, EAC translates the clustering information into pairwise relations. The translation is done by counting how many times do a pair of samples (x_i, $x_j \in X$) appear together in the same cluster. Repeating M times the clustering let us estimate the probability using a frequentist approach. Thus, the probability of any two points x_i and x_j appearing together can be measured by $t_{ij} = \frac{N_{ij}}{M}$, where N_{ij} counts the number of times that x_i and x_j are in the same cluster. All matrix entries (t_{ij}) of T meet the following conditions: $0 \le t_{ij} \le 1$; $t_{ij} = t_{ji}$ and $t_{ii} = 1$. As a result, Matrix T can be seen as an adjacency matrix, where t_{ij} tells the degree of connection of a pair of vertices.

Inspired by EAC we considered a similar but much faster method, which is based on the k-nn algorithm. This algorithm has a unique solution for a dataset X and a value k. Thus, to gather useful density information we need to subsample the dataset X. Removing a random fraction f of the data, where $0 \leq f \leq 1$, lets us find many variants of X that are both different to themselves and to the original dataset. Hence, we may find different groups of neighbors when k-nn is applied repeatedly M times. Mimicking EAC we create a matrix T' analog to T. The value of an element (i, j) of T' is found by $t'_{ij} = \frac{N'_{ij}}{M'_{ij}}$, where N'_{ij} counts the number of times that x_i and x_j are mutual neighbors and M'_{ij} counts the total number of times that x_i and x_j appear simultaneously ($N'_{ij} \leq M'_{ij} \leq M$). Hence, when two samples (x_i and x_j) appear in a subsample of X a unit is added to M'_{ij} and to N'_{ij} only if the two samples are mutual neighbors. Thus, $0 \leq t'_{ij} \leq 1$; $t'_{ij} = t'_{ji}$ and $t'_{ii} = 1$. Matrix T' is an adjacency matrix too, where each element t'_{ij} tells the degree of connection of a vertex pair.

Next, we merge the Euclidean distance matrix (D^{euc}) associated with X and the density information gathered by T and T'. With these elements we define $D^L = \frac{D^{euc}}{T}$, where an element (i, j) from D^L can be written as $d^L_{ij} = \frac{d^{euc}_{ij}}{t_{ij}}$, i.e. elementwise matrices division. Analogously, D'^L is the elementwise division of D^{euc} and T' ($D'^L = \frac{D^{euc}}{T'}$). It is important to limit the maximum value of d^L_{ij}, otherwise $d^L_{ij} \to \infty$ when $t_{ij} \to 0$. We achieve this by bounding the minimum t_{ij}, thus, all elements t_{ij} with values lower than α are set to α. This rule penalizes non-local points $\frac{1}{\alpha}$ folds in terms of the Euclidean distance. We apply this thresholding to D'^L too. Fig. 1 shows the pipeline described above for both PK-D variants.

We use both versions (D^L and D'^L) of the Euclidean distance with local information as input to PKNNG. Finally, we arrive to the result we were looking for: a dissimilarity considering connectivity, achieved by PKNNG, and local density, achieved either by k-means ensembles or by k-nn ensembles. We use the notation PK-D I to refer to the dissimilarity using local density estimated by k-means ensembles and PK-D II to refer to the one estimated with k-nn ensembles. Now, we reconsider the Spirals Dataset with noise from Fig. 2b. There are two low dimensional representations, one illustrated by Fig. 2e (PK-D I) and the other by Fig. 2f (PK-D II). The structures corresponding to each cluster (black, red and green) are not compact in any of the two representations, however, the added density information makes it easier to partition the data with k-means. The section presenting our experimental results revisits these examples and supports our claim.

2.4. Ensembles matrix: a closer look

Below we discuss a few details about the ensembles matrices, in order to describe how the matrices are constructed and the meaning of their components. The objective of this subsection is to clarify the density information encoding used by the adjacency matrices T and T'.

A single k-means iteration results in the samples of X being tagged with a label, where the tagging process is controlled by the algorithm optimization rule. We assume that when the number of clusters K given to k-means is greater than the number of clusters existing in X, we are finding neighborhoods or high density areas rather than clusters. Thus, the value of K is related to the size of the neighborhood, as it gets bigger when the size of the neighborhoods becomes smaller and so does the number of samples in those areas. After assigning each sample to a cluster we impose the condition of neighbors to the samples inside each cluster, therefore, we increment their counter N_{ij} by one. At the end of the process N_{ij} counts how many times were (i, j) together. Hence, $\frac{N_{ij}}{M}$ is a normalized frequency bounded by (0, 1). By using N_{ij} we can find the weighted mean number of neighbors of any sample (i). For example, $\bar{N}_i = \sum_j \frac{N_{ij}}{M} = \sum_j t_{ij}$ is the sample (i) weighted mean number of neighbors. In the same fashion, we can derive $\bar{N}'_i = \sum_j \frac{N'_{ij}}{M'_{ij}}$ in spite of the more complex relation introduced by M'_{ij}. Expanding $\bar{N}'_i$ notation:

$$\bar{N}'_i = \sum_j t'_{ij} = \frac{\sum_j N'_{ij}}{\sum_j M'_{ij}} = \frac{\frac{\sum_j N'_{ij}}{M}}{\frac{\sum_j M'_{ij}}{M}} = \frac{\langle N'_i \rangle}{\langle M'_i \rangle}, \tag{3}$$

where $\langle N'_i \rangle$ is the mean value of N'_{ij} and $\langle M'_i \rangle$ is the mean value of M'_{ij}.

There is also a link between adjacency matrices T and T' and probabilities. An entry from T or T' represents the strength bonding sample (i and j). Thus, we can find the probability that a sample ($x_j \in X$) selects another sample ($x_i \in X$) as its neighbor, i.e. $p(i|j)$, by $p(i|j) = \frac{t_{ij}}{\sum_j t_{ij}}$ and $p'(i|j) = \frac{t'_{ij}}{\sum_j t'_{ij}}$. Using matrix notation we can write the later as $P = T \cdot D^{-1}$, where $p(i|j)$ is element (i, j) from P and D is a diagonal matrix having $\bar{N}_i = \sum_k t_{ik}$ as elements of its diagonal, i.e. $d_{kk} = \sum_k t_{ik}$. We can find P' in the same fashion.

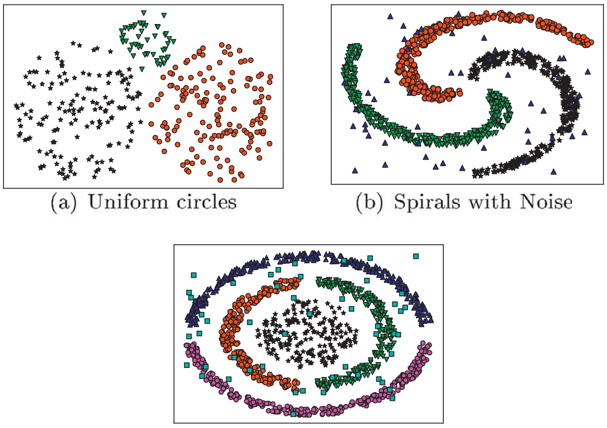
Matrices T and T' and their respective normalizations P and P' are known to be used by Spectral Clustering. Luxburg (2007) shows that adjacency matrices T and T' are related to the unnormalized Laplacian and the matrices of conditional probabilities P and P' are a particular normalization that can be interpreted as a random walk.

3. Experimental results

In this section we present the results of clustering our new vector space representation obtained by: (i) calculating the dissimilarities PK-D I or PK-D II, (ii) applying MDS and selecting a small group of principal directions. We also include a group of algorithms ranging from complex methods like Spectral Clustering and Affinity Propagation to more simple but commonly used methods like k-means and Hierarchical Clustering. We consider simple algorithms to contrast the performance given by clustering the original data and the representation derived from PK-D. We are also including complex algorithms to establish the general impact of our dissimilarity.

3.1. Toy data

Dissimilarities PK-D I and II are formed by two elements D^L, D'^L and PKNNG. We use this subsection to show the difference in performance between the blocks forming PK-D and PK-D working as a whole. The Toy Datasets used in this part are illustrated by Fig. 3. The results from our experiments are shown by Figs. 4 and 5. In our first experiment we followed the sequence: (i) find a lower data representation using one of D^L, D'^L, PKNNG, PK-D I, PK-D II and the original data; (ii) clustering using k-means or Hierarchical Clustering with Ward linkage (HC-w) and (iii) comparing the clustering labels with the true class labels using the Adjusted Rand Index (Hubert & Arabie, 1985) (ARI). Thus, the figure presenting our results is divided in two: left column presents the results of k-means and right column presents the results of HC-w. Each panel shows in the x-axis the name of a dissimilarity with an extra suffix (.a) or (.b), where (.a) was added to those clustered with k-means and (.b) to those clustered with HC-w. We use this naming convention throughout the paper. The y-axis has ARI value corresponding to a partition found by one of the two clustering algorithms. The first element of the x-axis is named either as k-means or HC-w and it corresponds to the clustering algorithm applied to the original data. Our experiment consists of: applying dissimilarity first, then MDS, and finally clustering the two principal directions. The



(a) Uniform circles

(b) Spirals with Noise

(c) Rings with Noise

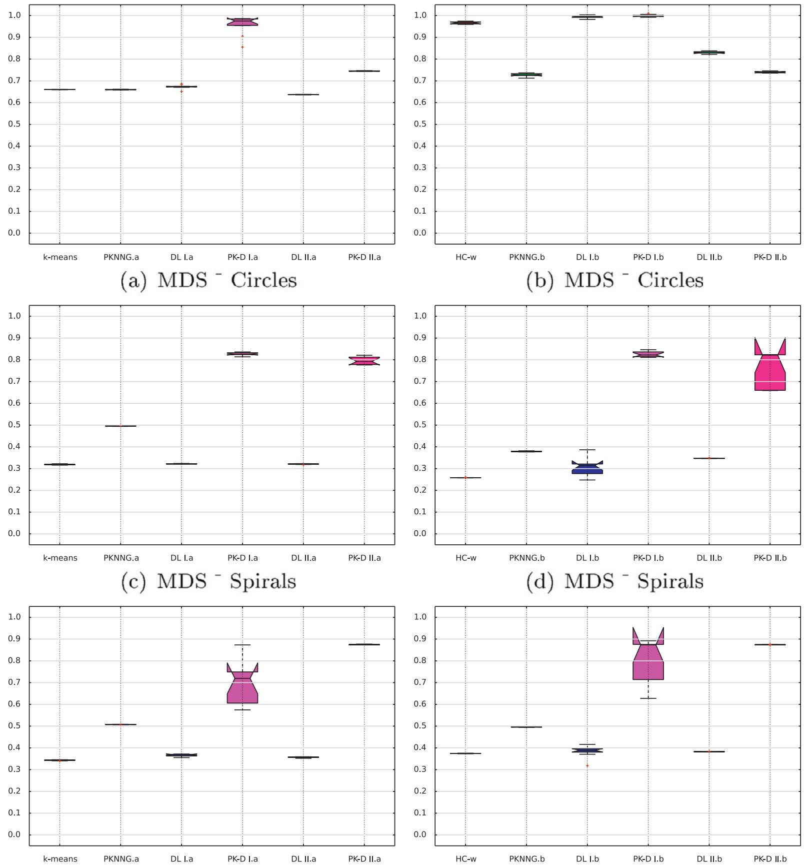
Fig. 3. Toy data. (a) Uniform circles, (b) Spirals with noise and (c) Rings with noise. The noisy version of the Spirals shown by panel (b) and the noise free dataset differ in the blue painted triangles, i.e. the samples representing noise.

procedure was done 30 times and the results are shown using box-plots in Fig. 4.

The results from Fig. 5 correspond to our second set of experiments. On one hand, the left column shows the value of the ARI vs. the number of neighbors used in PKNNG while the parameters of the ensembles are fixed. On the other hand, the right column shows the value of the ARI vs. the parameters of the ensembles while the parameter of PKNNG is fixed. The panels from the right column use a double *x*-axis, from top to bottom the first one is calibrated in number of clusters for the *k*-means ensembles and the second in number of neighbors for the k-nn ensembles. Finally, their *y*-axis is calibrated using the ARI. The panels from the left have the *x*-axis calibrated in number of neighbors used as a variable in PKNNG and the *y*-axis is calibrated using the ARI. These results are the outcome of averaging 10 runs of the following sequence: applying the dissimilarity, applying MDS and clustering a lower data representation with two principal directions.

The results of Fig. 4 show a consistent improvement in PK-D performance respect to its components D^L, D'^L and PKNNG. There is a single case in Fig. 4b, where PK-D II.b scores are lower than D'^L.b. Yet, this kind of results are not unexpected, since under certain conditions one of the two methods might hinder the overall performance of the dissimilarity. In this example, the choice of the clustering algorithm influences negatively the clustering outcome as it can be seen by contrasting PK-D II.a and PK-D II.b. Therefore, we can conclude that PK-D works better than its separate components, despite a single adverse result.

The second set of experiments are related to the stability respect to the parameters of PKNNG and D^L or D'^L. Except for the variability shown by Fig. 5e there are no noticeable performance loss for the range of parameters studied. Hence, we set for the remaining experiments the value *k* used by PKNNG to 17 and the value of *K* used by the *k*-means ensembles to $N/10$, where *N* is the number of samples in the dataset. There could be examples where $N/10$ leads to a low average number of samples per clusters, lower



(a) MDS ¯ Circles

(b) MDS ¯ Circles

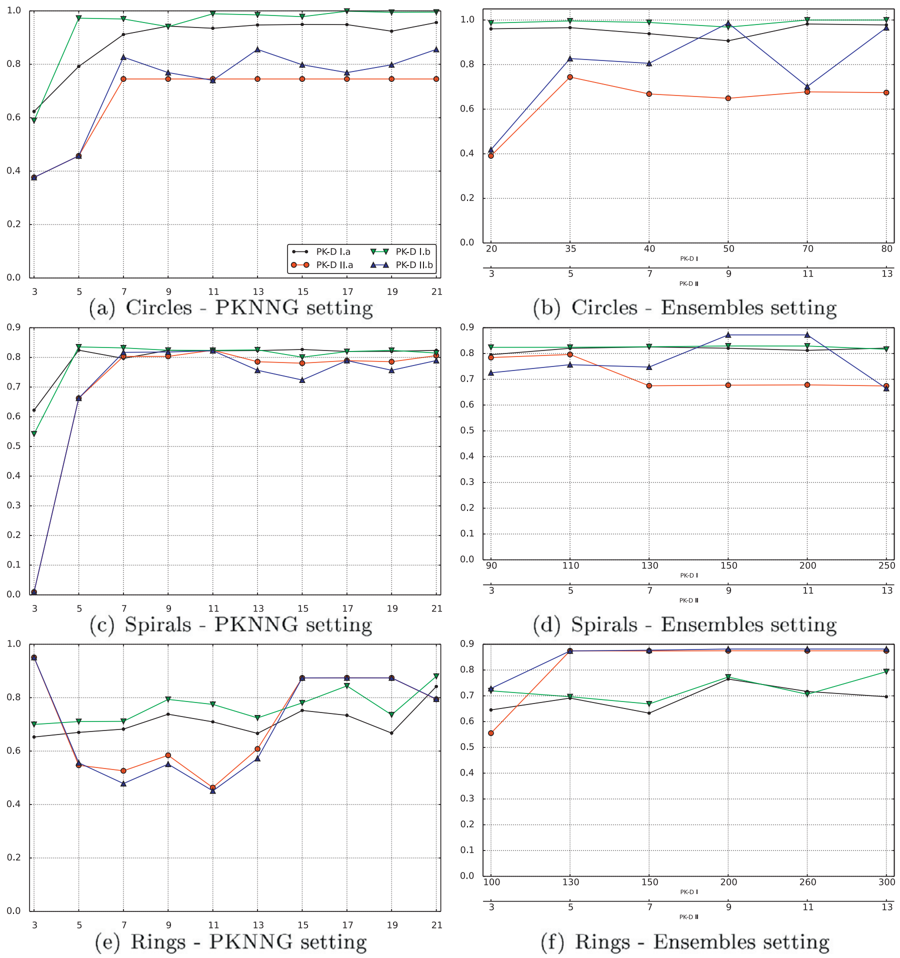
(c) MDS ¯ Spirals

(d) MDS ¯ Spirals

(e) MDS ¯ Rings

(f) MDS ¯ Rings

Fig. 4. Distribution of ARI values for the original data and five representations. Left column shows clustering with *k*-means and right column shows clustering with HC-w. The first element from each panel was notated as *k*-means or HC-w and it reports the clustering result of the corresponding algorithm when applied to the raw data. The name of each dissimilarity in the *x*-axis has an extra suffix (.a) or (.b) informing the clustering algorithm: (.a) for *k*-means and (.b) for HC-w. We use this naming convention throughout the paper.



(a) Circles - PKNNG setting

(b) Circles - Ensembles setting

(c) Spirals - PKNNG setting

(d) Spirals - Ensembles setting

(e) Rings - PKNNG setting

(f) Rings - Ensembles setting

Fig. 5. PK-D I and PK-D II settings. We include both clustering algorithms (a) *k*-means and (b) HC-w to evaluate the changes in clustering quality for various setups. In the left column we show ARI values (*y*-axis) vs. number of neighbors (*x*-axis) used by PKNNG. In the right column we show ARI values (*y*-axis) vs. number of clusters used to build the *k*-means ensembles (top *x*-axis) and number of neighbors used to build the k-nn ensembles (bottom *x*-axis).

than 5. In the event of $N/10$ being too low, we recommend raising it to $N/5$. The parameters required for the k-nn ensembles are the number of neighbors and *f*. We use 5 neighbors unless we state otherwise and set *f* to 0.85 as we did in the above examples. Finally, α is set to 0.1. As we explained in Section 2.3, it limits the lower similarity value for both *k*-means ensembles and k-nn ensembles.

3.2. Real data

We consider two domains of application: Biological and Image data. Testing each variant of our dissimilarity requires applying PK-D I and II to the original data; decomposing the dissimilarity matrix with MDS and clustering a lower data representation with a simple algorithm (*k*-means or HC-w). Next, we compared the quality of our clustering results and the results of other methods. We repeated the clustering algorithm 30 times and measured its quality with the ARI. All datasets were clustered using *k*-means, HC-w, average linkage (HC-av), affinity propagation (Aff), Spectral Clustering with RBF (SC-RBF) and Nearest Neighbor Graph (SC-NN) as similarity. We used scikit-learn (Pedregosa et al., 2011) implementation of all clustering algorithms and quality measure (ARI) and present our results using boxplots. For all the competing algorithms we performed an exhaustive search to find the configuration leading to the best clustering quality, i.e. highest cRand.

As we did for the toy data, we combine PK-D I and II with *k*-means or HC-w. Hence, we code the type of dissimilarity and clustering algorithm with the following notation: PK-D I refers to the dissimilarity variant using *k*-means ensembles and PK-D II to the variant using k-nn ensembles. In both cases we use fixed values of *k* either as the number of clusters for *k*-means examples or as the value *k* used for the unsupervised k-nn ensembles, and we set the value α to 0.1. On the other hand, the number of neighbors used by PKNNG was set to 17 for all datasets. Table 1 condenses

Table 1
Main features of the real datasets. Nine datasets described by four main features: type of data (oligonucleotide, Grayscale or RGB), number of samples (*n*), number of classes (Classes), dimensionality (*p*), number of clusters used for *k*-means ensembles in PK-D I (PK-D I), value of *k* used for k-nn unsupervised ensembles in PK-D II (PK-D II) and ensembles size (ES), which is equal for both variants. The number of neighbors used by PKNNG was set to 17 for all datasets. *Each COIL image is vectorized considering the three RGB channels to a (128 × 128 × 3) vector.

Dataset	Type	*n*	Classes	*p*	PK-D I	PK-D II	ES
CNS	Oligo	42	5	989	10	3	500
LEU	Oligo	248	4	1000	20	3	500
AML-ALL	Oligo	38	3	999	6	3	500
BPLC	Oligo	103	4	1000	15	3	500
LUNG	Oligo	197	4	1000	20	3	500
Olivetti	Gray	400	40	500	100	5	200
Leaves	Gray	866	3	100	100	7	200
MNIST	Gray	2000	10	784	200	7	200
COIL	RGB	1440	20	49,152*	200	2	200

the most important information about the data, which includes the parameter settings for each dataset. Additionally, we use the suffixes (a) and (b), as we did in the Toy data subsection. Each suffix indicates the clustering algorithm: (a) *k*-means or (b) HC-w. For example, the notation PK-D I.a refers to the dissimilarity variant using *k*-means ensembles and *k*-means for clustering. Finally, we selected lower data representation with a dimensionality equal to the number of eigenvectors used by Spectral Clustering. We find this to be the most simple and fair setting to compare both methods.

For the sake of completeness we have included in an additional file an expanded version of our experiments. This version includes two Density-Based algorithms: DBSCAN (Ester, Kriegel, Sander, & Xu, 1996) and an automated clustering method by Sander, Qin, Lu, Niu, and Kovarsky (2003) based on OPTICS (Ankerst, Breunig, Kriegel, & Sander, 1999). Both algorithms are used to cluster the original high dimensional datasets and the low vector space representations PK-D I and PK-D II. We close this section with a discussion concerning all clustering methods.

3.2.1. Biological data

Each one of the sets: ALL-ALT, LEU, CNS, BPLC and LUNG (Broad Institute, 1999) have been preprocessed by one or more bioinformatics methods aimed to select a number of significant genes (Monti, Tamayo, Mesirov, & Golub, 2003). Each of these sets correspond to a different type of cancer or tissue (BPLC). The clustering results are aimed to find if the genes selected can uncover the types of cancer or tissues hidden within the genetic data. Table 1 condenses the most important information about the data. For our tests we have used a publicly available version of the Biological datasets with the significant genes already selected as described by Monti et al. (2003). The files are available at Broad Institute (1999) cancer section. Before clustering all datasets were normalized to zero mean and unit standard deviation by row and then by column (Kluger, Basri, Chang, & Gerstein, 2003).

3.2.2. Image data

We used three publicly available and well known datasets: MNIST digits (Lecun & Cortes, 1998); Olivetti faces (AT&T Labs, 1992) and COIL (Nene, Nayar, & Murase, 1996). We have also used the Leaves dataset, which is a non-public set coming from an agricultural study by Larese et al. (2014). In the case of MNIST digits (MNIST), we randomly selected a subset of 200 digits per class due to the size of the whole set. For the same reason we selected the first 20 classes from COIL-100 (COIL). In both datasets the images were vectorized, thus each row of the dataset corresponds to an image and each column to a pixel. No other preprocessing was applied to these two sets.

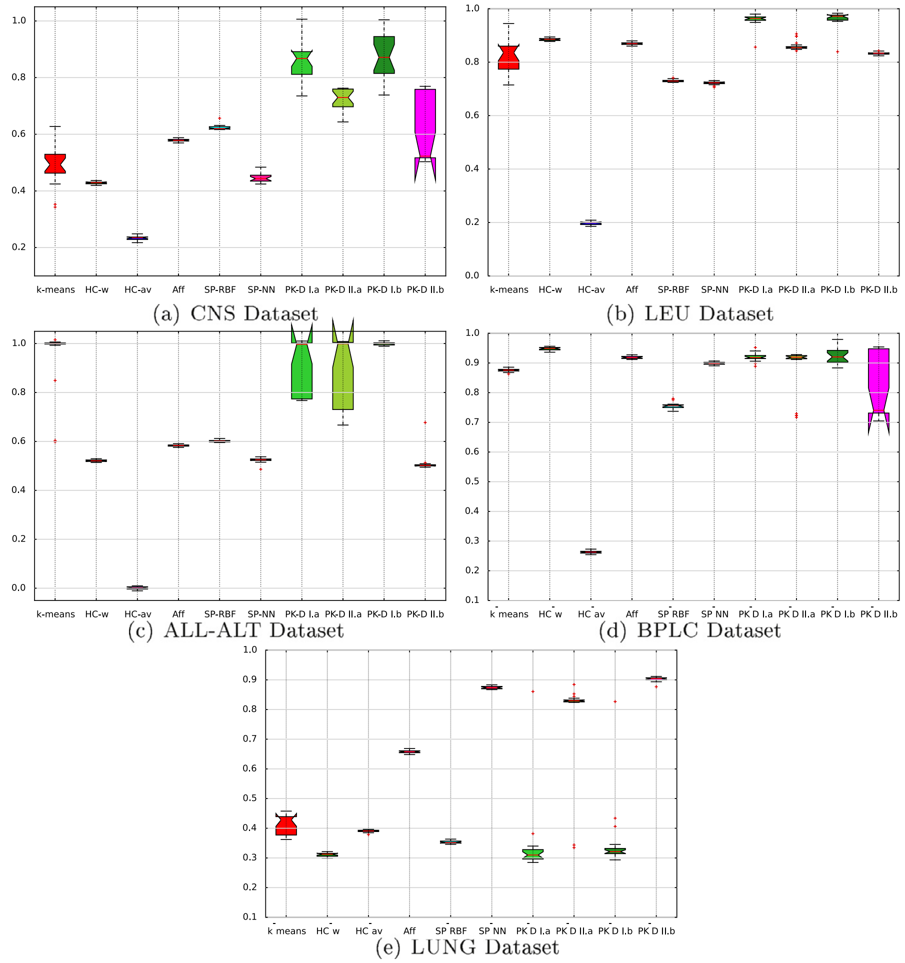
The remaining examples, Olivetti faces and Leaves dataset, were preprocessed before the clustering phase using bag of words (BoW) (Fei-Fei & Perona, 2005). Our BoW procedure involved first using SIFT (Lowe, 2004) to detect and compute points of interest (*poi*) in each image. The number of *poi* in each image is variable. Then, we created two temporary representations, one for Olivetti and another for Leaves dataset, consisting in all the images' *poi* together, i.e. we concatenated the *poi* from each image in order to extract a dictionary of "graphical terms". The concatenation of *poi* is a simple matrix concatenation since all *poi* have the same length, i.e. the length of a SIFT descriptor. The dictionaries are found by clustering the temporary representations with *k*-means. The dictionary size, i.e. the number of clusters used by *k*-means, was of 500 words for Olivetti and of 100 for the Leaves dataset. Once the dictionaries are determined we transform each image into a vector by associating a dictionary term with a position in the vector. So, we use each bin of the vector to count the number of times a term appears. Hence, according to Euclidean distance we assign each *poi* to its closest term and add one to the vector position associated with the term. Thus, if a point of interest poi_i is assigned to a word represented by *Cluster*10, we add one to position 10 in the vector. Finally, we apply clustering to this new BoW representation without any further preprocessing. We used the OpenCV 2.4 SIFT implementation with default parameters to generate each temporary representation. Table 1 condenses the most important information about the data and Fig. 7 shows a diagram of the BoW preprocessing method.

3.2.3. Real data results

Fig. 6 illustrates our experimental results on biological data. There are two aspects that have a negative impact in the clustering quality: (i) the density variability within each cluster and (ii) the high dimensionality of the data. Both factors make these examples to be challenging clustering tasks. As a result, we have found that in some examples it is difficult to set the value of K required for the ensembles methods. In the case of *k*-means ensembles, mostly we used a value near $K = \frac{N}{10}$ as a rule of thumb. However, on occasions this value can be too near to the actual number of clusters, or the number of samples in the dataset could be too small. Thus, for BPLC, AML-ALL and CNS we had to change $K = \frac{N}{10}$ to a number near $K = \frac{N}{5}$. In our tests there are two sources of variability: (i) the ensembles methods and (ii) the clustering algorithms. The effect of ensemble variability is shown by the high variability of PK-D II.b in BPLC and CNS. This variability cannot be attributed to HC-w because it is a deterministic algorithm. The cause is related to f, the parameter used in PK-D II to sample the data. We can summarize this effect as follows: the subsampling procedure leads to an ensemble that reduces the separation between clusters, however, this effect depends on the clustering algorithm too. This becomes clear as we contrast the PK-D II.a and PK-D II.b results in BPLC and CNS. In both cases PK-D II.a has higher median and less interquartile distance. Also, the ensembles variability is related to its size, Fred and Jain (2005) studied this parameter thoroughly for *k*-means ensembles. We have set the size of the ensemble to 500 to avoid variations due to low size. The second effect, variability of the clustering algorithms, can be found in *k*-means and it can only be noticed in ALL-ALT. We can see that results for PK-D I.b (HC-w clustering) have less interquartile distance than for PK-D I.a (*k*-means clustering), thus, different results under the same data point to the clustering algorithm.

Most image data examples (Leaves, MNIST and COIL) show interesting results (Fig. 8). In those cases affinity propagation and spectral clustering can only perform as well as *k*-means and HC-w. On the other hand, PK-D I shows that it can match clusters to classes more accurately, thus, letting us infer that PK-D I is finding the natural structure within the data. In a lesser degree we can say the same from PK-D II. Finally, Leaves dataset shows that methods relying on connectivity, either ensembles of unsupervised *k*-nn (PK-D II) or SC, are not successful to associate clusters to classes. In the remaining example, Olivetti faces, there is a marginal improvement of PK-D II over the rest of the methods.

For the sake of completeness we have included an additional file with expanded tests results. These experiments include two algorithms based on density: DBSCAN (Ester et al., 1996), noted as DBS, and an automated clustering method by Sander et al. (2003) based on OPTICS (Ankerst et al., 1999). Common knowledge indicates that as dimensionality increases estimating density becomes more challenging, classical texts like Mitchell (1997) discuss this fact. Thus, we expect that an algorithm based on density probably would not perform well. Our suspicions, except for two cases (BCLP and COIL), were confirmed. Results from the additional file, see Figs 1–4, show that only DBSCAN has an acceptable performance.



(a) CNS Dataset

(b) LEU Dataset

(c) ALL-ALT Dataset

(d) BPLC Dataset

(e) LUNG Dataset

Fig. 6. Biological dataset. On each panel we show the result of one dataset. The *y*-axis holds the ARI values of the clustering algorithms named in the *x*-axis.

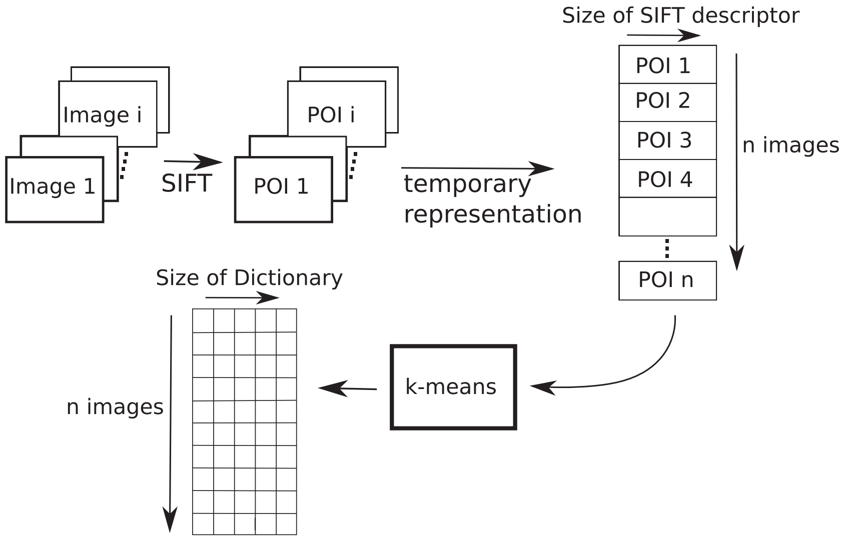
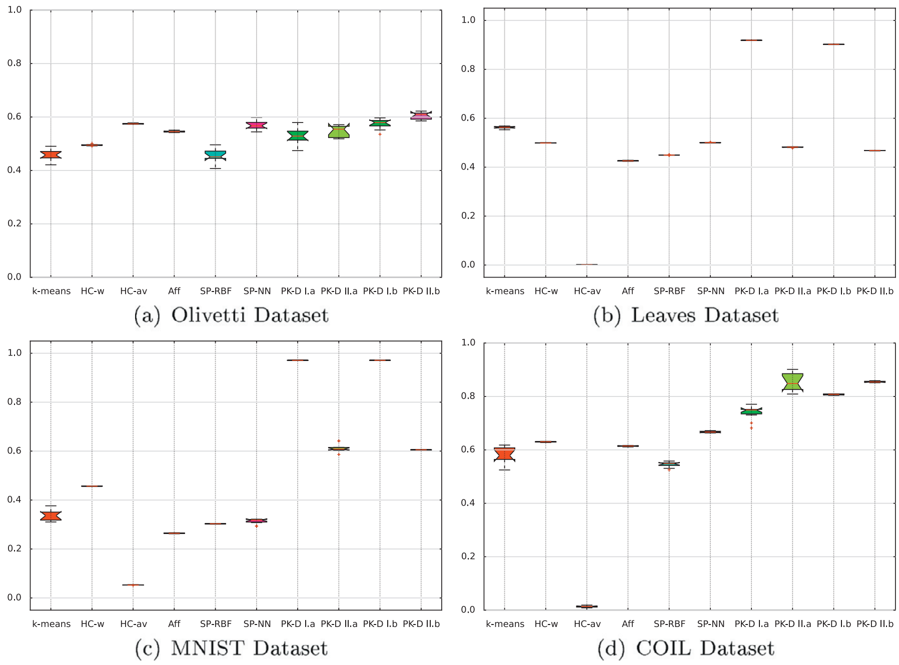


Fig. 7. BoW preprocessing. The diagram shows the BoW preprocessing pipeline used for Olivetti faces and Leaves dataset. See Section 3.2.2 for a description of the process.

We complete our test by applying both DBSCAN and OPTICS to PK-D I and PK-D II. In almost all datasets PK-D improves the clustering quality of DBSCAN and OPTICS alone; however the results of PK-D + OPTICS and PK-D + DBSCAN are not as good as those from PK-D + *k*-means and PK-D + HC-w. Nonetheless, we have found two unfavorable cases: COIL, where PK-D I + OPTICS (PK-D I.c) and PK-D I + DBSCAN (PK-D I.d) did not work and Olivetti, where only PK-D I + OPTICS (PK-D I.c) did work. COIL and Olivetti have the vector representations with highest dimensionality, 20 and 40 dimensions, respectively. Yet, DBSCAN alone worked well on COIL but not in combination with PK-D I, which strongly suggests that data density is being distorted by the PK-D.



(a) Olivetti Dataset

(b) Leaves Dataset

(c) MNIST Dataset

(d) COIL Dataset

Fig. 8. Image dataset. On each panel we show the results for one dataset. The *y*-axis holds the ARI values of the clustering algorithms named in the *x*-axis.

4. Discussion and concluding remarks

In this paper we presented a new dissimilarity based on connectivity and density. To this end we introduced a new procedure to aggregate connectivity and density information. This procedure allowed us to mix PKNNG, a connectivity method, with *k*-means ensembles and k-nn ensembles. We showed on toy data that the combination of connectivity and density was more resilient to noise since it increased clustering quality respect to clustering using connectivity and density alone. We did also show that our approach PK-D + MDS boosted clustering quality, i.e. the same clustering algorithm without PK-D + MDS had poorer results. Moreover, PK-D + MDS combined with *k*-means and HC-w surpassed Spectral Clustering with RBF and NN dissimilarities and other well known algorithms based solely on density like DBSCAN and OPTICS. Also, we showed that combining DBSCAN and OPTICS with PK-D + MDS has better results than using DBSCAN and OPTICS alone. However, these results are not as good as using PK-D + MDS combined with *k*-means or HC-w. The reason behind such improvement in clustering quality is that our procedure PK-D + MDS performs a non-linear dimensionality reduction. The operations done to obtain D^L, D'^L are non-linear and so it is PKNNG. Thus, PK-D + MDS output is able to retain the most relevant components even if the data is not linearly separable. As a result, clustering algorithms respond better to PK-D + MDS than to the original data. Spectral Clustering also divides a low dimensional representation, yet the most common similarity do not consider both density and connectivity, therefore, on many test cases it fails to divide the data according to the classes.

There is a constant interest in enhancing clustering methods to tackle new and more challenging problems. Chen (2015) proposed improving density clustering by using a search based on near neighbor influence, an idea partly inspired by near neighbors. Similar to Chen (2015) we did explore the use of neighbors to estimate data density. However, our work has a different direction. First, because we merge density and connectivity, and second because our approach, PK-D + MDS, can use any clustering algorithm to search for clusters. The notion of simplifying a clustering problem by using an appropriate dimensionality reduction method is not new and has shown interesting results in the past (Song, Yang, Siadat, & Pechenizkiy, 2013). Thus, it is not uncommon to find similarities with other authors. For example, Inkaya (2015b) developed a new similarity for spectral clustering by using density and connectivity information. Contrary to PK-D this method has no parameters, which is highly positive, however, the author reports some issues with noisy data. Also, we have found that density and connectivity information was successfully used for classification tasks (Inkaya, 2015a).

We have tested two variants of PK-D, each resulting from a different density estimation procedure and four clustering algorithms. All together we have shown that PK-D + MDS boosted clustering quality. Yet, we did not discuss how to choose the “best” PK-D variant and clustering algorithm. This could be the greatest weakness of our approach. Yet, selecting these parameters is a clustering validation problem, and since PK-D + MDS can lead to non-Gaussian data distributions the problem is not an easy one. In the past, we had addressed the problem of validating arbitrary shaped partitions (Baya & Granitto, 2013). However, the current problem requires to compare solutions in order to select a parameter configuration, PK-D variant and clustering algorithm. This problem cannot be solved by our validation procedure. We leave as future work to develop an automated procedure to select the best PK-D settings, clustering algorithm and number of clusters leading to the “best” data partition. This problem is still one of the most interesting and difficult unsolved problems in clustering.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.eswa.2015.12.037.

References

Ankerst, M., Breunig, M., Kriegel, H.-P., & Sander, J. (1999). Optics: ordering points to identify the clustering structure. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, *28*(2), 49–60.

AT&T Labs (1992). *The Olivetti faces dataset*. Cambridge: AT&T Lab. http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html. Accessed June 2015.

Bayá, A. E., & Granitto, P. M. (2011). Clustering gene expression data with a penalized graph-based metric. *BMC Bioinformatics, 12*(1), 2. doi:10.1186/1471-2105-12-2.

Baya, A. E., & Granitto, P. M. (2013). How many clusters: A validation index for arbitrary-shaped clusters. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 10*(2), 401–414. doi:10.1109/TCBB.2013.32.

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation, 15*(6), 1373–1396.

Broad Institute (1999). Broad Institute Cancer Datasets. http://www.broadinstitute.org/cgi-bin/cancer/datasets.cgi. Accessed June 2015.

Chen, X. (2015). A new clustering algorithm based on near neighbor influence. *Expert Systems with Applications, 42*(21), 7746–7758. http://dx.doi.org/10.1016/j.eswa.2015.05.007.

Cox, T. F., & Cox, M. A. A. (2000). *Multidimensional scaling* (2nd ed.). Chapman & Hall/CRC.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines: And other kernel-based learning methods*. New York, NY, USA: Cambridge University Press.

Dhillon, I., Guan, Y., & Kulis, B. (2004). Kernel *k*-means, spectral clustering and normalized cuts. In *Proceedings of the tenth ACM SIGKDD international conference on knowledge discovery and data mining, KDD-2004* (pp. 551–556).

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In E. Simoudis, J. Han, & U. Fayyad (Eds.), *Proceedings of second international conference on knowledge discovery and data mining* (pp. 226–231). AAAI Press.

Fei-Fei, L., & Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition, CVPR 2005: Vol. 2*. (pp. 524–531).

Forgy, E. W. (1965). Cluster analysis of multivariate data: Efficiency versus interpretability of classifications. *Biometrics, 21*, 768–769.

Fred, A., & Jain, A. K. (2005). Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 27*(6), 835–850.

Hinton, G., & Roweis, S. (2003). Stochastic neighbor embedding. In *Proceedings of international conference on advances in neural information processing systems*.

Hubert, L., & Arabie, P. (1985). Comparing partitions. *Journal of Classification, 2*(1), 193–218.

Inkaya, T. (2015a). A density and connectivity based decision rule for pattern classification. *Expert Systems with Applications, 42*(2), 906–912. http://dx.doi.org/10.1016/j.eswa.2014.08.027.

Inkaya, T. (2015b). A parameter-free similarity graph for spectral clustering. *Expert Systems with Applications, 42*(24), 9489–9498. doi:10.1016/j.eswa.2015.07.074.

Jain, A. K., Murty, M. N., & Flynn, P. J. (1999). Data clustering: A review. *ACM Computing Surveys, 31*(3), 264–323.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2014). *An introduction to statistical learning: With applications in R*. Springer Publishing Company, Incorporated.

Kim, S.-W., & Duin, R. (2010). An empirical comparison of kernel-based and dissimilarity-based feature spaces. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics): Vol. 6218* (pp. 559–568). Springer Berlin Heidelberg.

Kluger, Y., Basri, R., Chang, J. T., & Gerstein, M. (2003). Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Research, 13*(4), 703–716.

Larese, M. G., Namías, R., Craviotto, R. M., Arango, M. R., Gallo, C., & Granitto, P. M. (2014). Automatic classification of legumes using leaf vein image features. *Pattern Recognition, 47*(1), 158–168.

Lecun, Y., Cortes, C. (1998). The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist. Accessed June 2015

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision, 60*(2), 91–110. doi:10.1023/B:VISI.0000029664.99615.94.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing, 17*(4), 395–416. doi:10.1007/s11222-007-9033-z.

Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., & Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. In *Proceedings of international conference on advances in neural information processing systems* (pp. 536–542).

Mitchell, T. M. (1997). *Machine learning* (1st ed.). New York, NY, USA: McGraw-Hill, Inc.

Monti, S., Tamayo, P., Mesirov, J., & Golub, T. (2003). Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Machine Learning, 52*(1–2), 91–118.

Nadler, B., & Galun, M. (2007). Fundamental limitations of spectral clustering. In *Proceedings of international conference on advances in neural information processing systems 19* (pp. 1017–1024). MIT Press.

Nene, S. A., Nayar, S. K., Murase, H. (1996). Columbia Object Image Library (COIL-100). http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php. Accessed June 2015

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., … Duchesnay, E. (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research, 12*, 2825–2830.

Pekalska, E., & Duin, R. (2008). Beyond traditional kernels: Classification in two dissimilarity-based representation spaces. *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews, 38*(6), 729–744.

Pekalska, E., Paclik, P., & Duin, R. P. W. (2002). A generalized kernel approach to dissimilarity-based classification. *Journal of Machine Learning Research, 2*, 175–211.

Roweis, S., & Saul, L. (2000). Nonlinear dimensionality reduction by locally linear embedding. *Science, 290*(5500), 2323–2326.

Sander, J., Qin, X., Lu, Z., Niu, N., & Kovarsky, A. (2003). Automatic extraction of clusters from hierarchical clustering representations. In *Proceedings of the 7th Pacific-Asia conference on advances in knowledge discovery and data mining* (pp. 75–87). Springer-Verlag.

Schölkopf, B. (2001). The kernel trick for distances. In *Proceedings of international conference on advances in neural information processing systems* (pp. 301–307).

Song, M., Yang, H., Siadat, S., & Pechenizkiy, M. (2013). A comparative study of dimensionality reduction techniques to enhance trace clustering performances. *Expert Systems with Applications, 40*(9), 3722–3737. http://dx.doi.org/10.1016/j.eswa.2012.12.078.

Tenenbaum, J., De Silva, V., & Langford, J. (2000). A global geometric framework for nonlinear dimensionality reduction. *Science, 290*(5500), 2319–2323.

Vega-Pons, S., & Ruiz-Shulcloper, J. (2011). A survey of clustering ensemble algorithms. *International Journal of Pattern Recognition and Artificial Intelligence, 25*(03), 337–372.

Williams, C. (2002). On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning, 46*(1–3), 11–19.

Xu, R., & Wunsch II, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks, 16*(3), 645–678.

Xu, W., Hancock, E., & Wilson, R. (2014). Ricci flow embedding for rectifying non-Euclidean dissimilarity data. *Pattern Recognition, 47*(11), 3709–3725.